AGARD-CP-455

AGARD-CP-455

AGARD

ADVISORY GROUP FOR AEROSPACE RESEARCH & DEVELOPMENT

7 RUE ANCELLE 92200 NEUILLY SUR SEINE FRANCE

AGARD CONFERENCE PROCEEDINGS No.455

Advances in Techniques and Technologies for Air Vehicle Navigation and Guidance

NORTH ATLANTIC TREATY ORGANIZATION

DISTRIBUTION AND AVAILABILITY
ON BACK COVER

AGARD-CP-455

NORTH ATLANTIC TREATY ORGANIZATION

ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT

(ORGANISATION DU TRAITE DE L'ATLANTIQUE NORD)

AGARD Conference Proceedings No.455

ADVANCES IN TECHNIQUES AND TECHNOLOGIES FOR AIR VEHICLE NAVIGATION AND GUIDANCE

Papers presented at the Guidance and Control Panel 48th Symposium held at the Instituto da Defesa Nacional in Lisbon, Portugal from 9th to 12th May, 1989.

THE MISSION OF AGARD

According to its Charter, the mission of AGARD is to bring together the leading personalities of the NATO nations in the fields of science and technology relating to aerospace for the following purposes:

- Recommending effective ways for the member nations to use their research and development capabilities for the common benefit of the NATO community;
- Providing scientific and technical advice and assistance to the Military Committee in the field of aerospace research and development (with particular regard to its military application);
- Continuously stimulating advances in the aerospace sciences relevant to strengthening the common defence posture;
- Improving the co-operation among member nations in aerospace research and development;
- Exchange of scientific and technical information;
- Providing assistance to member nations for the purpose of increasing their scientific and technical potential;
- Rendering scientific and technical assistance, as requested, to other NATO bodies and to member nations in connection with research and development problems in the aerospace field.

The highest authority within AGARD is the National Delegates Board consisting of officially appointed senior representatives from each member nation. The mission of AGARD is carried out through the Panels which are composed of experts appointed by the National Delegates, the Consultant and Exchange Programme and the Aerospace Applications Studies Programme. The results of AGARD work are reported to the member nations and the NATO Authorities through the AGARD series of publications of which this is one.

Participation in AGARD activities is by invitation only and is normally limited to citizens of the NATO nations.

The content of this publication has been reproduced
directly from material supplied by AGARD or the authors.

Published December 1989

ISBN 92-835-0535-2

Printed by Specialised Printing Services Limited
40 Chigwell Lane, Loughton, Essex IG10 3TZ

PREFACE

In the 1990's and beyond, air vehicles will be designed on the basis of functionally integrated systems and the operation of such aircraft will also be highly integrated.

Past experience has clearly shown that interfacing classical subsystems can introduce critical compromises in overall systems performance, availability, safety and cost.

Future systems will be designed and built by means of a new process where the physical characteristics of the resulting. systems may not have any resemblance to today's subsystems or line-replaceable units.

The navigation and guidance aspects of future air vehicles cover a broad spectrum of important technical issues to meet the operational requirements with improved capability and increased survivability for reasonable costs. The systems increasingly depend upon the use of advanced sensor and signal processing techniques and particularly on software to provide the "intelligence" required for their functional performance.

This symposium was intended to deal with advances in techniques and technologies to design, build and test such navigation and guidance systems.

AVANT-PROPOS

Dans les années 1990 et au-delà, les véhicules aériens seront conçus sur la base de systèmes multi-fonctions et la mission de tels aéronefs sera aussi hautement coordonnée.

L'expérience acquise démontre que l'interconnexion des sous-systèmes classiques risque d'amener des compromis critiques en ce qui concerne les performances globales, la disponibilité, la sécurité et le coût des systèmes.

Les systèmes futurs seront conçus et construits au moyen de procédés nouveaux où les caractéristiques physiques des systèmes ne pourront avoir aucune ressemblance avec les sous-systèmes actuels qui sont remplaçables aux divers échelons de maintenance.

Les performances en navigation et en pilotage qui seront demandées aux aéronefs futurs afin de satisfaire aux besoins opérationnels, tout en assurant des capacités améliorées et un niveau de survivabilité plus élevé à des coûts raisonnables soulèvent bon nombre de questions techniques importantes dans des domaines divers. Les systèmes en question dépendent de plus en plus de la mise en oeuvre de techniques sophistiquées de détection et de traitement du signal et, en particulier, des logiciels qui fournissent "l'intelligence" nécessaire à leur fonctionnement.

Le symposium a examiné les progrès réalisés dans le domaine des techniques et des technologies demandées pour la conception, la réalisation et les essais de tels systèmes de navigation et pilotage.

GUIDANCE AND CONTROL PANEL OFFICERS

Chairman: Ir Pieter Ph. van den Broek
Delft University of Technology
Department of Aerospace Engineering
Kluyverweg 1
2629 HS Delft, The Netherlands

Deputy Chairman: Professor Edwin B.Stear
Director
Washington Technology Center
University of Washington
376 Loew Hall — FH 10
1013 NE 40th Street
Seattle, WA 98195
United States

TECHNICAL PROGRAMME COMMITTEE

Chairman:	Mr U.Krogmann	GE
Members:	Mr B.Chaillot	FR
	Dr H.Winter	GE
	Pr Dr P.Sanz-Aranguez	SP
	Mr R.Dunn	UK
	Pr J.T.Shepherd	UK
	Dr J.Niemela	US
	Dr G.T.Schmidt	US

PANEL EXECUTIVE

From Europe:
Commandant M.Mouhamad, FAF
Executive, GCP
AGARD—NATO
7, rue Ancelle
92200 Neuilly-sur-Seine, France
Telephone: (1) 4738 5780
Telex: 610176 F — Fax: (1) 4738 5799

From USA and Canada only:
AGARD—NATO
Attention: GCP
APO New York 09777

HOST PANEL COORDINATOR

Engineer Antonio L.Alves-Vieira
Instituto Politécnico de Setúbal
Escola Superior de Tecnologia
Largo dos Defensores da República 1
2900 Setubal, Portugal

ACKNOWLEDGEMENTS/REMERCIEMENTS

The Panel wishes to express its thanks to the Portuguese National Delegates to AGARD for the invitation to hold this meeting in their country and for the facilities and personnel which made the meeting possible.

Le Panel tient à remercier les Délégués Nationaux du Portugal près l'AGARD de leur invitation à tenir cette réunion dans leur pays et de la mise à disposition de personnel et des installations nécessaires.

CONTENTS

* Published in classified volume CP-455 (Supplement).
** Not available at time of printing.

SESSION IV: NEW TECHNIQUES AND ALGORITHMS
Chairman: Dr J.Niemela (US)

SESSION V: SENSOR TECHNOLOGY
Chairman: Dr G.Schmidt (US)

* Published in classified volume CP-455 (Supplement).
** Not available at time of printing.

* Published in classified volume CP-455 (Supplement).
** Not available at time of printing.

RECENT DEVELOPMENTS IN TERPROM

by

Denis M Grey and Robert S Dale

British Aerospace (Dynamics) Limited
Downshire Way
BRACKNELL
Berkshire
RG12 1QL
United Kingdom

SUMMARY

Terrain Reference Navigation Systems (TRNSs) began development some 30 years ago but since then their appeal has broadened considerably. The BAe Terrain Matching Profile system (TERPROM) is one of the most mature. It has 3 basic modes of operation and a number of features of which navigation is but one. Terrain Following (TF) and 'intelligent' ground proximity warning are just two of several benefits which result from TERPROM's ability to interpret the digital map data base. These features have all been tested in a variety of fast-jet aircraft. Recent developments include TERPROM's integration with other systems such as Doppler, Scene Matching Area Correlation (SMAC), and Global Positioning System (GPS) to improve navigation performance even further and moreover, a redesign of the system architecture has been proposed which enables the best information from a range of sensors to be used both for navigation and weapon aiming.

INTRODUCTION

The concept of terrain profile uniqueness and its use as a navigation technique began to be developed in the US in 1958. The US Air Force awarded industry a research contract in 1960 and as a result of this work, a number of terrain contour matching systems were conceived. In the 1960s, several programs were sponsored to investigate the application of these techniques to ballistic missiles and in the early 1970s, a contract was awarded to integrate them with a low altitude drone. The idea here was to make an operational system capable of being deployed. Later, studies established the feasibility of a terrain comparison aided INS on strategic cruise missiles. However, several developments since then have broadened the appeal of terrain referenced navigation systems (TRNS) into both aircraft and missiles.

The increased availability of digital map data has made it possible for vehicles employing TRNs to use a large proportion of potential target areas. The US Defence Mapping Agency (DMA) leads a consortium of countries of which each member is responsible for producing Digital Land Mass System (DLMS) data files for its own country and other areas. DLMS data exists for approximately 20 million square miles of the earth's terrain. Consortium countries continue to expand, update and improve the accuracy of their data bases to cope with the increasing requirement for a range of digitized displays for civil and military use.

However, even 20 million square miles does not cover the whole of the earth's terrain and where data does not currently exist, topographical paper maps can be digitized using automatic methods or they can be manually traced. Similarly, satellite or aircraft reconnaissance photographs can be used to generate DLMS data.

The development of TRNS has also been facilitated by advances in electronic technology including the increased capability of modern data processors and increased data storage density. TERPROM uses these advances to provide a mature, covert, day/night, all weather capability.

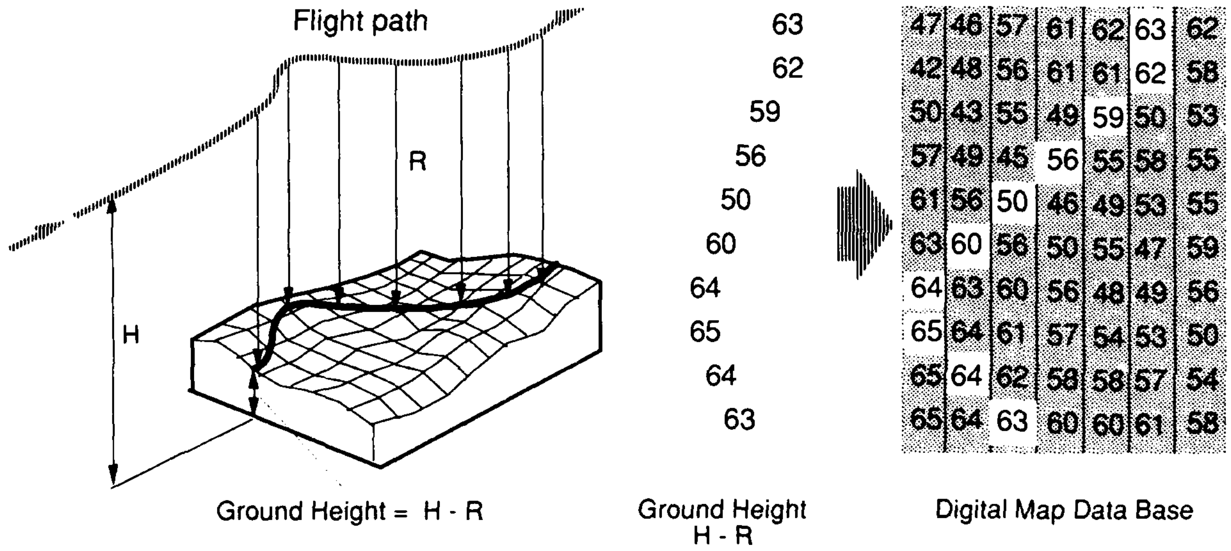
Several detailed papers have been presented on the technical aspects of TERPROM (Ref 1). For completeness, a brief description will be given here.

TERPROM MODES OF OPERATION

TERPROM is based on an extended Kalman Filter which accurately models the barometric/inertial height, the stored digital map data and the errors in the vehicle's dead reckoning system (usually an inertial navigation system (INS)). This error model is updated by radar altimeter (radalt) measurements of height above ground; the information is processed using 3 basic modes Single Fix, Continuous Fix and Prediction Mode.

Single Fix Mode

The first method, which is similar to that used on early TRNS, is called the Single Fix Mode and is illustrated diagrammatically at **Fig 1**. The algorithm constructs a profile of the ground over typically 5kms and compares this with the profiles contained in its stored terrain model. The point at which the best fit is obtained is a measure of the aircraft's true track.



47	46	57	61	62	63	62
42	48	56	61	61	62	58
50	43	55	49	59	50	53
57	49	45	56	55	58	55
61	56	50	46	49	53	55
63	60	56	50	55	47	59
64	63	60	56	48	49	56
65	64	61	57	54	53	50
65	64	62	58	58	57	54
65	64	63	60	60	61	58

Fig. 1 TERPROM Single Fix Principle

The Single Fix Mode is tolerant of large errors and is used when the system is uncertain of its exact position. For instance it is sometimes used to provide the initial fix after take off, and to reestablish position after long periods over flat, characterless terrain or over the sea.

Occasionally, another strip of ground has a similar profile to the one overflown and a false fix could result. To ensure that this occurrence has no adverse effect on the system, a second single fix is performed and correlated with the first. Only when two correlating fixes are found is the information processed in the Kalman Filter to give updated position and velocity estimates and the system switched to the Continuous Mode.

In all TERPROM flights processed by BAe, the Single Fix technique has never passed an erroneous fix to the Kalman Filter from which the continuous mode has been unable to converge.

Continuous Mode

The Continuous Mode is the normal mode of operation and is more accurate than the Single Fix mode. An understanding of the Continuous Mode can be obtained by considering the simplified example at **Fig 2**. Having processed the altimeter information at Update 1, the Kalman Filter has an estimate of the vehicle's position and height (point X) and a measure of the uncertainty in its estimate (ellipse E). The ellipse will in general lie with its major axis parallel to the terrain surface. Between updates, the Filter propogates the position, height and their uncertainty information forward to update 2 using a dynamic error model of the INS. This predicted position is marked Y and the uncertainty region is shown by the dashed ellipse. The altimeter measurement by itself would indicate an uncertainty region (G) parallel to the new terrain tangent. However, the Filter optimally merges this new information into an improved position and height estimate (Z) with a smaller uncertainty region (hatched area) than that of either measurement taken separately. In this way, small corrections are continually applied every 100m or so. For high speed vehicles, this corresponds to an update rate of about 3 Hz. The Kalman Filter thus sustains an estimated position which is more accurate than the batch processing of the Single Fix.

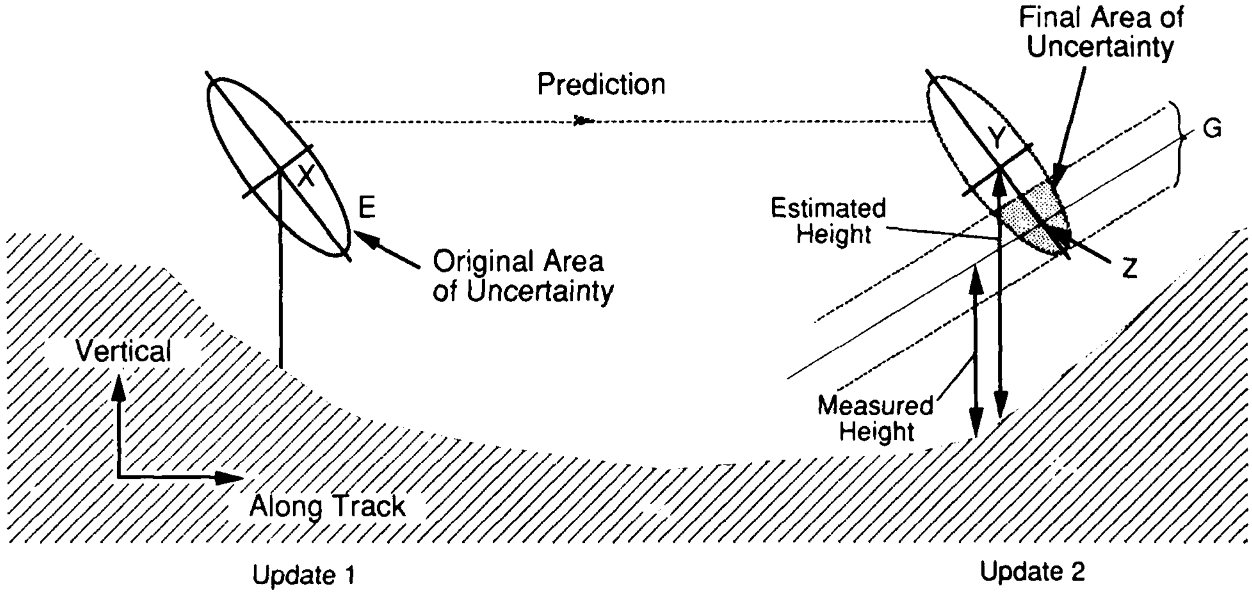


Fig. 2 The Kalman Filter Principle Used in Continous Mode

Prediction Mode

The Kalman Filter estimates errors in the dead reckoning system and retains knowledge of their values. The filter then predicts the effect of these errors using a dynamic model of propagation. So for an INS based system, the KF will establish the drift rate and compensate accordingly even if some update data are interrupted. This will happen if there is no radalt or terrain data available. Therefore an INS which is updated by TERPROM will be significantly more accurate, even if the radalt becomes temporarily unserviceable, than an INS without TERPROM.

INTEGRATING TERPROM MODES IN A MISSION

The method by which these basic Modes are integrated into a typical mission is illustrated at **Fig 3**.

After aircraft take off in the case of an aircraft system, or launch for a missile, the navigation system will drift at the INS drift rate. When the vehicle is being fed meaningful radalt information at least two Single Fixes are performed until they correlate. The Kalman Filter will switch to the Continuous Mode and remain there unless the system can no longer compare meaningful radalt information with the digital map - for instance when over long stretches of water. The system will then carry out Single Fixes or revert to the Prediction Mode. In Prediction, it will drift at a significantly lower rate than before because the INS drift errors have been modelled and eliminated. On flying over land again, TERPROM monitors the magnitude of the residual drift. If this is sufficiently small, TERPROM changes to the Continuous Mode. If not, Single Fixes are carried out as before and the system reverts to Continuous Mode after a correlating fix has been found. That said, for the majority of envisaged overland sorties, including flight over lakes, the system will stay in Continuous Mode after the initial Single Fixes.

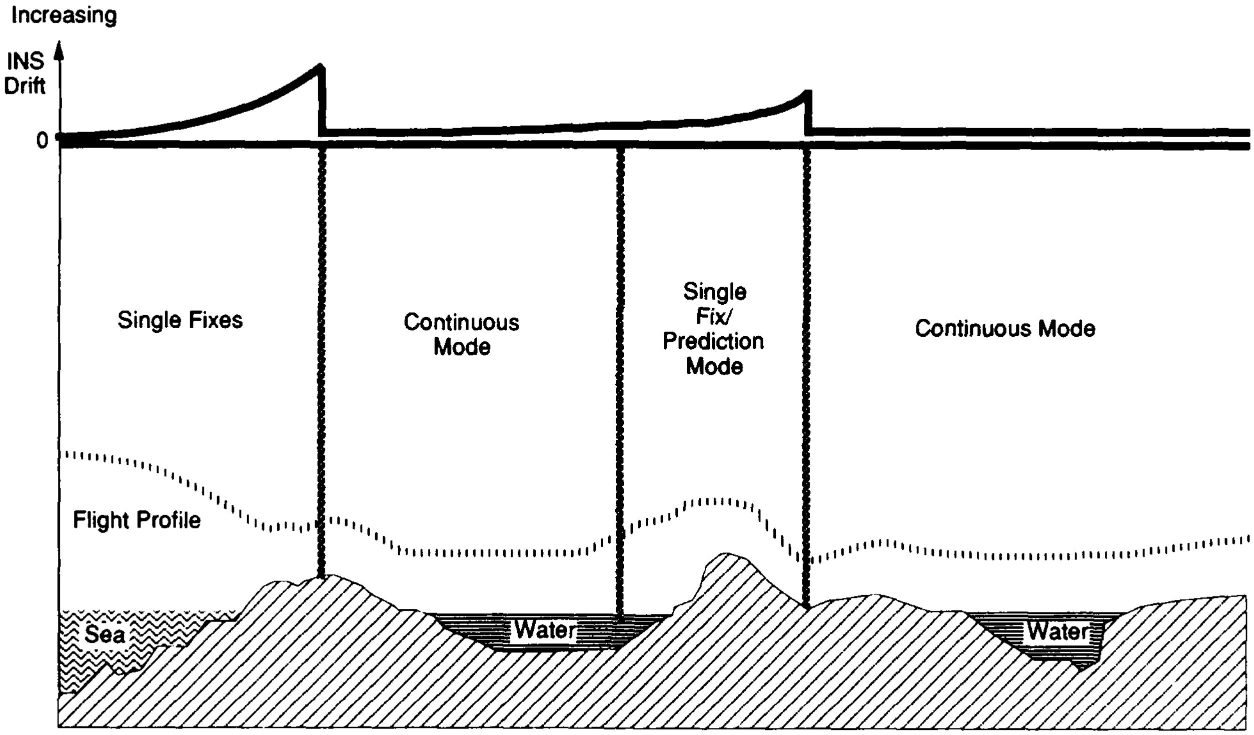


Fig. 3 Integration of Basic TERPROM Modes

TERPROM FEATURES

Some TRNS are promoted simply as a method of updating the INS and hence the aircraft's position on a moving map. However, TERPROM has been used in trials for years as a total terrain reference system (TRS). It is important to understand that it is a mathematical device for extracting a variety of information from the digital data base of which navigation is but one element. It is therefore not affected by acceleration and no 'g' restrictions have been applied by the operators in flight trails.

Navigation

Reliable, autonomous and covert navigation has been demonstrated over a wide variety of terrain types in the UK, USA and mainland Europe and over different coverings of snow, sand, trees etc. Unlike earlier TRNS, no specific fix areas are required and unlike Global Positioning System (GPS), TERPROM is immune to countermeasures. It is totally independent of satellites and does not therefore depend on the supply by a friendly government of the essntial P codes. Neither does it depend on Doppler radar information.

TERPROM trials have proved that no significant degradation occurs for at least 5 minutes even if the radalt is switched off or the vehicle is over the sea. Nevertheless, for continued accuracy, terrain slopes are required. **Fig 4** illustrates that navigation performance only slowly degrades when passing from undulating to flat terrain with less than 2 per cent slope. The degradation is slow because the INS will already have been calibrated. However, navigation accuracy increases rapidly as the vehicle overflies featured terrain.

As TERPROM models the errors in the INS, it can detect out of specification INS behaviour which may be indicative of an impending fault. Indeed, on two consecutive flights is 1987, TERPROM did just this on the next flight the INS failed.

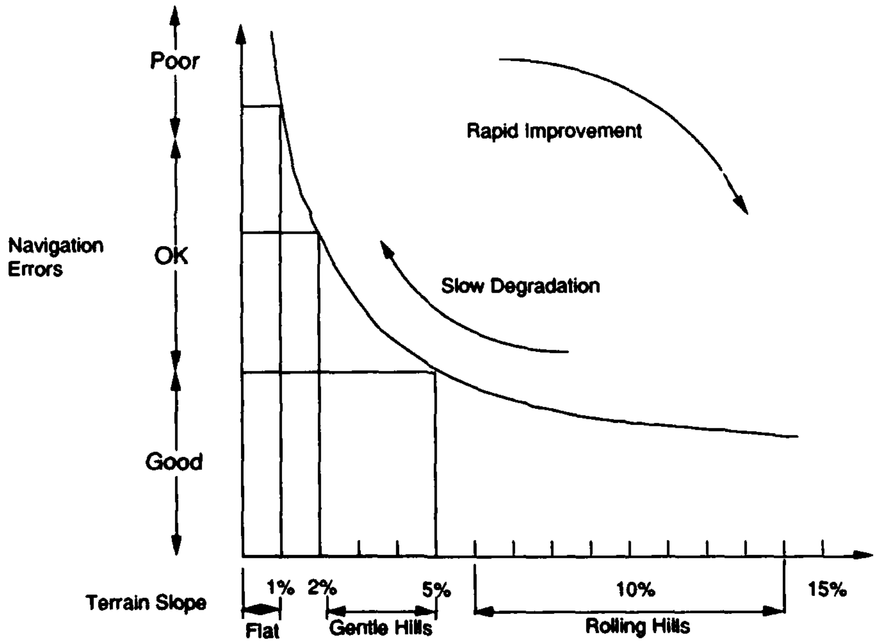


Fig. 4 TERPROM Navigation Performance

Terrain Following

Terrain Following Radars (TFRs) have drawbacks in modern warfare. The radar advertises the aircraft's presence especially if the aircraft manoeuvres fiercely and in any event, manoeuvre limits are imposed due to the mechanical constraints of the radar antenna. TERPROM terrain following (TF) is covert (you can even switch the radalt off if you wish) and is regularly used in full air combat manoeuvring.

The algorithms used by TERPROM for TF vary from those typically used by TFR in straight flight - these being useful to cross monitor a TFR and TRN system in peacetime operations - to those which take account of over-the-hill information to promote a more terrain hugging profile in rough terrain. Both types have been successfully flown in the TERPROM system on the F16 aircraft.

Fig 5 illustrates the principle of TERPROM TF. A mathematical beam is projected in front of the vehicle. In plan, the size and orientation of the beam is related to the expected cross track error (as continuously predicted by the Kalman Filter) and the projected flightpath. In elevation, the terrain and obstacles are mathematically lifted to take into account map errors and spread longitudinally for along track errors and combined to create a worst case profile. The vehicle's flight profile will depend on the clearance height set by the aircrew or missile mission planner.

One spin-off of a mathematical TF system as opposed to one based on radar returns is that it is unaffected by terrain cover (e.g. jungle or snow) or by flying height. (TFRs are limited to line of sight which can be short at very low level and they are adversely affected by low grazing angles.)

This means that operational experience and confidence is built up very quickly so reducing system certification times and pilot training. Indeed it is not unusual for guest pilots on the F16 to be flying as low as 100 feet at 600 knots and at night on their first flight with TERPROM as the cornerstone of a suite of navigation aids.

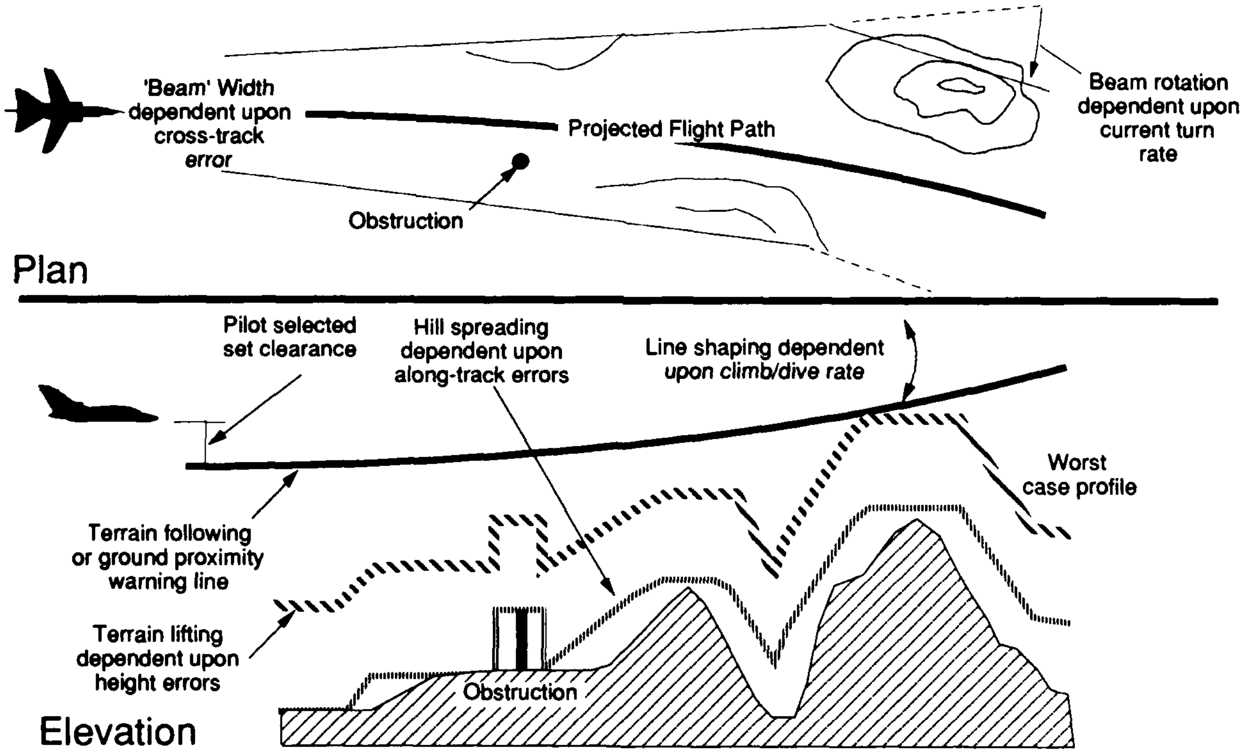

'Beam' Width dependent upon cross-track error
Projected Flight Path
Beam rotation dependent upon current turn rate
Obstruction
Plan
Pilot selected set clearance
Hill spreading dependent upon along-track errors
Line shaping dependent upon climb/dive rate
Worst case profile
Terrain following or ground proximity warning line
Terrain lifting dependent upon height errors
Obstruction
Elevation

Fig. 5 TERPROM Terrain Following

Terrain Masking

Horizontal and vertical guidance in current TERPROM systems are uncoupled. Whilst each has been mechanised automatically by feeding TERPROM commands to the aircraft's autopilot, they can also be displayed to the pilot for him to interpret. Algorithms have been developed which combine horizontal and vertical commands and choose a TF route to a specified way point which makes best use of terrain screening **(see Fig 6)**.

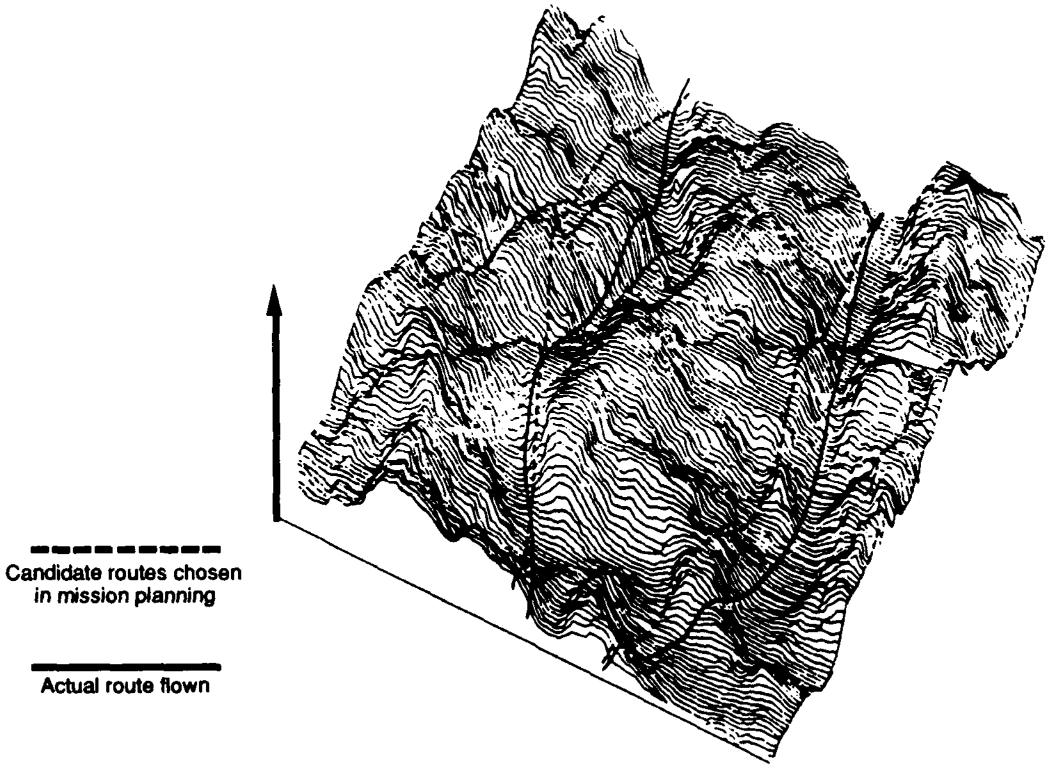


Fig. 6 TERPROM Terrain Masking

There is still considerable ergonomic work yet to be done to determine the best way information can be presented to the pilot to avoid increasing his workload or overwhelming his display whilst keeping him informed of the intended track. As a first step, an aircraft has been guided automatically using an algorithm which is computed partly on the ground in mission planning and partly in real time in flight.

Ground Proximity Warning

Traditional warnings based on radalt readings are satisfactory when the terrain is flat or gently rolling and when the aircraft is not manoeuvring fiercely enough to prevent a radalt reading. A much more flexible and reliable ground proximity warning is provided by using map data to predict approaching hills either directly ahead or into a turn **(Fig 7)**.

Moreover, by using the navigation system's accurate height estimate, rather than directly using the radalt, visual or audio warnings can be given whatever the height or attitude of the aircraft or the 'g' force applied. Those of you who have spent time at 500 or 600kts 'in the weeds', looking over your shoulder for other formation members, for the target, or for the bounce will readily appreciate the value of a predictive intelligent ground proximity warning.

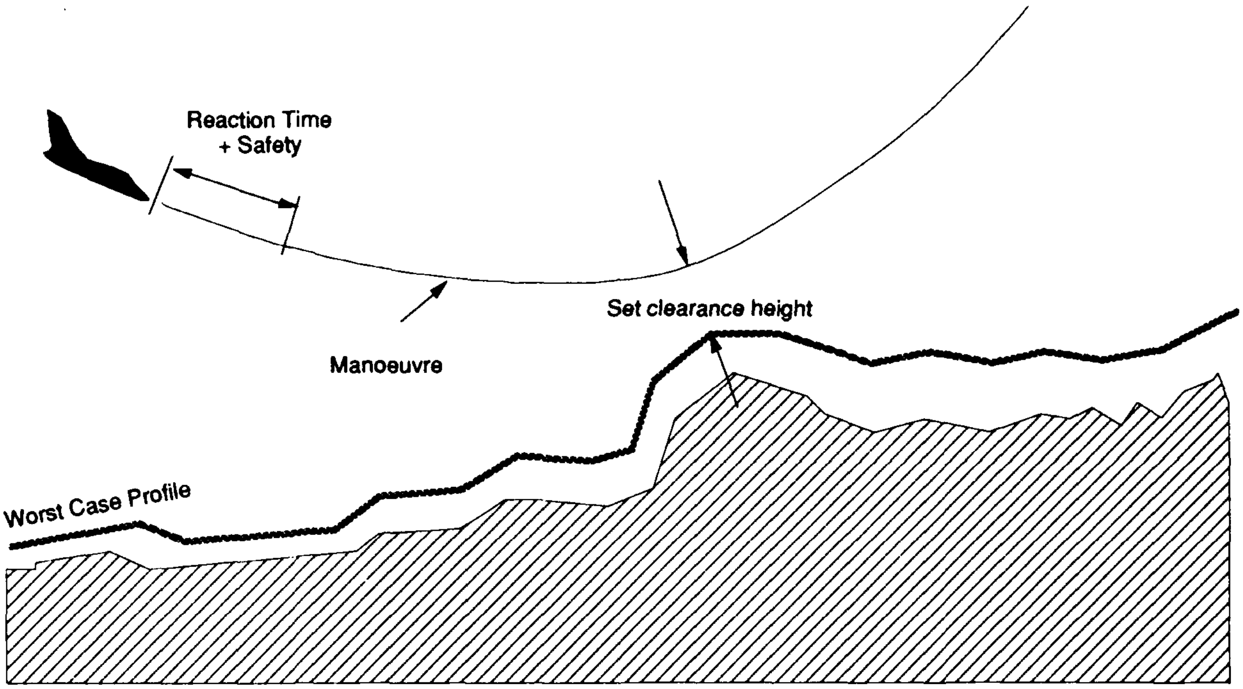


Fig. 7 TERPROM Ground Proximity Warning

Obstacle Warnings

In its simplest form, an obstacle avoidance warning can merely warn of an obstacle directly on track but this can be extended to advising of obstacles to right or left so as to avoid turning into them.

More complex still is the ability to overlay a warning marker on an image of the obstacle in the HUD although, as with terrain masking information, there is much ergonomic study yet to be carried out in this area to avoid swamping the pilot with information when he is searching for a target in a highly industrialised area for example.

Criticisms have been levelled at TRNSs due to the limited availability of obstruction data. Ten years ago, similar criticisms were voiced about the quality of terrain data but in that time TERPROM has been tuned to work with all qualities of map data, and the mapping agencies have improved the quality of their products.

Three classes of DLMS are used. Digital Terrain Elevation Data (DTED) is used for ground profile, Digital Feature Analysis Data (DFAD) is used for features such as towns and Digital Vertical Obstacle File (DVOF) is used for obstacles. It is the improving quality and quantity of DFAD and DVOF which will make it possible to fly lower and lower with TERPROM. There will continue to be a requirement for some form of active obstacle warner such as laser or other radar for nap of the earth low flying, but TERPROM will be an important complementary system.

Target Acquisition and Weapons Aiming

The current debate over the type of aircraft required to perform the close air support role in the USA has highlighted the difficulty of pilots locating camouflaged, mobile targets in the battlefield. At high speed and low-level there may not be enough time to detect, acquire and classify the target while manouevring the aircraft into a tracking and firing position.

A speedy method of transferring target information from a Forward Air Controller (FAC) to the pilot is for the FAC to pass the location via an Automatic Target Hand Over System (ATHS) to the attacking aircraft so that the target is indicated by a marker in the Head Up Display (HUD). Another approach is for the lead attack aircraft to acquire the target using a steerable FLIR. As he looks at the target he can fix it in azimuth and elevation. Range information would normally be provided using an active laser rangefinder or radar, but the point here is that TERPROM provides the target coordinates or range passively, without any forward emission from the aircraft. The ATHS automatically transfers the target position to the remainder of the formation who can then attack with virtually real-time information.

To acquire fixed targets, the coordinates are usually inserted into the navigation computer before the attack and an acquisition aid such as a cross or box is displayed in the HUD.

Unfortunately, due to INS drift, an unaided system needs to be updated manually and accurately immediately prior to the target run which creates a significant extra workload when the pilot or crew least need it. TERPROM not only removes this workload, but ensures that the aircraft is exactly positioned at the start of the target acquisition phase.

As far as weapon aiming is concerned, TERPROM will feed the weapon aiming computer with passive three dimensional range information to the target and feed this to the HUD for pilot interpreted release or for automatic release **(Fig 8)**. Traditional weapon aiming computers are unable to forecast the terrain ahead since they are fed information directly from the radalt reading below the aircraft. TERPROM will indicate correct bomb impact point even on the side of a hill by working cooperatively and iteratively with the computer.

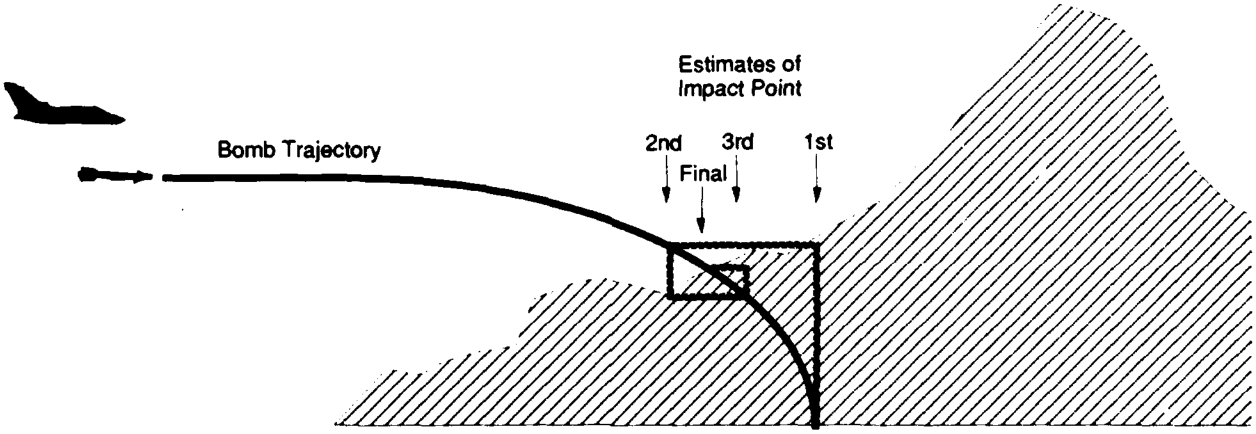


Fig. 8 TERPROM Weapon Aiming

TRIALS RESULTS

TERPROM algorithms began to be developed in 1975 with the aim of providing the Long Range Stand Off Missile (LRSOM) project with an autonomous navigation update system **(Fig 9)**. Initial flight trials were carried out on Jaguar and Comet aircraft, but the main trials aircraft was a Jetstream belonging to the Cranfield Institute of Technology. It was fitted with an INS, a continuous wave TRT radalt and a downward looking Vinten camera to determine accurately the aircraft track. The track was estimated using the TERPROM algorithms and compared with the photographs.

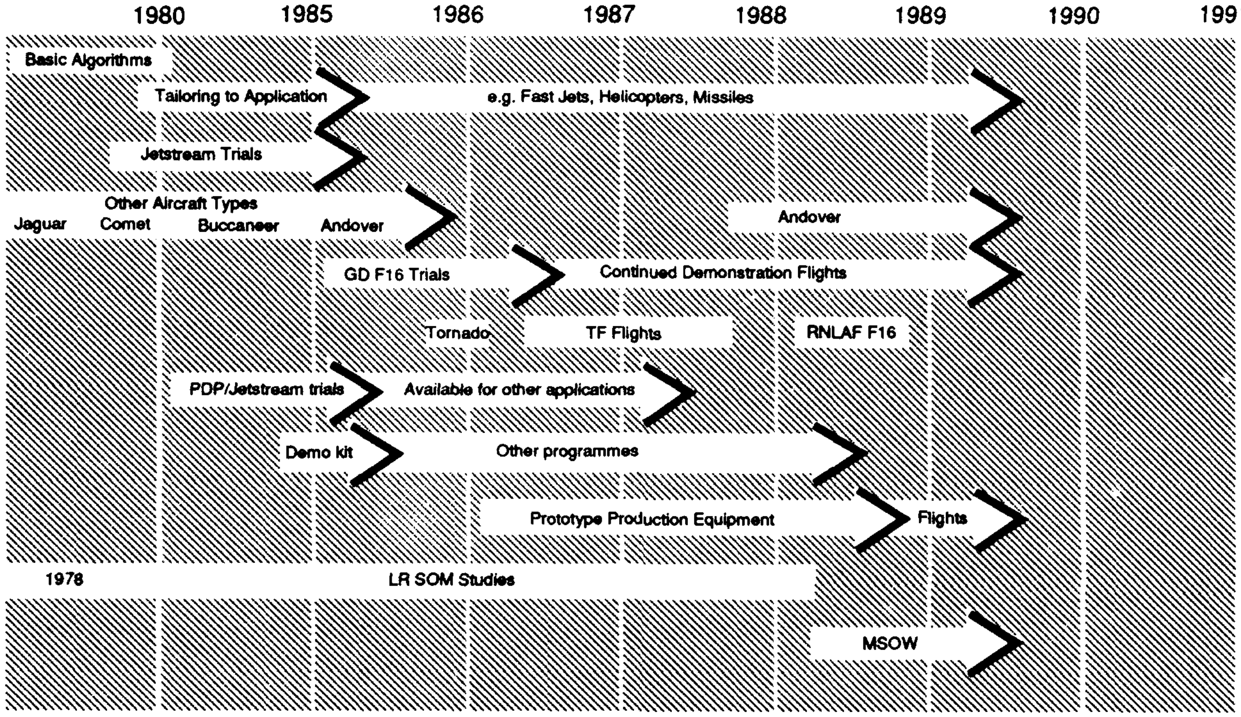


Fig. 9 TERPROM Development Activities

In 1980, a second, pulsed radalt, was also fitted. Although both worked at 4.3 GHz, they responded differently to cultural features such as houses, trees and power lines.

Real time trials began in 1981. Pre flight, a planned route and height profile was fed from a small cassette tape into a magnetic bubble memory in a dedicated computer on the aircraft. At first, guidance commands were fed to an instrument landing system type display for the pilot to follow in azimuth and elevation. In July 1982, however, a parallel set of commands was fed to the aircraft autopilot and the aircraft flew with TERPROM in full control.

BAe built in 1984 a ruggedized computer system with a variety of interfaces to allow it to be fitted to a range of aircraft types.

In 1985, the system was delivered to General Dynamics Fort Worth Division for demonstration trials on the F16. The aim of the trials is to evaluate low cost navigation and night systems to demonstrate a more versatile low altitude night attack capability on Close Air Support and Battlefield Air Interdiction (CAS/BAI) missions. Hundreds of sorties have since been flown, terrain following down to 100ft.

In the General Dynamics magazine 'Code One', senior test pilot Joe Bill Dryden says of TERPROM, with which he has been flying since that first F16 flight 4 years ago 'The accuracy, without the requirement to make any updates, regardless of the length of the mission, is astounding. The reduction in cockpit workload is staggering.' And his advice for choosing systems for night operations 'It goes without saying that TERPROM should come first on everybody's list! With such a system you can do everything better, day or night, rain or shine, regardless of what your mission is.'

In March 1989, Robert Ropelewski reported of his sortie in the GD F16 in Armed Forces Journal International that, 'Throughout our 1.7hour flight in the F16, the TERPROM system kept us within 20 to 30 metres of our preselected course..'- and this was over the relatively flat and not particularly well mapped Texan countryside.

During 1985 and 1986, trials on the Tornado aircraft from BAe Warton were carried out in conjunction with the MOD Aeroplane and Armament Experimental Establishment at Boscombe Down in Wiltshire. The trials were very successful. Not only did they demonstrate excellent navigational accuracies and the ability of TERPROM to reduce significantly the height channel errors in the BARO/IN mode, but extensive TF and automatic TF sorties were flown using TERPROM.

The Royal Netherlands Air Force (RNLAF) observed these trials with increasing interest and in 1987 used TERPROM to enhance their own F16 Night Falcon 2 project. They confirmed that Forward Looking Infra Red (FLIR) systems and Night Vision Goggles (NVGs) are not always very effective in poor European weather. Cloud, fog and rain adversely affect these systems whereas TERPROM would aid target acquisition and provide a passive TF facility. In 'Defensiekrant', test pilot Major Tom Bakker of the Tactical Air Command in Zeist describes TERPROM as 'the biggest step forward in the last ten years.... This system is so accurate that hardly any margins remain..... The big advantage of this system is that it not only offers improved navigation but also a terrain following capability which GPS cannot offer.' In the same article, Major Dick Berlijn, Head of Weapons Systems Department, Describes his flight as 'A fantastic experience. The workload related to accurate navigation drops considerably. Also, my weapon system is improved as the computer always knows exactly how high I am above the [target], that produces an incredible accuracy when I drop my bombs. Conclusion the total [effectiveness] of the F16 improves.'

In late 1988 and early 1989, trials took place on a Sea King helicopter using two different grades of INS. At the time of writing, the results are still being analysed but initial indications are that they were fully successful.

EQUIPMENT DEVELOPMENT

Developments in the electronics field have meant that a TERPROM system can now be housed in a very small space, even though the software to perform all the computations for navigation, TF etc is considerable. The current system, whose software is written in ADA, is housed in a 3/4 ATR unit - about 12 litres in volume - and uses two 1750A architecture processors, a 1553B interface and numerous incidental interfaces for tailoring to the avionics fit and for communicating with ground loading and test equipment. The store for the map data is made up of 4 modules each of 4 megabytes of Electrically Erasable Programmable Read Only Memory (EEPROM); each module is capable of holding over 50,000 square nautical miles (150,000 sq kms) of terrain and obstruction data. EEPROM has been chosen for its non-volatility, reprogrammability and maturity, making it the ideal technology for a self-contained TRN system. Also, flight tested in 1987 was an optical disc system holding the map data for TERPROM. Whilst currently there are environmental limitations on optical disc systems, they have the potential to achieve an order of magnitude improvement in storage density.

QUANTITATIVE IMPROVEMENT

Quantifying TERPROM's advantage over existing navigation systems is difficult but its main strength on aircraft lies in improving their night and adverse weather capability.

Penetrating to targets using autonomous, stealthy techniques is only possible at the moment in good weather and in daylight.

For what proportion of the time would TERPROM permit covert navigation and attacks which would otherwise not be possible unless an active system were used?

Good weather means one thing to the F16 pilot over the Mojave desert where any cloud at all or visibility less than 20 miles may be unusual. It means something else to the Tornado or AMX pilot fighting his way across the North German plain in winter.

So the definition of good weather is debatable but let us assume that we are leading a formation of 4 aircraft against a target which requires a closely coordinated attack. The area is heavily defended and you want to penetrate and attack with minimal transmissions, not using TFR.

You really need 300ft clearance from cloud to ensure that you stay visual with the others and if we assume an average penetration height of 200ft, we would require a minimum cloud base of about 500ft. To acquire and attack the target successfully may need a visibility of 5km.

With reference to **Fig 10** we can see that the joint probability of this combination existing over the North Germany in winter is 60 per cent. It is therefore probable that for 40 per cent of the time in winter, TERPROM would be likely to permit covert operations when at present they are only possible with an active insecure TFR system. In summer, the improvement in operating capability would be smaller but significant. If we were to overlay on this the hours of darkness you can see what major benefits TERPROM provides.

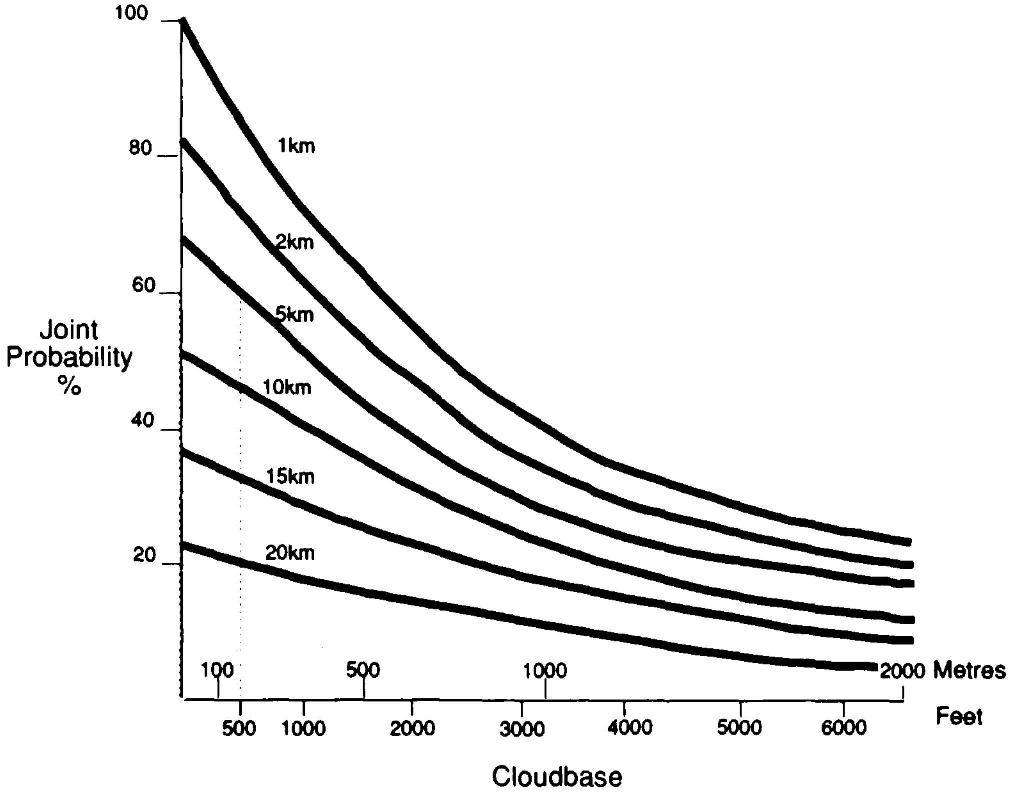


Fig. 10 Joint Probability of Occurrence of a Specified Cloudbase and Visibility in the Central Region in Winter

TERPROM DEVELOPMENTS

Although extensive TERPROM trials have demonstrated excellent performance over a wide range of terrain, developments have taken place to improve its navigation performance, especially over extensive areas of flat ground **(Fig 11)**. However, it must be borne in mind that while enhancements which rely on additional sensors may make the system more accurate, they tend to increase cost and system complexity.

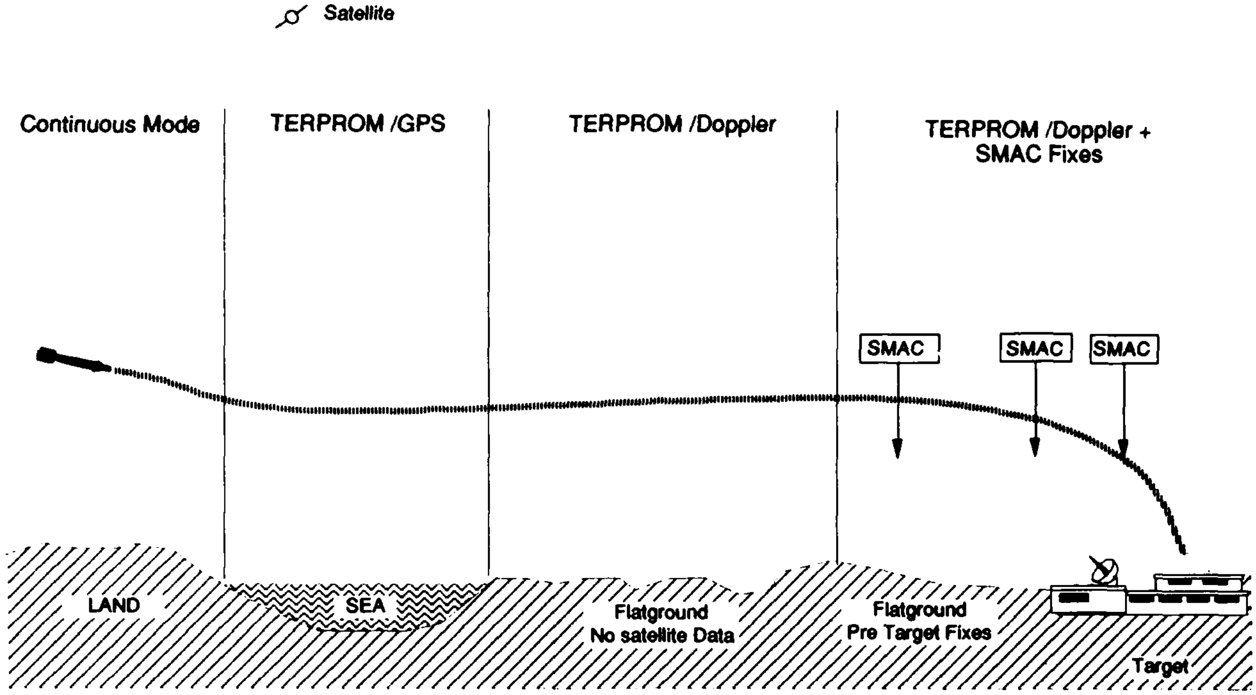


Fig. 11 TERPROM Enhancements

TERPROM/Doppler

One of the earliest extensions to basic TERPROM was the supplementing of radalt data with velocity measurements from a Doppler radar. As with the radalt in the normal Continuous Mode, the Kalman Filter now can also predict vehicle velocity and compare it with the actual Doppler radar output to enable estimated quantities of position and velocity etc to be corrected. The filter processes the radalt and Doppler information independently so that over well featured terrain the Doppler radar may be muted whilst the radalt information is used. Over moderately undulating terrain, both inputs can be used independently and asychronously to provide the best information. Over flat ground, the Doppler radar supplies all the information to reduce the drift which would exist if only the radalt were used. This enhancement has been tested against many sets of flight data.

TERPROM/SMAC

Another way of improving the system is to use other sensors to supplement or replace Doppler radar. One of these is infra red (IR) SMAC which is applicable mainly to long range missiles. It preserves the autonomy of basic TERPROM and, unlike Doppler, is covert. By comparing the IR image of a way area with a stored digital 'map', the position of the vehicle can be established very accurately. The positional information is passed to the Kalman Filter where it is combined with the radalt- derived navigation data.and is processed in much the same way as the Doppler measurement.

The particular advantage of this combined system is that, whereas TERPROM performance is best over well-featured terrain, SMAC operates at its best over flatter terrain, where there tend to be more recognisable linear features such as roads, rivers, railway lines and field boundaries. Also, flat ground will not distort the captured IR image.

Flight trials have shown that the combined navigation performance is better than that of either TERPROM or SMAC alone and in particular it is more robust against defects of the digital data bases of either system.

TERPROM/GPS

Similarly, GPS can be successfully integrated with TERPROM. In many ways, these two systems represent an ideal combination. Over water and flat terrain ,TERPROM tends to drift but GPS does not suffer from satellite screening and so the optimum constellation may be chosen. In hilly terrain, satellites may be screened from a low flying vehicle but here TERPROM's highly accurate vertical channel enables the integrated system to choose the satellite constellation which provides the best horizontal navigation performance rather that that which gives the minimum Geometric Dilution of Precision (GDOP).

Overall, by using a variety of sensors, a composite system can be provided which accurately navigates at all times.

Alternative Forms of TERPROM

One of the major requirements of the original TERPROM system is an accurate and reliable INS. However, INSs tend to be heavy, large and expensive, and for unmanned air vehicles (UAVs) and other applications such as reversionary aircraft systems, an alternative dead reckoning may be all that is available. BAe has recently developed other forms of TERPROM which can be used with lower cost dead reckoning systems such as Doppler/Attitude and Heading Reference System (AHRS), an Air Data System (ADS)/AHRS or with ADS/Gyro Magnetic Compass (GMC).

In tests against data recorded in flight, the systems were shown to navigate well, although not quite as accurately as TERPROM and INS because error sources such as wind are far harder to characterize. However, accuracies were adequate to identify waypoints, targets and airfields and to terrain follow.

In general, adding new sensors means that new system errors have to be calculated and the Kalman Filter redesigned accordingly. However, British Aerospace have designed a new algorithmic architecture in which models of each sensor, including the INS, are treated as measurement models peripheral to the main system model. Now new sensors can be added or deleted with ease and even whilst the filter is running, sensor measurements can be processed or ignored so allowing failures to be tolerated. Details of the filter were presented by BAe to the American Institute of Navigation in 1988 (Ref 2).

Whilst the system has not yet been flight proven, it has been successfully tested against both simulated and flight data input from numerous sensors.

CONCLUSION

Having now accumulated hundreds of hours of flying trials experience, it is clear that TERPROM is a mature and robust terrain reference system which provides a range of operational benefits to manned and unmanned aircraft. All pilots who have flown the system have enthused about the concept and its implementation.

The most widely publicized feature is its covertness and its ability to update accurately an INS, but of growing importance are its other benefits such as TF, intelligent ground proximity warning and improved target acquisition which have been demonstrated on hundreds of flights. Recent developments include TERPROM's ability to work with a range of low cost navigational systems and the design of a new algorithmic architecture which allows the addition of any number of update sensors for improved performance and to cope with sensor failure.

REFERENCES

1 R S DALE. 'TERPROM - Automatic Precision Navigation and Guidance in All Weather'. AGARD Conference Proceedings 387 - May 1985.

2 A J ROBINS. 'Recent Developements in the TERPROM Integrated Navigation System'. American Institute of Navigation 1988.

POSITION FIXING BY DETECTION OF ROADS AND RIVERS

by

Prof. Dr.-Ing. E.Mühlenfeld
Inst. f. Elektr. Informationstechnik, TU Clausthal
Leibnizstr. 28
D-3392 Clausthal-Z
Federal Republic of Germany

Summary

An image sensor, looking straight downward from an aircraft, detects line-shaped patterns of roads, rivers or railroads with a detection rate of 75%. Combining these measurement with stored map data on these traffic routes and with INS data yields reliable position estimates within ± 8 m.

List of Symbols

ε : probability, that the detected contour has not been stored in the map.

W_{pred} : predicted distribution.

W_{meas} : *measurement distribution.*

$W(\underline{x}/\underline{y})$: a-posteriori-distribution of the actual position $\underline{x}$ when $\underline{y}$ has been measured.

c_1 : the overal distribution of $\underline{y}$.

1. Introduction

The goal of this paper is autonomous position fixing, independent on cooperative devices on earth or in space. There has to be some kind of INS-system on board, which must not necessarily have the precision of an inertial system, though we need a gyro as an angular reference, so that angles can be measured and compared in earth coordinates. The position indication of any such system requires occasional updating. For autonomous position fixing in the two dimensions of the earth's surface we need a two-dimensional map, containing a measurable local feature z as a function of position coordinates x and y. Any function z(x,y) can generally be considered as an image, and position is fixed from the peak of some kind of correlation function between the measured image and the map-image. To avoid ambiguities, z(x,y) should be rather unique and identifying for a local neighbourhood. A sharp peak of the correlation function is obtained for best accuracy, if high spatial frequencies are contained in the local variations of feature z.

In many areas spatial variation of local altitude, used as feature z, is not very identifying, especially in its high-frequency content. But the amplitude distribution z(x,y) of the local emission or reflection of electromagnetic waves in the IR, visible or radar wavelength-region mostly containes sharp, time-invariant image elements. Line-shaped patterns are selected as elements for an image-supported navigation:

- Traffic routes, such as roads, rivers and railroads.

Any one of these patterns is not identifying the position by itself. But if their geometrical context is compared to the context of routes, stored in a digital map, we get an unambiguous and precise position fix.

To this end we had to solve two tasks:

1) Traffic routes have to be detected and data on their absolute direction and their position relative to the vehicle have to be acquired.

We solved this task for airborne navigation using the visible region of the electromagnetic spectrum:
An image-processor looks straight downwards and uses all available a-priori-knowledge to adapt its electrooptical preprocessing and digital filtering for optimal detection of line-shaped patterns with weak contrast in spite of local disturbances in highly structured background.
Such a Contour-Sensor has been tested inflight under various wheater-conditions. It is now commercially available for measuring edge-positions for a variety of tasks in the control of production processes and automated visual inspection. It is also used for the acquisition of map-data from road-maps or aerial photographs. So I would like to introduce the principle of this sensor independently from our particular aim of image-supported navigation later on.

For our second task

2) Position fixing from the spatial context of line-shaped patterns

contour-data might be acquired using other regions of the visual spectrum or by completely different means. In a land-vehicle the system might be told the direction of a traffic route, which the vehicle is using or crossing, by manual input, for example.

If we know, that we are on a traffic route of a particular direction, we can look into the map for routes of equal direction and know that we are somewhere on one of these lines: we have a line-shaped probability-distribution of our position. More routes, detected subsequently, yield more probability distributions, which have to be merged somehow, to obtain the actual position at the point of maximum probability. The probability-density-functions can be stored and merged as twodimensional image-functions, or they can be described by statistical parameters, which can be handled mathematically in the way of a Kalman-Filter, to reduce computational requirements.

Experimental results on the reliabilty of this method of position fixing will be presented.

2. Acquisition of contour data by a contour sensor

The common procedure for image processing starts by bringing the entire image into a computer after scanning it in a fixed raster. This yields two problems:

- a large amount of information has to be transferred, stored and processed.
- sampling noise is generated by any fixed raster converting a smooth contour into a staircase.

A finer sampling raster reduces sampling noise but increases computer workload.

As relevant information for image analysis and classification is mainly contained in image contours, we developed an electrooptical preprocessor, called 'contour-sensor', with a sampling pattern, adapted to line-shaped image contours. Contours are defined rather simply:
" A contour is a line shaped pattern, formed by the maxima of the gradient of average intensity (or other statistical moments).

Averaging on either side of a contour should cover as wide areas, as possible, to eliminate drop-outs and other local disturbances of the pattern and its background. On the other hand it should not extend to regions outside the homogeneous image areas. In our case the inner area of a traffic route is very narrow and limited by it two contours. The sampling area itself thus must be narrow and line-shaped.
If we know neither direction nor position of the contour, which has to be detected, the line-shaped sampling-pattern must be rotated and shifted either sequentially or in parallel by an array of line-shaped sampling-patterns. As the aircraft is moving continuously there is a natural translational shift. The star-shaped sampling pattern of fig.1 is formed by the surfaces of photosensitive semiconductor elements and is adapted to different directions in a parallel mode [1].

Output data represent integrals over radial paths, corresponding to low-pass-filtering in radial direction. By this way of scanning, the two-dimensional light-intensity-distribution has been converted to a one-dimensional array of sensor-data. This array can be conveniantly filtered by appropriate digital algorithms, matched to the contour-profile. As the difference between output data yields the gradient of average intensity, this filtering will have a highpass characteritic, essentially. Radial integration by the shape of sensor elements and tangential differentiation by digital filters combine to form a two-dimensional unisotropic filter.

> It is not necessary for the image of a line-shaped pattern to cross the center of the sampling star-pattern precisely. Depending on contrast and local image defects, contours are detectable even if the star-center has a distance of up to half the star-radius from the contour [2]. When the aircraft crosses a traffic route, it can be detected several times with a scanning period of 20 ms and velocities less than mach 3. Computation yields several samples of position data of the same route. The probability distribution describing these measurements is gaussian-like in the orthogonal, and uniformly distributed in the direction of the contour. It can be visualized as a dam with gaussian profile and forms a base line for navigation, the width of which corresponds to the variance of the measurements.

3. Flight test of contour detection

The contour-sensor-system was mounted in a DO 228 aircraft of the DFVLR in Oberpfaffenhofen (FGR) looking straight downward through a hole in the fuselage together with an ordinary video camera. Its images were recorded together with detected traffic-route-data. These data were evaluated in the laboratory by comparing them with photographic and strategic maps.

We had 7 flights in summer and in 2 winter in an altitude of 1000 m. Flighing in an altitude of 30 m demonstrated low-level capability. The visible spectrum did not permit operation above clouds or fog. For later feasibility studies on the all-weather-capability of the sensor principle, radar data from a SLAR system of DFVLR were recorded during the same flight.

Not all traffic routes were detected, of course, because some of them were hidden by forrests or happened to be bordered by an acre of the same brightnes: Detection rate was 75 %.

Due to randomly generated line-shaped patterns, like traces of farm carts or rests of snow in a ditch, ε = 7 % of the detections did not correspond to map data.

4. Navigation updating

Any system supplying data on traffic routes will omitt some of them, which only means, that they are just not available for a position update. And any system will come up with routes which are not contained in the map. Those routes might be confused with nearby routes of the same direction. But this is no particular problem of unknown detected routes, as for any detected route there might be more than one route of corresponding direction in the map, and it could therefore be confused with a wrong route. So the matching algorithm has to cope with wrong or multiple matches, anyhow.

It is not necessary, but for the explanation of the principle let us assume, that we start from a well known position. Our initial probability distribution, depicted over the two position coordinates thus is a δ-puls. As our vehicle proceeds, an INS-system attempts to predict the actual position, but taking its drift into account, the

predicted distribution W_{pred}

will be a gaussian fuction, its variance increasing with time.See fig 2.

If a traffic route is detected, this yields position data, taken from a route of equal direction in a map. Its distribution, the

measurement distribution W_{meas}

has the shape of a dam with gaussian profile according to chapter 2, if there is one and only one corresponding traffic route in the map.

For optimal estimation of the actual position $\underline{x}$

$W(\underline{x}|\underline{y})$, the a-posteriori-distribution of the actual position $\underline{x}$,
when $\underline{y}$ has been measured,

is computed from Bayes'equation

$$W(\underline{x}|\underline{y})\ W(\underline{y}) = W(x,y)=W(\underline{x})\ W(\underline{y}|\underline{x})$$

with

$W(\underline{y}) = c_1$, the overal distribution of $\underline{y}$, which is independent of $\underline{x}$,
so that it does not influence the decision for a particular
$\underline{x}$ and can thus be considered as a constant,

$W(\underline{x},\underline{y})$, the joint distribution,

$W(\underline{x}) = W_{pred}$, the a-priori-probability before $\underline{y}$ is measured, and

$W(\underline{y}|\underline{x}) = W_{meas}$, the measuring-distribution of $\underline{y}$, when the actual
position is $\underline{x}$.

We thus get the a-posteriori-distribution

$$W(\underline{x}|\underline{y}) = W_{pred}(\underline{x})\ W_{meas}(\underline{y}|\underline{x})\ /\ c_1\ ,$$

and can take its maximum, for instance, as position estimate. It is then shifted by the INS-system according to the vehicles further movements and convolved in fixed intervals with a gaussian function, corresponding to low-pass-filtering in the spatial domain, to take the INS-systems drift into account. After this treatment it is used a a-priori-distribution, as soon as a new measurement is available.

ε is the probability, that the detected contour has not been stored in the map and thus is useless; ε was 7 % in the flight test. Thus with probability ε the measurement $\underline{y}$ is useless and the actual position $\underline{x}$ is equaly distributed with constant probability densitiy c_2, while Wmeas has to be weighted with $(1-\varepsilon)$ to yield

$$W(\underline{y}\,|\,\underline{x}) = \varepsilon\ c_2\ +\ (1-\varepsilon)\ W_{meas}(\underline{y}\,|\,\underline{x})\ .$$

Using this expression in Bayes' equation,

$$W(\underline{x}\,|\,\underline{y}) = W_{pred}\ [\ \varepsilon\ c_2\ +\ (1-\varepsilon)\ W_{meas}\ ]\ /\ c_1\ .$$

For a more careful deduction of this result using the methods of Kalman filter theory, see reference [3]. If both W_{pred} and W_{meas} are gaussian, $W(\underline{x}\,|\,\underline{y})$ is gaussian, too.

If the detected route has a match in the map which is far apart from the predicted position, $W_{pred}(\underline{x}) \cdot W_{meas}(\underline{y}\,|\,\underline{x})$ becomes very small. We therefore only look for matches in the map within a distance of 3 σ. If there are more than one matches in the map within a distance of 3 σ, Wmeas is a multimodal distribution of several parallel gaussian dams. In this case $W(\underline{x}\,|\,\underline{y})$ is a multimodal gaussian distribution. This can only happen in urban areas with a dense net of traffic routes, provided that recent position fixes have rendered prediction rather accurate with a small σ. Urban areas are therefore excluded from position fixing, assuming that there are enough routes in the suburban surroundings. After a long flight without position fixing, only, multimodal distributions might have to be coped with, as Wpred than will be very flat before the first new fix.

There are two ways of computing $W(\underline{x}\,|\,\underline{y})$:

1) All distributions are taken as normal or gaussian, respectively, and are described parametrically by its statistical moments.

If the system has to cope with multimodal distributions, some of them have to be combined and others have to be omitted, to reduce computational workload and eventually yield a monomode distribution again.

Modern digital electronics allows a new approach:

2) Probability distributions are stored as two-dimensional functions in and image processor and convolved, added or multiplied pixel by pixel.

This does not require any approximation of real distribution functions by mormal, preferably monomode distributions. It allows to store all a-priori-knowledgle, obtained sofar, in a fractured probability mountain and to exploit it accordingly.

The first approach has been choosen for experimental verification of reliable performance.

6. Experimental performance of update-simulation

Experimental simulations of position fixing were performed on a computer system with a given random drift of the INS-system and the measuring-distribution, which had been observed in the flight tests [3]. The simulation was based on worst case data:

- maximum random drift 0.3 o/h; any low-cost inertial system is better;

- detection rate 60 %, which was 75 % in the flight test;

- 10 % of the detected contours are not to be found in the map (7 % in the flight test);

- random distance of traffic routes 400 m, which is typical for central Europe.

Under these conditions

+ position fixing was absolutely reliable and was never disturbed irrevocably (fig.3) and

+ the error of the position estimation was never greater than 8 m.

References

[1]. Therburg,R.-D. , Image-Supported Navigation; Localization and Orientation in Biology and Engineering ed.
by Varju/Schnitzler, Springer Verlag Berlin Heidelber1984, Page: 307-318.

[2]. Mühlenfeld E. Prof.-Dr.-Ing., Fricke U. Dipl.-Ing., Grafikprozessor beschleunigt Bildverarbeitung, Elektronik 4/17.2.1989

[3]. Rehkopf A., Simulation und teilweise Erprobung eines kontugestützten Navigationssystems bei Verwendung von unimodalen und multimodalen Verteilungsdichten im Kalmanfilter ,Diplomarbeit TU Clausthal 1988

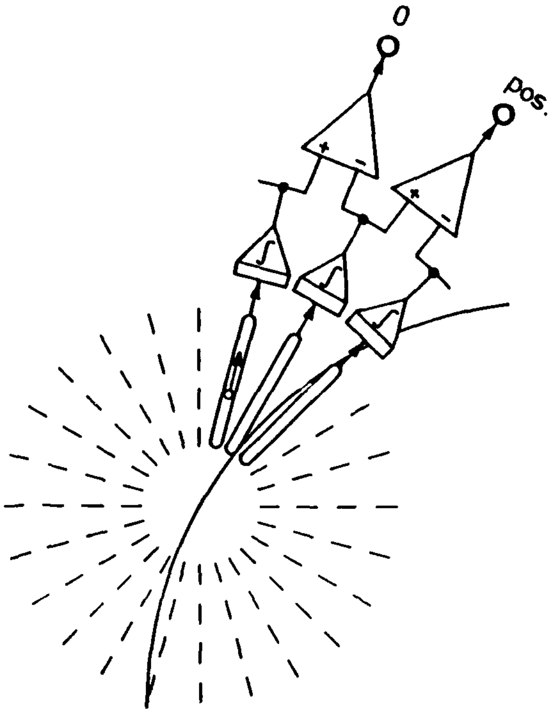
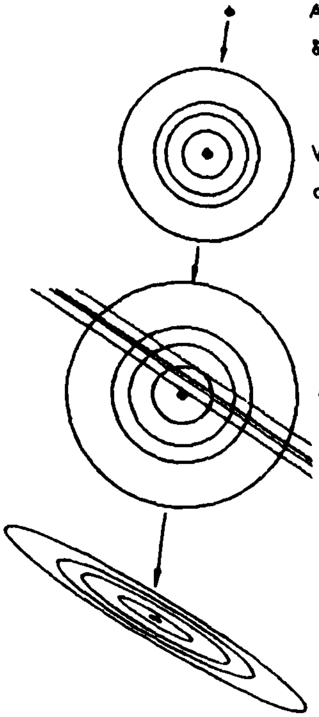


Fig. 1: Scanning pattern



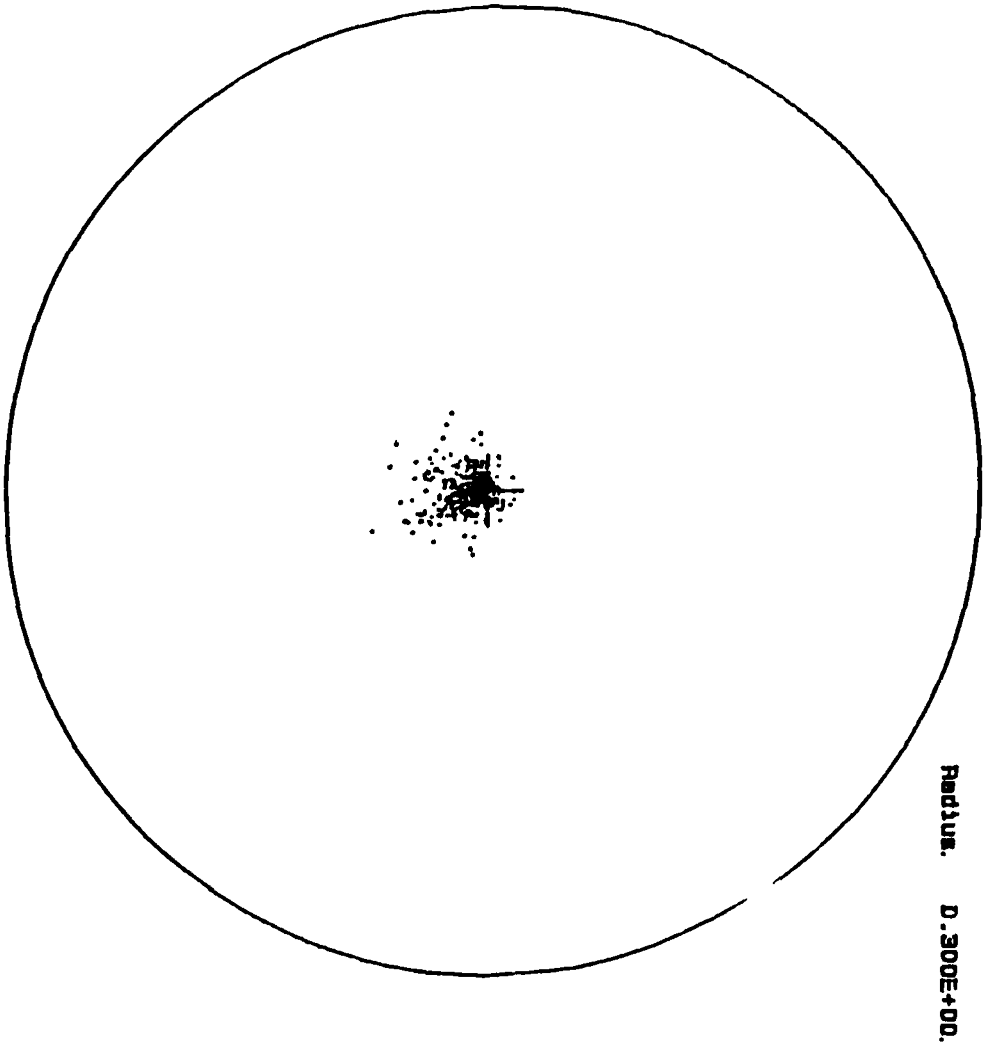
At the start, position-estimate has a δ-distribution of probability.

While travelling, the distribution is spreading due to drift of the INS.

Crossing a traffic route and retrieving its position from a map yields a gaussian dam as measurement distribution.

Merging of both distributions generates a new gaussian distribution of position.

Fig. 2: Position fixing



```
*********** End-Statistik *************
 <Rw-Rs>x =  -2.3425826E-04
 <Rw-Rs>y =   1.1895980E-02
--------------------------------------
 <(Rw-Rs)'*(Rw-Rs)> :
 1.1092079E-04 -1.7113392E-05  2.9561837E-04
--------------------------------------
 < P > :
 1.1629071E-04 -5.4891339E-06  1.2249120E-04
```

Fig. 3: Estimated flight position

Autonomous Automatic Landing through Computer Vision

R. Schell, E.D. Dickmanns

Aerospace Technology Department
Universität der Bundeswehr München
W. Heisenberg Weg 39, 8014 Neubiberg
Federal Republic of Germany

Abstract

The automatic autonomous landing approach through computer vision has been investigated in a simulation loop with real image sequence processing hard- and software. The use of integral spatio-temporal world models is the presupposition to achieve real time performance with the microprcessors currently available. Results achieved for a business-jet aircraft demonstrate that this set up is powerful enough to solve the problem of autonomous unmanned landing approach.

Introduction

An experienced pilot has the ability to carry out a landing approach by processing information gained hrough vision only. He has learned to control the motion of the aircraft in all three dimensions in space and time by evaluating time-variant, two-dimensional image sequences gathered from his eyes. To transfer this ability to a machine, it needs a lot more computing power than available at the moment. By using expert knowledge and the laws of perspective mapping it is possible to reduce the data rate to an acceptable degree which can be handled by today's microprocessors. With these restrictions and the 4-D-world-model, developed and described in [2], a landing approach controled by computer vision only is possible.

By applying HF-technology a secure flight guidance is possible today, independent of weather. For navigation and landing approach seperate systems have been developed, but these systems need ground installations and communication between airport and airplane (ILS, MLS). For airports and planes supplied with the appropriate high performance equipment the navigation problem is thus solved satisfactorily.

On the other hand, the capability of performing an autonomous landing approach may be of advantage for smaller airports that do not possess an ILS and where the approach may be difficult because of adjacent mountains. Here an autonomous system could help to increase the safety of the approach by reducing the workload of the pilot.

A further application is the landing approach of small unmanned airplanes. So their missions can be made much more flexible and adaptable to the actual situation because the exchange of information with remotely piloted vehicles may be reduced drastically with only a minor reduction in information exploitation for control determination, then happening autonomously on board.

Problem statement

The following assumptions were made to realize the simulation of a completely autonomous landing approach by computer vision: the airplane is about 3 km distant from the runway and should have not more than 300 m altitude above ground. The deviation from the course is assumed to be small enough, that it can be controled by using the linearised equations of motion. Aerodynamic velocity is the only value measured conventionally; all the other state variables should be determined by computer vision. Based on these assumptions a landing approach shall be performed.

The landing approach has been chosen as an entry problem for aircraft control through computer vision, because the image processing is relativly simple, but the control task is rather pretentuous: with the operation of the four control variables a precision manoeuver has to be carried out in all six degrees of freedom.

A business jet aircraft was selected for the simulation, because all its technical data were available. The approach speed amounted from 70 to 60 m/s and flaps and spoilers were used too.

Simulation loop

Fig. 1 shows the simulation loop used for the experiments. A 32-bit'minicomputer (PE) simulates the dynamics of the aircraft. This is done by integrating the 12 nonlinear equations of motion every 50 msec. The three angles are then transferred to a three-axis rotational motion simulator (DBS) on which a camera platform is mounted. This platform can hold two cameras (one with a tele the other with a wide-angle lens). The viewing direction can be rotated in azimut and elevation. Thus, the cameras are able to fixate a certain point in the image.

The other result of the integration, the position of the plane according to the runway is sent to a 3-D graphics system for fast computer generated images. The view of the runway is now projected to a cylindrical screen with a radius of 2.5 m, from where the cameras can pick up the picture.

The cameras are connected to the BVV [3], the real-time image sequence processing system, where the entire state estimation and calculation of the control parameters is done.

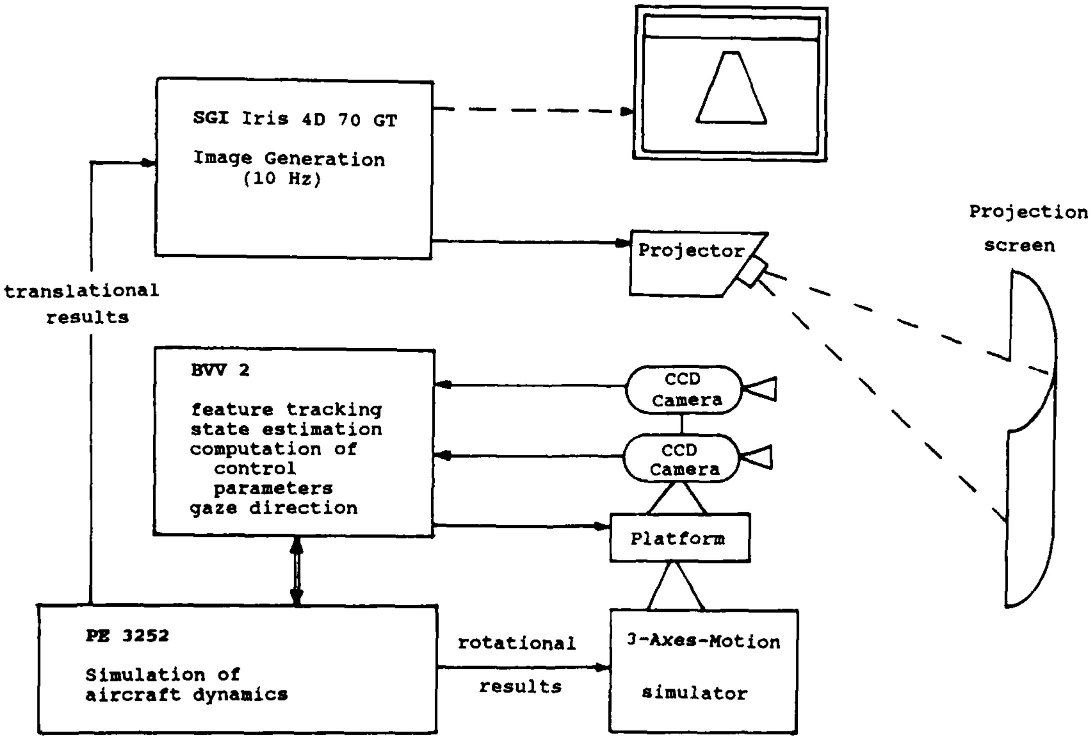


Fig. 1: Simulation loop for automatic aircraft landing approach through computer vision

Architecture of the real-time vision system

Fig. 2 shows the system architecture of the real-time vision system BVV. The video signal is digitized and fed to a video bus. Special interface hardware [3] allows n parallel microprocessors to have independent access to these signals. Each of them may accept picture elements (pel) belonging to a rectangularly shaped subimage, called window, containing up to 4 kpel, not necessarily densely distributed over the image. The position of the windows is software controlled from frame to frame, thus allowing a supervised feature tracking. Each microprocessor is initially assigned a feature by the main processor, for example the horizon or the borders of the runway.

The measured feature positions are reported to the main processor via the "system processor" which does all communication handling in the system.

The BVV image sequence processing system used has three Intel 8086 parallel processors (PP) evaluating three contour elements each in every video cycle of 16 2/3 msec; the main processor (general purpose processor GPP) consists of a 80386 32-bit single board computer. The interpretation and motion control activity runs at 100 msec cycle time (6 video cycles). The activity on the PP-level is directed by the main processor during runtime; this includes repositioning the window, changing search direction or a switch in feature tracking (see a) and b) in fig. 2).

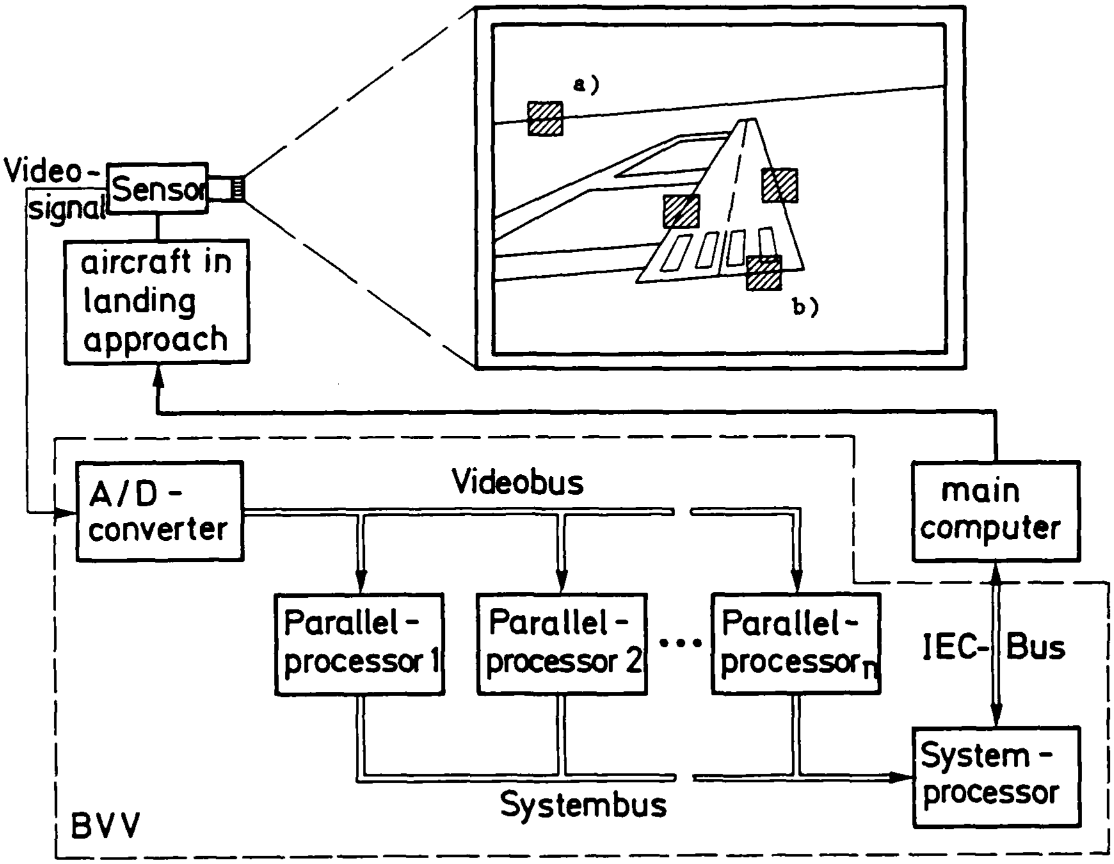


Fig. 2: Image sequence processing architecture (BVV 2)

Dynamic models

The motion of an airplane in all three translational and rotational degrees of freedom can be represented by a twelfth order system of difference equations, which after linearisation around an assumed working point is approximated by two loosely coupled sixth order systems: the longitudinal and the lateral motions.

These two dynamical models are selected to reconstruct the state vector by estimation in an optimal manner through Kalman Filter [4]. With this technique it is possible to measure only a few output variables and to estimate the whole state vector in a numerically efficient recursive manner. In addition the method has the advantage that it allows the rate quantities (temporal derivatives) to be obtained by the smoothing numerical operation of integration.

In this approach [5] the chosen longitudinal state vector has the components: speed in direction x and z: u and w, pitch rate q, pitch angle Θ, altitude H above ground and distance to the runway base line x; the controls are the elevator angle η and the thrust level δ:

$$x_L = [\, u \;\; w \;\; q \;\; \Theta \;\; H \;\; x \,]^T \; ; \quad u_L = [\, \eta \;\; \delta \,]^T \quad . \qquad (1)$$

The lateral motion has the components speed in direction y: v, roll rate p, yaw rate r, bank angle Φ, heading angle Ψ and lateral offset y, the last two relative to the vertical reference plane containing the runway center line; the controls are the aileron angle ξ, and the rudder angle ζ:

$$\mathbf{x}_S = [\, v \quad p \quad r \quad \Phi \quad \Psi \quad y \,]^T \; ; \qquad \mathbf{u}_S = [\, \xi \quad \zeta \,]^T \quad . \qquad (2)$$

The two state vectors are estimated by two instationary Kalman Filters of sixth order. The bulk of the measurement values are abtained solely by vision, applying the laws of perspective mapping. Velocity V is the only value measured conventionally. Thus, a discrete dynamical model of the form

$$\mathbf{x}[(k+1)T] = \mathbf{A}\,\mathbf{x}(kT) + \mathbf{B}\,\mathbf{u}(kT) + \mathbf{w}(kT) \qquad (3)$$

can be obtained, where T is the sampling time of 100 msec, and the vector $\mathbf{w}$ is additive process noise.

Perspective mapping

As mentioned above, all optical measurements are done according to the laws of perspective projection. The runway is assumed to be rectangular with known length L and a breadth B at a heigth H = 0. Fig 3 illustrates the different transformations that are needed to map a point on the runway into a point in the image plane of the camera [6]: first a point P (x_L, y_L, z_L) in the plane of the runway is transformed by translation $\mathbf{T}_S$ into the geodetical coordinate system (x_g, y_g, z_g), at the center of gravity of the airplane

$$\mathbf{s}_g = \mathbf{T}_S \mathbf{s}_L \quad . \qquad (4)$$

Then this point is mapped into the system fixed with the airplane (x_f, y_f, z_f), by performing a rotation $\mathbf{R}_S$ with the angles Φ, Θ, Ψ.

$$\mathbf{s}_f = \mathbf{R}_S \mathbf{s}_g \quad . \qquad (5)$$

The camera is mounted on a platform, which has the distance l_x, l_y, l_z from the center of gravity, so this point is mapped into x_f', y_f', z_f' by performing a translation $\mathbf{T}_K$

$$\mathbf{s}_f' = \mathbf{T}_K \mathbf{s}_f \quad . \qquad (6)$$

The platform is able to rotate with the angle ψ_k around the outer yaw axis and the angle ϑ_k around the inner pitch axis, so x_f', y_f', z_f' is mapped into x_c, y_c, z_c by rotation $\mathbf{R}_K$

$$\mathbf{s}_c = \mathbf{R}_K \mathbf{s}_f' \quad . \qquad (7)$$

A final projection is done by mapping point P into the perspective plane of the camera (x, y, z), including the parameters of the camera (focal distance, etc.) and further scaling values of the simulation loop

$$\mathbf{s} = \mathbf{P}\,\mathbf{s}_c \quad . \qquad (8)$$

The whole transformation of a point on the runway into a point in the perspective plane of the camera is obtained by multiplying the five matrizes:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = P\, R_K\, T_K\, R_S\, T_S \begin{bmatrix} x_L \\ y_L \\ z_L \end{bmatrix} \qquad (9)$$

The image coordinates y z represent one set of measurement values for the point P to feed the Kalman Filter. Several measurement points are needed to yield a stable behavior for the relative state estimation process.

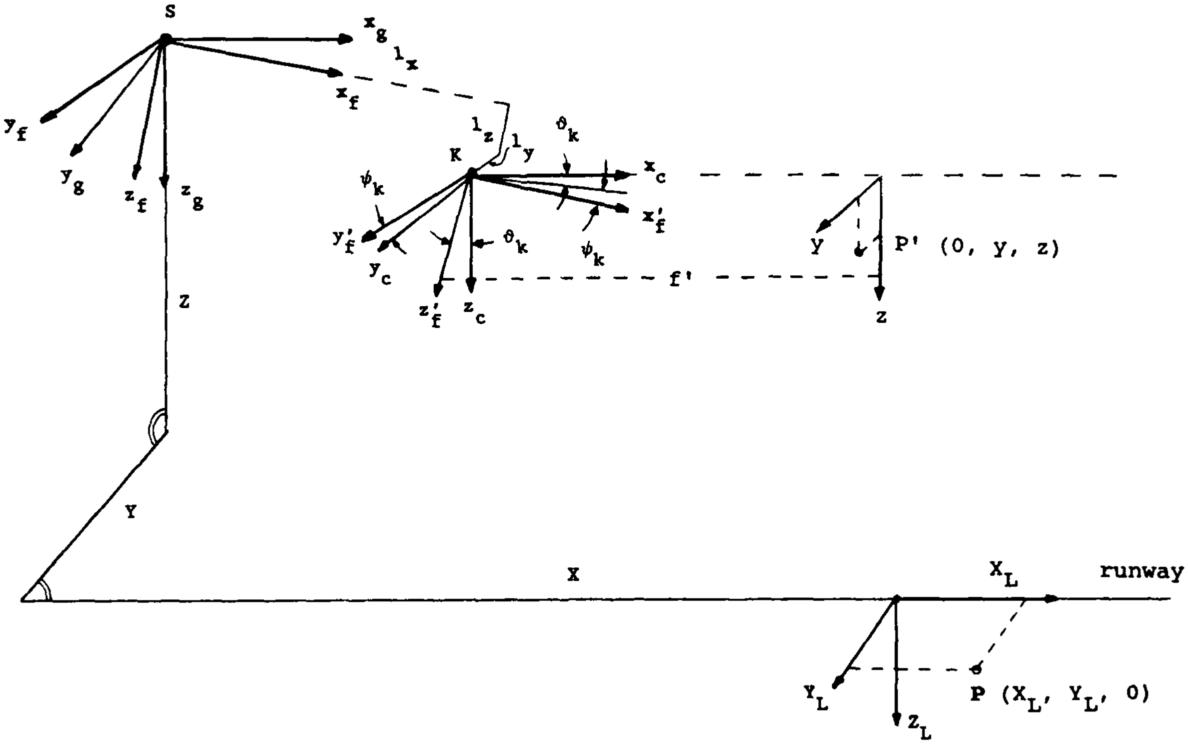


Fig. 3: Coordinate systems for deriving the imaging equations

State feedback control

The nominal trajectory shape in the vertical reference plane for autonomous visual landing, as selected in [5], is given in fig. 4. At point A, a transition into a stationary descent is initiated by reducing the lift by a constant amount such that the resulting stationary glide slope intersects the runway baseline point E. At the flare initiation altitude H_D, an exponential vertical deceleration is started. When flying over the runway base line the thrust is reduced to minimum, and the elevator is set to maximum. That leads to touchdown with a vanishingly small vertical velocity.

This schedule is precomputed in the initialization phase so that the state variables of the desired trajectory are available for every step in the landing approach and can be compared with the state variables estimated by the Kalman Filter every 100 msec. The differences of this comparison are the input to the controler.

Three feedback loops are used for compensatory control (fig. 5). Both the longitudinal and lateral controlers have been designed by Riccati-methods, the latter one being a pure regulator. For the longitudinal motion a digital integrator forces the static errors in altitude and velocity to zero.

The controlers have been designed with the following goals: show a good dynamical behavior of the airplane in the whole range of the approach, but do not cause excessive amplitudes of the control elements. The behavior achieved has been proven in the simulation loop with the complete nonlinear equations. Iterations by varying the weighting matrices of the Riccati-design have been performed. Stationary gain factors turned out to be sufficient for the landing approach.

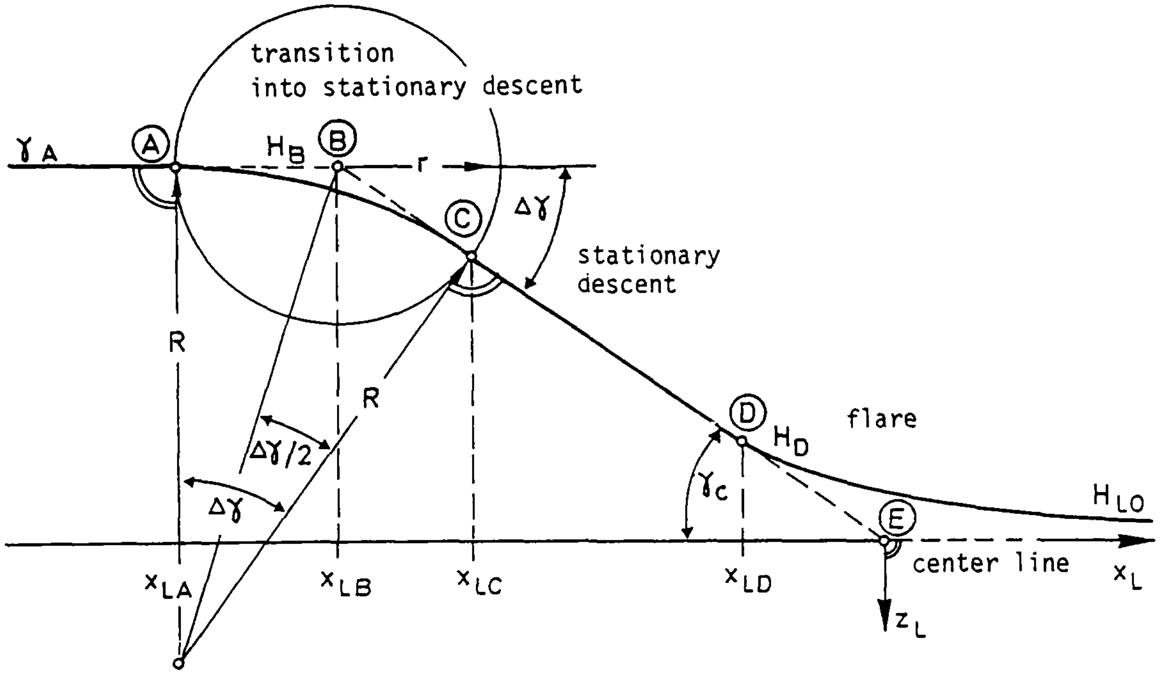


Fig. 4: Verticcl trajectory profile, from [5]

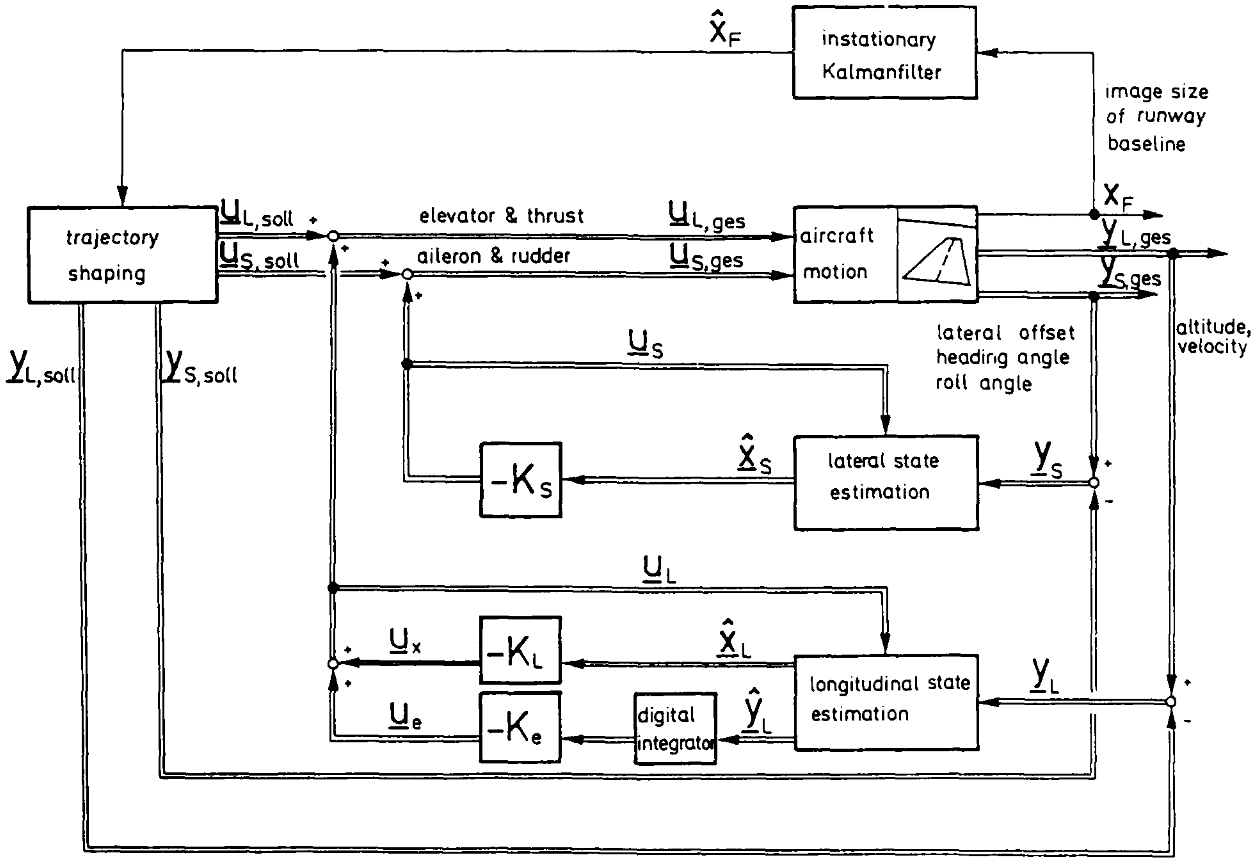


Fig. 5: Block diagram for model based feedback control, from [5]

Results

As described above, the system has been tested in a fixed base real-time simulation loop with computer generated calligraphic image sequences at 10 Hz update rate. Fig. 6 shows several snapshots as seen from the TV-camera in the loop. The digitally evaluated image areas, in which one PP determines the position of three linearly extended intensity transitions by correlating shifted and rotated templates in each case, are brightened for visual monitoring by the human operator.

Fig. 6a (upper left) taken at a range of about 1400 m and an altitude of about 100 m shows one window on the horizon line for roll and pitch determination and one window each on the lateral borderlines of the runway. Initially the borderlines are tracked horizontally until they appear under an angle of about 45° (slope 1, fig 6, 2c); for smaller slopes in the final approach phase they are tracked vertically.

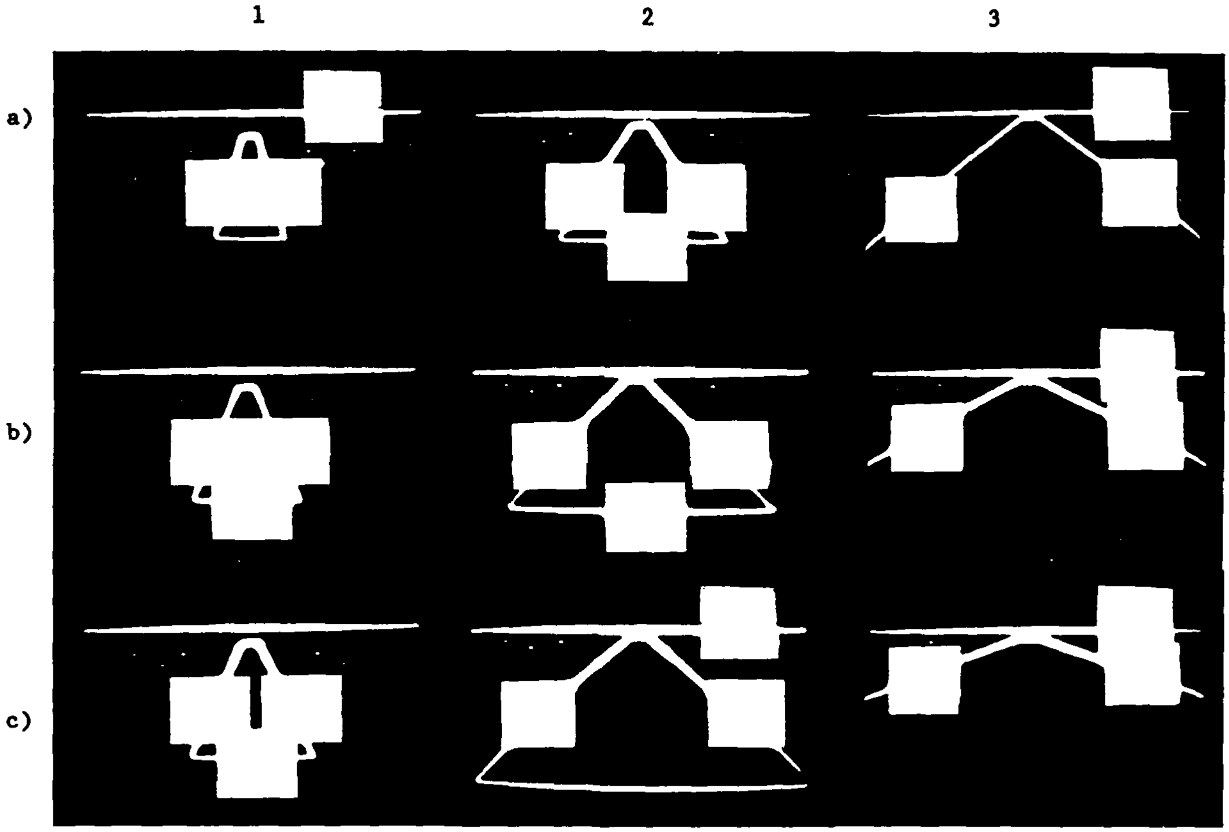


Fig. 6: Snapshots for autonomous landing approach in the simulation loop from [5]

When the baseline appears large enough, the third window is repositioned from the horizon line onto the baseline (fig 6, 1b) in order to accurately estimate the distance to the runway threshold (top in fig 5). Shortly before the baseline leaves the image due to the close approach, the third window is shifted to the horizon again (fig 6, 2c). At large distances and altitudes the runway borderlines appear close to vertical in the image. During approach these angles become smaller and smaller. Due to the elevation of the camera above the ground when the landing gear touches the runway, there is a limit angle which can be reached (fig 6, 3c).

Figure 7 shows the improving results of range estimation as the aircraft approaches the runway improving the optical measurement conditions. After the runway threshold has disappeared, longitudinal distances can only be determined from velocity integration over time until the end of the runway becomes clearly visible.

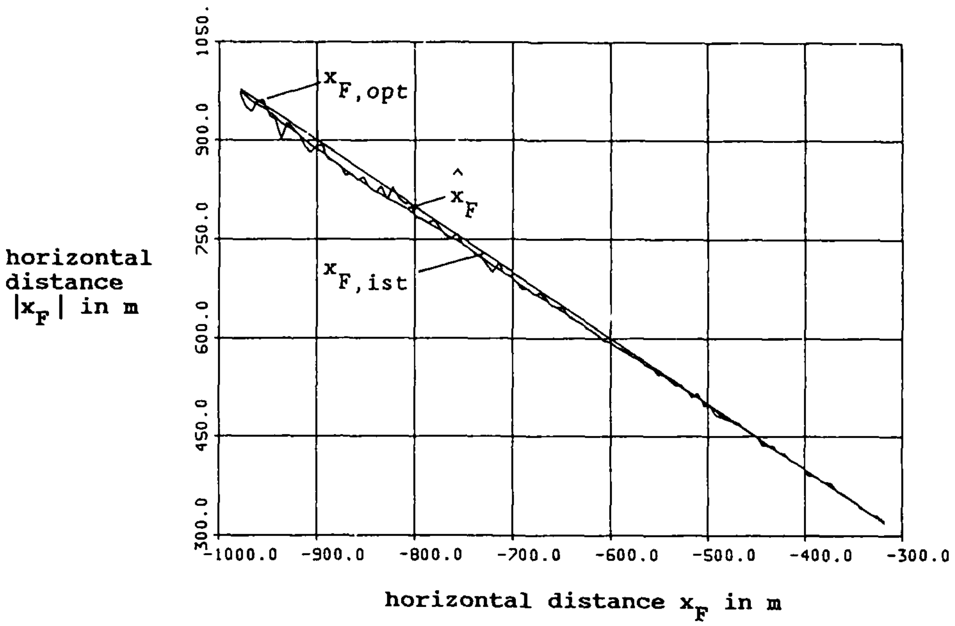


Fig. 7: Range determination, as a function of range, from [5]

Figure 8 shows histories of some state variables plotted over the distance to the runway threshold x_F (for almost constant speed this roughly corresponds to time histories). The subimages always show the value as determined in the simulation computer (index "ist"), the noise corrupted optical pseudo-measurement (index "opt") and the estimated value (shown with ^). It is seen that the estimates, on the one hand, are in good agreement with the true values "ist" and, on the other hand, are well smoothed as compared to the direct pseudo-measurements "opt".

Test runs with perturbations (both off-nominal initial conditions and wind-gusts) have shown that the vision based autonomous landing approach controler as developed is well able to cope with small disturbances.

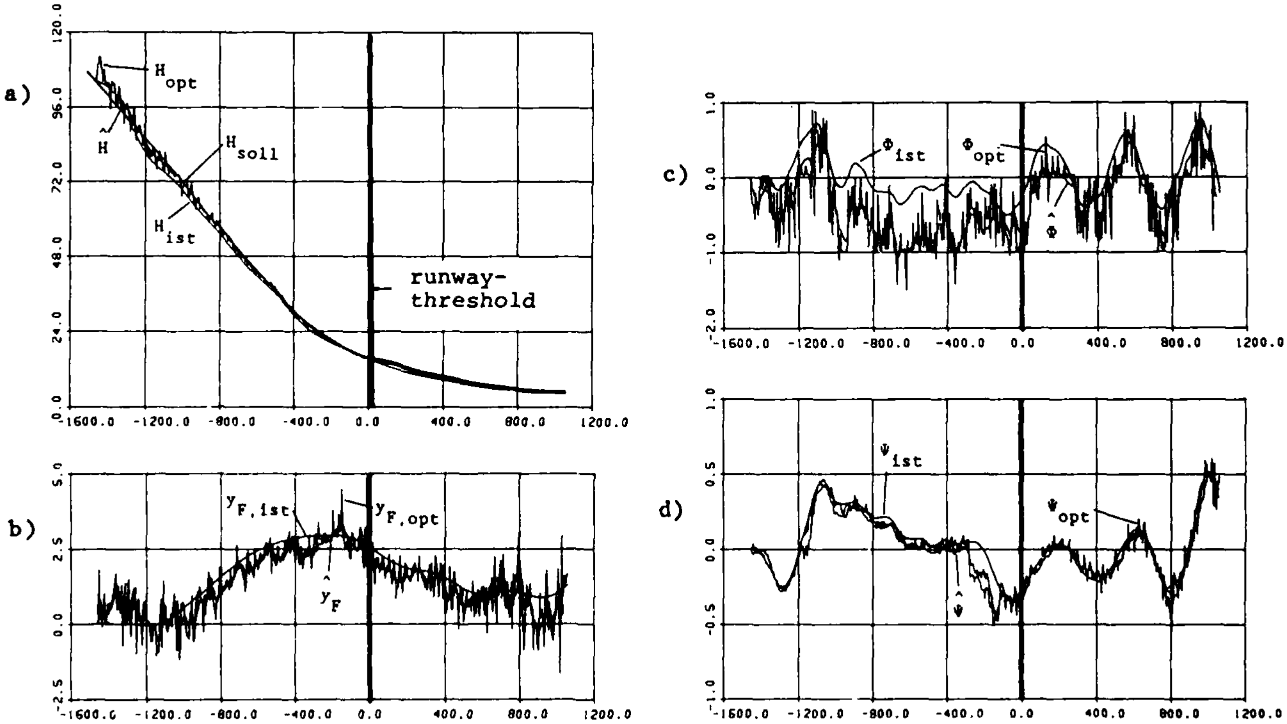


Fig. 8: Simulation results of some state histories against runway coordinates: a) altitude H in m; b) lateral offset y_F in m; c) roll angle Φ in degrees; d) heading angle Ψ in degrees; from [5]

Conclusions

By combining the dynamical models of modern control theory, shape representations of objects, their relative spatial orientation and the laws of perspective projection as part of the measurement model, a numerically efficient recursive scheme for real-time image sequence processing has been obtained [1]. Its performance capability has been demonstrated at a task as complex as the fully autonomous final landing approach of an aircraft in six degrees of freedom in a real time hardware-in-the-loop simulation [5] with moderate computer processing power. Improvements are being implemented presently for achieving more robust performance under more severe perturbations.

Literature

1. Dickmanns, E.D.: 4-D dynamic scene analysis with integral spatio-temporal models. Proc. 4th international symposium on robotics research (ISRR), Santa Cruz, 1987.

2. Dickmanns E.D., Graefe V.: Dynamic monocular machine vision. Int. J. Machine Vision and Application, Vol 1, No 4, 1988, pp. 223-240.

3. Graefe, V.: Two multi-processor systems for low-level real-time vision. In: Brady, M.; Gerhardt, L.A.; Davidson, H.F. (eds.) Robotics and artificial intelligence. Berlin, Heidelberg, New York: Springer 1984.

4. Kalman, R.E.: A new approach to linear filtering and prediction problems. Proc. IFAC-Congr. 1960, Moscow, Vol. 1, 481-492.

5. Eberl G.: Automatischer Landeanflug durch Rechnersehen. Dissertation UniBw München 1987.

6. Schell R.: Bewegungsgleichungen und Jacobi-Matrizen für die Längs- und Seitenbewegung. Interner Bericht, UniBw München, LRT/WE13/IB/88-4.

INTEGRATED FLIGHT GUIDANCE SYSTEM USING DIFFERENTIAL-GPS FOR LANDING APPROACH GUIDANCE

Dipl.-Ing. Thomas Jacob

o. Prof. Dr.-Ing. G. Schänzer
Institute of Guidance and Control
Technical University Braunschweig
Hans-Sommer Str. 66
D-3300 Braunschweig, West Germany

SUMMARY

Using the satellite based Global Positioning System (GPS) for precision flight guidance, accuracy problems arise due to the influence of dynamic manouvers on GPS receivers. The error behaviour in stationary as well as in dynamic applications is explained. From the error behaviour a system concept of an Integrated Flight Guidance System is derived. Different concepts of system integration are explained. The results of a closed loop mechanization of a Kalman Filter coupling GPS and INS implemented in the Integrated Flight Guidance System were checked by simulation and flight test results in approach and landings up to CAT II.

List of Symbols

ILS	Instrument Landing System	LOC	ILS Localizer
MLS	Micro Wave Landing System	GP	ILS Glide Path Equipment
GPS	Global Positioning System	INS	Inertial Navigation System
AHRS	Attitude Heading Reference System	ECL	Extended Center Line
x,y,z	North, East, Vertical Position	γ_1	Nominal Flight path angle
$\underline{\varphi}(\varphi\lambda h)$	Position (Lattitude, Longitude, Height)	$\underline{v}$	Inertial Velocity
$\underline{\Phi}(\phi\theta\psi)$	Attitude angles (bank, pitch, azimuth)	$\underline{a}$	accelerations (body fixed)
$\underline{\varepsilon}$	Coordinate axes misalignement		
$\underline{R}_i$	Range Satellite$_i$ - User Antenna		

1. INTRODUCTION

Today Global Positioning System (GPS), a satellite navigation system, is available as a new position finding aid. It seems to be possible, from a technical point of view, to realize landings under bad weather conditions by using GPS as an "Instrument Landing Aid". The advantage in performing landings even in bad weather conditions, without the help of any ground instrument located at the airfield is fascinating. That is the reason for a discussion whether the Microwave Landing System (MLS) will be out of date before it is generally installed or whether GPS can be used instead of MLS. If the total configuration of the satellite based worldwide available position finding system GPS is installed, it now appears possible, that a guidance of an aircraft from any terminal A to a terminal B including:

- runway
- take off
- cruise
- approach
- and landing

guidance could be achieved by one Integrated Flight Guidance System as it is presented in this paper. In developing an Integrated Guidance System for landings, the following two aspects:

- the accuracy
- the dynamic error behaviour

of the complete system must be tolerable for a flight guidance system.

2. Accuracy Requirements

The accuracy and performance requirements for the ground instruments of Instrument Landing Systems have been defined by the ICAO in Annex 10 /1/. The requirements are variated for the horizontal and vertical guidence instruments. For each, a maximum bias value has been defined. That is, an angular shift of the mean nominal path and a maximum beam bend, which is mainly due to multipath. (The following values are calculated for a standard runway with a 3 degree glide path and 1500m/3000m runway length for CAT II / III.)

ILS-Ground equipment of Cat II / IIIa (fig. 1)
(visibility horizontal 400m / 200m
vertical 30m / 0m)

has to fulfill the following values:

CAT	II		III	
	Offset	Bend	Offset	Bend
horizontal	4.5m	4.1m	3.0m	2.4m
vertical	1.2m	0.5m	0.6m	0.5m

The requirements for the guide beam characteristic are set so that the deviation of the aircraft due to ground instrument are less than (fig 2):

CAT	II	III
horizontal	5.0m	5.0m
vertical	1.2m	1.2m

Additionally, the roll and pitch attitude should not deviate more than 2 degrees from the reference values at the threshhold due to course bends. For CAT III the signal quality has to be good enough to provide automatic flight.

As we have no requirements available for military aircrafts, the ICAO accuracy values may be applicablᵉ to this system to. If the Integrated Flight Guidance System should be applied in a commercial airplanᴌ ɔr landings, it must satisfy the requirements of the ICAO depending on the weather conditions.

Although GPS in the presented system is just one of many different sensors it is this sensor, that is responsible for stationary accuracy. Therefore, the error characteristic of a high precision 5 channel GPS C/A code receiver in stationary as well as in dynamic flight tests have been analyzed. Specially flight tests are important in order to check the error behaviour in normal application conditions. From these tests the necessary information for an accurate system integration can be obtained.

3. Error characteristics of the Global Positioning System

For a high precision landing approach guidance the following errors will be recognizably disturbing.

- the measured GPS-position has an offset against the real position (Fig. 3).
- the measured GPS-position changes, even when there is no aircraft movement, due to the movement of the satellites (change of constellation, GDOP, selective availability) (Fig. 3).
- each measurement has a noise, of which the amplitude is receiver dependent; an oscillation is heterodyned (Fig. 3).
- when the selected satellite constellation is changed due to the rising of a new satellite, the position measurements react nearly like a step function with an amplitude of several meters (errors in the order of 8 m have been measured) (Fig. 3)

This error behaviour in stationary applications can be modelled in the following way:

Offset	15 m
Drift	0.1 m/min

two oscilations with

Amplitude	A1	0.2 m	Period	T1	16 s
Amplitude	A2	0.6 m	Period	T2	250 s.

- depending on the receiver type there will be dynamic errors, which occure in flight phases with longitudinal accelerations as well as in phases with leteral accelerations. The reasons for the dynamic errors are:
  - the influence of acceleration on the receiver clock (crystal oscillator).
  - the influence of changes in the measurement signals to the code tracking loop (delay lock loop) and the phase lock loop.
  - receiver integrated software and filtertechnique using low pass filters to reduce noise (fig. 4).
  - time lack due to signal processing; when the antenna is at time t1 at position x1 the receiver need in the order of 0.4s untill the position measurement is physically available at the output.
- flying a turn with a bank angle greater than the elevation of a satellite, a masking of the antenna may be produced for two reasons:
  - the hemispherical antenna is focused in such a direction where the locked satellites are undetectable.
  - parts of the aircraft (e.g. wing or body) move into the line of sight satellite - user antenna. In the most successful cases this effect produces an error of only several meters (about in the same order when changing the constellation) or at the worst case a total loss of GPS position information (fig. 5 point A).

Because the reaction of the chrystal oscilator, the reaction of the loops and the integrated filters differ with the receiver type, the parameters of the error model will also differ for each receiver type. However, there is one effect of the masking which each GPS receiver is exposed to. GPS receivers differ only in the time they need to relock and in the accuracy of the first position measurement after the shadowing. In different receiver performances, the time to lock on range from ca. 15 s up to several minutes. The error of the first measurments after lock on again can be in the order of hundred meters. For a high precision flight guidance for a landing phase in CAT II conditions, none of these errors can be accepted. Summed all errors they must be at least less than 5m in the horizontal and 1.2m in the vertical. This can be achieved by system integration as it has been done in the Integrated Flight Guidance System developed at the Institute of Flight Guidance and Control of the Technical University of Braunschweig.

4. Systemconcept

As generally known, the offset and the time dependency of the offset can be extensively eliminated by using Differential GPS - Technique (fig 6)/5/. Receiving the satellite signals at the ground station, the position of each satellite is known. As the position of the ground antenna is known, the range from the ground to each satellite can be calculated. By comparing the computed range with the measured range the actual system error can by determined. Transmitting this error to the aircraft, the error can be corrected in the onboard position finding computation. This technique can reduce the position error up to the receiver residuals which are in the order of a meter or a centimeter depending on the receiver type. However using this correction measures, the dynamic error characteristic is not improved.

The stationary tests, as well as the flight test, have proved to be excellently accurate in the long term, however, a lack of accuracy in dynamic manouvers has been detected. Therefore, and at least for security reasons during a landing approach but also for accuracy and for improvement of the dynamic behaviour of the system, it is necessary to generate additional position information without using GPS.

In dynamic manouvers the aircraft position, groundspeed, attitude and acceleration can be measured / calculated by using an inertial measurement unit (IMU). IMUs have a good short term accuracy, however, in the long term they have recognizable drifts. The long term accuracy is dependent to the gyro drift which determines the coordinate axes misalignment /7/. Obviously a system concept which utilizes the good long term accuracy of the GPS and the good short term accuracy of an IMU would produce a good overall accuracy.

This concept is realized in the "Integrated Flight Guidance System" (fig 7), which has been developed and also succesfully tested at the Institute during several tests. Investigations using a complete simulation of strapdown sensor systems as well as a simulation of the GPS system have been carried out. In this investigations differend integration concepts have been tested. Additionally the Integrated Fligth Guidance System has been checked by flight tests. In these flight tests, several landings up to CAT II accuracy have been performed using the institute´s fligth test and research aircraft a DORNIER DO128.

The developed Integrated Flight Guidance System is in the base concept composed of two parts:

- a position finding system and
- a guidance generator.

The position finding system computes the aircrafts position using Differential GPS, as well as, available sensor information currently being used in aircrafts.

To improve the dynamic characteristic of the entire system and to get sufficient information about the flight path during a breakdown of satellite information, the integrated flight guidance system is coupled with Kalman filter technique with inertial sensors - gyros and accelerometers. Radio or baro altitude sensors are used for the vertical guidance aditionally to GPS. Although the best estimation of position is calculated using this technique, no pilot is able to follow a nominal flight path using only a position information in latitude, longitude and height. It is necessary to convey the position information to the pilot in a way he is used to. Therefore, one must calculate a nominal flight path consisting of standard rate turns and linear parts using known coordinates of the target place (airport threshold, paratrooper drop point, tanker airplane, rendevous point,...). Calculating the nominal flight path from the precisely known actual position, the substantial advantage is programmed not to fly an intercept manouver.

Flight guidance data are calculated from the information of the actual position, the deviation, and the attitude relative to the nominal flight path. With this method the pilot receives information on how to stay on the nominal flight path, which may be curved horizontally or vertically and how to rereach it if he deviates from it. For the indication during the flight test an ILS cross deviation indicator or a flight director can be used. A coupling to an autopilot for automatic flight and landing is actually flight tested in the Institute.

5. GPS / INS System Integration

Combining GPS and INS, different depths of integration can be realized. The integration and the Kalman filter mechanization is dependent on

- the tasks of the integration,
- the accuracy limits,
- the robustnes,

- the computer time capacity,
- the IMU sensor concept (platform / strapdown / ring laser gyro),
- the stand alone capacity of each subsystem in the emergency case of system failure
- ...

For this tasks the basic concepts for system integration can be devided into the following topics:

- open loop GPS aided INS (fig. 8)
- closed loop GPS aided INS (fig. 9)
- fully integrated INS / GPS (fig. 10).

In an open loop or closed loop GPS aided INS mechanization, both systems are operating autonomously. Whereas both systems compute their own position estimation, this integration concept is relatively robust. If one of both systems or the Kalman filter fails, a position is still attainable. The Kalman filter estimates the elements of the state vector by using realistic error modells, from the dynamic of the measurement signal. The position, velocity, attitude and sensor errors of first order must be modelled in this state vector.

The open loop implementation can be used for platform systems. The advantage of the open loop implementation is, that in case of inacurate measurements just the Kalman filter is influenced, however, not the inertial system calculations itself.

As the sensors of the platform system are seperated from the body of the aircraft by gimballs, they are operating normally at their reference point zero. The attitude angles are measured by the angles of the gimballs. The Kalman filter can run external to the INS. In the filter the errors of the inertial system have to be modelled, in addition the sensors errors have to be regarded /8/.

In contrary to platform systems, sensors of strapdown systems are not uncouplied from the aircraft body. They are operating in a dynamically more disturbed environment as there are vibrations, angular accelerations, angular oscilations which result in an additional negativ influence to the system performance. In addition sensors are not operating at a reference point zero /9/. Therefore the errors of the system will increase very rapidly and problems of numerical inaccuracy in an open loop Kalman filter may soon increase. Therefore it is advantageous to loop back the estimated sensor errors to the strapdown calculations to compensate for the sensors errors. Consequently, the erros of the INS will be kept low and linear error models can be used. However this system implementation can be unstable.

The system integration of best accuracy will be of course the total integration of both systems. As the Kalman filter theory asks for uncorrelated measurements, it is optimal from the estimation theory to use the raw measurements of the GPS as there are:

- the range measurements to at least four satellites,
- the ephemerides to calculate the Keppler laws for computation of the satellite positions,
- the parameters to correct for the Iono- and Troposphere errors,

and to estimate the receiver clock errors (time offset and frequency) as a part of the filter model. The measurement equations are more complex. This implementation needs much more computation time. Additionally this concept is more sensitive to system failures. In the case of a malfunction of the Kalman filter or of the INS, the navigation capability would be totally lost. As the raw measurements of the GPS (range and ephemerides) are necessary, no standard GPS receiver providing normally just position outputs can be used.

As in the Integrated Flight Guidance System, strapdown sensors are used, because the system has to be robust and because compatibility to different GPS receivers and strapdown sensors of different accuracy classes should be achieved, the closed loop GPS/INS integration has been used in this system.

The described concept of the Integrated Flight Guidance System has been already thoroughly investigated in numerous flight test in which landings have been performed using a procedure similiar to the ILS. Here the sensor system and also the procedure shouldn´t be changed. Therefore the pilot finds a system reaction which is similiar to the ILS.

6. Flight-Test Results

Figure 11 shows the ground track of the same flight as that shown in figure 5. The GPS-position and the position output of the position-finding part of the "Integrated Navigation System for Aircraft" are shown here. Based on the sensor errors, which are determined by the system, the position-finding part is able to determine the flight curves at point A faultlessly. The position output during the remaining phase, - the landing approach and landing, can be determined reliably by the locations part of the "Integrated Flight Guidance System".

Even with a breakdown of the GPS signals, the filter algorithmns still give the position, speed and Euler angle with a high precision for a limited time. A decisive prerequisite for that is a realistic mathematical model of the dynamic error characteristics of the inertial systems.

The result of the command value generator is plotted in figure 12 and 13 relative to the touch down point. The vertical and horizontal deviation from a nominal glide path are given to the crosspointer indicator, and an indication is generated, which the pilots are alredy accustomed to from previous ILS approaches. The vertical deviation grows up to 90 degree when the pilot leaves the nominal glide path and is passing the nominal touch down point in a minimum height, as it previously happend during the flight test.

Which overall position accuracy can be achieved using the "Integrated Flight Guidance System"? In figure 14 the ground track of two take-offs, two landings and one back track are plotted in meters relative to the take-off position. During these aircraft movements, the pilot tried to fly so as to roll with little deviation from the centerline as possible. It can be recognized from the plots, that the ground tracks deviate from each other less than 1.3 m. This deviation is so small, that one cannot determine - without using a highly precise flight path tracking system - if the 1.3 m is the position accuracy of the Integrated Flight Guidance System or if the aircraft really deviated 1m from the centerline which is possible during landings.

The vertical accuracy had been checked by using a laser range finder, which measured the distance from the aircraft to the ground. After transformation from the body fixed into the navigation coordinate system, and comparing the laser height with the height of the Integrated Flight Guidance System, the profile of the terrain is computed including the measurement errors of the laser and the integrated system. Comparing this terrain profile with a topographic geodetic map, an accuracy of less than +/- 0.8m could be verified for discrete points (fig 15).

7. Summary

The flight tests, which have been made with the "Integrated Flight Guidance System", developed at the Institute for Flight Guidance and Control of the Technical University of Braunschweig have shown good results by combining two sensor systems with different, time dependent, signal qualities: the inertial sensors, with their excellent short-term characteristics; and the GPS with excellent long-term characteristics. With the Kalman filter technique it is possible, even in high dynamic flight phases, to determine a position of high precision and reliability. The position determined is better, than the precision of each system standing alone. While in real- time application the vertical accuracy of the "Integrated Fligth Guidance System" is sufficient for CAT II landings the horizontal accuracy seems to fulfill the demands of the ICAO for CAT III. In offline applications of the position finding part of the "Integrated Flight Guidance System" accuracies in the order of 10cm can be achieved, by using special differential techniques (double differential).

It is possible to use the presented system in all high precision navigation applications. By using this system an accuracy can be reached which is similiar to the accuracy which is necessary for airborne cartography. Up to now this quality has been possible only by airborne photography, however by a using much more time to interpret the air photographs. For an application of this system for terrain profiling (e.g. TERCOM) the reference for computing digital maps can be reached in nearly real time.

8. Literature

/1/ Aeronautical Telecommunications
International Civil Aviation Organisation (ICAO) ANNEX 10

/2/ Avionics Navigation Systems
Kayton, M.; Fried, W.; John Wiley & Sons 1969

/3/ Richtlinien für den Allwetterflugbetrieb nach Betriebsstufe II
Nachrichten für Luftfahrer NfL I-350/72 Bundesanstalt für Flugsicherung

/4/ Principle of Operation of NAVSTAR and Systemcharacteristics
Milliken, R.J; Zoller, C.J. AGARDograph AG 245

/5/ The NAVSTAR GPS System
AGARD Lecture Series No. 161

/6/ Differential Operation of NAVSTAR GPS
Kalafus, R.M. Journal of the Institute of Navigation Vol. 30, 1983

/7/ Fligth Guidance II
G. Schänzer, Technical University Braunschweig

/8/ Techniques of the development of Error Models for Aided Strapdown Navigation Systems
Lechner, W. AGARDograph AG 256

/9/ Genauigkeitsanalyse von Trägheitsnavigationssystemen
N. Lohl, Technical University Braunschweig 1981

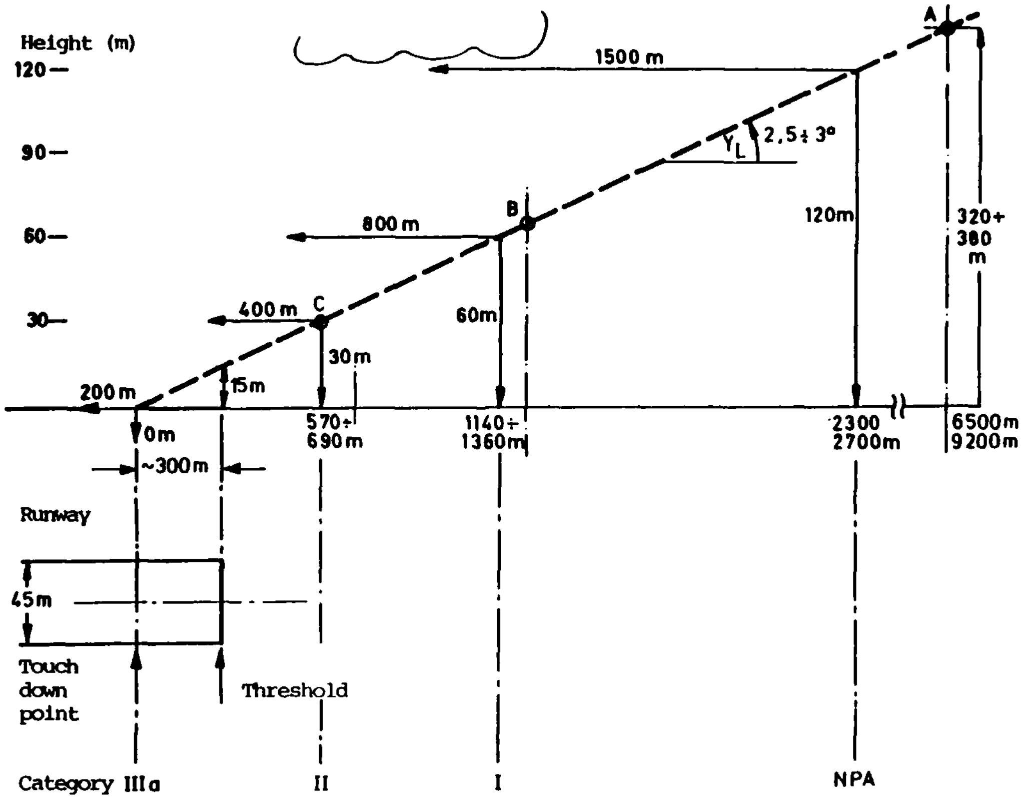


Figure 1: Categories and visual conditions in IFR

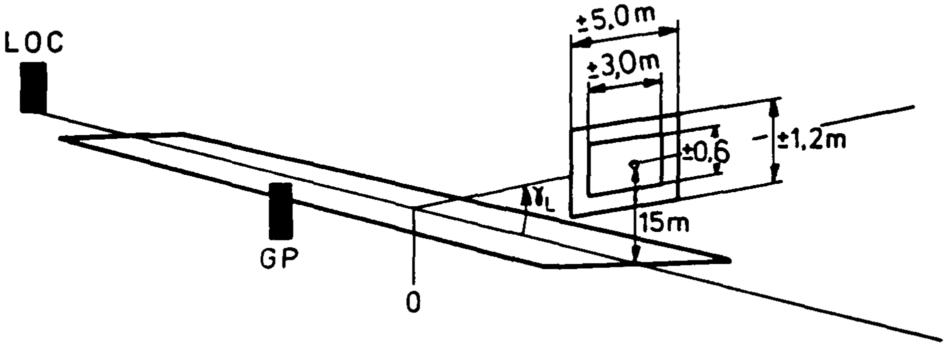


Fig. 2: ICAO - ILS accuracy requirements CAT IIIa

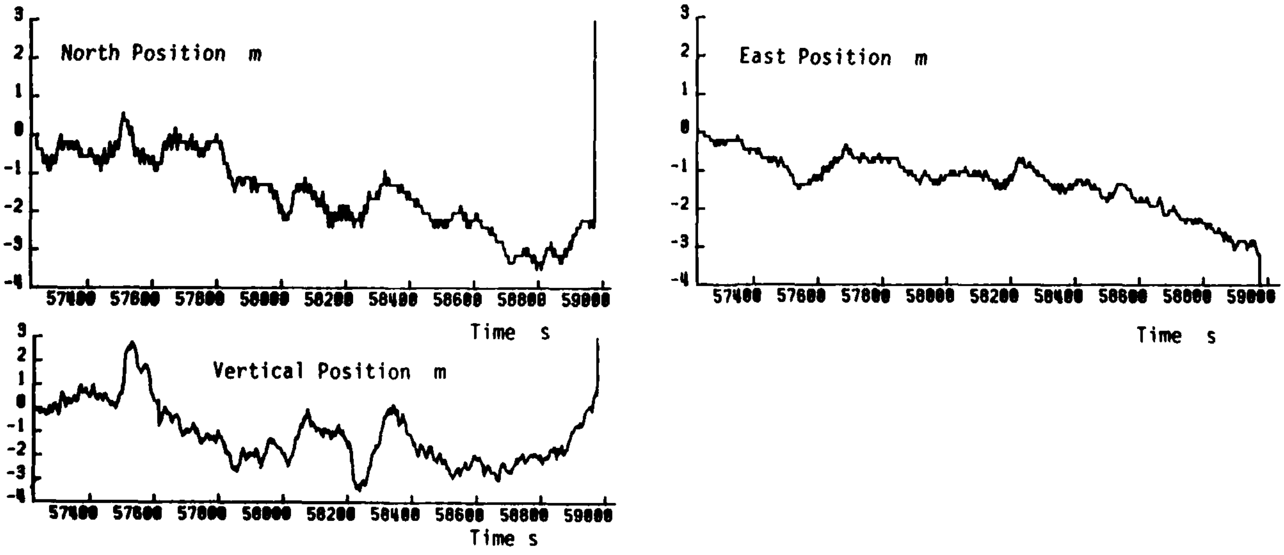


Fig. 3: Deviation of North- East- and Vertical Position (v=0.0m/s, 5Channel C/A Code)

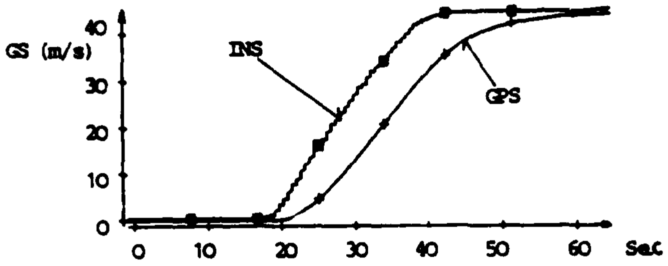


Fig. 4: Comparison of ground speed measured by INS and GPS

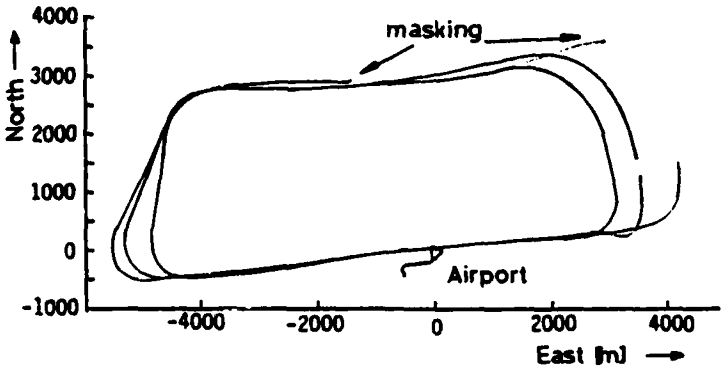


Fig. 5: Flight test - ground track measured by GPS

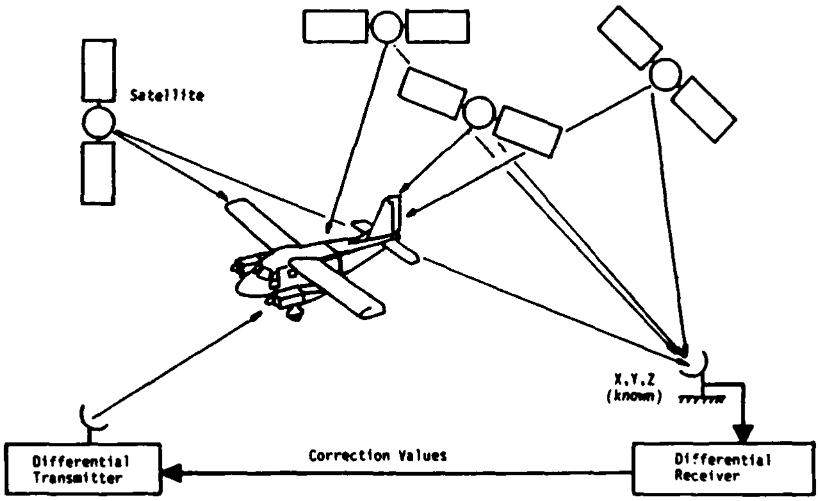


Fig. 6: Differential GPS

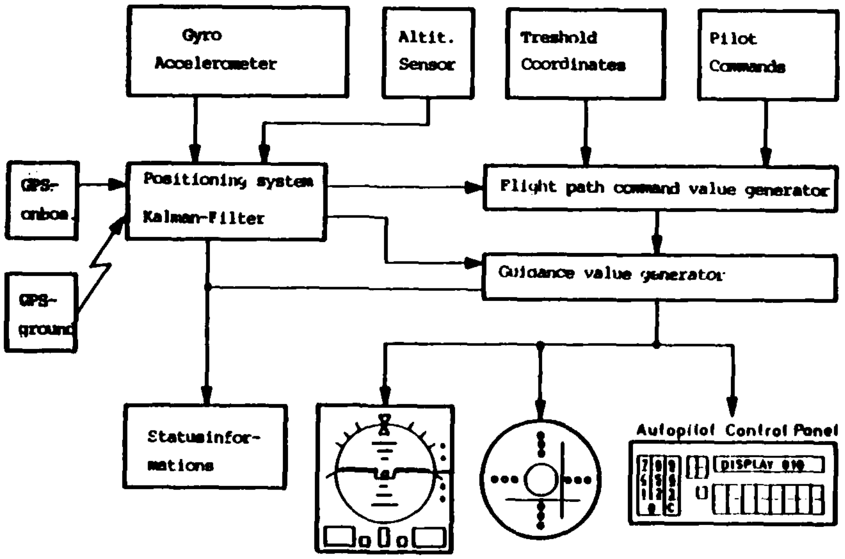


Fig. 7: Integrated Flight Guidance System

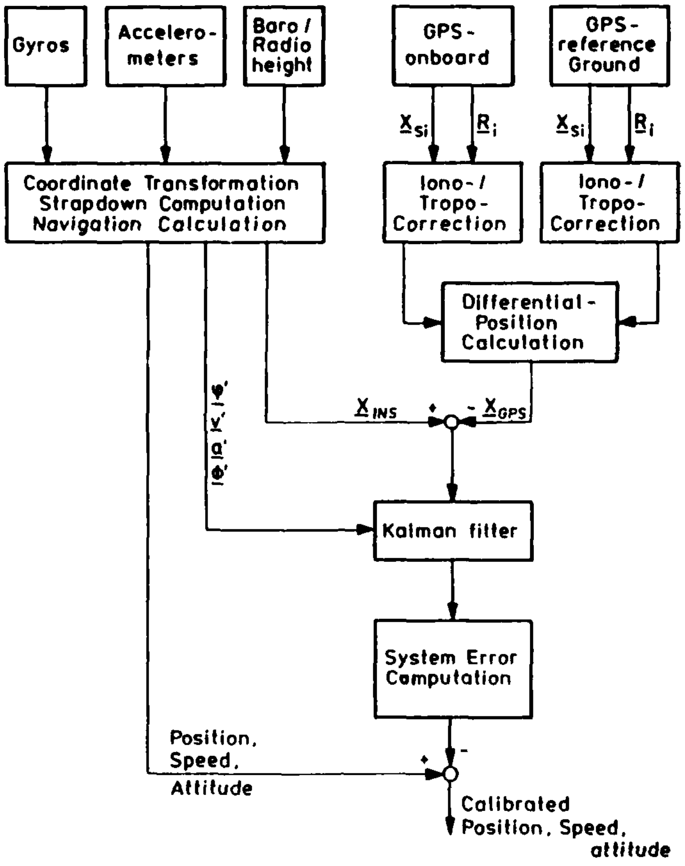


Fig. 8: Open loop GPS aided INS

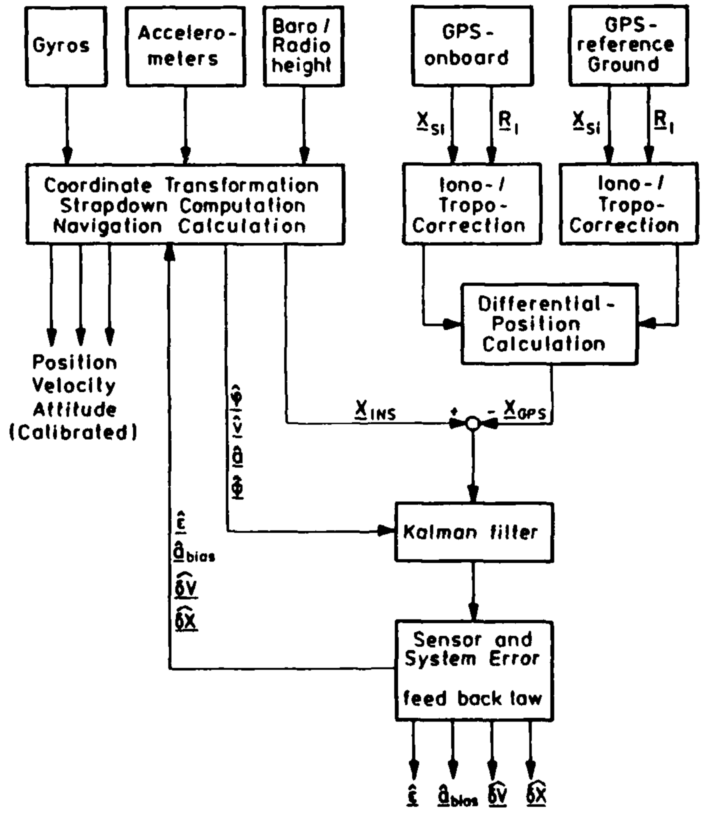

Gyros
Accelero-meters
Baro / Radio height
GPS-onboard
GPS-reference Ground
$\underline{X}_{si}$
$\underline{R}_i$
$\underline{X}_{si}$
$\underline{R}_i$
Coordinate Transformation Strapdown Computation Navigation Calculation
Iono-/ Tropo-Correction
Iono-/ Tropo-Correction
Differential-Position Calculation
Position Velocity Attitude (Calibrated)
$\underline{\hat{\varphi}}$ $\underline{\hat{v}}$ $\underline{\hat{a}}$ $\underline{\hat{\Phi}}$
$\underline{X}_{INS}$
+
-
$\underline{X}_{GPS}$
$\underline{\hat{\varepsilon}}$ $\underline{\hat{a}}_{bias}$ $\underline{\widehat{\delta V}}$ $\underline{\widehat{\delta X}}$
Kalman filter
Sensor and System Error feed back law
$\underline{\hat{\varepsilon}}$ $\underline{\hat{a}}_{bias}$ $\underline{\widehat{\delta V}}$ $\underline{\widehat{\delta X}}$

Fig. 9: Closed loop GPS aided INS

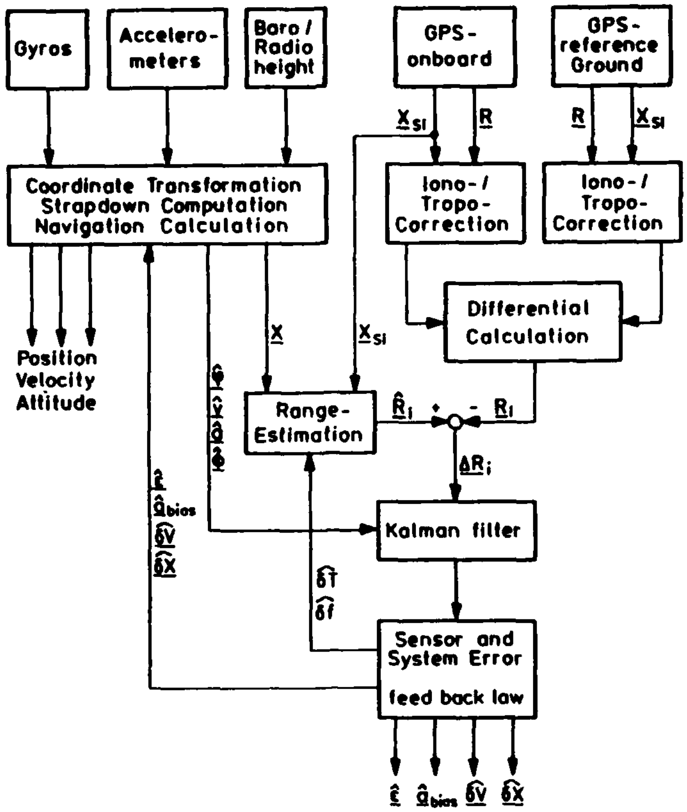


Fig 10: Fully integrated GPS / INS

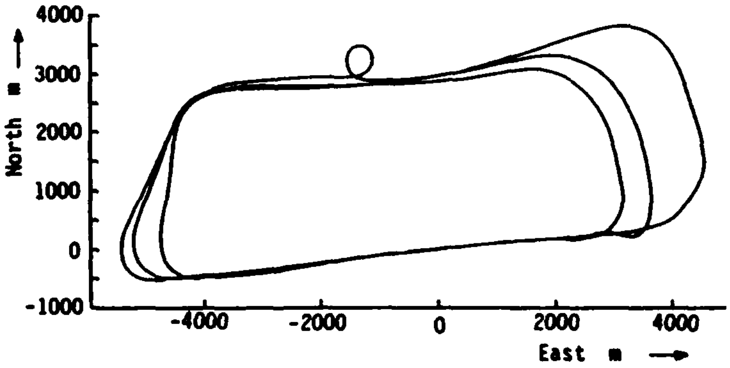


Fig 11: Ground Track of Integrated Flight Guidance System

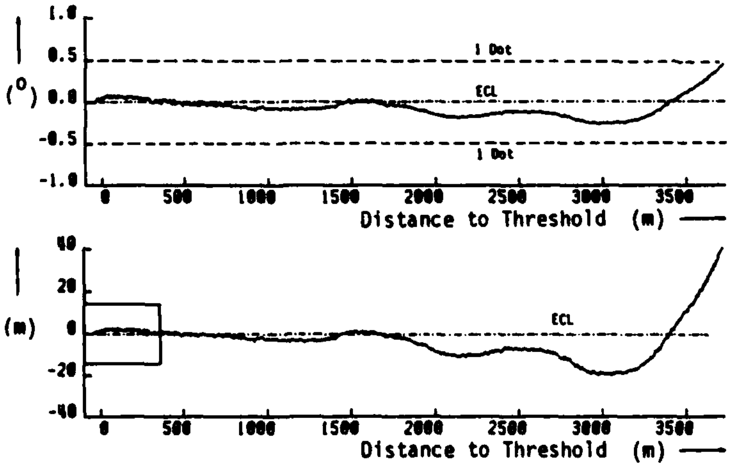


Fig. 12: Crosspointer indication: Horizontal deviation calculated by command value generator (flight test)

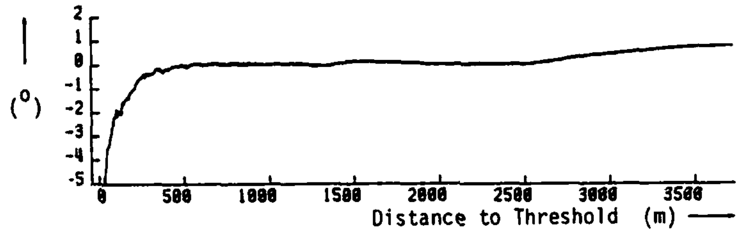


Fig. 13: Crosspointer indication: Vertical deviation calculated by command value generator (flight test)

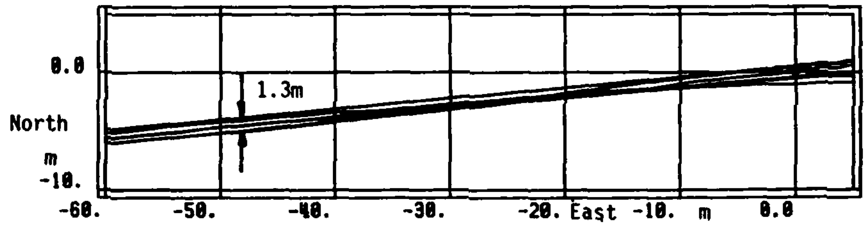


Fig. 14: Ground track calculated by Integrated Flight Guidance System while taxiing, starting and landing (flight test)

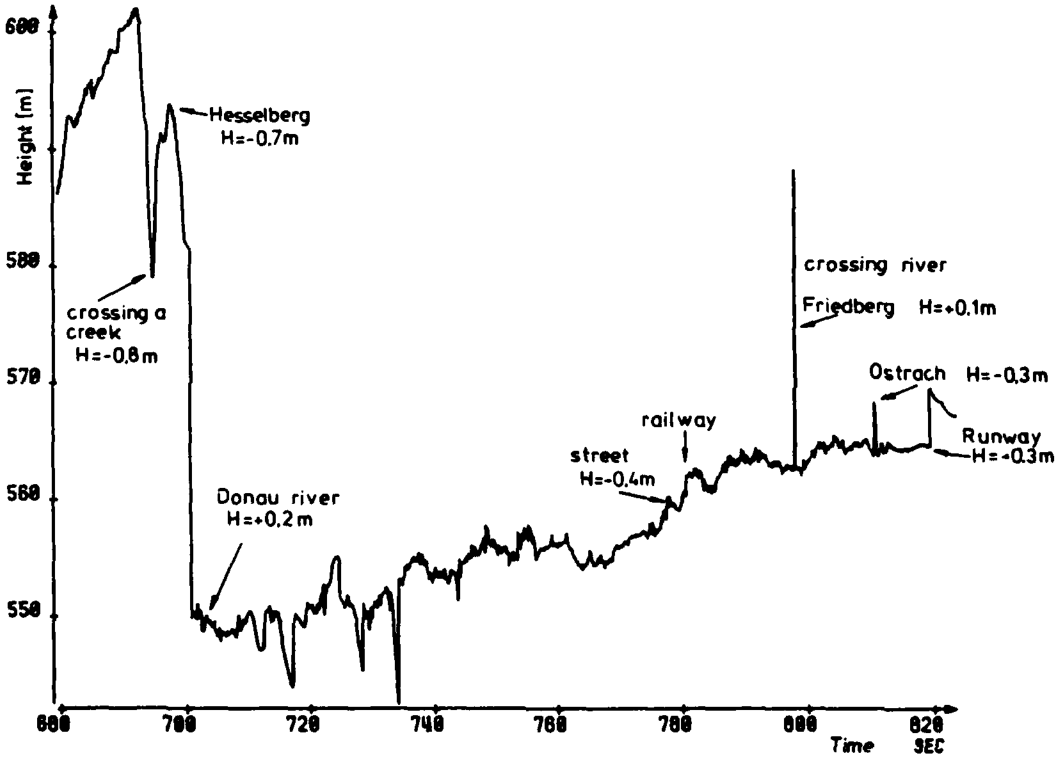


Fig. 15 Comparison of terrain profile calculated by Integrated Flight Guidance System with topographic geodetic map for discrete points

MINIATURE GPS-BASED GUIDANCE PACKAGE

by
L. Stotts (DARPA)
Program Manager
Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209 USA

J. Aein (RAND)
Senior Staff Scientist
The RAND Corporation
2100 M Street, N.W.
Washington, D.C. 20037 USA

N. Doherty (MITRE)
D-81 Department Head
The MITRE Corporation
Burlington Road
Bedford, Massachusetts 01730 USA

SUMMARY

The technical goals, issues, and status of the GPS-based Guidance Package or GGP effort within the Aerospace and Strategic Technology Office of DARPA are presented. The GGP exploits the synergisms achieved by combining inertially sensed (IMU) movement with externally sensed GPS reference signals. The goal is to produce a combined GPS/IMU navigation grade system which will be miniaturized for easy insertion to any host vehicle and inexpensive for use by expendable vehicles (weapons and platforms). Efforts already under way within the DoD community based on integrating conventional navigation technologies result in systems on the order of 65 lbs, 160 watts, 1800 cu.in., and $65 K per unit. The GGF effort aims to match the navigation performance of the conventional technologies but fit within an envelope of 10 lbs, 20 watts, 120 cu.in., and $15 K per unit. The GGP builds upon the integrated circuit technology from the preexisting DARPA mini GPS receiver (MGR) program combined with the following: (a) solid state linear accelerometers and fiber optic rotation rate sensors (gyros) for three axes inertial sensing, and (b) a data processor and associated software to implement a Kalman filter to integrate the sensor outputs and provide the navigation solution as well as any filtered velocity, acceleration, and orientation data needed by the host vehicle. Major cost reduction breakthroughs are offered by FOG sensors which employ integrated optic chips for light wave processing along with the polarization preserving fiber optic rotation sensing coil and laser diode optical source. GGP host vehicle insertion is also facilitated by its packing/customizing achieved through modularity of MGR, IMU, and navigation microprocessor subsystems. Modularity is achieved with standardization of (1) the Kalman filter architecture in the navigation processor and (2) data transfer points (ports) interfacing the MGR and IMU sensors to the navigation processor. Technical detail is provided on the following topics: functional architecture, technology, and status of the MGR chip set; desired performance, approach, status, and technology issues for a FOG sensor, and system level integration and performance issues.

INTRODUCTION

This paper reviews the technical goals, issues, and status of the navigation technology efforts within the Aerospace and Strategic Technology Office of DARPA. The primary objective of the DARPA/ASTO navigation effort, known as the GPS Guidance Package, or GGP, is to exploit the synergisms[1] achieved by combining inertially sensed (IMU) body movement with externally sensed radio reference signals from

[1] The IMU drift coefficients can be calibrated through an extended "initialization" provided by GPS sensing while GPS receiver tracking during high dynamic maneuvers can be aided by the IMU. Additionally, the IMU provides

multiple satellites comprising the Global Positioning System (GPS). The DARPA goal is to produce a combined GPS/IMU system, the GGP, which will be miniaturized for easy (if not trivial) insertion to almost any host vehicle and inexpensive for use even by expendable vehicles (weapons and platforms).

Several efforts are already under way within the DoD community based on integrating conventional navigation technologies employing tuned rotor gyros and analog GPS receivers. To date, these configurations result in somewhat bulky systems on the order of 65 lbs, 160 watts, 1800 cu.in., and $65 K per unit. These require sizable host vehicles and at best are only infrequently expendable.

The DARPA GGP effort aims to match the navigation performance of the conventional technologies but fit within an envelope of 10 lbs, 20 watts, 120 cu.in., and $15 K per unit. To succeed, the GGP must maximally utilize solid state (like) devices, fabrication, and assembly methods, i.e., minimize labor input. Consequently, the GGP builds upon the integrated circuit technology base developed from the preexisting DARPA mini GPS receiver (MGR) program sometimes referred to as "Virginia Slims" for its packaging resemblance to a product of the same name [1].

To an MGR-type GPS sensor (hereinafter referred to as an MGR) must be added the following: (a) solid state linear accelerrometers and rotation rate sensors (gyros) for three axes to provide inertial sensing, and (b) a data processor and associated software to implement a Kalman filter to integrate the sensor outputs and provide the navigation solution as well as any filtered velocity, acceleration, and orientation data needed by the host vehicle. The MGR and "solid state" IMU may each have its own embedded processor chip dedicated to running the necessary real time, lower level, sensor signal processing. The subsystem structure is shown in Fig. 1.

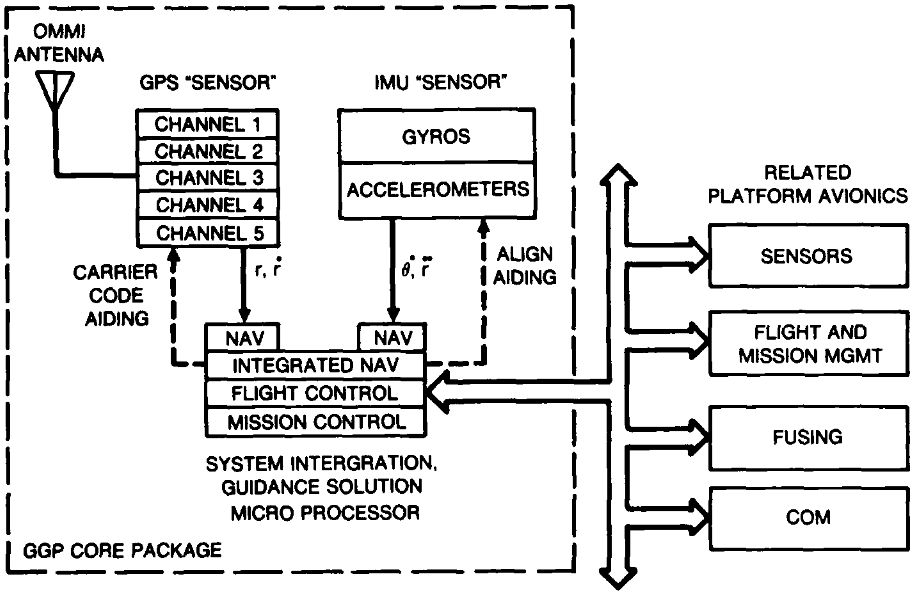


Fig. 1— Low-Cost GPS Guidance Package

As there are several silicon accelerometer (SiAccel) efforts already under way upon which the GGP can draw, the primary GGP technology push is on developing a solid state like rotation sensor (gyro) for the IMU. A review of the gyro technology base indicates use selections of a rotation sensor based on the Sagnac effect; either the Ring Laser Gyro (RLG) or the Fiber Optic Gyro (FOG). Of the two technologies, the RLG is by far the more mature, e.g., RLGs which have very high performance are now going into production. However, the prospects for meeting the low cost goals for the GGP using high performance RLGs do not look promising.

navigation during periods of GPS signal loss; remember four separate satellite signals are required to navigate with GPS alone. Moreover, mutually combining GPS with inertial sensing allows slower GPS receiver sequencing amongst the satellites even in high dynamics. This then provides fault tolerance amongst receiver channels.

There are major cost reduction breakthroughs offered by FOG sensors which employ integrated optic chips for light wave processing along with the fiber optic rotation sensing coil and laser diode source. The objective is to provide FOG peformance levels comparable to RLGs. Consistent with DARPA's acceptance of technology risk in order to induce jump advances in technology, the FOG rotation sensor was selected for GGP development. The integration of a GPS receiver with an RLG-based IMU is a lower risk approach which is likely to be pursued elsewhere in the DoD community.

Insertion of the GGP to host vehicles is facilitated through its very small size (to minimize host burden) and flexible packaging/customizing achieved through modularity of MGR, IMU, and navigation microprocessor subsystems. The modularity with interface control between the data processor and sensor subsystems will allow swapouts of either MGR or IMU sensors without redesigning the whole system. The modularity is facilitated through standardization of (1) the Kalman filter architecture in the navigation processor and (2) data transfer points (ports) interfacing the MGR and IMU sensors to the navigation processor. This will allow flexible choices in the selected number of MGR processing channels (e.g., two-channel sequencing MGR for low dynamics vice six-channel parallel MGR for high dynamics) and easy IMU upgrades with improving FOG sensors.

This paper is organized as fol'ows: Section II outlines functional architecture, technology, and status of the MGR chip set; Section III discusses the desired performance, approach, status, and technology issues for a FOG sensor, and Section IV discusses system level integration and performance issues.

MINI GPS RECEIVER MGR

The following section reviews the specific Rockwell Collins MGR technology of reference 1 as an example only of the MGR needs, capabilities, and issues for a GGP. A DARPA GGP will employ a similar MGR technology but need not be identical with that described here.

The physical MGR chip partitioning developed in reference 1 is shown in Fig. 2. A summary of the technical parameters for each chip with their respective power consumption is shown in Table 1. The performance requirements for the MGR are exactly the same for conventional GPS receivers as specified in SS-US-200 [2]. [Summary shown in Table 2] Referring to Fig. 2, the three key new chips developed (out of five) in order of their technical difficulty are as follows:

- o GaAs RF/IF/AD Receiver Front End
- o CMOS Hi-Speed Digital Signal Processor (VHISC Technology)
- o Silicon Bipolar Synthesizer

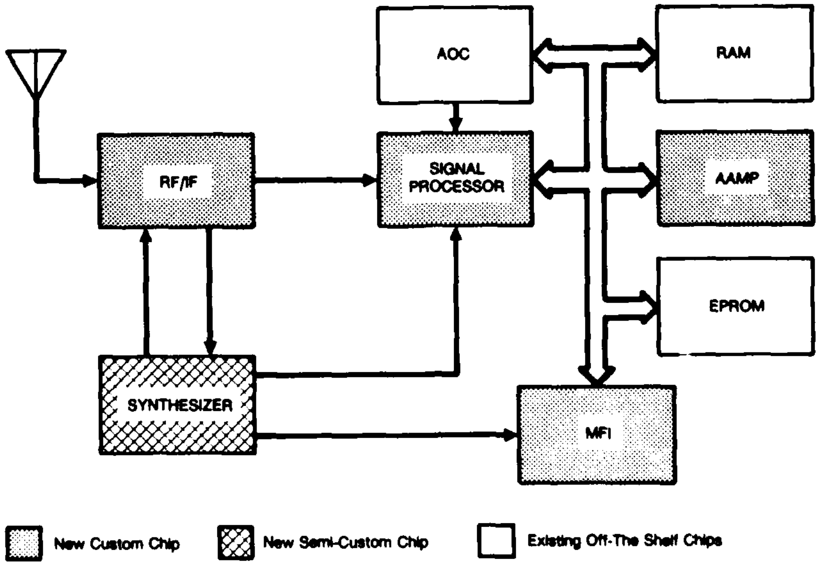


Fig. 2—Generic MGR Architecture

Table 1

CHIP SET SUMMARY

Chip Type	Device Count	Dimensions (Inches)	Implementation Technology	Power (mW)
RF/IF translator	300–400	0.200 x 0.240	GaAs	1700
Signal processor	20,000	0.185 x 0.220	1.25 micron bulk COMS	90
Multifunction Interface	29,000	0.370 x 0.370	1.6 micron bulk CMOS	20
Micro processor	60,000	0.214 x 0.261	2 micron bulk CMOS	80
Frequency synthesizer	600–700	0.250 x 0.250	biploar silicon	500

Table 2

USER EQUIPMENT REAC AND TTFF TIME REQUIREMENTS

	REAC 1 (MIN) REAC 2 (MIN)	TTFF 1 (MIN) TTFF 2 (MIN)	UNCERTAINTY			MAX. ACCEL. (M/SEC^2)	MAX. JERK (M/SEC^3)
			POSITION (KM)	VELOCITY (M/SEC)	TIME (SEC)		
LOW DYNAMIC SET	10.5 NOT REQUIRED	5.5 NOT REQUIRED	100 (3σ)	25 (3σ)	20 (1σ)	6	20
MEDIUM DYNAMIC SET	9.0 NOT REQUIRED	4.0 NOT REQUIRED	100 (3σ)	75 (3σ)	20 (1σ)	10	20
HIGH DYNAMIC SET	7.0 6.5	2.0 1.5*	100 (3σ) 10 (3σ)	150 (3σ) NEGLIGIBLE	20 (1σ) 10 (1σ) (μ SEC)	10 NEGLIGIBLE	100 NEGLIGIBLE

NOTE: The probability of success in achieving the stated REAC and TTFF times shall exceed 0.9. These REAC and TTFF requirements are applicable over the –40°C to +55°C temperature range.

*TTFF 2 can be reduced to 0.5 minutes if position and time uncertainties are also negligible.

The advanced data processor, control chip--AAMP--was developed by Rockwell Collins to reduce power consumption in the conventional technology manpack GPS receiver while the multifunction interface or glue chip was produced with a straightforward (silicon) foundry. The AAMP is an embedded MGR data processor and will not be used for MGR/IMU sensor integration and control. A summary description of these chips is given in tables 3 and 4.

Modularity needs for multiple GPS signal tracking (2 to 6 channels) are determined by the host vehicle dynamics and fault tolerance requirements. Multiple channel modularity is obtained through the MGR chip partitioning chosen with a matching software structure provided. Each GaAs MMIC chip processes either one of the two possible L-band frequencies (L1/L2) identically radiated by all GPS satellites. Each digital signal processing chip fully processes one C/A and P-coded transmission. Using mix and match the minimal configuration for low dynamics uses a one MMIC two-si¸.al processor configuration with L1 and L2 and 4 P-codes sequentially processed. The maximal configuration for a high dynamics set, which simultaneously processes two frequencies and six channels, will use two GaAs MMIC and six signal processor chips. The rest of the high dynamics set remains the same as the two-channel set (synthesizer, MFI, AAMP) although the data processing throughput requirements on the AAMP increase to where two AMMP chips may be required.

The most unique chip technology challenge for the MGR has been the GaAs MMIC chip whose layout is shown in Figs. 3 and 4. On this common die two unique suites of circuits must be processed in the GaAs foundry. Specifically, a low noise (3 db NF) high quality analog RF amplifier path with over 90 db of gain must be accomplished along with gigabit digital FET gates to implement the necessary on-chip frequency dividers and the analog-to-digital converter. Moreover, the digital FETs employ balanced enhancement and depletion modes for ultra low power drain.

The GaAs foundry must find a compromise process to fabricate in common both the analog and digital devices with adequate yield. An optimized "conventional" GaAs foundry process for exclusively digital or analog transistors cannot be employed. Moreover, more than 90 db of gain must be accomplished across only 0.2 in. without any parasitic oscillations.

The GaAs MMIC RF chip outputs fully digitized (2 bits/sample) both in-phase and quadrature samples of the full 10 MHz IF bandwidth to the digital signal processor CMOS chip shown in Fig. 5. On this custom CMOS chip, carrier frequency and PN-code tracking along with C/A and P-code correlation is achieved in hardware. The signal processor chip outputs every 1 m-sec both in-phase and quadrature code cross-correlates (received signal with receiver stored reference) for the prompt, early, and late correlation channels used by the AAMP for data symbol detection and code tracking.

Table 3

MGR SUMMARY: MMIC AND SIGNAL PROCESSOR CHIPS

CHIP	FUNCTION
• A GaAs MMIC that contains all RF, IF, mixing and signal quantizer functions along with some high speed digital dividers for the synthesizer. This chip provides nearly 100dB of gain and has a noise figure under 3 dB.	Analog • Antenna/filter input • On-chip RF overload limiter • Low noise amplifier • SPDT switch to off-chip image-rejection filter • SPDT switch from off-chip image-rejection filter • 1st downconversion mixer (dual-gate FET) • 1st IF stage and lowpass filter • 1st IF amplifier • In-phase and quadrature 2nd downconversion mixers • 2nd IF stage with active band limiting filter • 2nd IF amplifier and AGC • Signal digitizer • Buffers to drive off-chip signal processor. Digital • L-band voltage-controlled oscillator • Buffer amplifiers—1st mixer • On-chip digital dividers to synthesze second IF I and Q injections • On-chip SPDT switch logic.
• A single digital signal processor chip that demodulates the GPS signal and provides 1 ms signal integrations to the data processor. The chip contains all code VCO and code generation functions.	• Carrier phase rotation • Carrier VCO • Code generation • Code removal • Code VCO • Signal integration • Built-in test • Timing and bus interface.

Table 4

MGR SUMMARY: SYNTHESIZER, MICROPROCESSOR, AND BLUE CHIPS

CHIP	FUNCTION
• A frequency synthesizer chip that includes the intermediate speed dividers and locks the VCO to a frequency standard. An on-chip dual-mode oscillator or an external oscillator can be used as the frequency standard	• Frequency standard • Phase detector • Loop filter
• A single-chip advanced architecture microprocessor (AAMP) that contains on-chip floating point operations and has sufficient throughput to handle all the processing for a two-channel GPS set	• Real time executive • Application software initialization • Control/display drivers • Navigation • Receiver manager • Receiver processing • Signal preprocessing • Satellite database manager • Satellite position/velocity processing • Built-in test • Utilities.
• A single multifunction interface chip that incorporates all the functions required for controlling one- two- and five-channel GPS sets. It also contains the memory and interrupt controllers, as well as low frequency set timing functions.	• memory control • Interrupt control • Frequency/time generation • Serial interface • Built-in test.

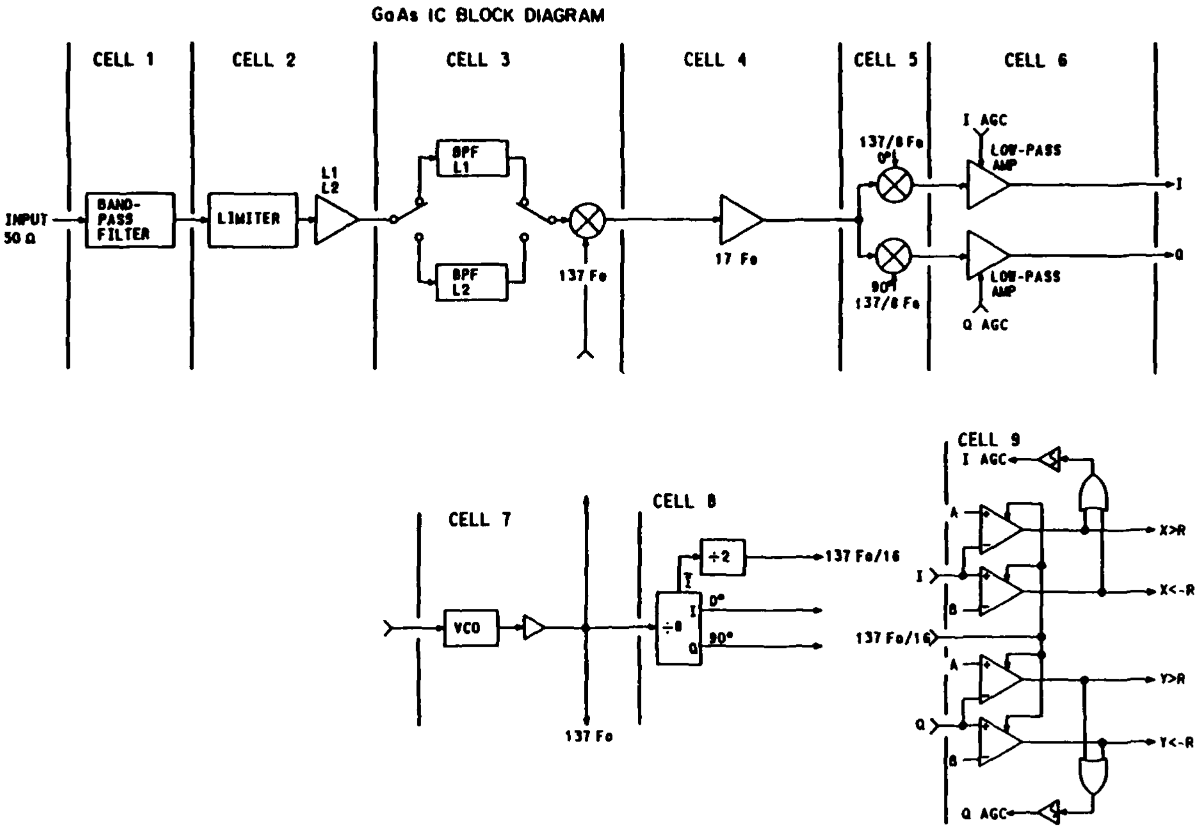


Fig. 3—GaAs MMIC Block Diagram

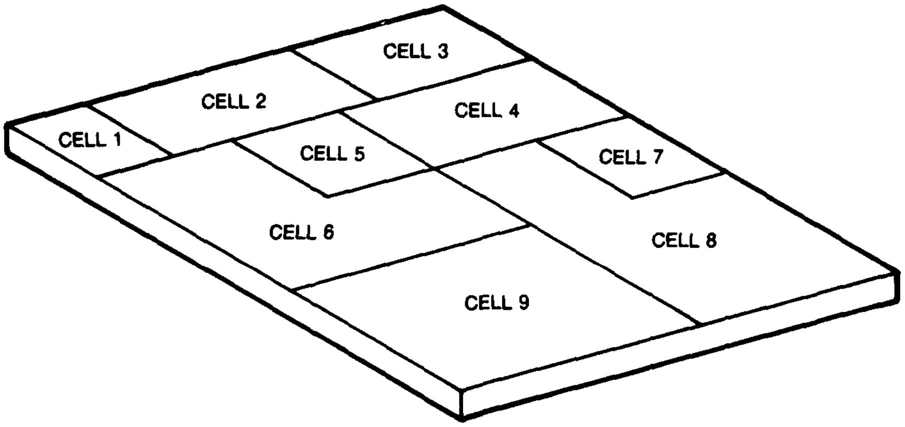


Fig. 4—GaAs MMIC Floor Plan

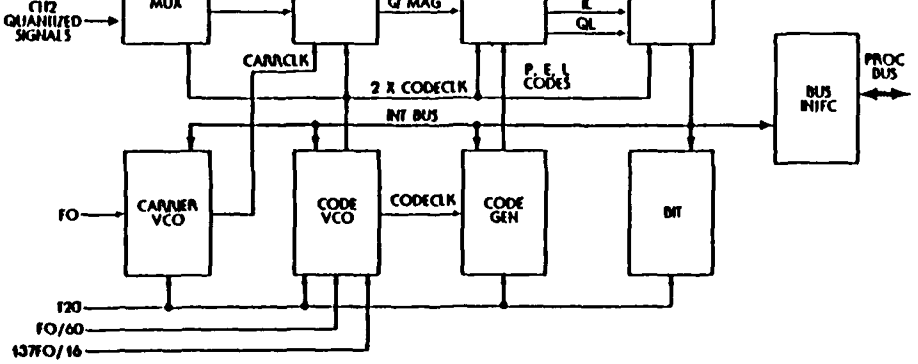


Fig. 5—Signal Processor Function Block Diagram

The signal processor output data are supplied every millisecond to the AAMP microprocessor where real time signal processing software further processes this 1 KHz data down to 50 Hz bandwidth (20 msec correlates) or less for GPS system data detection as well as signal carrier and code tracking. The error null steering commands determined by AAMP tracking loop software modules are returned by the carrier and code VCO steering control lines on the signal processor chip.

Examples of all the Rockwell Collins 2-channel MGR chips set are now functioning in a navigating breadboard, including the GaAs MMIC chip. The challenge of realizing low noise high RF gain (without oscillation) concurrent with gigabit digital logic gate speed on a common MMIC has now been successfully demonstrated.

FIBER OPTIC GYRO

Two sets of performance goals for the IMU components of the GGP are shown in Table 5. These values are based on improving the current generation of Fiber Optic Gyros (FOG) and the availability of Silicon Accelerometers. The early time frame performance in FOG is almost achievable now but with discrete fiber parts instead of the desired integrated optics. The early DARPA goal is to incorporate integrated optics and a modest improvement in current performance. The later goal is to match navigation grade Ring Laser Gyro (RLG) performance.

Table 5

IMU PERFORMANCE GOALS

	EARLY DEMONSTRATION	MIDTERM DEMONSTRATION
GYRO DRIFT BIAS	0.01 DEG/HR	0.003 DEG/HR
GYRO SCALE FACTOR	50 PPM	10 PPM
GYRO RW COEFFICIENT	0.005 DEG/$\sqrt{HR}$	0.0015 DEG/$\sqrt{HR}$
ACCELEROMETER SCALE	50 μG[a]	10 μG[a]
ACCELEROMETER SCALE FACTOR	100 PPM	50 PPM
ACCELEROMETER RW COEFFICIENT	0.03 (M/S)/$\sqrt{HR}$[b]	0.03 (M/S)/$\sqrt{HR}$[b]

[a] Micro G's of gravity.

[b] Meters per second per root hour.

Sagnac effect [3,4,5] based rotation rate sensors establish a pair of contra-propagating beams of light in a planar light guide circuit having exquisite optical symmetry (reciprocity) between the clockwise and counterclockwise propagation paths around the light circuit. Mechanical rotation rate measurably upsets this symmetry which can be photoelectronically detected and processed to provide a rotation rate output. A broad characterization of Sagnac effect rotation rate sensors (RLG, IFOG, RFOG) is shown in Fig. 6. The ring laser gyro (RLG) is now becoming commercially available with a very high grade of inertial measurement quality. Although a major improvement over mechanical gyro technology, it was judged to suffer the following drawbacks with respect to the DARPA GGP goals:

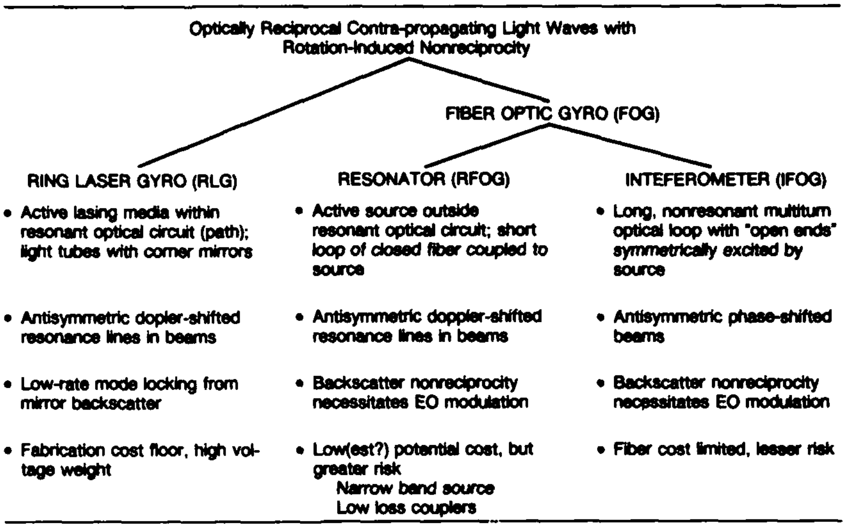
(a) Not amenable to employing integrated optics

(b) Requires complex glass machining with very high quality corner mirrors

(c) High voltage discharge needed to excite laser

(d) Mechanical motion dither needed to break up measurement deadband caused by common mode locking between contrapropagating beams

Of these drawbacks, the second item, (b), may through extensive development and production experience reduce present costs by creating a large industrial robotic glass machining base. Mechanical path dither, currently used, may be replaced by an electro-optic dither technique. However, these evolutionary RLG tech base improvements are deemed not commensurate with an early time frame, nor are they expected to reduce the rotation sensor costs to the level expected of a successful (albeit risky) fiber optic gyro, FOG, effort.



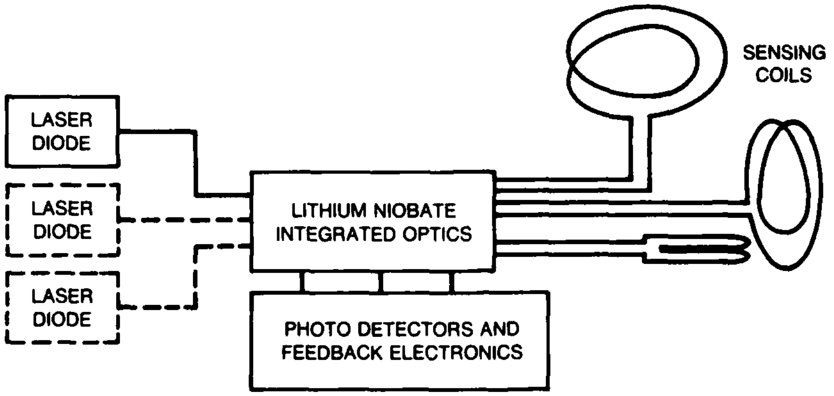
Optically Reciprocal Contra-propagating Light Waves with Rotation-Induced Nonreciprocity

FIBER OPTIC GYRO (FOG)

RING LASER GYRO (RLG)	RESONATOR (RFOG)	INTEFEROMETER (IFOG)
• Active lasing media within resonant optical circuit (path); light tubes with corner mirrors	• Active source outside resonant optical circuit; short *loop of closed fiber coupled to* source	• Long, nonresonant multiturn optical loop with "open ends" *symmetrically excited by* source
• Antisymmetric dopler-shifted resonance lines in beams	• Antisymmetric doppler-shifted resonance lines in beams	• Antisymmetric phase-shifted beams
• Low-rate mode locking from mirror backscatter	• Backscatter nonreciprocity necessitates EO modulation	• Backscatter nonreciprocity necessitates EO modulation
• Fabrication cost floor, high voltage weight	• Low(est?) potential cost, but greater risk *Narrow band source* Low loss couplers	• Fiber cost limited, lesser risk

Fig. 6—SAGNAC effect

From these considerations DARPA selected the FOG sensor approach to be assembled with an integrated optics chip, using crystalline lithium niobate and conceptually shown in Fig. 7 Initially, each rotation sensing axis will be composed of its own sensing coil, IO chip, diode light source, and photodetector. With IO chip fabrication improvements, one chip can be shared by all axes. With increased optical source intensity, one source (with one IO chip) can be power-divided amongst all axes.



- SAGNAC NULLING FEEDBACK ARCHITECTURE — LARGE MEASUREMENT RANGE
- POLARIZATION PRESERVING FIBER (PPF) — INCREASED OPTICAL RECIPROCITY
- LONGER WAVELENGTH — ELIMINATE PHOTODARKENING & REDUCE BACKSCATTER
- INTEGRATED OPTICS — LOW COST

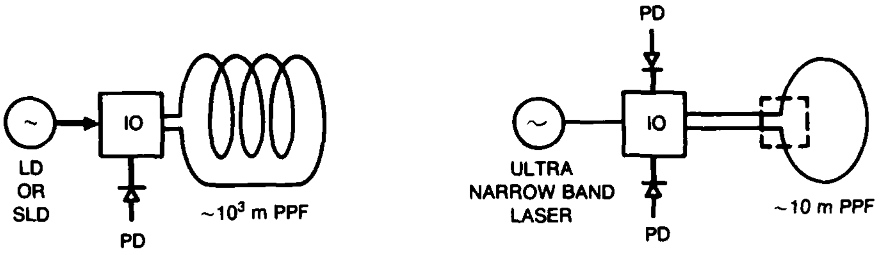
Fig. 7—Three-Axis Fiber Optic Rotation Sensor

There are two fundamentally different implementations of a FOG sensor; a resonant structure or RFOG, and an interferrometric structure or IFOG. An oversimplified comparison of the properties of the RFOG and IFOG are summarized in Fig. 8.

The RFOG [6] utilizes a short loop of fiber as an extremely high Q resonant light circuit. Ideally, when there is no rotation input, each of the two contra-propagating light beams remain trapped in the fiber sensing coil at the resonant light frequency and no light escapes[2] the coil coupler structure. Mechanical rotation doppler offsets the beams from the coil resonance frequency and light begins to escape from the coil. The special optical coupler (Fig. 8, "dashed box"), together with the lithium niobate IO chip, route the escaping coil output light

[2] Ideally, the optical source power is fully absorbed with carefully matched loss elements in the resonant fiber loop.

(under mechanical rotation) for each circulating beam to photodetectors; one for each beam direction. Since the beams are doppler shifted off resonance in opposite (light) frequency directions, the corresponding photodetectors provide skew symmetric outputs.



- NON-RESONANT STRUCTURE
- COUNTER PROPAGATING BEAMS MIXED ON COMMON PHOTO DETECTOR
- OPTICAL RECIPROCITY REQUIRED

- OPTICALLY RESONANT STRUCTURE
- COUNTER PROPAGATING BEAMS PROCESSED ON SEPARATE PHOTO DETECTOR'S
- ELECTRICAL SYMMETRY REQUIRED

RISK/COST UNCERTAINTIES

- PERFORMANCE
- SLD
- 10^3 m FIBER

- PERFORMANCE
- COIL COUPLER
- NARROW BAND LASER

Fig. 8—IFOG or RFOG

Inasmuch as the short fiber loop is a delay line, the input-output optical system transfer function, H(jw), of the resonant optical structure has a periodic null pattern in frequency spaced about 40 MHz apart. Referring to Fig. 9, a single, laser diode optical source is physically divided in two to launch each of the contrapropagating beams. Each beam is then individually heterodyned by optical frequency modulators in order to place beams in separate (usually adjacent) resonant nulls of the sensing coil. In this manner, the RFOG effectively operates as two separate gyros multiplex sharing, in optical frequency and propagation direction, a common coupler fiber coil sensing structure. The two balanced and symmetric gyros operate in a push-pull mode so that even order channel impairments cancel.

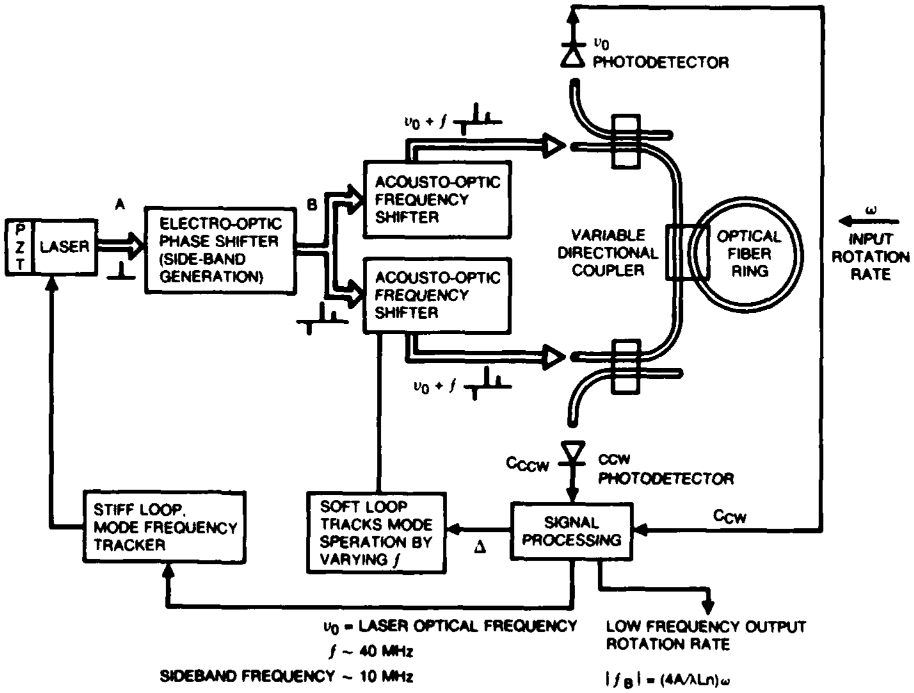

ν_0
PHOTODETECTOR
$\nu_0 + f$
ACOUSTO-OPTIC FREQUENCY SHIFTER
A
B
P Z T
LASER
ELECTRO-OPTIC PHASE SHIFTER (SIDE-BAND GENERATION)
ACOUSTO-OPTIC FREQUENCY SHIFTER
VARIABLE DIRECTIONAL COUPLER
OPTICAL FIBER RING
ω
INPUT ROTATION RATE
$\nu_0 + f$
C_{CCW}
CCW PHOTODETECTOR
STIFF LOOP, MODE FREQUENCY TRACKER
SOFT LOOP TRACKS MODE SPERATION BY VARYING f
Δ
SIGNAL PROCESSING
C_{CW}
ν_0 = LASER OPTICAL FREQUENCY
f ~ 40 MHz
SIDEBAND FREQUENCY ~ 10 MHz
LOW FREQUENCY OUTPUT ROTATION RATE
$|f_B| = (4A/\lambda Ln)\omega$

Fig. 9—Resonant Ring Gyro Control and Measurement Approach

In contrast to the RFOG, the IFOG structure [5,9] is a single gyro with one photodetector to which both contrapropagating beams additively output. The long (many turns) fiber coil is a nonresonant, very broadband structure. Consequently, a broadband diode light source can be used, which the present IFOG technology requires, as will be explained below.

The IFOG photodetector physically performs a mathematical crosscorrelation between the two contrapropagating output beams which were launched from a common light source. Ideally, with no path imperfections or asymmetries (optical reciprocity), the photodetector then produces the autocorrelation function of the light source after passage through the fiber coil. With no rotation, this outputs the peak of the diode source autocorrelation function. With mechanical rotation the path propagation times change producing an optical phase shift (as opposed to doppler frequency shift) between the two beams and an off-peak value of the autocorrelation function is read at the photodetector.

This simple, interferrometric, configuration produces electronic difficulties with the output signal. First, since the autocorrelation functions are even, rotation direction cannot be directly read. Second, the output measurement is centered around d.c., requiring more complex d.c. electronic amplification circuits. This situation can be corrected by introducing a subcarrier frequency modulation [7] at a multi-kilohertz frequency (100 KHz) on each propagating beam prior to photodetection as shown in Fig. 10. This is accomplished with an optical phase modulator on one arm of the coil. The phase modulator device periodically (at the FM subcarrier frequency) changes the optical path delay[3] through itself. Note that one beam is frequency modulated prior to entering the fiber sensing coil while the other beam is modulated after exiting the coil. This produces optical frequency modulations time shifted with respect to each other by the beam propagation time through the coil.

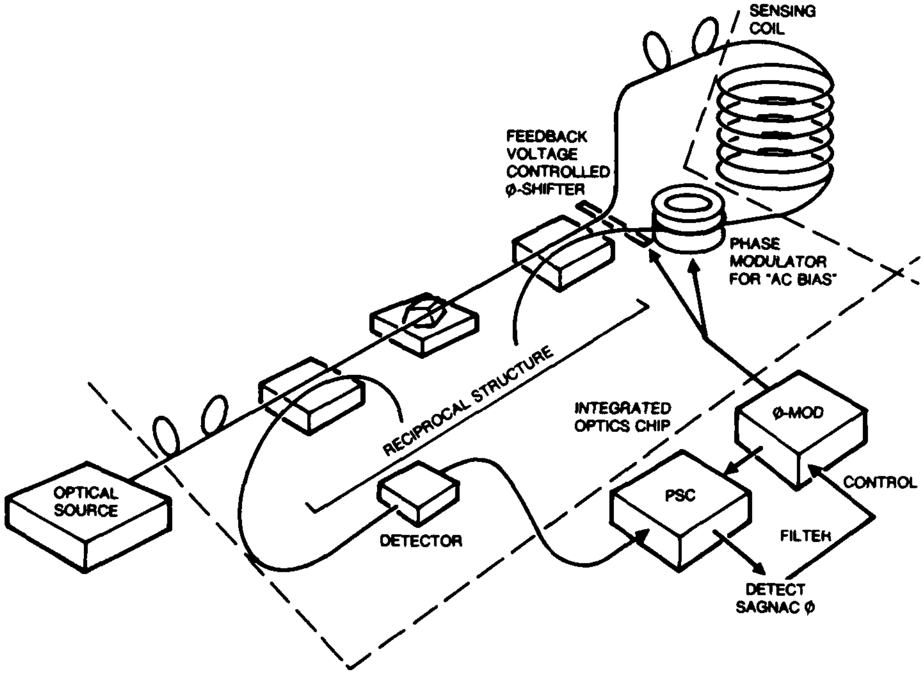


Fig. 10—Interferometer FOG with Feedback

The photodetector correlation output can be narrowband processed around the first harmonic of the FM modulation frequency and then synchronously detected by multiplying and smooothing the product of the first harmonic output signal with the same periodic signal driving the optical phase modulator. The consequences of this modulation/signal processing arrangement are to allow use of inexpensive a.c. photodetector output amplifiers while producing a skew symmetric detector output signal which is zero for zero rotation input rate. Rotation direction is then immediately given by the sign of this output.

[3] Time base variation is mathematically equivalent to imposing a frequency modulation.

A brief discussion on optical sources is now in order. Recall, that for the RFOG, employing the very high Q light circuit sensing coil, an extremely narrow laser diode optical source bandwidth is required to fit within the resonance linewidth of the resonant optical structure. Now, in general, the autocorrelation function of any narrowband waveform is an attenuated cosine function with a period equal to that of source (optical) wavelength and a rate of envelope decay equal to the reciprocal of the source bandwidth (i.e., "spectral linewidth"). The effective time duration of this correlation envelope time decay when multiplied by the speed of light produces a "length" referred to as the coherence length of the source. The coherence length of an RFOG source must be in excess of a mile!

Next consider the long optical path length (on the order of 1 km) used by a high sensitivity IFOG. Discrete imperfections in the optical path (for example, due to splices or connectors, nicks in the fiber cladding, impurities in the fiber core, etc.) will cause small-scale (1) optical reflections (backscatter), and (2) crosspolarization mode coupling (forward scatter). These unwanted optical interference signals are analogous to the multipath interference in a radio link. In direct analogy with radio multipath, spread sectrum sources [8] can be used to combat the multipath.

All such "optical multipath" interference sources separated from the desired optical signal or from each other by less than the coherence length of the source (i.e., autocorrelation envelope decay time) will upset the measurement accuracy. This results from the photodetector crosscorrelating the exiting light beams (desired plus interfering) of the fiber coil. This crosscorrelation produces not only the desired autocorrelation function of the optical source but is also corrupted by crosscorrelations between the "multipath" generated beams and the desired beams. However, those crosscorrelations separated by more than a source coherence length are zero. Consequently, short coherence length sources (large instantaneous bandwidths) considerably reduce the number of small inteference signals corrupting the measurement accuracy.

The IFOG detector processing depends on the wavelength periodicity of the optical source autocorrelation function and not its envelope. Consequently, an optical diode source having large symmetrical instantaneous spectral bandwidth (e.g., Super Luminescent Diode, SLD) with a very stable center frequency produces an autocorrelation function still useful for IFOG operation but having a fast decay time (submillimeter coherence length).[4] Now only multipath source pairs within a coherence length (submillimeter) of each other can beat together to produce a degraded output and most of the multipath is rejected. Consequently, current IFOG technology is facilitat ' by use of broadband diode sources while the RFOG must use extremely narrowband lase. sources to excite their resonant structures.

In contrast to the IFOG, recall that the RFOG operates with a much shorter fiber and off-resonance frequency detection for each separately propagating beam and does not crosscorrelate the beams. Consequently, the RFOG is affected by optical multipath sources in a very different way than the IFOG. Here the resonance optical bandpass shape becomes distorted by multipath. The RFOG system phase transfer function must be kept skew symmetric about a small frequency region centered on the resonance point of the fiber coil.

Since the IFOG sensor has been in active development in industry for a longer time, it is much closer to being implementable in a nearer term, flight-testable GGP prototype. For example, the broadband optical sources, although expensive, currently exist while available narrowband solid state sources are still too wide (~1 MHz vice the 100 KHz or less needed). Consequently, the IFOG rotation sensor is the initial choice for the GGP based on early availability, but interest remains high at DARPA in the RFOG sensor technology.[5]

So far in the discussion, both FOG architectures suffer from a very limited measurement dynamic range. That is to say, the ratio of the maximum rotation rate usefully measurable to the random noise output is not as large as desired. The RFOG resonant structure is so narrowband that the higher rotation rate doppler shifts the

[4] In theory the source also could be broadbanded by electronically induced modulation. Alternatively, other signal processing techniques for multipath rejection might be employed.

[5] The reader is put on notice that DARPA views the RFOG and IFOG, although both rotation sensors, as distinctly different transducers with unique and different advantages, uses, risks, and cost. They are not viewed as a priori competitive technologies for the same universal application.

source out of the resonance while the IFOG ccsine autocorrelation function[6] becomes ambiguous for large Sagnac phase shifts resulting from higher rotation rates. These effects produce measurement (not device) failure.

Actual measurement dynamic range is set by even lower maximum permissible rotation rates. Namely, as the rotation rates increase the measured output becomes nonlinear with input mechanical rate; i.e., nonlinear scale factor. Consequently, for increasing overall measurement precision, there is a further decrease in useful dynamic range.

The means to increase the dynamic range is to employ measurement feedback (Figs. 9 and 10) creating a closed loop FOG sensor. As the Sagnac effect develops and is sensed at the photodetector(s), feedback electronics filter the signal and then input to a voltage-controlled optical phase or frequency shifter(s) on the integrated optics chip. In this way the Sagnac optical signal can be nulled out. The rotation rate measurement is taken from the drive voltage to the voltage-controlled optical shifter. The Sagnac optical signal is processed as the error signal input to a tracking feedback loop. This technique is the optical analogue to the Phase Lock Loop (PLL) for the IFOG, and Automatic Frequency Control (AFC) for the RFOG.

To date, laboratory IFOGs have been assembled using discrete fiber optical parts (e.g., couplers, polarizers, splitters, modulators) operating with light sources at 0.83 μ wavelength. The net drift rate bias has been at the 0.1°/hr or less regime. Some efforts have already installed the feedback architecture and employ polarization preserving fiber (PPF) in the sensing coil. The PPF coil considerably imroves optical reciprocity of the light path by drastically reducing the level of unwanted crosspolarized (nonreciprocal path) light being crosscorrelated on the photodetector. All DARPA efforts for future high performance IFOGs will incorporate these features.

Further efforts [8] at implementing individual or a few optical parts (y-splitter, coupler, polarizer, phase modulator) on lithium niobate chips have succeeded. The DARPA effort will emphasize accomplishing a high level of optical integration on one Integrated Optic (IO) chip. The long-term goal is to place all optical parts on the IO chip excepting the light source.

The IO fabrication method deposits etched strip lines of titanium on a lithium niobate crystal substrate. These titanium lines are then heat diffused into the crystal forming light guides whose interguide propagation coupling properties are carefully controlled. These optical couplers and splitters are passive linear "circuits." Optical modulators and switches are obtained by using the electro-optic effect in lithium niobate where the light propagation velocity (refractive index) is a function of the strength of an imposed external E field. Capacitive "plates" are deposited over desired portions of the chip with electric control lines to which off-chip drive electronics attach. The process is quite similar to that used by solid state integrated circuit foundries (except fewer foundry steps are needed). It should be noted that lithium niobate chips produce intrinsically polarization preserving light guides, which enhances the use of PPF sensing coils.

It has been well known that lithium niobate crystals are susceptible to optical damage. It is believed that this damage is caused by energetic photon collisions with atomic iron impurities embedded in the lithium niobate lattice structure. At appropriate photon energies (i.e., optical wavelength) the outer electron shells of the iron become excited with very long relaxation time. Unfortunately, the present 0.83 μ wavelength is such an appropriate photon energy level.

When the incident rate of the energetic photons (i.e., light intensity) reaches a threshold value, enough excited iron nuclei impurities are produced to distort the lithium niobate lattice structure and produce visible photo-darkening effects. Above onset threshold, the photo-darkening is proportional to time accumulated light energy usually taking tens of minutes. When the light is removed, the crystal structure can relax back to its normal condition, but may take a week or more to do so.

There appears to be no reasonable prospect to significantly reduce iron impurities in today's lithium niobate foundry processes. Current telecommunications

[6] It is made into a sine rotation detector characteristic by the phase modulator.

experience at the 0.83 u wavelength has detected photo-darkening onset at milliwatt light intensity levels, a light intensity level which most IFOGs would be well under. Unfortunately, this measurement for darkening onset was made based on intensity loss from input to output of the chip with primarily unidirectional propagation through the chip (in comparison with bidirectional for the IFOG). With the IFOG's critical dependence on optical reciprocity, the light intensity onset threshold could be considerably reduced, especially for high levels of functional chip integration (i.e., optical devices on the chip).

For the GGP, this risk to successful IO chip employment has been judged too high! Fortunately, longer wavelength fiber technology (1.3 to 1.55 µ) does not suffer from the effect as the individual photon energy is too low to excite the iron impurities. Even more fortuitously, there is considerably greater optical quality in the fiber parts at the longer wavelength (e.g., lower loss 1/2 db/km vice 3 db/km, reduced scatter, increased polarization extinction in PPF, etc.) while the lithium niobate chip fabrication is more tolerant in its critical dimension requirements. It is to these longer wavelengths that the telecommunications industry migrated successfully several years ago.

The 0.83 u wavelength was originally employed in IFOG research because of the availability of quality diode optical sources with adequate power level and photodetectors. Currently, quiet photodetectors are readily available at the longer wavelengths while broadband superluminescent diodes, SLD (remember a quality spread spectrum source is needed for multipath scatter reduction) are just now becoming available at these wavelengths. It is on this longer wavelength optical technology that the GGP will evolve for lower cost (using nonphoto-damaging IO chip) and obtain better performance (higher quality fiber).

Listed below are current research areas to the longer wavelength closed loop IFOG effort by DARPA for its GGP.

o Component Level
    Light source--SLD, LD, LED
    IO chip & optronic devices--voltage controlled optical phase modulator, high-level integration
    Physical characterization--reciprocity, scattering mechanisms, optical loss, g load & thermal gradient effects
    Connectors/splicers--alignment, reflections

o System Level
    Closed loop signal processing and sensor modeling
    Error analysis and performance prediction
    Parameter optimization

The current GGP IMU effort is completing several preliminary designs of a closed loop IFOG along with selection and integration of miniature accelerometers with performance at the 0.01°/hr level given in Table 3. In the next phase of the GGP program, two designs will be selected and parallel efforts to build IMU brassboards of the selected designs will be pursued. These brassboards are to commence flying-laboratory tests within three years of contract award. For this GGP flight testing, an MGR will also have been integrated, as discussesd in the following section.

GGP SYSTEM INTEGRATION

System integration effort for the GGP is being conducted at the functional levels, listed below:

o Mechanical Packaging
    Form/Fit, Temperature, Shock

o System Data Processor/Bus Hardware
    Thruput, Memory, Sensor Interface Control

o System Software Modules
    Filter Design, Resource Manager

o System Simulation/Trade-off Optimization
    Covariance and Monte Carlo Simulation

- User Mission Analysis
  Trajectories/Profiles, Performance Measures

With the exception of the mechanical integration, there is strong interaction amongst the other system integration functions, listed above.

Since the modular GGP is planned for a wide variety of host/mission applications on the one hand, and the system scope is both broad and interconnected on the other hand, the system's effort must be partitioned by an appropriate development philosophy in order to avoid needlessly locking out useful GGP applications while apportioning the system's effort into manageable assignments. The key requirements imposed by DARPA to achieve this goal is (1) to host guidance/navigation solution functions as well as top-level resource control with all attendant software modules strictly in the Navigation Systems Microprocessor(s) (Fig. 1), and (2) to maintain strict control over electrical/data exchange interfaces/standards between the two GGP sensor suites (MGR and IMU) and the system microprocessor.

A consequence of this approach may produce a fairly significant throughput, memory space, and coding burden on the system's microprocessor. This will be strongly driven by the size (i.e., number of states modeled) of the Kalman Filter employed [10]. However, the latest high-speed commercial microprocessors (possibly RISC machines) should be able to accommodate the expected thruput of several MIPS. The standard, multimegabit high-speed backbone data bus will be used with either 16 or 32 bit width.

Software development will initially focus on the following: (1) implementing the Kalman Filter, (2) separating out of the existing software modules in the MGR and IMU sensors their low-level real-time signal procesing modules from their resource management modules, (3) determining software data access and command methods from top-level resource management in the system's microprocessor to the low-level real-time software in the sensor microprocessor, and (4) reconcile data formats and timing actions. Resource management decision tree/execution action tables will be developed for various host applications. Finally, the software will be thoroughly tested and validated.

System simulations will be performed at two distinctly different, but highly interactive, levels. At the aggregate system modeling level, mathematical input/output models for the MGR, IMU, and physical environment (e.g., gravity anamoly) are combined together with a Kalman Filter in "Covariance" simulations for specified mission dynamic profiles. Selective "Monte Carlo" simulations will then be run. Such analysis relates instrument performance requirements, generates expected system behavior, and predicts mission performance. It provides a "specification dialogue" between the user application and the GGP modular design as well as eliciting what are critical system drivers and the measures of merit/performance appropriate to each class of application. An example of this will be provided in what follows.

The aggregate system modeling simulations will not test the adequacy of the real-time operational system-level software nor the specific behavior of the MGR and IMU components and their dedicated low-level real-time software. Moreover, the dynamical detailed interactions between the three subsystems needs to be verified. To do this a sequence of hybrid-brassboard simulations will be used wherein a mixture of real-time computer simulation with hardware simulation will be set up for a selected class of mission. As GGP sensor instruments become available they will be inserted in the brassboard. The end state of the sequence will be an all-GGP brassboard customized for the selected mission class.

Returning to the aggregate system-level modeling, a series of preliminary results were obtained for two classes of mission: (a) an Unmanned Vehicle or UMV and (b) a long-range strike weapon. Representative mission trajectories were "flown" postulating a high quality IMU in the 0.003°/hr gyro drift rate and 50 μg accelerometer bias category.

GPS received signal was then denied (e.g. jammer) at selected parts of the mission and IMU only guidance error buildup was calculated. In addition to the IMU error sources, earth gravity anomalies and vertical deflection were modeled as a first order Markov process.

The modeled MGR, IMU, vehicle motion, and gravity inputs were optimally combined in a 48-state system Kalman filter. Additionally, a 78-state model is executed as the "truth" model. The covariance simulation then flies the mission trajectory (in non-real-time) accumulating as a function of time the 1 σ state errors for both the truth model (78-state) and the system filter (48-state). The 48-state system filter is deliberately overstated in order to identify the most significant 17 to 23 states to retain in an operational filter formulation that can execute in real time.

Finally, a perfect IMU was postulated in order to benchmark the error floor introduced by the (unmapped) gravity anomalies and vertical deflections. The primary intent of this exercise was to build the capability to make these calculations and develop preliminary insight to relate IMU quality to mission needs. The mission trajectories are purely hypothetical and were used only to develop this system level skill. They have no validity as to any specific real missions.

Shown in Figs. 11, 12, and 13 are three sample mission trajectories used with GPS signal loss introduced. Figs. 11 and 12 are two unmanned vehicle trajectories in which the first passes by a single jammer and the second passes by a series of jammers. Fig. 13 shows in plan view a long-range strike weapon-type trajectory with a jammer in the immediate neighborhood of the target.

The model results plotting one sigma horizontal displacement errors versus time are shown in Figs. 14, 15, and 16. The solid lines are the modeled "true" 1 σ errors and the dotted lines are what the modeled system Kalman filter "says" are the 1 σ errors. The corresponding results for the gravity errors only case with zero instrument error showed only a few meter increase in σ_H. This establishes that additional improvements in IMU performance beyond the .003°/hr level (Table 5) will require maps of gravity anomaly and vertical deflection, thus increasing somewhat system memory and requiring increased microcomputer capacity. In all probability these considerations will be less demanding than the added mission planning burden, i.e., obtaining gravity surveys.

CONCLUSIONS

The DARPA GGP program objective to exploit GPS/IMU synergisms in an extremely small package with little burden to an extremely wide variety of host vehicles and applications is being actively pursued. The goal to achieve current navigation grade of performance (~ 1 nm/hr.) but with major per unit cost reductions in large-scale production should be achievable. Except for the IMU, solid state technology is now either off the shelf (system integration microprocessor) or becoming available (mini-GPS receivers). The silicon accelerometers are currently under active development by DoD. Consequently, the Fiber Optic Gyro (FOG) is the primary technology needed to produce a successful GGP.

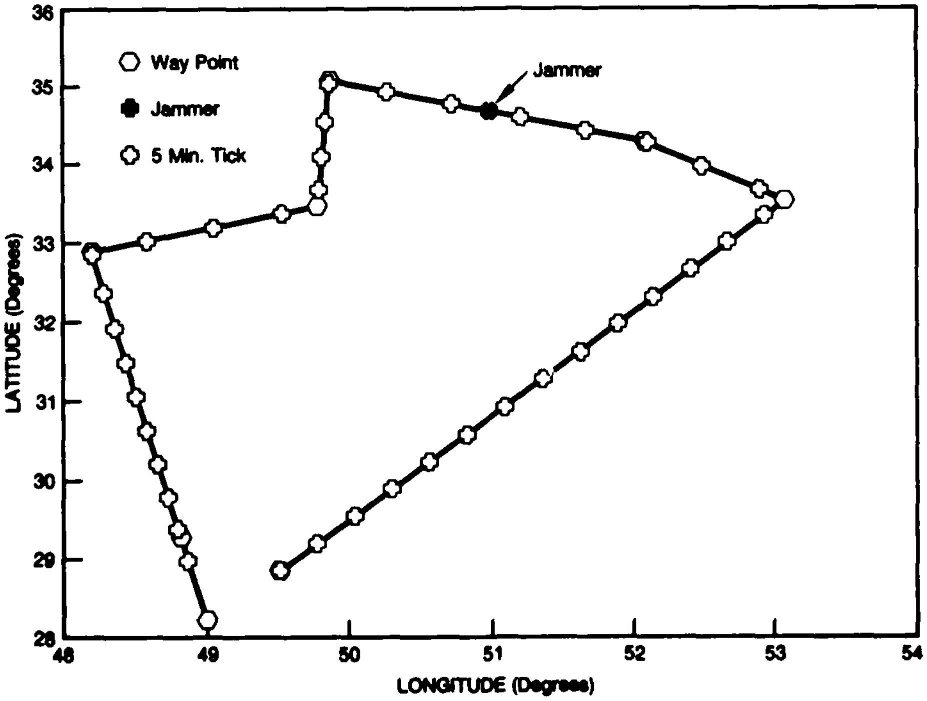


Fig. 11—Ground Track (UMV-1)

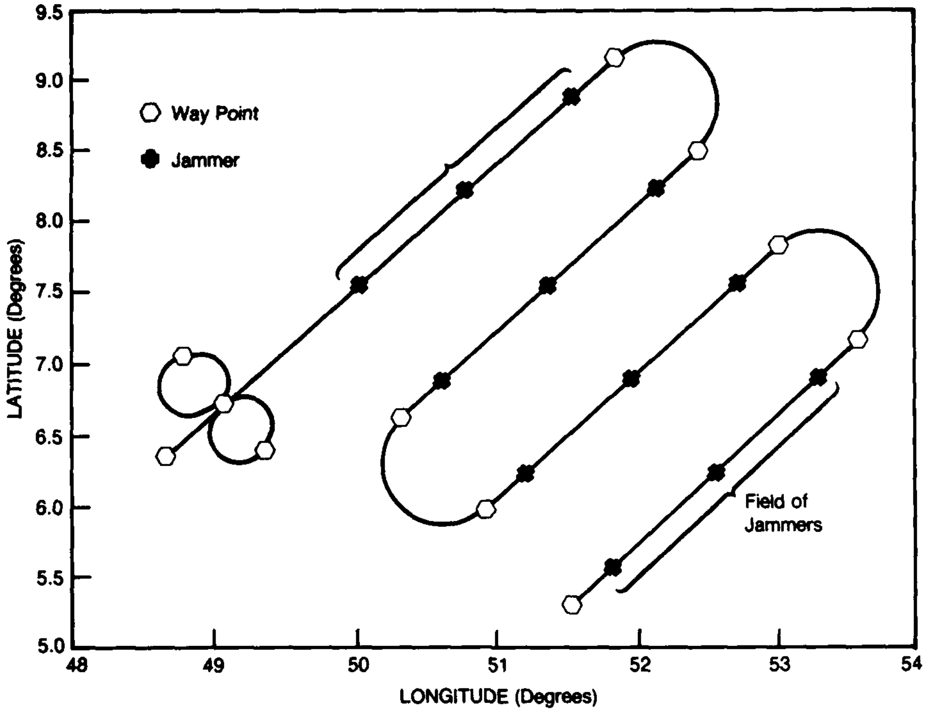


Fig. 12—Ground Track (UMV-2)

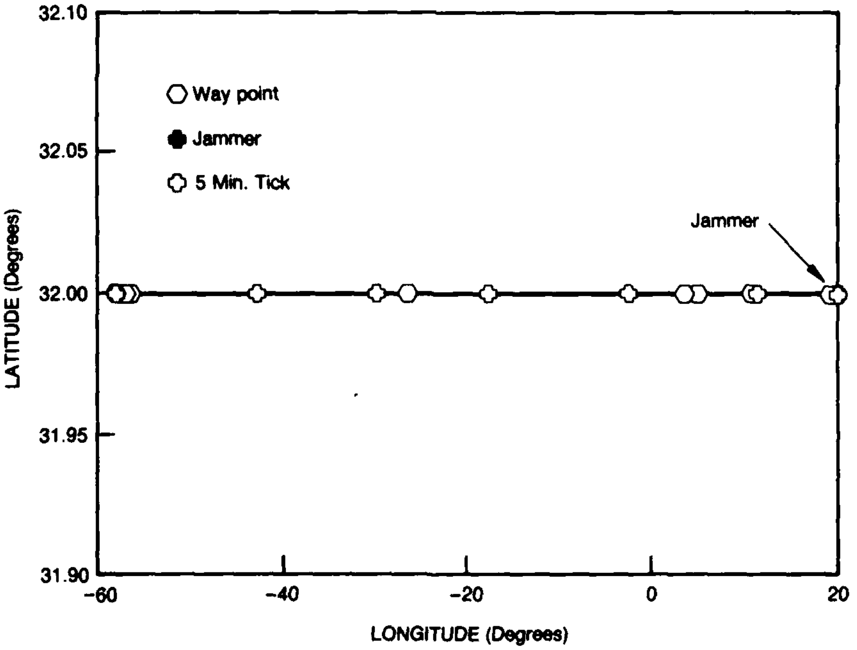


Fig. 13—Ground Track (SW-22)

In this paper the principal technical approaches being pursued in the DARPA GGP program have been reviewed along with discussion of the issues associated with these approaches. The performance and size goals of the GGP (i.e., FOG) can be met with low to moderate risk. The higher risk resides in making the cost reduction breakthroughs for the FOG instruments. This risk principally falls on the following FOG components: (1) integrating high-level optical signal processing parts on the integrated optics chip, (2) producing higher power, long wavelength optical sources, and (3) automated parts assembly machinery and infrastructure, e.g., sensing coil winders and automated polarization aligning connections/splices between the optical subassemblies of source, chip, and coil. These challenges are commensurate with the broad DARPA charter to significantly advance the DoD technology base.

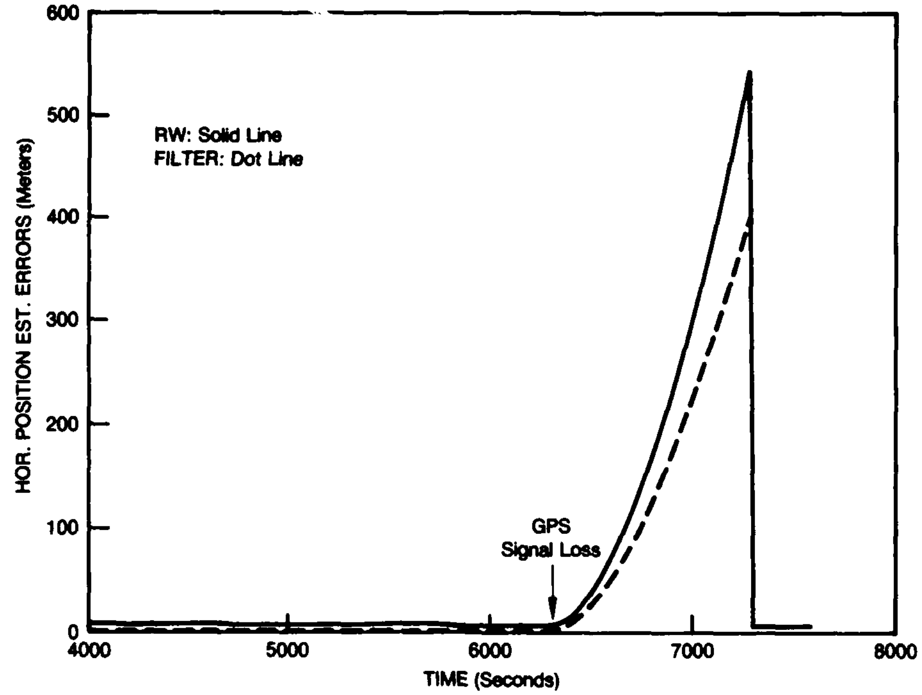


Fig. 14—Estimation Errors of Navigation States (UMV-1)

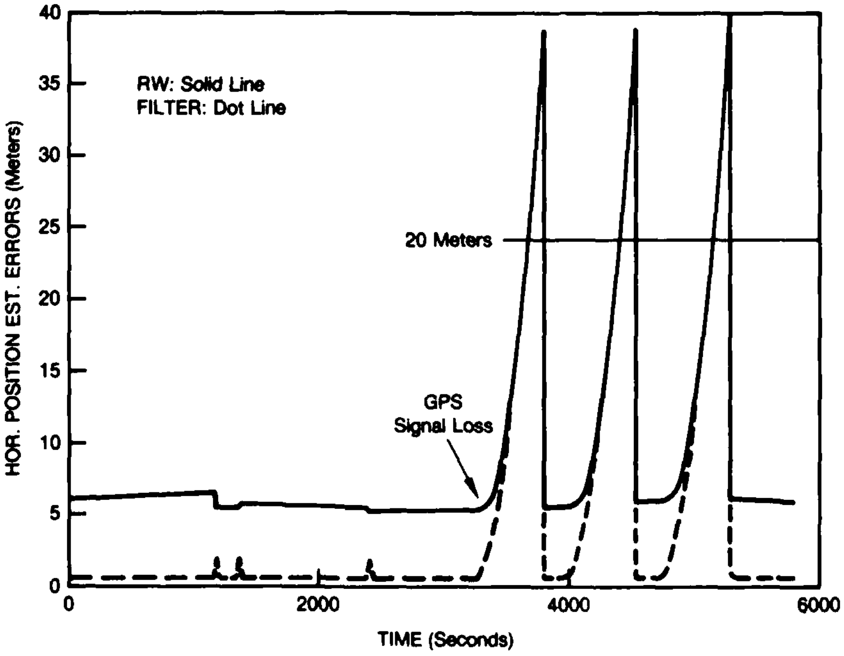


Fig. 15—Estimation Errors of Navigation States (UMV-2)

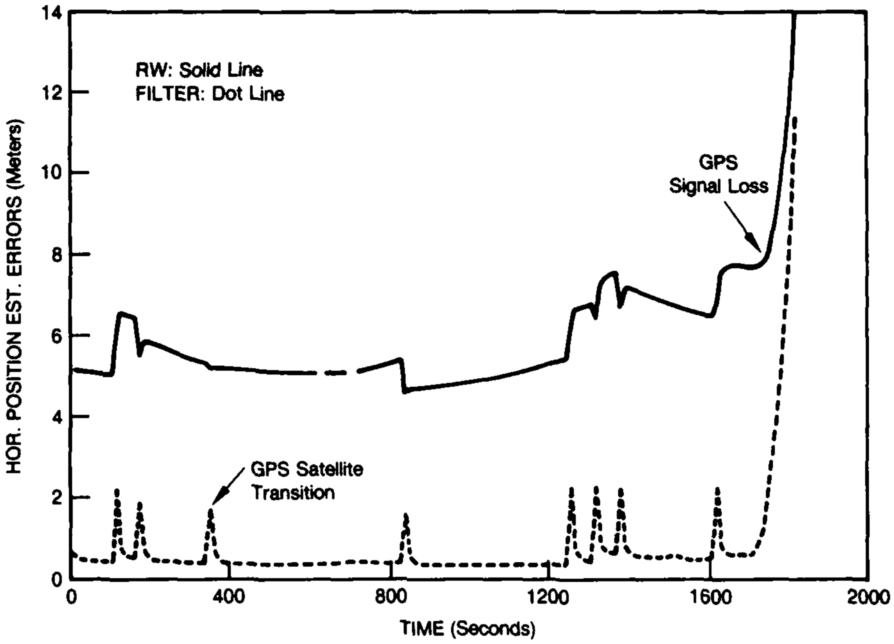


Fig. 16—Estimation Errors of Navigation States

REFERENCES

1. Hemesath, N. B., J.M.H. Bruckner, R. J. Weber, and J. P. Young, "A Miniature GPS Receiver," Microwave Journal, September 1987, pp. 1-104

2. U.S. Air Force, "System Segment Specification for the User System Segment, Navstar Global Positioning System, Phase II," 31 January 1979

3. Chow, W. W., J. Gea-Banacloche, and L. M. Pedrotti, "The Ring Laser Gyro," Reviews of Modern Physics, Vol. 57, No. 1, January 1985

4. Ezekiel and Arditty, "Fiber-Optic Rotation Sensors Tutorial Review," Springer-Verlag Optical Science Series, Vol. 32, pp. 2-26, November 1981

5. Bergh, Ralph A., H. C. Lefevre, and H. J. Shaw, "An Overview of Fiber-Optic Gyroscopes," J. Lightwave Tech., Vol. LT-2, No. 2, April 1984, pp. 91-107, F-1a 16

6. Carroll, R., C. D. Coccoli, D. Cardarelli, and G. T. Coate, "The Passive Resonator Fiber Optic Gyro and Comparison to the Interferometer Fiber Gyro," Fiber Optic gyros: 9th Anniversary Conference, SPIE, Cambridge, Mass., Vol. 719

7. Page, J. L.,"Fiber Gyro with Electro-Optic Phase Modulation," SPIE, Cambridge, Mass., September 1986

8. Fredricks, R. J., "Scattering Matrix Analysis on the Use of Wideband Laser Source in a Passive Fiber Rate Sensor," Springer-Verlag Optical Science Series, Vol. 32, November 1981, pp. 82-92

9. Bartman, R. K., B. R. Youmans, and N. M. Nerheim, "Integrated Optics Implementation of a Fiber Optic Rotation Sensor: Analysis and Development," Fiber Optic Gyros: 10th Anniversary Conference, SPIE, Vol. 719, pp. 122-134

10. Sorenson, Harold W. (ed.), Kalman Filtering: Theory and Application, IEEE Press, New York, 1985

RECHERCHES SUR LES SYSTEMES D'AIDE A LA NAVIGATION BASES SUR LES HEURISTIQUES DE PILOTES DE COMBAT

ADVANCES IN NAVIGATION SUPPORT SYSTEMS BASED ON OPERATIONAL PILOT'S HEURISTICS

DEBLON F. (*), GUENGANT A. (**), VALOT C. (*), AMALBERTI R. (*)

(*) Département d'Ergonomie Aérospatiale, CERMA,
5bis, Avenue de la Porte de Sèvres, 75731 PARIS 15 Air Cedex, FRANCE.

(**) Avions Marcel DASSAULT-BREGUET Aviation,
78, Quai Marcel Dassault, 92214 SAINT-CLOUD, FRANCE

SUMMARY

The conjoncture of the future battlefield (many threats, single seat combate aircraft) will impose the development of onboard decision support systems. The concept of pilot's assistant fits this need in assistance. While this concept covers different classes of aids, this paper focuses on the development of a navigation support system. Special attention has been paid to the quality of man-machine coupling of such a real-time aid. It is evidenced that, because the quality of the coupling is a vital goal, the best solution consists in computerizing the navigation expertise with the greatest proximity of human being (and not with optimal multi-expert and/or multi-constraint software). Thus, pilot expertise elicitation has been closely conducted during 4 years in nap of the earth penetration mission. A AI computer model of navigation is derived from this cognitive model with respect to the use of actors language. Extended description of this program is given in the paper, including direct connections with the future of the French's pilot assistant.

Key-words: pilot's assistant, cognitive model, expertise, schemata, actors language.

Résumé

L'environnement aérien et terrestre futur du champ de bataille imposera par sa complexité de munir les avions de combat de systèmes d'aides. Cela sera particulièrement nécessaire dans le cas du futur avion de combat français monoplace. Le concept de copilote électronique développé dans plusieurs pays répond à ce besoin. Il pose un défi technologique et ergonomique. Le problème du couplage d'un tel système avec le pilote fait l'objet ici d'une analyse détaillée et ses implications sur l'architecture d'un système d'aide sont analysées. Nous nous plaçons plus particulièrement dans le cas de la mission de pénétration basse altitude et nous montrons l'intérêt d'un modèle cognitif du pilote de combat. A partir de l'analyse de l'expertise de pilotes sur une telle mission, nous développons un projet d'automate de navigation basse altitude capable d'adaptation aux incidents du vol. Ce programme est calqué sur le modèle cognitif et utilise les notions de langage d'acteurs.

Mots-clés : copilote électronique, modèle cognitif, expertise, schéma, langage d'acteurs.

1 INTRODUCTION

Les missions aériennes impliquent, pour le pilote qui les conduit, de faire face à une complexité croissante l'environnement extérieur ainsi qu'à la nécessité de maitriser des systèmes embarqués sophistiqués. Ce constat, vrai à l'heure actuelle, prendra une tournure encore plus critique dans l'avenir. Les scénarios concernant le théâtre européen prévoient pour les forces aériennes un déséquilibre de 1 à 6, voire 1 à 10 ; la couverture anti-aérienne des corps de bataille sera extrêmement dense et diversifiée (**Congrès AGARD sur l'homme et les systèmes d'armes, Stuttgart, 1987**). Le domaine de réalisation des missions s'est lui aussi élargi : toujours plus vite, plus bas, de jour comme de nuit, par toutes conditions météorologiques.

Pour évoluer dans cette ambiance, le pilote des futurs avions de combat français, qui pour diverses raisons sera seul à bord, aura un besoin vital d'un système d'assistance à la conduite de mission. L'assistance à la navigation sera une partie de cesystème plus ambitieux appelé Copilote Electronique ou Pilot's Assistant.

L'objet de cette communication est en premier lieu de caractériser un tel système, et en second lieu, de présenter une architecture logicielle remplissant les spécifications précédemment définies, et orientée vers l'assistance à la navigation en mission de pénétration basse altitude.

2 SPECIFICATIONS POUR UN SYSTEME D'AIDE

2.1 Limites de l'automatisation

L'introduction massive de l'électronique à bord des avions d'armes a aidé le pilote en reculant considérablement son seuil de saturation. Seulement, il y a un revers à la médaille, et nous commençons à nous en rendre compte. La prise de conscience a eu lieu plus tôt dans d'autres domaines de conduite de processus, notamment dans le nucléaire, où fonctionnent déjà des automates régissant des tâches complexes. Or ce sont bien ces types d'automates qui devront être embarqués, vu la difficulté des processus à gérer à bord.

Associé à un automate effectuant une tâche complexe, l'opérateur humain ne suit pas en continu le déroulement du processus, mais contrôle uniquement de points en points. Donc très vite, il s'efface en tant qu'acteur impliqué dans la boucle de pilotage du processus, pour apparaitre comme un surperviseur [2] ; [3]. A la reprise en main du système, le passage de cet état de superviseur à un état actif peut prendre plusieurs secondes; à 600 kts et 200 ft/sol, ceci peut représenter tout l'écart entre la vie et le crash. Et des reprises en main, il y en aura, car outre leur complexité, les tâches de l'aéronautique militaire se déroulent dans un univers soumis à de rapides changements se traduisant par de fréquentes interruptions asynchrones (gestion de menaces, pannes, variations de la visibilité, ...).

Un autre effet pervers s'exerçant sur l'opérateur face à un automate est la rapide diminution de son niveau de connaissances, qui deviennent moins opérationnelles, car moins employées [4] ; [12] ; [17]. Au minimum, l'automatisation très poussée induit une standardisation des performances humaines, ce qui peut être souhaitable dans certains domaines, mais qui est loin de l'être en aéronautique de combat. Un couple appareil-pilote offrant une grande variabilité de la performances n'est-il pas préférable à un couple qui, en minimisant les risques de mauvaise performances, réduit aussi les possibilités de dépasser le standard ? [11].

2.2 Quelles aides et pour quel pilote ?

Le constat précédent sur l'automatisation nous amène à poser le probleme de l'interface homme/machine non pas comme un sous-problème à traiter en corollaire à l'automatisation des systèmes de bord, mais comme une réflexion centrale qui permettra justement d'organiser ces systèmes.

Il convient de définir la position du pilote dans l'appareil. A l'heure actuelle, force est de reconnaitre que nous sommes très loin de reproduire par des automates intelligents les capacités de flexibilité et d'adaptabilité de l'homme. Ses aptitudes à filtrer des informations, à poursuivre des buts en parallèle, à évaluer des situations à partir de données fragmentaires ou indirectes, à inventer des stratégies, imposent sa présence dans l'appareil.

Avoir un pilote dans l'avion, c'est choisir d'avoir un décideur plutôt qu'un exécutant: donc c'est organiser les aides à bord de telle manière qu'elles lui permettent le maximum d'anticipation et lui dégagent du temps pour se bâtir une représentation de la situation, seule condition pour maintenir le pilote en avance sur les événements.
L'aide peut être concue selon trois dimensions : (1) temporelle (court, moyen ou long terme), (2) nature de l'aide (acquisition des données, décisions, conduite), (3) destinataires de l'aide (pilotes experts ou

novices).

2.2.1 Dimensions temporelles de l'aide

Aider le pilote, c'est :

- à court terme, augmenter sa survivabilité en présentant l'évolution dans les instants qui vont suivre de la situation. On constate qu'une telle représentation imagée de ce qui va advenir dans l'immédiat si aucune action n'est entreprise est non seulement un moyen puissant d'appréhension de la situation, mais est aussi un levier pour le raisonnement. Ainsi, la symbologie du vecteur vitesse ne fait pas qu'indiquer le point d'impact de l'appareil, mais est aussi un instrument dans le raisonnement à court terme du pilote, qui en vol basse altitude, le positionne au ras des crêtes.
- à moyen et long terme, permettre l'évaluation globale de la situation et l'anticipation en présentant de manière synthétique les éléments pertinents pour le moyen et le long terme.

2.2.2 Nature de l'aide

Les aides à l'acquisition de données facilitent la formation chez le pilote d'une représentation correcte de la réalité. C'est un domaine où les technologies modernes peuvent beaucoup apporter avec par exemple la restitution de scène sur écran géant (option big picture) [1], le concept de tête moyenne [10], ou les jumelles solidaires des mouvements de tête [7]. Il s'agit là d'aides à l'acquisition fusionnant des informations d'origines parfois différentes sur un support unique.

Les aides à la décision et à la résolution de problèmes forment un pôle de recherches important avec les projets Pilot's Associate américains [16] et Copilote Electronique d'AMD-BA (communication personnelle [5]). Ces aides ne se placent plus au niveau de l'acquisition des informations, mais au plan de leur traitement et de l'élaboration des décisions. L'essor de l'intelligence artificielle a fortement contribué au lancement de tels projets d'aide au pilote. L'aspect technique est bien souvent le seul évoqué, laissant dans l'ombre le problème du couplage avec l'homme. C'est pourtant cet aspect qui doit déterminer les choix techniques si l'on veut laisser au pilote la primauté sur la machine et la responsabilité de la mission.

Un troisième type d'aide envisageable est l'aide à la conduite. Elle représente un prolongement naturel de la précédente, passant du conseil et de la définition de tactiques à l'exécution. Ce pilote automatique sophistiqué pourrait être branché sur ordre du pilote pour assurer la navigation suivant le plan de vol préparé, pendant une durée pouvant aller jusqu'à quelques minutes, laissant ainsi au pilote du temps libre pour des recalculs de route ou pour revoir l'approche de l'objectif. Le pilote doit pouvoir à tout instant reprendre la main et assimiler rapidement les actions réalisées par son système, celles en cours et celles abandonnées.

2.2.3 Destinataires de l'aide

L'aide est-elle destinée à un pilote spécialiste d'une mission dans un autre type de mission, ou dans sa mission habituelle ? A-t-elle pour but d'aider un novice, un expert, ou les deux ?

La littérature [18], [13] montre que les besoins en information sont différents entre débutant et expert, et les capacités d'actions et de mise en œuvre de tactiques ne sont évidemment pas les mêmes. Deux pilotes de niveaux de qualification différent auront besoin d'une expertise différente pour être aidés efficacement.

L'expertise elle-même pose problème : comment unifier les connaissances venant d'experts différents sur un même sujet ? Comme nous le constatons dans l'aéronautique de combat des experts différents utilisent des tactiques parfois opposées; où trouver l'expertise permettant de choisir ?

2.3 Architecture des systèmes d'aides et couplage avec le pilote

L'objet de nos travaux porte essentiellement sur les aides au raisonnement et à la conduite de mission, plus précisemment les aides à la navigation en pénétration basse altitude. Dans ce cadre, nous allons montrer comment l'architecture des systèmes d'aide peut découler logiquement d'une réflexion sur son couplage avec le pilote.

L'aide au raisonnement ou à la conduite de mission, de part sa nature, devra manipuler des connaissances de niveaux variés, les exploiter en temps réel, prendre en compte un univers changeant, et enfin tenir compte des stratégies du pilote. Elle relèvera donc au moins en partie du domaine de l'intelligence artificielle temps réel, et devra contenir divers types d'expertise (navigation, gestion des menaces, système d'armes, etc...).

C'est là généralement une chasse gardée des ingénieurs cogniticiens, et un domaine où rivalisent de sophistication les techniques de l'intelligence artificielle. Mais, avant le problème technique, il y a le problème de l'expertise. En effet, l'expertise mise en œuvre dépendra du but visé par le système d'aide.

Est-ce une aide à court, moyen ou long terme ? A court et moyen terme, si l'on veut toujours assurer au pilote son rôle de décideur, il importera au système d'aide de configurer son raisonnement, au moins dans sa présentation, de manière à s'insérer harmonieusement dans celui du pilote. Vu les contraintes temporelles, il convient en effet d'éviter que ce dernier adapte et contraigne son raisonnement afin de prendre en compte celui de son système embarqué.

Une des caractéristiques du fonctionnement cognitif humain est de privilégier le court terme, en prenant en compte une ou deux contraintes fortes pour résoudre un problème. On constate donc que l'homme ne raisonne pas de manière optimale au regard d'une échelle de temps plus longue, et prend difficilement en compte un nombre de variables et de contraintes élevé. Il faudra en tenir compte, et donc s'abstenir au moins pour le court terme de lui présenter un raisonnement, certes optimal à long terme, mais que le pilote mettra du temps à reconnaitre comme tel. Au contraire, à long terme, il importera d'utiliser les possibilités de calcul optimal et multi-facteurs d'un calculateur, pourvu qu'une présentation adaptée permette au pilote de la comprendre rapidement.

3 ARCHITECTURE D'AIDE A LA NAVIGATION

3.1 Le choix d'une modèle basé sur l'opérateur

En résumé de la réflexion précédente sur les aides, il nous est apparu que la possession d'un modele de fonctionnement mental du pilote dans l'exécution de sa tâche était indispensable pour l'élaboration d'un système d'aide. Il ne s'agit plus seulement de s'interesser de manière superficielle à ses besoins en informations et à ses capacités à les traiter, mais si l'on veut aider son raisonnement et même faire à sa place pendant un temps limité, il faudra tenir compte de ses manieres de résoudre les problèmes en vol.

Nous sommes allés plus loin en basant notre architecture de systeme d'aide à la navigation sur un modèle cognitif du pilote. Celà signifie que nous implémentons les manières de faire et les heuristiques non seulement à un niveau superficiel du système, mais qu'elles en forment la base.

Par ce choix, nous pensons améliorer le couplage du pilote avec son appareil, en assistant ce dernier par un système fonctionnant avec la même logique, améliorant ainsi la confiance envers l'aide.

3.2 L'application à la mission de pénétration basse altitude

En conséquence, nous avons entamé en 1984 un programme de recherche de 8 ans avec pour but de définir une architecture de système d'aide de ce qui pourrait s'appeller Assistant du Pilote du future avion de combat. Nous nous focalisons plus spécialement sur la définition d'un automate de navigation capable de s'adapter aux événements inattendus de la mission et dont la programmation est aussi proche que possible de la manière de faire du pilote. Pour résoudre le problème de l'unification de l'expertise entre pilotes, nous avons décidé de privilegier la cohérence d'une expertise et donc d'utiliser un seul style d'expert. Pour justifier ce choix, nous avons vérifié que de cette manière, l'utilisateur d'un tel automate, quelque soit son style de pilotage propre, aura une bonne prediction de ce que sera la reaction de l'automate dans une situation donnée.

La collecte de l'expertise et sa structuration ont demandé trois ans de recherche avec interviews de l'expert, analyses d'activités de pilotes en simulateur et en mission reelle. En parallèle sont menées d'autres études avec d'autres pilotes pour recueillir des données complémentaires, identifier différents styles de pilotage et vérifier nos hypothèses sur le modele cognitif.

L'informatisation du modele a commencé il y a deux ans. Le support physique d'expérimentation consiste en deux stations de travail en réseau. Une station Symbolics 3650 supporte la programmation de l'automate de navigation, qui interagit avec un Microvax sur lequel tourne un modèle de simulation d'avion des AMD-BA. Ce microvax supporte aussi un programme graphique qui permet de visualiser les raisonnements du système en animant des zooms sur des instruments de la planche de bord correspondant aux étapes de l'activité cognitive du pilote.

4 LA MISSION ETUDIEE

La mission qui nous intéresse est la mission de pénetration basse altitude sur Mirage F1Cr. Elle consiste classiquement à passer sur un objectif pour le traiter (destruction, photos, ...) à un temps bien precis, fourni au pilote, apres avoir navigue en tres basse altitude dans la portion du territoire sous contrôle ennemi jusqu'a ce point, et à revenir sur une base amie, toujours en basse altitude (Voir figure 1).

Le pilote dispose de son ordre de mssion comportant principalement une description breve de l'objectif avec ses coordonnées et l'heure de passage, et des détails fournis par son officier de renseignement (menaces, positon des lignes, ...). Il elabore un plan de mission au sol ou il tient compte egalement des facteurs meteo, avion disponible, etc... . Cette phase de preparation, aboutit au choix de son approche de l'objectif, puis a la détermination précise de sa route. Ces facteurs seront entrés dans les systemes avion. Il determine ses horaires d'entrée sur les lignes de front et de sortie, qu'il devra imperativement respecter.

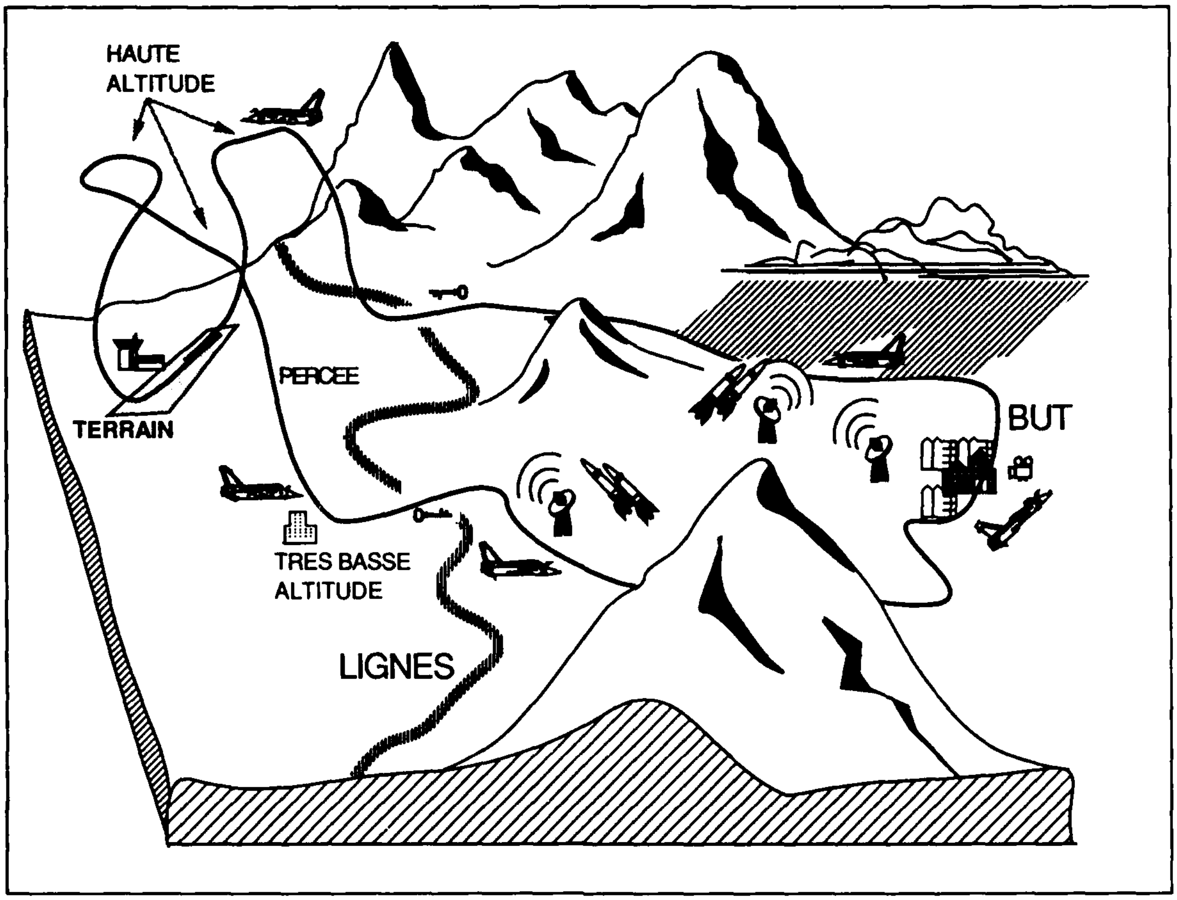


Figure 1 : MISSION DE PENETRATION BASSE ALTITUDE

Cette phase de préparation permet aussi au pilote de s'imprégner de sa mission, et d'imaginer les parades possibles aux incidents qui peuvent se produire pendant l'exécution.

Le plan de mission est précis et délicat à préparer, et il est impensable de la modifier profondément en vol. La tâche essentielle du pilote pendant le vol, sera dans la mesure du possible, d'accommoder ce plan pour le garder valide. Il est en effet très difficile de replanifier en vol, à 600 Kts et 200 fts sol.

L'exécution sous forte contrainte d'une tâche complexe de contrôle de processus rapide, suivant un plan préétabli, consiste finalement à traiter les évènements en gérant en permanence la distance et les distorsions imposées au plan original. Le problème central pour le pilote est d'arriver à l'heure sur son objectif, et à respecter également ses horaires sur les lignes de front.

D'un point de vue technique, la mission se décompose en phases quasi-autonomes pour lesquelles le pilote dispose d'une procédure d'exécution standart. Ainsi, classiquement la mission comporte une phase de décollage et navigation basse altitude, suivie de branches de navigation basse altitude, jusqu'à la phase de traitement de l'objectif. Ensuite viennent des branches basse altitude jusqu'au terrain.

5 LE MODELE COGNITIF

La base de notre modèle est référante à la théorie des schémas qui décrit l'organisation des connaissances d'une expert sous forme de procédures mentales, appelées **schémas**, et dotées d'attributs importants: procédure proprement dite, définition du but à atteindre par la procédure, incidents possibles, univers d'application de la procédure, etc. Nous pourrons donc structurer notre expertise à l'aide des notions psychologiques de **plan**, **schéma** et **script** telle que les définissent [14] et [13].

Le plan contient l'organisation abstraite des étapes à satisfaire pour atteindre le but, il correspond à la planification des grandes lignes de la mission et est utilisé lors de la conception générale de l'activité.

Le schéma est une structure de connaissances permettant d'atteindre un sous-but du plan, il assure la planification immédiate et le contrôle de l'activité. On peut structurer les connaissances d'un schéma en deux parties:

1. Le contenu exécutoire, appelé script, qui regroupe les procédures propres à satisfaire le but du schéma. Il est alimenté et actualisé par le contenu notionnel.

2. Le contenu notionnel renferme le but et les méthodes d'évaluation de ce but, les règles de décision

pour juger de la validité, les prérequis à l'exécution, les règles de contrôle de cohérence, les stratégies pour réaliser les différents sous-buts, les règles de savoir-faire, les contraintes du système, la description des incidents potentiels, il permet une paramétrisation en fonction de la situation réelle et une accommodation rapide en cas d'incident.

Ces connaissances permettent à l'opérateur de comprendre la situation et d'agir sur l'avenir immédiat en disposant de modèle d'anticipation, mais aussi de contrôler l'action en cours et de savoir quelles actions entreprendre si le but n'est pas atteint. Le plan de mission sera donc une séquence de schémas pour la réalisation de sous-but convergeant vers la réalisation de l'objectif de la mission.

Le modèle démontre particulièrement la flexibilité de ces procédures et leur capacité d'auto-adaptation aux contraintes imprévues lors de l'exécution de la mission. Cette auto-adaptation peut se faire localement au niveau du schéma en cours ou être répercutée sur les schémas à venir en les transformant. Une autre catégorie d'adaptation procède du chemin inverse, allant du plan vers les schémas, particulièrement lors des changements de stratégie ou de tactiques. Une représentation de l'auto-adaptation du modèle du fonctionnement cognitif du pilote est proposée en figure 2.

Un exemple inspiré d'observations réelles résume bien ces trois niveaux d'adaptation:

Un pilote se fait menacer par une conduite de tir adverse imprévue en début de mission. il accélère (accommodation locale du schéma en cours d'exécution).

Cette activité va contraindre la durée du schéma et un certain nombre de vérifications voire de réglages pour la phase de vol suivante ne seront pas exécutés. Il faudra les prévoir dans le ou les schémas suivants (accommodation des schémas suivants par diffusion amont des contraintes).

Les prérequis non encore exécutés du schéma suivant seront insérés en priorité.

Mais au delà de cette accommodation prioritaire à court terme, le pilote va prendre des mesures pour anticiper sur un dégradation de la situation en accélérant (méta-connaissance sur son propre savoir-faire). Comme il sait qu'il ne faut pas voler lentement, il va se détourner de son itinéraire prévu pour l'allonger (nouvelle stratégie). Cette stratégie va se répercuter sur les contenus de toutes les procédures du plan (diffusion du plan vers les schémas).

Le modèle décrit permet de s'auto-adapter rapidement et efficacement à grand nombre des contraintes avant d'en être réduit à des raisonnements beaucoup plus formels et coûteux en temps de type résolution de problèmes. Cette façon de faire nous paraît être un bon reflet de l'expertise du pilote de combat.

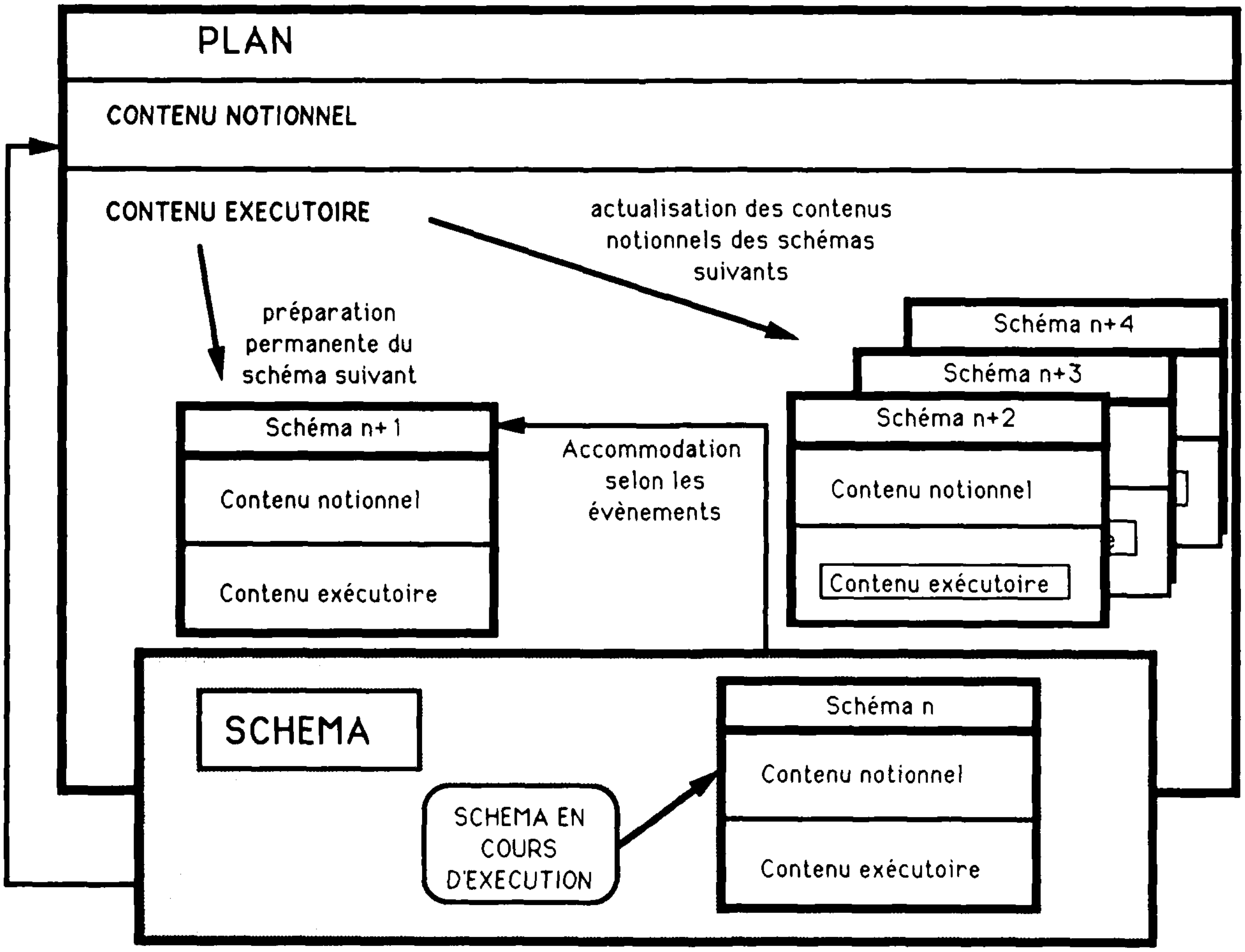


Figure 2 : MODELE COGNITIF D'AUTO-ADAPTATION

6 LE MODELE INFORMATIQUE

6.1 Introduction

Le modèle informatique est construit à partir du modèle cognitif et transpose les notions de plan, schéma et script [6]. La représentation par ***frames*** [9] est particulièrement adaptée pour la description d'entités de connaissances que nous manipulons dans le contenu exécutoire du plan ou des schémas.

Nous retrouverons les deux niveaux de fonctionnement du modèle cognitif, local (gestion à court terme) et global (gestion à moyen et long terme).

Quatre pôles de connaissances (conduite générale du schéma, exécution des scripts, traitement des évènements et gestion du temps) émergent de l'étude cognitive; chacun est modélisé par un agent spécifique possédant les connaissances propres à un domaine et capable d'échanger des informations avec les autres, nous parlerons de **moniteurs** (schéma, script, évènement et temps).

Pour traiter cette connaissance distribuée et rendre compte de la démarche de l'opérateur, nous avons choisi une programmation du modèle utilisant de techniques voisines de celles employées dans les langages d'acteurs [8].

Un langage d'acteurs diffère de la notion classique de langage orienté objet en ce sens que chaque objet du langage (chaque acteur) peut fonctionner en parallele aux autres. La communication et la synchronisation se font par envoi de messages. La variété de langage d'acteurs mis en œuvre s'inspire de ABCL/1 [19]. Les messages peuvent être de trois types :

- l'envoi de message asynchrone permet à un acteur d'envoyer un message et de continuer à travailler sans attendre de réponse.

- l'envoi de message synchrone représente une question d'un acteur à un autre. L'émetteur est arrêté tant qu'il n'a pas obtenu de réponse.

- l'envoi de message avec rendez-vous est le cas où un acteurpose une question à un autre, continue à travailler jusqu'au moment où il cherche à utiliser la réponse. Si celle-ci est arrivée, il continue, sinon il l'attend.

En plus de ces types, nous avons besoin de prendre en compte des interruptions asynchrones de l'environnement, et de modifier sur commande les traitements en cours dans les acteurs. Pour celà, chaque message a un champ urgence, instancié à "Ordinaire" ou "Express", ce qui nous permet dans le cas d'un "Express" de forcer son exécution en interrompant un "Ordinaire".

6.2 Le niveau local

Il modélise l'exécution d'un schéma. Avant d'étudier en détail l'architecture de ce niveau, nous allons présenter les différentes situations pouvant survenir dans l'exécution d'un schéma pour mettre en évidence les différents niveaux d'accommodation nécessaires. Nous allons nous intéresser à un schéma de navigation basse altitude sans visibilité.

Exemple 1. Le pilote détecte une défaillance de sa centrale à inertie, par comparaison de l'image radar et de sac carte radarisée, les paramètres de navigation étant normaux. Si cet incident se produit assez tôt dans la branche il y aura immédiatement insertion du schéma de correction dans le script en cours. Il s'agit ici d'une auto-adaptation purement locale.

Exemple 2. Le contrôle de carburant se fait habituellement en fin de branche, au moment ou le pilote a des actions importantes à effectuer: si un écart apparait, le traitement de l'incident devra être reporte dans la branche suivante faute de temps. Ce n'est plus une adaptation locale, il s'agit ici d'un legère modification du schema suivant.

Exemple 3. L'insertion d'un traitement ne pouvant être différé, a provoque un retard qui ne pourra pas être comblé dans la branche, le pilote doit recalculer son horaire sur les branches suivantes et donc éventuellement supprimer les actions non indispensables. Il s'agit donc là d une replanification partielle, consecutive à une adaptation locale bien qu'on ne remette pas en cause l'objectif de la mission.

Exemple 4. Le pilote se fait menacer par une conduite de tir adverse, c est le niveau d'accommodation le plus complexe qui a déjà été décrit dans la description du modèle cogn'tif, il met en oeuvre à la fois une adaptation immédiate au niveau local et une replanification avec de nouvelles stratégies pour la poursuite de la mission (changement de route, abandon de l'objectif initial pour un objectif secondaire...).

6.2.1 Le moniteur schéma a la responsabilité du niveau local, il est l'interlocuteur aupres du niveau global, en effet il en reçoit les schémas successifs déjà parametrés au niveau du plan et lui transmet en fin d'exécution le résultat. En cas d'incident depassant les compétences du niveau local, le moniteur schema alerte le niveau global. Il détient d'une part la description du script et des procedures (contenu executoire), d'autre part la description du but et des procédures d'évaluation de ce but, des heuristiques pour adapter ce script à la situation réelle, des procédures de contrôle de la coherence organisees en pile et inserees dans les temps morts du script (contenu notionnel). Il va donc pouvoir produire dynamiquement le script d'execution et le transmettre au moniteur script charge de l'execution proprement dite et qui fonctionne donc en permanence. Pour tout ce qui touche aux contraintes temporelles, il se fera aider par le moniteur temps. Son dernier auxiliaire est le moniteur évènement à qui il transmet au fur et à mesure les informations qu'il reçoit de façon à actualiser la liste des incidents possibles et qu'il alertera en cas d'incident ou d'interruption dans le déroulement du script. Il garde le contrôle du niveau local centralisant les messages importants, accommodant le script en cours, gérant la pile de cohérence, activant ses auxiliaires selon ses besoins.

6.2.2 Le moniteur script, possède la description du script qu'il doit dérouler ainsi que les méthodes d'exécution des différents types d'actions de ce script. En effet nous avons pu recenser trois types d'actions: les routines (i), les blocs (ii) et les attentes (iii) pour lesquels nous avons développé un utilitaire spécifique.

(i) Les **routines** correspondent aux actions du pilote de plus bas niveau: ce sont des prises d'information sur instrument, des actions de réglage ou des vérifications de valeur attendue. Les routines sont elles aussi représentées à l'aide de *frames*, on peut ainsi indiquer si elles sont obligatoires, ajournables, les limites de validité, les valeurs seuil, la durée moyenne, le code dessin associé, les zooms concernés...
Lors de son activation, elle interagit directement avec les paramètres du modèle avion et la visualisation associée apparait l'écran. Les routines sont l'élément de base pour modéliser la conduite, qu'il s'agisse d'actions de conduite du pilote, des vérifications consécutives ou d'attente d'un résultat.
Précisons un dernier point: Le pilote est continuellement en attente d'un résultat qui lui permettra de poursuivre sa mission. C'est par exemple atteindre le prochain point tournant. L'estimation du temps nécessaire pour atteindre ce résultat lui donnera le temps disponible dont il dispose pour effectuer ses tâches et éventuellement insérer des actions de cohérence, ce rôle d'*estimation temporelle* et de *micro-planification* est la tâche du moniteur temps local que nous reverrons. Lors de la routine de lecture si le résultat n'est pas atteint (valeur seuil), le moniteur schéma va activer le moniteur temps local et prévoir de réactiver cette routine de lecture ultérieurement.
Les routines sont aussi utilisées dans les procédures de contrôle de la cohérence dont la partie exécutoire est insérée dans le script du schéma.

Donnons quelques exemples de routines:

1. La routine pour mettre la pente a -10° en début de la phase de percée modifie le paramètre de la pente de l'appareil et provoque un zoom sur le CTH (viseur tête haute) et sur la poignée pilote.
2. La routine de vérification du taux de descente sur le variomètre provoquent zoom sur le variomètre actualisé en permanence par le modèle avion.
3. La lecture de la distance au but sur l'indicateur de navigation (IDN) provoque un zoom sur l'IDN et permet une évaluation du temps disponible.

(ii) Les **blocs** correspondent à un séquence de routines visant au même but et centrées sur le même instrument, le zoom sur l'instrument restera affiché pendant tout le déroulement du bloc. Certains blocs pourront être interrompus et repris par la suite, d'autres au contraire ne seront pas sécables. De plus à l'intérieur d'un blocs certaines routines pourront être interverties ou supprimées d'autres pas, il faudra donc en tenir compte dans l'organisation du script. Ils sont aussi représentés sous forme de *frames*.
Donnons quelques exemples de blocs:

1. Le bloc de contrôle de navigation se fait sur le CTH et comprend trois vérifications élémentaires (bon but, bonne route, bon guidage en vitesse). L'ordre bien que logique n'est pas obligatoire, en cas d'interruption par le contrôle aérien par exemple, il pourra être repris.
2. Le réglage du radar en mode PR (percée), comprend la routine de changement de mode et le réglage du site de l'antenne, il était initialement en mode HA (haute altitude). Ici l'ordre est imposé et les deux actions sont obligatoires.

(iii) Les **attentes** modélisent le moments ou le pilote ne fait rien, il s'agit de périodes pendant lesquelles le pilote se focalise sur l'attente d'un résultat, attend les effets de l'action précédente ou n'a pas le temps d'entreprendre quelque chose dans le temps libre restant. Pour un pilote ne pas avoir de temps libre est inquiétant car il ne pourra alors pas insérer facilement le traitement en cas d'incident, c'est pourquoi il s'en ménage toujours au départ. Cependant les attentes apparaissent rarement explicitement au départ dans le script prototypique, elles sont générées par les mécanismes d'auto-adaptation locaux.

6.2.3 Le moniteur temps local possède quatre fonctions principales de gestion du temps dans le déroulement du schéma :

1. Il détient l'expertise pour le calcul du temps nécessaire à l'exécution d'une routine ou d'un bloc ou à l'obtention d'un résultat.

2. Il est capable de réorganiser au mieux dans le temps imparti et en fonction des priorités relatives la séquence des blocs et routines du script. C'est une compétence de micro-planification. Ceci intervient dans la gestion des attentes de résultat, l'insertion des actions de coherence et des scripts de traitement en cas d'incidents.

3. Il échange des informations avec le moniteur événement pendant la phase de selection du sous-schéma de traitement, pour lui indiquer le coût temporel du traitement et sa faisabilité compte-tenu des contraintes temporelles existantes et des possibilités d'insertion.

4. Il contrôle la validité temporelle de la réalisation du schéma. En cas d'incident il alerte le moniteur schéma, le déroulement du schéma a demandé plus de temps que prévu: on est en retard, il va falloir combler ce retard dans les schémas suivants en supprimant ou en reportant les actions optionnelles ou moins urgentes. Il y a donc fonctionnement parallèle avec le moniteur script mais fonctionnement plutôt séquentiel avec les moniteurs schéma et événement du fait qu'il y a attente de réponse pour continuer le traitement.

6.2.4 Le moniteur événement local contient l'expertise pour le traitement des incidents. Son activité se décompose en deux phases successives: diagnostic et génération de traitement. Les incidents peuvent être de deux types :

- Le résultat ramené est différent de celui attendu.
- L'exécution de l'action en cours a été interrompue.

Le moniteur événement est capable pour chaque schéma de retrouver rapidement l'ensemble des incidents pouvant survenir à un instant donné, car il tient compte de l'historique du schéma pour actualiser dynamiquement cet ensemble et ne pas s'encombrer d'événements improbables. La phase de diagnostic est alors simple et rapide et reflète bien la démarche du pilote. Le moniteur événement statut sur la gravité de l'incident et si celui-ci dépasse ses compétences le niveau global est alerté.
Le moniteur événement possède pour chaque événement un ou plusieurs sous-schéma de traitement, il étudie alors les possibilité d'insertion avec le moniteur temps local (si l'on manque de temps on préférera une solution moins optimale mais rapide). Si le traitement n'est pas possible dans le schéma courant le niveau global sera alerté, on lui transmettra le traitement sélectionné qui devra être insérer dans le schéma suivant.
Un exemple d'accommodation locale est présenté en figure 3
On retrouve le parallélisme entre la continuité de l'exécution présente par le moniteur script et la préparation du futur plus ou moins proche à partir du passé par les moniteurs événement, temps et schéma.

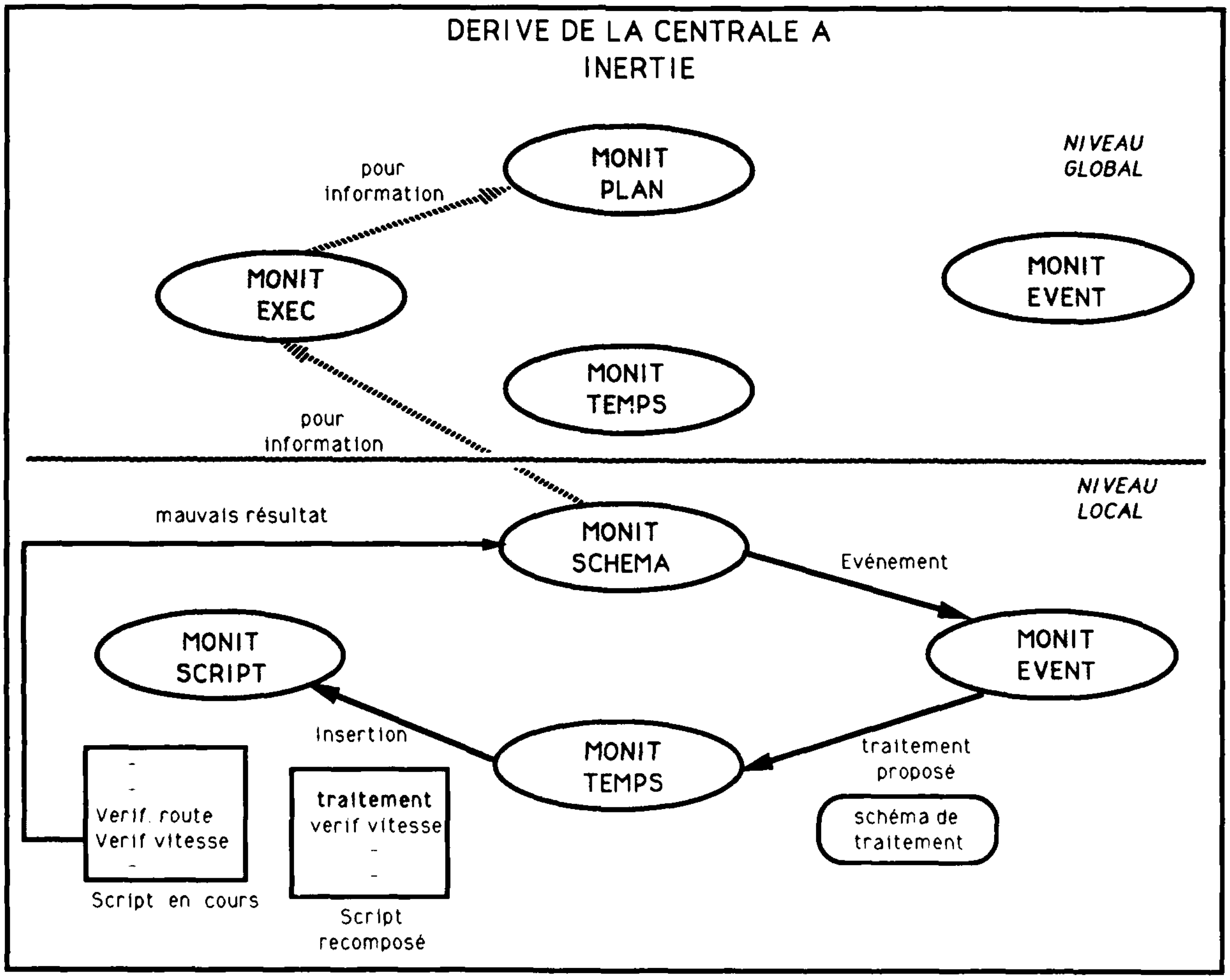


Figure 3 : ACCOMMODATION LOCALE

6.3 Le niveau global

Il modélise l'exécution du plan élaboré lors de la préparation. Il contient la séquence des schémas selectionnés pour satisfaire aux différents objectifs du plan et contrôle le niveau local. Il doit être capable d'accommoder le plan nominal en fonction de la situation réelle et éventuellement de replanification plus profonde. Son architecture est calquée sur celle du niveau local.

6.3.1 Le moniteur plan y joue un rôle analogue à celui du moniteur schéma pour le niveau local. Il manipule le contenu notionnel du plan. c'est à dire les connaissances d'univers de la mission.
Il est tenu au courant de tous les événements importants de l'exécution et transmis par le niveau local. Il décide alors s'il peut laisser agir la boucle d'auto-adaptation du niveau local ou s'il doit intervenir pour un traitement plus profond.

<u>Exemple:</u> Une accumulation d'incidents mineurs peut être le signe précurseur d'une situation grave.

Il possède dans son contenu exécutoire la séquence des schémas qu'il a sélectionnés lors de la préparation. Il actualise continuellement le contenu notionnel des schémas suivants en fonction de la situation. seul le schéma suivant est complètement instancié et prêt pour l'exécution. En effet il est inutile de savoir dans le détail ce qu'il faudra faire précisément dans trois branches par contre on devra tenir compte de tel événement qui influe sur ses objectifs .

Il est aidé pour l'aspect exécutoire par le moniteur exécution. pour les aspects temporels par le moniteur temps global et pour l'évaluation et le traitement de situations incidentelles graves par le moniteur événement global.

6.3.2 Le moniteur exécution contient à tout instant le prochain schéma prêt qui lui est transmis et actualisé par le moniteur plan. Il transmet ce nouveau schéma au moniteur schéma dès que le schéma en cours est réalisé.

6.3.3 Le moniteur temps global possède d'une part les mêmes fonctionnalités que son homologue du niveau global: Il évalue la durée des phases. le temps potentiellement disponible. il est chargé de faire tenir le modifications du plan dans les contraintes horaires impartie à la mission. Il doit en autre gérer le passif transmis par le schéma en cours et décider où l'insérer dans le schéma suivant.
'Il assure d'autre part la gestion des ressources du pilote. En effet une contrainte temporelle importante est que l'homme ne peut exécuter en parallèle plus de deux tâches au maximum. Il va donc falloir partager les ressources temporelles de notre modèle opérateur entre la réflexion (niveau global) et l'action (niveau local). On entend par réflexion les tâches de compréhension de situation à partir de l'étude de l'historique et d'accommodation ou d'élaboration de plan pour anticiper les situations à venir. Sous forte pression temporelle. le temps alloué à une replanification est restreint: on privilégie une solution simple mais rapide.

Cette contrainte peut. en fait. se révéler être une aide dans de nombreuses situations incidentelles où la recherche de solutions optimales est un problème complexe.
Le moniteur temps global exerce en continu un contrôle sur la cohérence temporelle de l'activité de l'opérateur.

6.3.4 Le moniteur événement global est chargé comme celui du niveau local d'assurer en liaison avec les moniteurs plan et temps. l'évaluation d'une situation incidentelle grave remettant en cause le plan de mission et la sélection des traitements pour y remédier. En effet la séquence des schémas sélectionnées devient obsolète et une rplanification ou tout au moins une accommodation importante s'impose.

Exemple 1: Dans le cas de menaces imprévue rendant l'objectif inaccessible. il y aura replanification pour traiter un objectif secondaire dont le plan connait déja les cractéristiques dans son contenu notionnel.
Exemple 2: Dans le cas de l'accrochage missile. il y aura abandon du schema en cours pour la mise en oeuvre d'une procédure d'évitement (break. accélération pleine Post-Combustion....) au niveau local et évaluation de la situation à partir des connaissances d'univers au niveau global.

Pour l'instant seules des stratégies d'accommodation du plan nominal sont implémentées.
Une illustration de la mise en œuvre du niveau global est en figure 4.

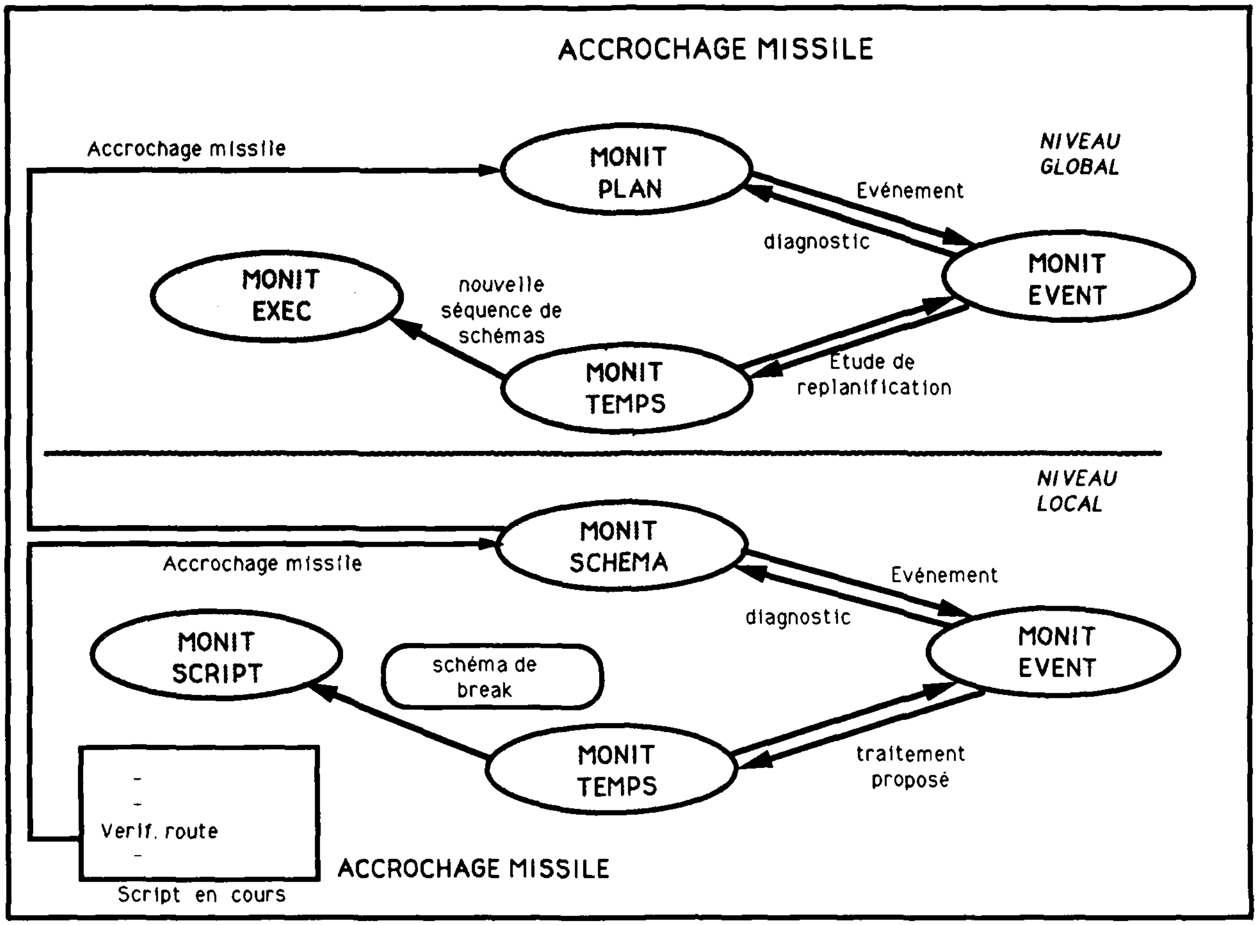


Figure 4 : ACCOMMODATION LOCALE ET REPLANIFICATION

' 7 Conclusion et perspectives

L'intérêt d'une architecture logicielle basée sur un modèle cognitif du pilote de combat nous semble triple:

(i) c'est une solution intéressante au problème du couplage du pilote avec un système d'aide: elle offre une logique de fonctionnement et de résolution de problème facilement compréhensible en temps restreint. La réalisation pratique d'une interface sophistiquée reposant sur un tel système est un nouveau champ d'études que nous explorons. Notre première préoccupation a consisté à définir une telle interface pour l'automate de navigation, offrant au pilote le résumé des actions effectuées et en cours pour faciliter sa reprise en main de l'appareil. D'autre part, nous nous penchons sur un modèle des intentions du pilote qui permettrait au système à tout moment de pouvoir se déclencher pertinemment dans le cours de l'action.

(ii) Le deuxième intérêt de cette architecture repose dans les solutions qu'elle propose à la gestion du temps réel par l'intelligence artificielle. En effet, l'architecture même de l'expertise humaine (schéma et métaconnaissances) apporte une réponse aux problèmes de modélisation de raisonnements temps réel. C'est l'organisation des connaissances qui permet un traitement rapide de peu de données, mais dont la pertinence est en général suffisante pour construire la réponse à court terme. La gestion des consequence de cette réponse à court terme (souvent non optimale) est réalisee de façon différée en dehors des periodes de fortes contraintes temporelles.

(iii) Enfin, cette méthode de résolution de probleme par une societe d'acteurs est une alternative aux architectures classiques de type moteurs d'inferences, et semble naturelle pour realiser ce genre de tâche.

BIBLIOGRAPHIE

[1] ADAMS (1988) The big picture concept, proceedings of the Aerospace medecine congres, New Orleans, N°45

[2] AMALBERTI R. (1987). Pilots as systems managers and supervisors, a risky new role according to man-

machine interface reliability. Acte du congrés AGARD GCP-FMP. Stuttgart (1987). 17-1:17-8

[3] **BAINBRIDGE L. (1987)** Ironies of automation. in "New technology and human errors". Rasmussen. Duncan & Leplat edts. Wiley publ.. 271-286

[4] **BISSERET (1984)** L'assistance à la résolution de problèmes dans la supervision des processus. rapport INRIA.

[5] **CHAMPIGNEUX G., GAUDRY Y. AUBRY P., HAVRE A., BRACQ D. (1988)** Expert system model designed for airborne use on fighter aircraft. Actes du congrès Human -Machine Interaction. Artificial intelligence. Aeronautics & space. 165-172

[6] **DEBLON F. (1988)** Modélisation de l'activité humaine en contrôle de processus rapide à partir d'un cas exemple: l'aéronautique de combat. Actes du congrès ERGO-IA. Biarritz. 48-63

[7] **FURNESS T. (1986)** The supercockpit and its human factors challenge. proceedings of the Human factors society. 48-52

[8] **HEWITT C.E. (1977)** Viewing Control Structures ad Patterns of Passing messages. Journal of Artificial Intelligence; Vol. 8-3.

[9] **MINSKY M. (1975)**: A framework for representing knowledge. in Winston P. edts. The Psychology of Computer Vision. N-Y. Mc Graw Hill.

[10] **MOREAU (1987)** Nouveaux concepts pour les cockpits futurs. Interavia. 6. 671-673

[11] **MORISHIGE R. (1987)** Cockpit automation. a pilot's perspective. Acte du congrés AGARD GCP-FMP. Stuttgart. 1987.

[12] **RASMUSSEN J. & GOODSTEIN L. (1985)** Decision support in supervisory control. proceedings of the 2nd conference on task analysis. 33-42

[13] **ROTH E.M. & WOODS D.D. (1988)**: Aiding human performance. 1 Cognitive analysis. Le Travail Humain. t 51. 1 . 39-64.

[14] **RUMELHART D. & NORMAN D (1983)** Schemata and frames. in representation in memory. Report of the University of California ONR 5302.

[15] **SHANK R. & ABELSON R. (1977)** Scripts. plans. goals and understanding. Lawrence Erlbaum ass. Hillsdale

[16] **SMITH D. & BROADWELL M (1988)** The pilot's associate. Actes des 8' journées Internationales d'Avignon. Conférences spécialisées de la Défense. 261-277

[17] **WIENER E.(1985)** Cockpit automation : in need of a philosophy. SAE tecnical paper series 851956

[18] **WOODS D. (1988)** Coping with complexity: the psychology of human behaviour in complex systems. in Tasks. errors & mental models. Goodstein. Andersen & Olsen edts. Taylor & Francis publish.. 128-148

[19] **YONEZAWA A., SCHIBAYAMA E., TAKADA T. & HONDA Y. (1987)** Modelling and Programming in an Object-Oriented Concurrent Language ABCL/1: Object-Oriented Concurrent Programming: MIT Press.

Remerciements : ces travaux sont soutenus par la Direction des Recherches et Etudes Techniques. groupe 9 (Ergonomie. facteurs humains). contrat 87.1037 et 89.

Ils donnent lieu à une étroite coopération entre le département d'ergonomie aérospatiale du CERMA et le département DSA (service IA) des Avions Marcel Dassault-Breguet Aviation.

NASA/RAE COOPERATION ON A KNOWLEDGE BASED FLIGHT STATUS MONITOR

G F Butler
Royal Aerospace Establishment, Farnborough, GU14 6TD, UK.

E L Duke
NASA Ames Research Center, Dryden FRF, Edwards, CA 93523, USA.

Summary

As part of a US/UK cooperative aeronautical research programme, a joint activity between the Dryden Flight Research Facility of the NASA Ames Research Center (Ames-Dryden) and the Royal Aerospace Establishment (RAE) on Knowledge Based Systems has been established. Under the agreement, a Flight Status Monitor knowledge base developed at Ames-Dryden has been implemented using the real-time IKBS toolkit, MUSE, which was developed in the UK under RAE sponsorship. The Flight Status Monitor is designed to provide on-line aid to the flight test engineer in the interpretation of system health and status by storing expert knowledge of system behaviour in an easily accessible form. In this paper, the background to the cooperation is described and the details of the Flight Status Monitor, the MUSE toolkit and the MUSE implementation are presented.

1. Introduction

A major concern during the flight testing of high performance aircraft systems is the timely and efficient monitoring of advanced avionics and digital flight control systems. These complex systems are crucial to flight safety and require engineering specialists on the ground to monitor and analyse system status information. Currently, such information is displayed on strip-chart recorders or CRT screens in the flight control centre and, when a failure occurs, it is difficult for any individual or group of individuals to assimilate the information in order to identify the probable cause and devise an appropriate response. Often flights are aborted in cases where it may later be determined that the severity of the failure was overestimated or data misinterpreted due to time pressure.

To help to alleviate this problem, the Dryden Flight Research Facility of the NASA Ames Research Center (Ames-Dryden) has investigated the application of Expert or Knowledge Based Systems (KBS) technology to monitoring the status of aircraft systems, (References 1-3). The resulting Flight Status Monitor (FSM) uses fault indicators from the onboard systems, which are telemetered to the ground and processed by a rule-based model of the aircraft failure management system. Such a system is designed to encapsulate the expert knowledge of experienced flight test engineers in an easily accessible form and hence reduce the time required to respond to failure indications. Currently, the FSM is implemented in Common Lisp and this version has served as a non real-time demonstrator of the FSM concept. In general, however, the time available to analyse and develop recommendations when a problem occurs in the flight environment is measured in seconds. Consequently, a KBS designed to assist in monitoring must also be able to respond in this time scale.

As part of a US/UK cooperative aeronautical research programme between NASA Ames-Dryden and the Royal Aerospace Establishment (RAE), methods of improving the performance of the monitor to allow effective on-line operation are now being explored. Under the agreement, part of the Flight Status Monitor Knowledge Base has been implemented using the software toolkit, MUSE, which was developed in the UK for real-time KBS applications. Over the last 5 years at RAE, there has been a substantial research effort into the reduction of pilot workload by the use of KBS and the MUSE package is one of the fruits of this activity. MUSE comprises a collection of knowledge representation languages coupled with a set of supporting tools for creating, testing and delivering KBS. It is aimed primarily at real-time and embedded applications, which are characterised by the need to respond rapidly to external asynchronous events during processing. MUSE was originally developed by Cambridge Consultants Ltd. under contract to RAE, although an extended version of the package is now also available commercially.

In this paper, the implementation of the Flight Status Monitor Knowledge Base using MUSE is described. Sections 2 and 3 gives an overview of the Flight Status Monitor and MUSE respectively, while Section 4 is concerned with the development of the prototype MUSE FSM. The paper concludes with a discussion of several issues raised during the programme and plans for future work.

2. The Flight Status Monitor

A Knowledge Based System capable of monitoring the health and status of flight-crucial control systems on high performance research aircraft has been developed at the Dryden Flight Research Facility of the NASA Ames Research Center. A conceptual view of such a Knowledge Based Flight Status Monitor is shown in Fig. 1. The system receives information on the status and health of the Flight Control System (FCS) from a telemetry downlink. This information is then processed by the FSM which determines whether any changes have occurred compared with the previous sample and, if so, evaluates the effect of these changes. A data-driven foreground loop determines the state of the system and informs the user, in this case the systems engineer, of the changes and consequences. If a failure occurs, a warning or caution is issued along with information about corrective or emergency procedures, if required. As a part of this evaluation, the FSM may ask the user questions regarding the state of the aircraft and similarly a background task allows the user to query the monitor for information on the FCS state or the rationale used to reach its conclusions. The FSM interrupts the background task when necessary to evaluate new data, but if the FCS failure and status information has not changed, the system does not re-evaluate the state.

The Knowledge Base contains both aircraft specific rules, e.g.

If AC Power is failed
Then Analog Reversion Mode is failed,

and metarules, the rules that the systems engineer uses to determine the correct action for a failure situation, e.g.

If All Downlink Data is failed
Then there is a Telemetry failure.

The FSM uses these rules to model the failure detection system of the FCS and compare the state generated to that of the aircraft. If the monitor's conclusions disagree with the aircraft state, a warning is issued and the user is able to ask the system to resolve this conflict. The conflict resolution is processed as part of the background task so as not to interfere with the higher priority task of evaluating the aircraft status information as it is received.

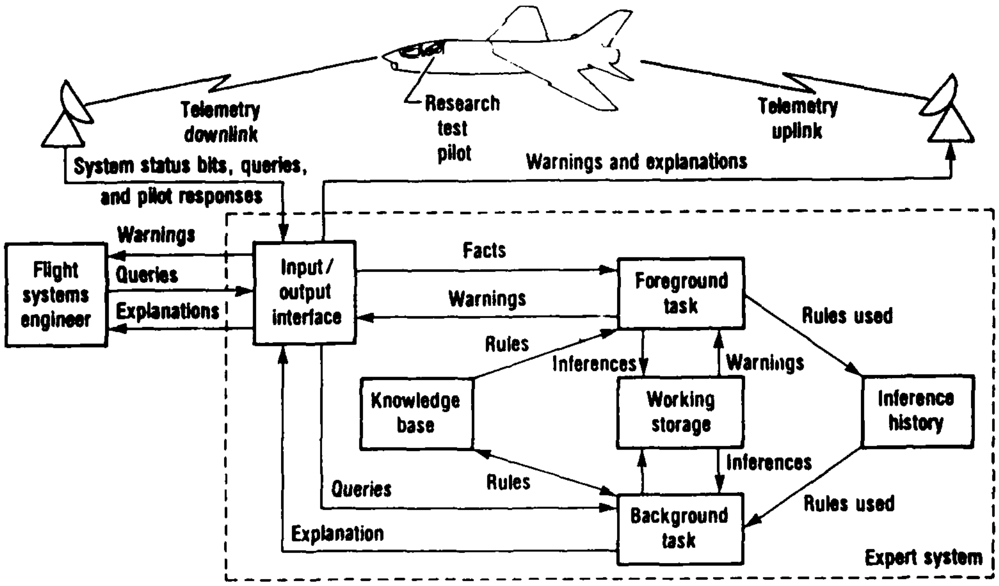


Fig.1 Overview of Knowledge Based Flight Status Monitor

2.1 Structure of Flight Status Monitor

The complete Flight Status Monitor consists of several cooperating Expert or Knowledge Based Systems, each with its own inference mechanism, as shown in Fig. 2. These inference mechanisms are predominantly forward-chaining, data-driven processes. The **Aircraft Sensor and Failure Management (ASFM)** expert system uses a forward-chaining mechanism to model the aircraft failure management system and to deduce conditions of concern or danger based on the failure indicator information. A **Metamonitor** expert system identifies situations of concern by detecting conflicts between the deductions from the ASFM system and the information available from the aircraft failure management system. These conditions are analyzed further by a **Fault Isolation** expert system that infers probable causes of conflicts, recommends corrective actions, and issues warnings. In total, these systems provide detailed status information and perform a function comparable to that of an expert flight systems engineer.

The **System Operability** expert system uses knowledge of the system effectiveness and the detailed system status information to provide a high-level assessment of the the ability of the FCS to control the aircraft, complete a specific mission, or function in a given mode. This assessment is performed by a backward-chaining mechanism using hypotheses in an order established by the user. The order of the hypotheses is important because it allows the KBS to determine the highest level at which the system is operable and to provide this information to the user. The System Operability rules are also used to establish the worst consequences of any additional failure in the **Next Worst Failure Monitor** expert system. Finally, the **Procedural Aide** expert system advises the user on the corrective or emergency procedures to be used when specific problems have been encountered.

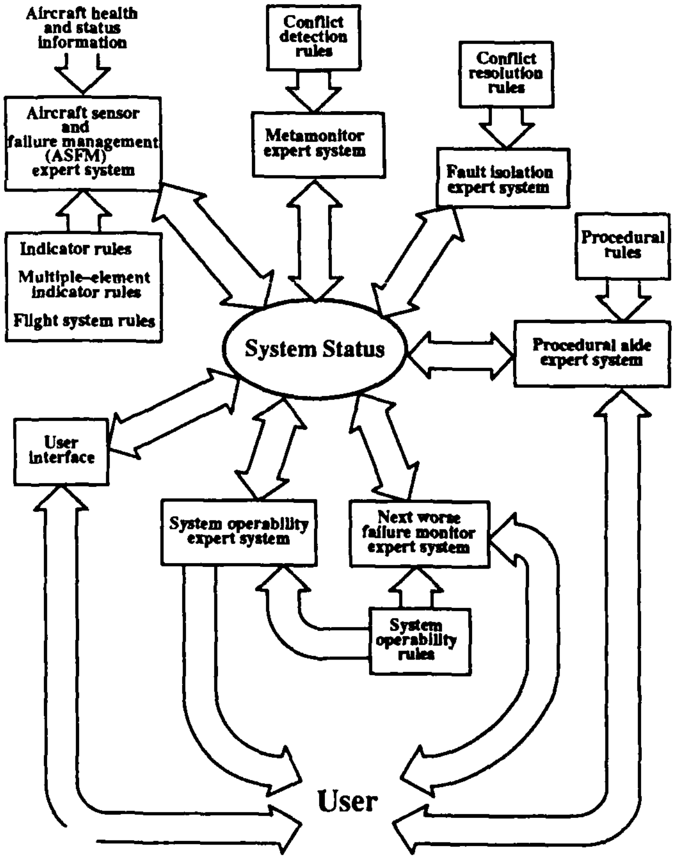

Aircraft health and status information
Conflict detection rules
Conflict resolution rules
Aircraft sensor and failure management (ASFM) expert system
Metamonitor expert system
Fault isolation expert system
Indicator rules
Multiple-element indicator rules
Flight system rules
Procedural rules
System Status
Procedural aide expert system
User interface
System operability expert system
Next worse failure monitor expert system
System operability rules
User

Fig.2 Internal structure of Flight Status Monitor

2.2 ASFM and System Operability Rules

The rules used in the FSM serve to characterize the flight control system of a redundant digital-fly-by-wire vehicle. This characterization includes a definition of the FCS health and status information, a definition of redundant system elements, a model of the vehicle's failure management system, and a definition of emergency procedures associated with FCS failures. The FSM uses several different representations of rules (Fig. 2). Some of these representations are in the form of traditional if-then production rules. Many of the rules are defined in unusual formats, however, to facilitate definition of the knowledge base and to increase execution speed of the inference mechanisms. [see Reference 2]. Since the MUSE implementation has been primarily concerned with the ASFM and the System Operability systems, the particular rules for these systems are described in more detail below.

For the kernel of the ASFM Knowledge Base, nearly all the rules depend upon certain basic words or indicators. These are names used to identify bits in the telemetry stream or flight system time history. Three distinct types of basic words are used: failure indicators, status indicators, and cross-channel assessment indicators. Failure indicators represent knowledge of the failed state of aircraft subsystems. For example, in a telemetry stream there may be a bit that represents an input sensor to the FCS, such as a pitch rate gyro. If this bit is on, it could indicate that the subsystem has failed. Status indicators are similar to failure indicators in nature, except that they represent the status of particular parts of the system, rather than a failure. In modern redundant flight control systems, it is not uncommon for each computer to contain an assessment of the health of itself and the other computers. These are the cross-channel assessment indicators.

Multiple-element indicator rules are concerned with groups of indicators that are similar in function. The primary purpose of these rules is to accommodate redundant elements. When these rules are applied, a fact is added to the main system status repository, identifying the number of failures of the appropriate type. There are two types of multiple-element rules: **Intrachannel** rules are used to identify failures of redundant elements within a single channel of the FCS, while **Interchannel** rules are used to identify failures in redundant elements within the FCS. Traditional if-then production rules are used to model the vehicle's failure management system. These rules are also used to model the interconnections and dependencies within the flight system. Again, two types of rule are used : **Intrachannel** and **Interchannel.** These rules use the facts derived from the basic indicators, cross-channel assessment rules, and multiple-element rules to deduce information about the vehicle's system state.

System Operability rules are used in general to provide high-level information, not only on the health and status of the vehicle flight system, but also on the particular control system mode being used. These rules are structured as traditional if-then production rules and are arranged in a hierarchical manner. Each of the system operability rules is evaluated until one is satisfied. It then displays the information to the systems engineer. This is the only backward-chaining mechanism in the Knowledge Based Flight Status Monitor. The consequents of these rules are also used to establish a hierarchical set of hypotheses for determining the next worst failure condition (see Reference 1).

2.3 Current Status

A working non-real-time version of the Flight Status Monitor has been implemented in Common Lisp and demonstrated at NASA Ames-Dryden. The ultimate goal in the development of a Knowledge Based FSM is, however, to demonstrate the capability of a real-time expert monitor providing "intelligent" interpretation of system status information and to apply this in the flight test control room environment. A major concern in the development is the capability of operating in real time, since information is available from the aircraft at speeds of 40-50 Hz and the number of indicators to be processed can be in excess of 100. The introduction of data structures which improve the real-time performance and yet allow easy modification and adaptation is seen to be essential for an operational system. In addition, different inference techniques are being investigated together with the data and rule representations required for more efficient operation. As part of this investigation, a substantial component of the FSM Knowledge Base has been reimplemented at RAE using the software package, MUSE, which was specifically designed for real-time KBS development and delivery.

3. The MUSE Knowledge Based System Toolkit

MUSE is a toolkit for the development of real-time Knowledge Based Systems, (Reference 4). It comprises a package of knowledge representation languages, coupled with a set of supporting tools for creating, testing and delivering applications. MUSE also includes features particularly suited to real-time operations, such as agenda-based priority scheduling, interrupt handling and fast data capture. The development environment for MUSE is the Sun workstation, but it can also be tailored to deliver prototype systems on compact solid-state hardware.

The heart of the MUSE system is an integrated package of languages for knowledge representation (see Fig 3). These languages all share the same set of database and object structures, thus allowing them to be freely mixed within a given application. PopTalk, which provides the central component of the package, is a block-structured procedural language which supports symbolic processing and therefore much of the AI programming style used in languages such as Lisp. It is derived from the Pop AI language which has been extended to include Object Oriented programming environment in the style of such languages as Smalltalk, and is implemented in C for compactness and portability.

Two rule-based languages are provided within MUSE. The first is a Forward Production System, referred to as FPS. Rule firing is governed by pattern-matches on objects in the MUSE databases and the rule actions may create or modify database objects or may in turn call upon embedded procedural code to perform special actions. Pattern matching is performed by a modified version of the Rete algorithm used by many production rule systems (see Reference 5). The second rule-based language is a Backward Chaining System, known as BCS. Like the FPS, it can pattern match on objects in the databases and can, as a result of matching, modify or create objects, or call procedural code. Unlike the FPS, however, the BCS rules can perform searching using a depth-first backtracking strategy, similar to that used in Prolog.

The second major component of the MUSE package is a set of architectural support facilities. With many AI toolkits the user is restricted to working with a single set of rules and a single database. A key aim of MUSE, however, is to permit the user to partition the application into a collection of well-defined modules. Depending on the nature of the problem to be tackled, a MUSE application can be structured as anything from a single production rule system up to a complex network of cooperating knowledge sources with both shared and private databases. Also included is a powerful data capture system of virtual channels, which allow access to either real or simulated data streams external to the MUSE process.

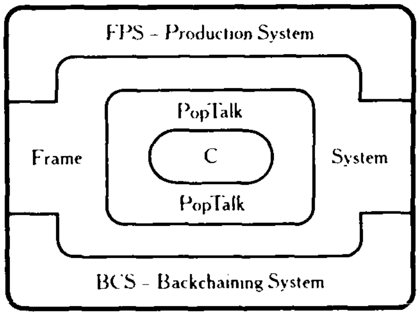


Fig.3 The MUSE Language Package

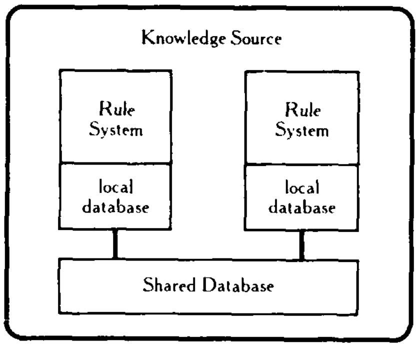


Fig.4 Typical Knowledge Source structure

The module structures are implemented by using the support for Object Oriented programming which is built into the PopTalk language. The basic structural element is referred to as a Knowledge Source and, as shown in Fig 4, a standard Knowledge Source comprises a collection of one or more rule sets each with its own private local database. Separate rule sets are linked by access to a shared database, which is visible to each rule set in the Knowledge Source. A given MUSE application consists of one or more Knowledge Sources, linked by shared access to databases. The control of the scheduling of these separate Knowledge Sources is carried out by a priority-based "agenda", which maintains an ordered list of tasks to run. The tasks can be scheduled explicitly by rules or procedural code, or implicitly by monitoring changes in the databases.

Two further supporting packages which are part of the normal MUSE system are a Structured Editor and a Run-time Browser. The editor provides a convenient window-based editor for creating and modifying MUSE source code. It allows the source code to be viewed as a hierarchy of modules in a highly structured fashion. It also supports access to unstructured ASCII files so that code originally developed with a conventional editor can be reused from within the structured editor. The MUSE editor is specially designed to ease the task of editing large collections of hierarchically nested objects. This is achieved by exploiting the notion of display folds. Each object on the screen can be shown in either closed form, in which case only the basic name and type of the object are shown, or in open form, where all the internal details of the object are visible. By selectively opening and closing objects, the user can view the code at different levels of detail. The second supporting package, the run-time browser, allows the user to examine the state of the MUSE internal data structures during a run. It also provides an interface for some other important MUSE functions such as the debugging system.

4. Implementation of the Flight Status Monitor in MUSE

The FSM Knowledge Base is written in Lisp and, as such, is not immediately suitable to use as input to the MUSE system. Furthermore, the part of the Knowledge Base selected for MUSE implementation contains well over 400 rules. Hence, the potential for transcription errors was considerable and it was decided to translate the rules from Lisp to MUSE format automatically. This process is described in more detail below. The basic structure of the Lisp FSM system has mapped across successfully to the MUSE implementation, although it has proved necessary to include an additional rule to perform voting across three channels. The rules and data have been segmented, where possible, along the lines of the original FSM structure.

4.1 Rule Translation

Compiler techniques and formal grammars were used to generate a parser for the original Lisp code, via the standard Unix programs YACC and LEX. The Lisp rules were then examined and a specification for a lexical analyser determined. This was given to the LEX lexical analyser generator which produced an appropriate C program. The original code was then further examined, the underlying grammar was specified in the modified Backus-Naur Form (BNF) used by YACC, and the actions to be taken at various stages of the parsing procedure were added. This BNF specification was given to YACC, which generated the C code for a parser for the Lisp rules. The combined lexical analyser and parser were then compiled along with a small amount of supporting C code to produce the final translator. In operation, the translator is fed with the appropriate section of the Lisp ruleset and generates MUSE code in a format that can be spliced into a standard MUSE structured editor file.

4.2 Structure of MUSE Flight Status Monitor

Each item of data in the MUSE system is represented by an object. These objects are created once on initialisation, rather than being created and destroyed dynamically as required, since this could cause a large performance overhead due to garbage collection. The MUSE objects are divided into four groups: **Basic, Intra, Inter** and **FCS** to correspond with the FSM rule categories defined in Section 2.2. Basic objects represent the failure and status bits and provide the system with the base level information upon which to initiate the reasoning process. These are triplicated to correspond with the three information channels of the FSM. Intra and Inter objects represent the partial stages of reasoning. The Intra objects are associated with the **Intrachannel** rules and are also triplicated. The Inter objects are associated with the **Interchannel** rules and are not triplicated, since they contain merged information from all three input channels. FCS objects represent the output of the **System Operability** rules.

The structure of the translated rules was designed to mirror as closely as possible the original Lisp rules, as illustrated by the example given in Fig. 5. The following points should be noted:

1) The automatic translation process inserts "a_" at the start of each item name to ensure that the name commences with an alphabetic character, as required by MUSE.
2) Values 1 and 0 are used in the MUSE code to indicate on and off respectively.
3) In the MUSE version, the variables beginning with I are "handles" which are used to reference the particular consequents to be changed. This is necessary since items of data are being reused, rather than continuously created and destroyed.

4.3 MUSE FSM Rulesets

The rulesets in MUSE correspond to those in the original Lisp Knowledge Base with the addition of the Combining Rule which performs a voting function. Communication between rulesets is managed via shared databases referred to as Notice Boards. Four notice boards are accessed: the basic_nb, the intra_nb, the inter_nb and the fcs_nb, corresponding to the object groups defined above. The rulesets and their relationships to the notice boards are described below. The incoming data updates are taken from the basic_nb, are reasoned upon by the **Intra Channel** rules and the consequents are then placed in the intra_nb notice board. As there are three data channels in the system, the data in the basic_nb and intra_nb are triplicated. The **Intra Multi-Channel** rules monitor a number of failure and status indicators in a specific channel. Whenever an indicator is modified, the rule fires and calculates the total number of indicators set in the group that the rule is monitoring. The **Combining** rule communicates with the basic_nb, intra_nb and inter_nb notice boards and executes a unanimous voting operation across the three channels. It fires if it has three separate objects, in different channels, having the same name and value. The **Inter-Channel** rules are the first rules which operate on the combined output of each of the channels, i.e. on data which has been generated by the Combining rule. The results are placed in the inter_nb for for the final stage of the analysis. The **Inter Multi-Channel** rules operate in the same manner as the Intra Multi-Channel rules. Instead of monitoring failure and status indicators in one channel, however, they monitor indicators for all three channels. Thus, if an indicator is set in any channel that the rule is monitoring, the rule fires and recomputes the total number of indicators set. The **System Operability** rules are the final stage in the reasoning process. They take the output from the other rulesets and generate the system status indicators.

4.4 System Interface

The MUSE FSM system is interfaced to the Sun Unix environment by means of the MUSE block sockets mechanism. This provides the capability for connecting any data transfer process to the input end of the sockets. The system is configured to use a total of twelve MUSE data channels, with four data channels associated with each of the three FSM channels. A pair of data channels is used for each of the status and failure sections, one for setting bits on and the other for setting bits off. The interface checks that it is only propagating changes and, hence, will not notify the rule systems if the value received is the same as the existing value.

A support harness has also been developed to allow the user to set or unset failure or status bits in any of the three FSM channels. This enables, for example, the setting of all the wheels down without having to issue individual weight on wheel commands for each of the three wheels in each of the three FSM channels. A facility to reset the MUSE system is also provided to eliminate the need to recompile the FSM code for each run.

Original Lisp Rule

```
(setq INTER-CHANNEL_RULE_8
 (make-Production_Rule
   :Name          "AR Condition"
   :Kind          "Conjunctive"
   :ANTECEDENTS '(
              ("AR Mode Computed" "is" "off" )
              ("Monitor Strakes" "is" "on" )
             )
   :CONSEQUENTS '(
              ("AR Condition" "is" "on" )
              ("deduced AR Mode Indicator" "is" "off" )
             )
   :Certainty     1.0
   :Explanation   "None given" ))
```

Translated MUSE Rule

```
/* Inter_Rule_8:  - AR Condition */
|NASA explanation - None given
|
   if
     there is an item
       -tname "a_ar_mode_computed",
       -channel "i",
       -value 0
     and
     there is an item
       -tname "a_monitor_strakes",
       -channel "i",
       -value 1
     and
     there is an item I0 ,
       -tname "a_ar_condition" ,
       -channel "i"
     and
     there is an item I1 ,
       -tname "a_deduced_ar_mode_indicator" ,
       -channel "i"
   then
     assert {item I0: -value 1}
     and
     assert {item I1: -value 0}
     and
     do (
       printf('Rule Inter_Rule_ 8 AR Condition has fired') ;
     )
```

Fig.5 Example Lisp and MUSE rules

5. Discussion

Although many of the features of the MUSE package were not tested by the FSM implementation, the flexibility of MUSE in handling different knowledge representations was demonstrated. The variety of rulesets involved in the FSM Knowledge Base were all successfully converted into appropriate MUSE structures. The Flight Status Monitor, at over 400 rules, is the largest system yet implemented in MUSE at RAE, but apart from a requirement to increase the size of certain system buffers, few adverse effects were experienced.

The need to devise a special-purpose translator to convert the original Lisp rules to MUSE format, is symptomatic of a general problem in comparing the performance of different KBS development environments. As yet, there is no standardisation in the way that essentially the same knowledge is expressed in different systems. With small systems, manual translation is feasible, but as the size of the Knowledge Base increases, automatic translation becomes essential. In general, the development of such translators demands a high level of skill and a considerable commitment of resources. On this cooperative project, for example, well over 50% of the effort supplied by RAE was devoted to the development of the translator.

In order to assess the performance of the prototype MUSE implementation of the FSM fully, further analysis of the initial conditions and of representative failure modes is required. Since the MUSE rule system is only invoked when an input data item changes, there is no difficulty in maintaining real-time performance in this state. When the input data changes and the rule system is brought into play, however, the analysis time has been found to be generally less than 1 second. This is short enough to be acceptable to a user, but still longer than the time between successive data updates. A more detailed consideration of the foreground/background task management aspects of the FSM and its potential interaction with the MUSE prototype is needed to resolve this issue.

6. Concluding Remarks

The application of Knowledge Based Systems technology to flight test status monitoring is particularly appropriate. The monitoring task is manpower and information intensive and is fairly well understood. The capabilities of such a system to monitor data downlinked from the flight test aircraft and to generate information on the state and health of the system for the test engineers provides increased safety during the flight testing of new systems. Furthermore, the Knowledge Based Flight Status Monitor gives the systems engineers ready access to the large amount of information required to describe the aircraft system, thus enhancing engineering capabilities in both understanding and developing complex systems.

The time available to analyse and develop recommendations when a problem occurs in the flight environment is usually measured in seconds, however, and hence a Knowledge Based System designed to assist in monitoring must be able to respond in a similar time scale. The implementation of a substantial part of the Flight Status Monitor Knowledge Base in MUSE has demonstrated that this time scale can be met, although a considerable amount of further work would be required to integrate the MUSE FSM with other elements of the flight monitoring system into a fully operational online facility.

Both NASA and RAE have a continuing interest in the application of KBS to the monitoring of advanced avionic systems, and this joint project has brought benefits to both. RAE have been able to evaluate MUSE on a much larger Knowledge Base than those used for previous tests, while NASA now have access to a flexible and efficient implementation of part of the Flight Status Monitor, which offers wider development opportunities for the future. Further cooperative activities in this area are currently being planned and are expected to focus on some of the verification and validation issues pertinent to the introduction of Knowledge Based Systems into the flight environment.

References

1. Disbrow J.D, Duke E.L, Regenie V.A.
Development of a Knowledge Acquisition Tool for an Expert System Flight Status Monitor.
AIAA Paper 86-0204, 1986.

2. Duke E.L, Regenie V.A.
Description of an Experimental Expert System Flight Status Monitor.
NASA Tech Memo 86791, 1985.

3. Regenie V.A, Duke E.L.
Design of an Expert System Flight Status Monitor.
NASA Tech Memo 86739, 1985. (also AIAA Paper 85-1908).

4. Reynolds D.
MUSE: a toolkit for embedded real-time AI.
in "Blackboard Systems", (Ed. R.Engelmore & T.Morgan), Addison-Wesley, 1988.

5. Forgy C.L.
Rete: A fast algorithm for the many pattern/ many object pattern match problem.
Artificial Intelligence **19**, 17, 1982.

Acknowledgements

The authors would like to acknowledge the contributions to the paper made by J.D.Disbrow, A.T.Graves and M.Bowyer.

EXPERT SYSTEM FOR KALMAN FILTER SUPERVISION : APPLICATION TO AUTONOMOUS SATELLITE NAVIGATION*

D. Berton, T. Codron, R. Horak, S. Sallé**
SAGEM
Missile and Space Guidance Systems Unit
Eragny Center BP 51
F-95612 Cergy-Pontoise Cedex

Abstract :

We present the realization of a rule based supervisor which ensures the good performance and the robust behaviour of a non linear Kalman filter. Together with the description of the functional architecture of this supervisor, we explain the different sources of knowledge at out disposal and how we used them to program the rules. We analyze the typical characteristics of the Real Time Expert System which is the heart of the supervisor, and also how the emission of actions in real time led us to select an original tool to generate the Expert System. Rules and results are presented, they are applied in the case of a navigation filter for satellites and show considerable improvements in performances of the Kalman filter once coupled with the Expert System.

1 - INTRODUCTION

1.1 Introduction

Applicability of Artificial Intelligence to automatic control theory has been largely studied [AST87, FRA87, GEN87, HIG87, DUQ88]. Industrial, spatial, aeronautics and now military applications are emerging and confirm a general interest in these new techniques. However most of them apply to process control which is only one aspect of the automatic control field.

The study we are presenting in this paper deals with the supervision of a Kalman filter (an optimal estimator) by a rule-based expert system.

The example chosen concerns the autonomous navigation of a satellite based on the supervision of a non-linear Extended Kalman Filter (EKF) estimating position and speed of the satellite. The filter is based on Kepler gravitational equations and is updated by discrete measurements of the direction of known landmarks obtained by on-board image processing. These measurements are acquired on a real time basis and are asynchronous (not periodic).

Together with the description of the functional architecture of the supervisor, we explain the different sources of knowlege at our disposal and how we used them to program the rules.

Specific constraints due to the operation on real time events and the emission of actions in real time led us to select an original tool to generate the Expert System. We will analyze with the help of some examples the typical caracteristics of this expert system.

1.2 Goals of the study

Our approach is to realize a rule based supervisor which ensures both the good performance and the robust behaviour of a Kalman filter.

A classical Kalman filter requires that a certain number of mathematical hypotheses on its model, measurement updates, and associated noises be verified. Should this not be the case the filter would give under-optimal performances or could even diverge.

When an Extended Kalman filter is used or when the Kalman filter is non-linear either in its model or measurements, the complexity of the algorithms and the limitation of available analytical tools make the determination of the filter behaviour a very difficult task. It can even be impossible to determine an analytical solution.

Practically the implementation of such a filter in an operational system implies numerous computer simulations in order to estimate its performances and assure the necessary adjustments.

The problem related to this "trial and error" method is that : 1) it is static, settings are fixed once and for all and at most a few additional algorithms [constant trace, MID88, JAZ68] with fixed parameters are used, 2) all disturbances are not taken into account in the model, and if they were it would rapidly become very difficult to integrate them into a classical program.

* : This work was realized under D.R.E.T. contract 87/464

** : also at the Laboratoire d'Automatique de Grenoble E.N.S.I.E.G., BP46, F-38402, Saint Martin d'Heres Cedex

We think there is an interest in using an expert system that scrutinizes and adapts the parameters of the Kalman filter subsequently to the interpretation of a situation. This intelligent adaptation of the filter makes it more robust, give better performances and be able to deal with complex situations.

In our application we considered different disturbances which could destabilize the filter :

- a modeling error, the on-board model can be for reasons of computing load restrictive compared to the reality (non-linearities, or simplification of higher order terms),

- unmodeled variations of the parameters of the dynamic model,

- false measures which give wrong updates,

- a sudden and inopportune disturbance bringing a sudden change in the state vector,

- a disturbance of visquous type,

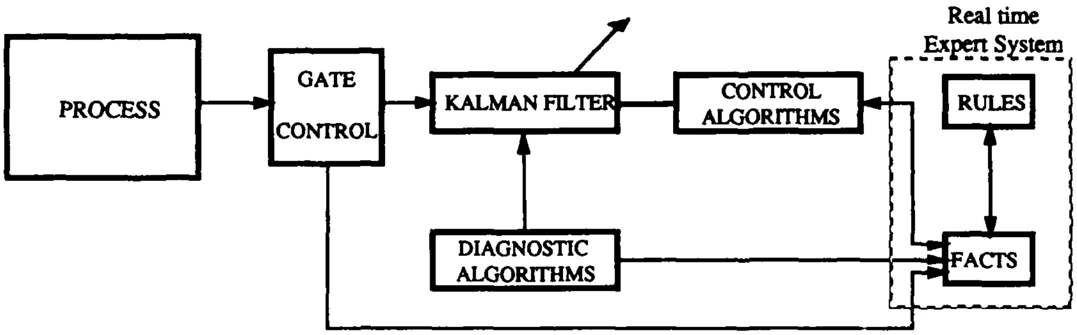
- a noise on the model with abnormal spectrum and level,

- biases on the measurements ...

1.3 Architecture of the system

To ensure proper functioning of the filter an on-line action on the parameters is necessary. This action can be decided after a diagnosis of the problem has been inducted ; further actions and a finer analysis can also be undertaken. Different strategies can be implemented but must remain general and flexible in order to take into account both uncertainty on the type of disturbance and its time of arrival.

A general schematic gives further insight on the proposed architecture :



The process (in our case a simulation of the trajectory of the satellite and of the landmarks sighting) emits the measurements toward a gate control which checks their validity before sending them to the Kalman filter.

A series of diagnostic algorithms analyzes in real time different parameters outgoing from the Kalman filter and generates initial facts. These facts trigger the inference process realized within the Expert System. Eventually some newly created facts will trigger one or several "control algorithms" which adapt the filter.

The library of control and diagnostic algorithms serves as an interface between symbolic data on the Expert System side and numerical data handled by the filter. As indicated below the algorithms vary from the setting of a simple threshold to sophisticated tools of Automatic Control.

1.4 Sources of knowledge

The knowledge at our disposal (formalized as rules within the Expert System) can be broken down into three categories :

a) Rules that rely on the knowledge of the Kalman filter

They characterize our knowledge relative to the "behaviour" of the filter. They will allow, in particular, to analyze innovation residuals in order to detect abnormal updates or the occurence of a perturbation.

b) Rules that rely on a physical knowledge of the process

They translate our knowledge about the physical laws underlying the observed phenomenon. For instance, in our application, we know that th· energy of the satellite is necessarily decreasing if the effect of the drag is augmented.

c) Rules that rely on the expertise acquired

Those rules represent shallow knowledge compared to the preceding ones. They result for example of the observation of the behaviour of the filter after having imposed new settings (various experiences with different simulated disturbances show for example a relationship between time of convergence and increase of model noise covariance, or evolution of performance with time and duration of those settings). They characterize our knowledge about the behaviour of the sytem obtained by numerous computer simulations.

2 - THE RULE-BASED SUPERVISOR

2.1 Functional organization of the supervisor

The functions of the rule-based supervisor can be divided into five parts :

a) *Elaboration of a criterion*

Before any reasoning the expert system elaborates a criterion which depends on the specific situation. Adapting the criterion facilitates the task of interpretation.

Experimentation and expertise allowed us to select the best criteria depending on the situation (context). For example, when trying to detect or identify a model disturbance (shock, visquous force) the time derivative of the estimated energy was chosen. In opposition when trying to detect a permanent error on the measurements or false updates, simple reasoning on the residual innovation was preferred.

b) *Monitoring*

The evolution of the different criteria is continuously monitored in order to detect abnormal behaviour of the filter, identify a disturbance or analyse in a finer way a diagnostic already suggested.

For example, tracking innovation residuals of the filter taking or not into account the precedent update (rule explained further on) allows the detection of false measurements. Also a negative time slope of estimated energy immediately followed by a positive slope is significative of a shock in the direction of the speed vector of the satellite.

c) *Identification*

A first or gross interpretation assured by monitoring can be further refined by calling different algorithms which will confirm or disprove the previous diagnostic(s).

Confirmation helps estimate more accurately the state of the filter, the type of the disturbance, its intensity or time of application. It usually utilizes final decision algorithms and necessitates fine tuning of those. Adaptative tuning of algorithms is a very powerful way of refining a diagnosis, sometimes complementary algorithms may also be used.

When the results of the algorithms called to refine the analysis of the perturbation are not consistent with the hypotheses the original diagnostic is cancelled. In that case the interpretation phase takes over again.

d) *Correction*

Once a disturbance is coarsely or finely identified, it is necessary to adjust the filter. There is usually a two stage analysis.

In a first stage, when the disturbance has been coarsely identified, which means that it has been detected and roughly classified, an immediate action on the filter is triggered. It consists of a very coarse adaptation of the filter like multiplying the covariance of the model or measurent noise by a set factor, which is based on off-line expertise. This correction doesn't assure complete optimal correction of the filter but at least allows for better overall performances and augments the speed of convergence. This last fact is pertinent for further analysis because future state estimates will be closer to reality.

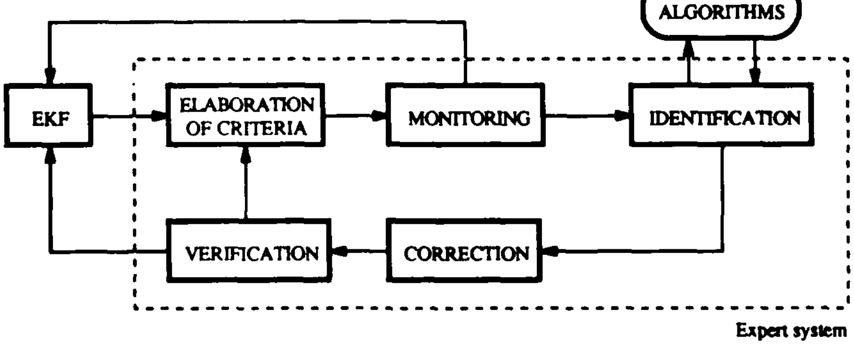
In a second stage, when the disturbance has been more finely identified, and if further adaptation of the filter is deemed necessary, more radical modifications of the filter are undertaken. For the case of satellite navigation it meant, for example, including in the dynamic matrix of the filter the model for atmospheric drag with the correct coefficient if such a perturbation had been identified.

e) *Verification*

After the Kalman filter has been corrected rules are used to verify the good adequacy of this correction with reality. For example after correcting the dynamic model of the filter in order to take into account a presumed visquous force (that was identified by the Expert System), the Expert rules verify that : 1) the time slope of the energy remains negative, and 2) the slope corresponds to the predicted evolution of the perturbation when correction is taken into account.

These verifications allow the expert system to "forget" the disturbance and start with a new set of criteria which will eventually allow the detection of other disturbances or an abnormal behaviour of the filter.

The schematic below resumes the above described functions.



2.2 Description of the algorithms

The main algorithms used by the Supervisor are the test of Page-Hinkley [BAS86], a multimodel analysis similar to the multimodel adaptative filter test used in [GRIM88], an algorithm that checks the coherence of the filter updates based on comparison of innovation residuals with or without update,the "Constant Trace" algorithm which ensures a constant value for the trace of the matrix of the covariance of state estimates errors,... Numerous other algorithms simply consist of the thresholding of numerical data, the setting of parameters of the filter, or the modification of the dynamic matrix.

We describe below two algorithms used for identification.

a) Detection of false measurements

This algorithm allows the detection of false measurements, inconsistent with the current estimated position of the satellite.

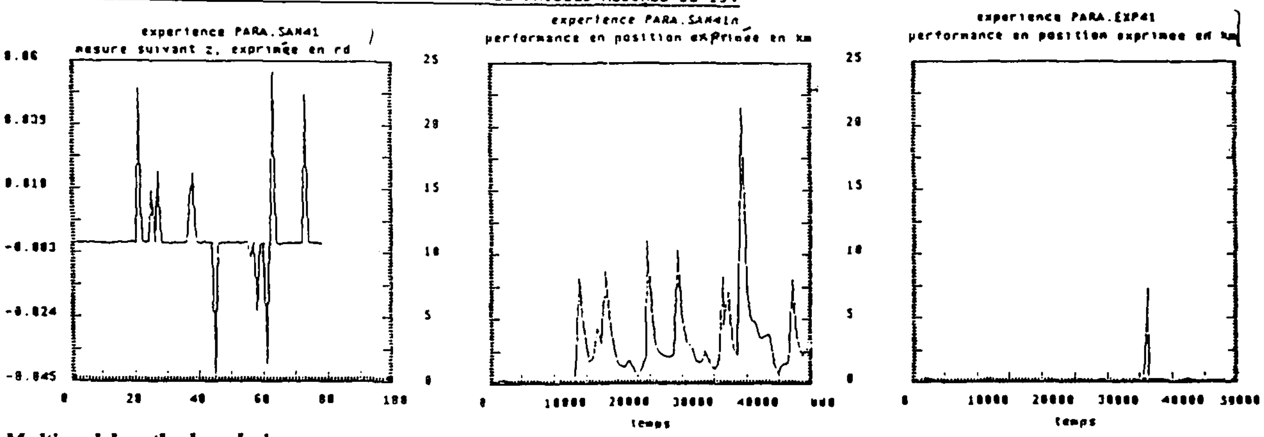
The idea is to consider two filters, one which takes into account updates, triggered as soon as a problem is detected (filter 1), and one which doesn't take into account any update, starting also when a problem is detected for the first time (filter 2).

A problem is detected when the innovation residual is bigger than what expected without false measure (typicaly < 4.10^{-4} rd in our case). Starting from that moment (u) the two filters described above are propagated until next update (u + 1). Then, if the innovation residual of filter 1 is bigger then a certain threshold while the innovation residual of filter 2 is smaller than a second threshold, a false measure is said to be confirmed at update u ; if both innovations residuals are bigger than the two thresholds, then u and u + 1 are considered as possible candidates for erroneous measures but cannot yet be confirmed as such. The idea is to wait until innovation residuals of filter 2 becomes smaller than the prescribed threshold.

The algorithm generates a couple of boolean values describing the state of the innovation residuals of the two filters. Expert system rules reason on successive couples of values in time and consider particular cases (as for example rule MESAPO2 described in chapter 3).

When false mesures are confirmed, all measures preceding confirmation and following first detection are eliminated.

PERFORMANCE AVEC/SANS SYSTEME EXPERT POUR UN TAUX DE FAUSSES MESURES DE 15%



b) Multimodel method analysis

This algorithm consists in running N Kalman filters in parallel, affecting a probability to each of the resulting innovation residuals and finally in choosing the best filter according to the evolution of the probabilities and comparisons between them.

The probability chosen is the Bayes estimate obtained from the innovation residuals :

$$(1)\quad Pi(n+1) = \frac{Pi(n) * P(y(n+1) / Hi, In),}{\sum_{j=1}^{N} Pj(n) * P(y(n+1) / Hj, In)} \quad \text{, with}$$

$$(2)\quad P(y(n+1) / Hj, In) = \frac{\exp(-0.5 * Pj(n+1)^{t} * Vj(n+1)^{-1} * Pj(n+1))}{(2\pi)^{m/2} * \det(Vj(n+1))^{0.5}}$$

This probability estimation is completely described in Basseville's book [BAS86].

The smaller the innovation residual, the better the model estimate and the closest the probability to one (see (2)).

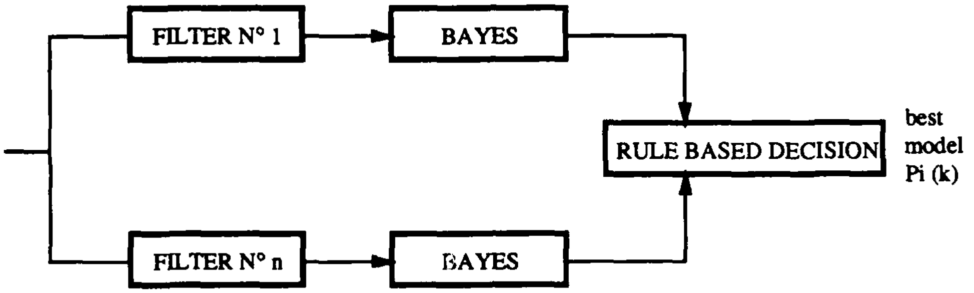
The algorithm consists in testing different filters, selecting the one which appears to best represent the underlying phenomenon. This selection uses the probability criterion above described.

For our Supervisor (in the satellite navigation case) this algorithm is used to refine our knowledge of the perturbation when an evolution of the drag is detected. The Page-Hinkley test gives us a good approximation of the time of appearance of the perturbation. The value of the time slope of the estimated energy is a coarse indication of the viscosity coefficient. The "Multimodel" test is used to refine those two parameters The different filters correspond to various viscosity coefficients and various time of appearance centered around the preliminary values.

Considering the fact that computation of covariance matrix and gain at each update takes a tremendous amount of time, we reduced the computation load by reducing the number of tests, by using filters with no updates who were only propagated, and by calculating a mean covariance for all filters. The resulting bad convergence of the Bayesian probabilities implied a slight modification of the preceding algorithm by introducing a power coefficient α in (2) which reduced the sensibility of the method. In most cases α was first chosen to be equal to 0.3 and was upgraded to 0.6 when the analysis was refined with fewer models.

The mechanism of decision is done by rules and concerns the 4 biggest probalities at each update. Corresponding rules are described in chapter 3.

The following diagram illustrates the method :



We note that the algorithm works both with values stored in the past and with incoming future values (but with limited depth of search in time).

2.3 Constraints on the Expert System and solutions

The different functionalities required by the supervisor implie that certain constraints on the expert shell used to develop it be verified.

- The Expert System must communicate in real time with an outside process (in our case a computer simulation). The facts generated (outputs of the diagnostic algorithms) must be dated to enable the Expert System to take into account time in its reasoning. It is useful to have a datation mechanism of facts which is intimately related to the shell and doesn't have to be managed explicitely by the user.

- It must be possible to include time constraints within the rules. In our application the rule "ENERGIE_MOINS_FROTTEMENT" (see § 3.2) checks that the fact "energy slope < set value" is true for at least 2000 sec. before being triggered.

- For the final application (on board filter) reasoning with time implies the synchronisation with a real clock. In our case, for simulation purposes, a virtual time mode is needed. The simulation explicitly defines the time progress.

- The supervision process must work continuously over time, which implies specific management ᴗ ᴗne facts base. The garbage collection of dated facts should not interrupt the reasoning.

- Whenever a new fact is created the E.S. should check if this fact does not alreadt exist and is currently valid (as indicated by the validity interval associated to the fact) This prevents triggering several times the same rule.

- In order to avoid a saturation of the facts base while being able to operate on past facts an "obsolescence date" should be associated with each fact. Once the real or virtual time has reached this date the fact is automatically eliminated.

- The acquisition of new facts must be taken care of at any time, occasionaly interrupting the reasoning.

- In some cases these new facts must trigger prioritary rules and thus suppress the current inference. In our Supervisor checking the validity of a new update stops, for example, a diagnostic phase.

CHRONOS* is a shell that satifies all preceding constraints. It is a real time expert shell which allows real-time firing of rules uppon reception of a new fact, it allows datation of facts and has internal mechanisms to manipulate dated facts. The concepts of priority, obsolescence, invalidation, time intervals and overlapping are all inherent to this expert shell.

The first order logic (with logical operators no, for any, exists, loop, ...) and right hand procedural language (allowing communication with other langages) is very useful to reduce the number of rules, assure a better reading and, most important, assure a very flexible maintenance of the knowledge base.

We found it very easy to modify the rules. Internal dictionnary and automatic graph maintenance allowed accelerated development. The 70 rules were built on a one rule-one day basis.

*CHRONOS is a product that was developped by EURISTIC SYSTEMS and SAGEM .

3 - KNOWLEDGE DESCRIPTION

3.1 Structure of the knowledge base

The knowledge base is broken down into four classes.

- *Servitude rules (12) :*

These rules concern the creation of the facts from the simulations results. These facts can be the value of the filter parameters (as the innovation residual, state vector, covariance matrix), or the result of certains transformation (e.g. energy estimate, covariance and slope).

- *Measurement rules (6) :*

These rules are used to detect erroneous measurements and permanent oscillations on measurements.

- *Diagnostic rules (30)*

. Multimodel
These rules call the "multi model algorithms". They allow the deduction of the date of appearance of a visquous type disturbance and the determination of its intensity.

. Selftuning
Those rules determine the optimal tuning of the measurement and model noise covariance after a model disturbance.

. Evolution of energy
Those rules allow a first classification of the disturbances (i.e. stable energy, decreasing energy, increasing energy) and permit their identification (caracteristic signatures of shocks and visquous forces, recognition of an oscillating energy caracteristic of an inertial thrust type).

- *Control rules (15) :*

They act on the filter parameters, or on the diagnostic algorithms themselves :

. rule for tuning of Q (model noise covariance),
. rule for tuning of F (dynamic matrix),
. rule for tuning of R (measurement noise covariance),
. rule to eliminate erroneous updates,
. rule to modify the filter parameters (page Hinkley threshold, multimodel power coefficient),
. rules for going backwards in past and modifying filter parameters.

3.2 Examples of rules

We present two diagnostic rules.

1) The rule **ENERGIE_MOINS_FROTTEMENT** is fired when the energy decreases for more than 2000 sec., providing that no correction was done on the filter for the last four updates. It puts back Q to its nominal value and autorizes a finer diagnostic of the perturbation.
The fact "valeur (energie totale) = decroissante" (decreasing total energy) results from a computation on the estimated energy.
The expression "clock > !te1 + 4 * !ir" (!ir is the period between two updates equal to 500 sec., clock is the current date, and !te1 defines the beginning of the logic appearance of the event "valeur (energie totale) = decroissante") imposes that the fact "the energy is decreasing" has been true for at least 2000 sec..
The condition part of the rule also checks that there has been no correction for 4 updates with the expression "clock > =!t3 + 4 * !ir" and that no shock diagnostic has been started during the last 10 updates with the expression "no (exists recal (choc _ ou _ frottement) = !rcf [!t5, !t6] such that !nact - !rcf < 10)".

```
rule ENERGIE_MOINS_FROTTEMENT :

as soon as

        numero (recal) = !nact ;
        variation (energie_totale) = decroissance [!t1, !t2] {!te1, !te2} ;
        pente (energie_totale) = !pet ;
        pente_courante (energie_totale) = !pcet ;
        correction (frottement) = vrai [!t3, !t4] ;
        intervalle (recalage) = !ir ;
        clock >= !t3 + 4 * !ir ;
        clock < !t4
        clock > = !te1 + 4 * lir ;
        !t2 = !te2 ;
        no (exists recal (choc_ou_frottement) = !rcf [!t5, !t6] such that !nact - !rcf < 10) ;
```

```
then

    recal (detection_frottement) := !nact ;
    recal (petit_Q) := !nact ;
    valeur_probable (frottement) :=5.0e+7 * (!ept - !pcet) / (3 * !ir) ;

end rule ;
```

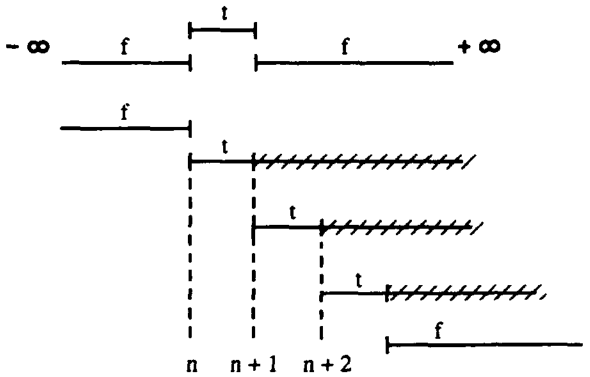
2) The rule **MESURE_ABERRANTE** (erroneous measurement) is used to cancel an isolated erroneous measurement, that is one which appears only in one update; the high innovation residual corresponding to a potential false measurement, when detected, creates the fact "mesapo (recal) = true".

The fact mesapo (recal) = t [!t1, !t2] and clock > !t2 allows the rule to be fired only when the event "erroneous measurement" disappears.

The duration of one update is expressed by numero (recal) = !n [!t5, !t6] with !t1 = !t5 and !t2 = !t6.

The fact numero (recal) saves the differents update numbers. When the rule is fired, it suppresses the corresponding update by assigning the (n + 1) th update to the nth in the facts base and in the filter simulation (call function).

Evolution of the fact "mesapo (recal)" (internal representation) :



isolate erroneous measurement

erroneous measurement on 3 updates

. three intervals are successively created with the time value, each one invalidate the previous

. ∞ or infp means infinity

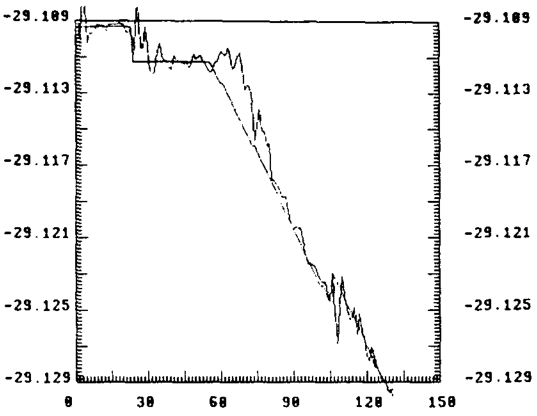
The rule also stops all multiple model reasonning that would have been triggered with the fact "arret (multimodel) = vrai".

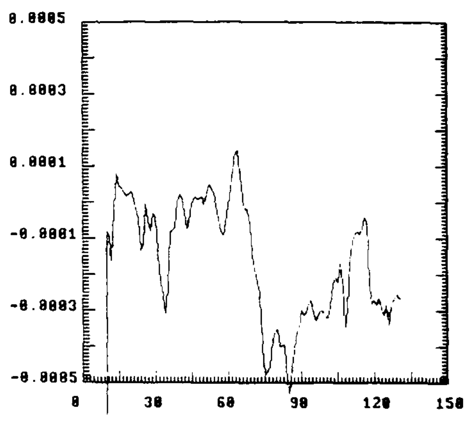
```
rule MESAP02

as soon as

    mesapo (recal) = t [!t1, !t2] ;
    coeff_vrais (recal) = !cv [!t3, !t4] ;
    numero (recal) = ! n [!t5, !t6] ;
    clock >= !t2 ;
    !t1 >= !t3 ;

    !t4 >= !t2 ;
    !t5 = !t1 ;
    !t6 = !t2 ;
    !cv < 0.5 ;

then

    !x := 2 ;
    !nact := !n + 1 ;
    !reponse := arr ;
    call "com_kalm.exe" (!x:in, !reponse:in, !n:in, !pdd:out) ;
    point (dialogue) := !pdd ;
    numero (recal) := !n [!t5, infp] ;
    temps (!n) := temps (!nact) ;
    mesapo (recal) := f [!t5, infp] ;
    !cvr := coeff_vrais (recal) ; coeff_vrais (recal) := !cvr [!t3, infp] ;
    !incr := incertitude (recal) ; incertitude (recal) := !cvr [!t3, infp] ;
    !incr := ecart_angulaire (recal) ; ecart_angulaire (recal) := !e [!t3, infp] ;
    arret (multimodel) := vrai [clock, clock + 1] ;

end rule ;
```

3.3 Session example

The session presented here shows the behaviour of the expert system in the case of two disturbances, shock and visquous force, which are described on the left figure. This figure shows the evolution of real and estimated energy, the two diagnostics, shock and visquous force, are identified by (1) and (2).

The right figure shows the energy slope used as a criteria for detecting and identifying the disturbances





Session output :

EXPERT SYSTEM MESSAGES EXPLANATORY COMMENTS

Le recalage 24 est valide au temps 16050

. The update 24 has been validated, but a decrease of energy is detected. The fact "pente (energie_totale) = - 1,7.10-4" created by the rule "DEUXIEME_POINT_DE_DIALOGUE" fires the rule "ENERGIE_DECROISSANTE", which creates at date 16050 the fact "variation (energie_totale) = decroissante". Consequently, rule GROS_Q is fired, which triggers a one hundred time increase of the model noise covariance.

Une perturbation a ete detectee au recalage 24 par une forte decroissance del'energie: difference de
pente =-1.3747340E-04. On regonfle Q d'un facteur 100 : nouvelle valeur de Q = 10000e-15.
Le recalage 27 est valide au temps 18150.

The "ENERGIE_MOINS_FROTTEMENT" rule is fired 2000 sec later subsequent to the fact that the energy slope remains negative (this one indicating that the disturbance detected at update 24 was probably an additional visquous force of intensity 1.7582.10^{-4} SI).

L'energie totale decroit depuis plus de 2000 secondes (T0 + 16050.00). Un
multimodele identifie 1 est donc lance entre les recalages 11 et 20 autour de la valeur
1.7581955E+00e-9.
Les coefficients de frottement testes sont : 8.7909775E+00e-9, 3.5163910E+00e-9,
1.7581955E+00e-9, 8.7909775E-01e-9, 1.7581955E-01e-9,
La perturbation la plus probable du multimodele 1 est la perturbation 6 apparue au
recalagè 11, avec une probabilite de 9.5920607E-02. Q est remise a sa valeur courante
: 100e-15 le recalage 28 est valide au temps 18825.

Firing of the rule "ARRET_ET_MULTIMODELE" consecutive to the determination of a null viscosity coefficient. This rule stops all multimodels algorithms.

Le frottement relatif est nul => la perturbation n'est pas un frottement.
Le frottement le + probable est nul => interruption des multimodeles en cours.
le recalage 29 est valide au temps 1955,
le recalage 30 est valide au temps 20275,
le recalage 31 est valide au temps 20975.

New firing of a multimodel analysis (identified 2) four updates after the multimodel identified 1 (update 27), consecutive to a persisting decrease of the estimated energy.

```
L'energie totale decroit depuis plus de 2000 secondes (T0 + 16050.00). Un multimodele identifie
2 est donc lance entre les recalages 11 et 20 autour de la valeur 4.7014206E+00 e-9 .Les
coefficients de frottement testes sont : 2.3507103E+01e-9, 9.4028413E+00e-9, 4.7014206E+00e-
9, 2.3507103E+00e-9, 4.7014207E-01e-9, 0e-9. La perturbation la plus probable du multimodele
2 est la perturbation 3 apparue au recalage 20, avec une probabilite de 1.2010048E-01le recalage
32 est valide au temps 21000.
La perturbation la plus probable du multimodele 2 est la perturbation 3 apparue au recalage 18,
avec une probabilite de 1.9741303E-01
Le recalage 33 est valide au temps 21675
La perturbation la plus probable du multimodele 2 est la perturbation 3 apparue au recalage 18,
avec une probabilite de 3.0650529E-01
le recalage 34 est valide au temps 22375
La perturbation la plus probable du multimodele 2 est la perturbation 3 apparue au recalage 19,
avec une probabilite de 2.6336715E-01
le recalage 35 est valide au temps 23075.
```

When examining the previous results (probabilities) of the multimodel test for the four last updates, the "STABLE_OU_OSCILLANT" rule determines an "oscillating" result, and decides to fire another multimodel analysis (rule "ENCORE_UN_MULTIMODELE").

```
Apres 4 recalages, le multimodele 2 n'ayant pas determine la pertubation, un multimodele affine,
identifie 3, est lance entre les recalages 16 et 21
Les coefficients de frottement testes sont : 4.7014206E+00e-9, 4.1137430E+00e-9,
3.5260655E+00e-9, 2.9383879E+00e-9, 2.3507103E+00e-9, 0e-9
La perturbation la plus probable du multimodele 3 est la perturbation 1 apparue au recalage 21,
avec une probabilite de 2.1264811E-01
le recalage 39 est valide au temps 25200
La pente de l'energie redevient superieure a sa valeur courante avant la perturbation. Celle-ci etait
donc un choc au recalage 24, et non un frottement comme annonce au recalage 31. => la detection
de perturbation n'est plus possible jusqu'au troisieme recalage suivant la stabilisation de ce choc.
La direction du choc detecte au recalage 24 reste a determiner.
Le frottement le plus probable est nul => interruption des multimodeles en cours
```

Interruption of the current multimodels analysis because a positive slope of the estimated energy implies that the disturbance cannot be a visquous force (conclusion of rule "ENERGY_MOINS_CHOC"). Taking into acount the abnormal null result of the previous multimodels analysis it is deduced that the cause of the disturbance is a shock.

```
le recalage 40 est valide au temps 25900
La direction du choc detecte au recalage 24 reste a determine:....
le recalage 43 est valide au temps 28025.
```

End of diagnostic of the first disturbance. Determination of the direction of the shock by the rule "SENS_DU_CHOC".

Consecutive to the firing of the rule "MESAPO2", corresponding to an erroneous measurement, disturbance detection is inhibited for 10 updates.

Firing of the "ENERGIE_DECROISSANTE" rule, consecutive to a negative energy slope, increasing model covariance by a 100 times factor.

Firing of the "ENERGIE_MOINS_FROTTEMENT" rule.

The rule "STABLE_OU_OSCILLANT" decides that preceding results are stable, and fires the two rules "MULTIMODELE FIN" and "MULTIMODELE _VRAIMENT_FIN" in order to have a finer assessment of the viscosity coefficient.

Comparaison of results of the two fine models analysis concludes to a viscosity coefficient of intensity 6.36 x 10^{-9}. The rule "ENVOIE_FROTTEMENT" corrects the filter in consequence and increases the model covariance by a 10 times factor in order to improve convergence.

End of diagnostic of the second disturbance. The Expert System verifies that the estimated energy slope corresponds to the determined viscosity coefficient.

4 - RESULTS

The simulations were realized in FORTRAN on a SUN 4 WorkStation. Software modules include trajectory generation, disturbances and measurements simulations, extended Kalman filter implementation and a library of algorithm routines both for diagnostic and control. Rules were implemented with the help of the Real Time Expert System shell CHRONOS, which allows use of first order logic.

We chose the case of a low orbit satellite, typically a satellite with perigee of 400 km and an excentricity of 0.01. Landmark sightings were taken each time the landmark line of sight was less than 27° from the local vertical : it corresponds to a 600 s mean time interval between updates. Modeling errors were assimilated to a 5 µg random white noise and measurement errors on the direction of the line of sight taken equal to 20".

Six types of disturbances were considered :

- frontal shocks (direction opposite to the velocity vector ofthe satellite) ,
- non-frontal shocks (for example an unwanted thrust during an orbit maintenance maneuver),
- visquous force (due to drag),
- inertial thrust of long duration (due to the effect of solar radiation),
- false measurements (due to an error in the landmark recognition algorithm),
- sinusoidal perturbation of measure (due to thermal distorsions in the landmark sensor).

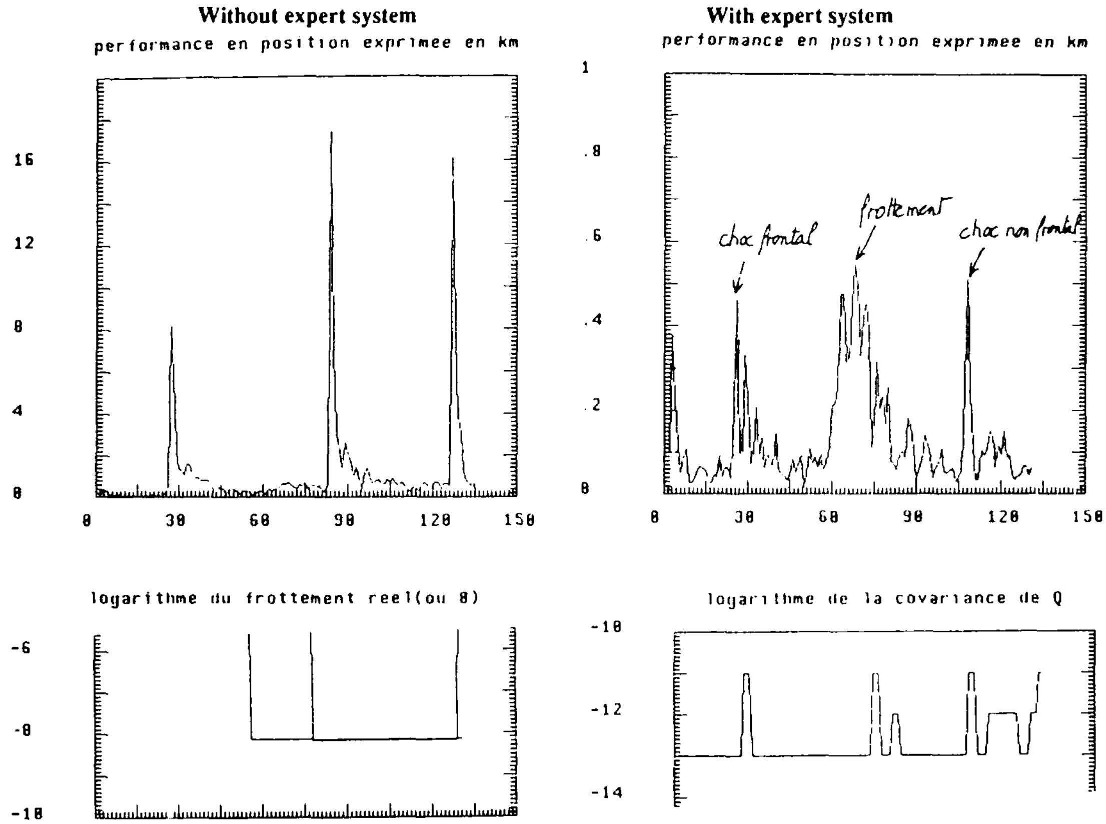
The shocks were taken between 0.1 and 50 µg with a time duration comprised between 25 and 50 s. They correspond to typical satellite orbit corrections. Visquous forces correspond to unpredictable solar activity which modifies drastically the density of atmosphere, they have amplitudes comprised between 5.10^{-10} and 5.10^{-8} SI (0.5 µg to 50 µg). Inertial thrust corresponds to solar pression of radiation, we chose values between 10 and 200 µg with durations comprised between 5000 and 15000 s. False measurements were rated to be able to attain 30 km and were able to have up to 30 % occurence, the occurence being parametrizable. Thermal noise was modelised by a sine wave in the order of one mrd which was overimposed on the measurements.

The filter has been nominally set with a compromise between robustness and precision. However the range of the possible disturbances is too important to avoid the divergence of the filter without Supervisor.The values chosen for model and measurement noise covariance are respectively 10^{-13} (km/s^2) and 2.10^{-8} (rd)^2.

Performances with and without the Expert System were compared for each type of disturbance taken seperately and for different intensities. The combination of several disturbances was then simulated and performances were again evaluated. The table underneath resumes some of the experiments :

	WITHOUT EXPERT SYSTEM			WITH EXPERT SYSTEM		
	Peak	Performance	Time of convergence	Peak	Performance	Time of convergence
Altitude shock (10^{-5} = 1 mg)						
+ 5.10^{-6}	/	peaks at 190 m	/	417 m	peaks at 145 m	5 updates
+ 10^{-5}	365 m	peaks at 200 m	8 updates	410 m	peaks at 120m	6 updates
+ 5.10^{-4}	18.82 km	140 m	34 updates	18.9 km	110 m	19 updates
Frontal shock						
- 5.10^{-6}	800 m	180 m	12 updates	800 m	125 m	14 updates
- 5.10^{-5}	8.34 km	125 m	16 updates	7 km	125 m	14 updates
Non frontal shock						
+ 5.10^{-6}	925 m	190 m	12 updates	890 m	190 m	12 updates
1.10^{-5}	1.78 km	140 m	14 updates	1.47 km	125 m	9 updates
Visquous force (S I)						
1.10^{-9}	peaks at 250 m	145 m	/	250 m	145 m	/
3.10^{-9}	peaks at 400 m	300 m	∞	360 m	170 m	∞
5.10^{-9}	peaks at 550 m	400 m	∞	407 m	125 m	20 updates
7.10^{-9}	peaks at 725 m	600 m	∞	470 m	125 m	10 updates
% of good measures						
70 %	23 km	diverge : (7.7 km)	∞	16.45 km (4 peaks)	200 m	start of divergence (slow)
75 %	22.9 km	diverge : (7.4 km)	∞	7.25 km (1 peak)	110 m	5 updates
85 %	21.6 km	peak at 8.36 km	∞	7.31 km (1 peak)	100 m	10 updates

The multiple disturbance session described in chapter 3 can also be analyzed in terms of performances (position error with respect to time) :


Without expert system
performance en position exprimee en km
16
12
8
4
0
0
30
60
90
120
150
With expert system
performance en position exprimee en km
1
.8
.6
.4
.2
0
0
30
60
90
120
150
choc frontal
frottement
choc non frontal
logarithme du frottement reel(ou 0)
-6
-8
-10
logarithme de la covariance de Q
-10
-12
-14

CONCLUSION

Results have been very convincing. The Real-Time Expert System built around an original tool and containing 70 rules of first order logic has proven to be very robust. We selected different kinds of disturbances, like a variation of the dynamic model (for example atmospheric drag), the rapid variation of some states (example of a shock), or the presence of false updates. The Kalman filter gave poor results when no expert rules were applied, especially in the case of false updates or permanent variations of the dynamic model. But once coupled to the Expert System the EKF showed improved performances both in time of convergence and in absolute residual error on state estimate.

We have shown that in the case of false updates and when no expert system was used,the error on estimated position attained 7 km with peaks over 20 km. With the expert system the error in position, after having detected and eliminated bad measurements, was below 120 m. Also with permanent variations of the dynamic model like an atmpospheric drag of 7 µg the positioning error was improved from 700 m to 120 m. When no disturbance occurs the expert rules can still improve the performances of the filter by continually adapting the different elements of the filter (i.e. noise and model covariances), the performance thus attains 80 m, without adaptive control it would be 160 to 200 m.

The use of a Real Time Expert System shell allowed a fast and flexible development of the Supervisor. It is clear to us that the approach we selected should be beneficial for other applications requiring a combination of a knowledge based system and numerical algorithms together with real time constraints. Such applications abound in the military domain,in process control, navigation or decision support system.

BIBLIOGRAPHY

[HIG] "A self-tuning controller based on expert systems and artificial intelligence "E.H Higham, Foxboro.

[FRA] "Intelligent knowledge-based process control".
J.C. Francis, RR Leitch. Heriot - Watt University, Edinburgh.

[BER 87] "Use of a rule based system for process control".
John A. Bernard (MIT), IEEE IECON 1987

[AST 87] "Expert control".
K.J Aström, J.J. Arton, K.E. Arzen (LIT)
AUTOMATICA VOL 22, N° 3 1986.

[DUQ 88] "Real time supervised adaptive control"
M. Duque, M. Samaan, M. M'Saad, I.D. Landau (LAG).

[GEN 87] Qualitative modeling for process supervision systems".
S. Gentil et S. Féray Beaumont.
1st European meeting on cognitive science approches to process control, Oct. 87

[GRIM 88] "Kalman filter residual expert system".
J. Grimshaw, P. Amburn, NAECON 88.

[NAV 86] "Navex : a convincing testimory of A.I. viability".
DJPE, May 8

[BAS 86] M. Basseville, A. Benveniste Ed. (1986) : "Detection of abrupt changes in signals and dynamic systems". LNCIS n° 77, Springer-Verlag.

[GOOD 85] G. Goodwin, D. Hill, M. Palaniswami (1985) : "Towards and adaptive robust controller" 7th IFAC Symp. on Identification and System Parameter Identification, York.

[ZAR 87] "Trajectoires spatiales" Olivier Zarrouaki. CNES 1987.

[CAM 84] CAMMAN project, Final report SAGEM/MATRA July 1984

[HOR 89] "Supervision d'un filtre de Kalman par système expert temps réel.
Application à la navigation d'un satellite". D. BERTON, R. HORAK, S. SALLE.
SMAI Colloquium 1989, Paris.

APPLICATION OF MULTIFUNCTION INERTIAL REFERENCE SYSTEMS TO FIGHTER AIRCRAFT

Authors: John M. Perdzock
Wright Research and Development Center
Wright Patterson Air Force Base, Ohio 45433
United States of America

Jack Jankovitz
Naval Air Development Center
NADC/CODE 4011
Warminster, PA 18974-5000
United States of America

Carlos A. Bedoya
McDonnell Aircraft Company
P.O. Box 516
St. Louis, Missouri 63166-0516
United States of America

SUMMARY

Advanced high performance fighter and transport aircraft depend on reliable, accurate, inertially derived data to perform such subsystem functions as fly-by-wire automatic flight control, fire control, weapon delivery, navigation, guidance, and cockpit displays. These data are currently obtained from inertial reference sensors which are essentially dedicated for each major functional application. Future inertial reference systems must meet the requirements of the functional users and have lower life cycle cost, increased survivability to combat damage, and improved reliability and accuracy. One way to meet these requirements is to use the same sensors to provide kinematic data to all applications. The feasibility of such a system was investigated in the Multifunction Inertial Reference Assembly (MIRA) study. The MIRA is defined by combined inertial requirements for flight control, cockpit display, weapon delivery, and navigation resulting in a common reference package for multiple users. A history of development activity is shown in Figure 1.

INTRODUCTION

The Multifunction Flight Control Reference System (MFCRS) program was initiated to develop and flight demonstrate the following flight control techniques of MIRA in a high dynamics fighter aircraft: (1) use high grade dynamic strapped down inertial reference instruments as flight control reference sensors, (2) use control laws to compensate for the effects of clustered sensors (gyros and accelerometers) at non-optimal locations, and (3) develop redundancy management for skewed and dispersed sensor clusters.

The MFCRS laboratory evaluation was followed by a ground structural mode interaction test and a two phase flight test program. The objective of the first flight test phase was to verify MFCRS air worthiness, to compare and evaluate MFCRS flying qualities with the flying qualities of the basic F-15, to verify proper MFCRS redundancy management operation, and to verify that MFCRS sensors were of navigation quality. Conducted during February 1984, the Phase I flight test evaluated the MFCRS redundancy management operation, revealed a low damping problem in the MFCRS Control System response at medium to high dynamic pressure flight conditions, and verified the navigation accuracy of the normal and skewed sensors. When the MFCRS low damping problem appeared it became necessary to account for the differences between expected and actual performance. The low damping analysis identified the problem to be time delays present in the system. The second part of the flight test program was conducted in July 1984 with time delays reduced within the capability of the system without major modifications. The details and results of both phases of the flight test program, together with the causes of the low damping problem are discussed in Volume II of this report.

After Phase II of the flight test, changes to the MFCRS hardware and software structure were identified which would improve system performance and expand the MFCRS flight envelope. These changes were designed and evaluated as part of the "Enhanced MFCRS" (EMFCRS) study which began in the fall of 1984. The EMFCRS Program involved the development, implementation, and laboratory evaluation of the necessary hardware and software changes to expand the MFCRS flight envelope. Following the laboratory evaluation, additional flight testing was planned for the fall of 1986 to verify that the

EMFCRS configuration would result in level 1 handling qualities in both supersonic and subsonic flight as well as in tracking of target aircraft.

Unfortunately, during ground testing prior to flight in August of 1986, a 22 Hz structural mode interaction was found in the control system pitch channel. This mode was unexpected as it had not been observed previously and was not in the available structural model. For a successful application of the Multifunction concept, an accurate aircraft structural model would obviously be required. Different hardware would also be required.

It became evident at the conclusion of the EMFCRS Program (1985/86), that to fully evaluate the multifunction concept, specially designed hardware and a digital flight control computer were required. To achieve the Level 1 handling qualities over the entire F-15 envelope the flight control system using the ring laser gyro/accelerometer assembly must be designed from the beginning with digital flight control requirements in mind. Also, modifications must be allowed in both the forward and the feedback loops of the flight control system, and the digital flight control system must operate at least at an 80 Hz iteration rate to achieve the bandwidth required and to reduce phase lags to a minimum. At the same time, the Navy was seeking a cost effective approach to test its ring laser gyro Integrated Inertial Sensor Assembly (IISA) hardware. The IISA hardware had digital outputs and the sensor block was designed for navigation and flight control, it had reduced sensor quantization, low dither noise and fast (8 MHz Z8002) microprocessors. The Ada Based Integrated Control System test aircraft (F-15B S/N 77-0166) had a digital flight control system with mechanical backup and an Integrated Flight and Fire Control (IFFC) system already on board. The modification of the digital flight control computer I/O to add the required interface to the IISA sensors was the only major hardware change required. As a result, the joint ABICS III program emerged.

MCAIR was also involved in evaluating the Ada High Order Language (HOL) in real time embedded systems applications programs since 1983/84. To take the next step in the evaluation of embedded systems using Ada, the navigation algorithms and the redundancy management would now be programmed and flight tested. The test aircraft would now have Ada software to implement the flight control laws, IFFC, redundancy management of the flight control sensors, and the inertial navigation. A five channel Global Positioning System (GPS) would also be on board to help score the navigation performance. Figure 2 shows the aircraft configuration. In September 1986 MCAIR received a contract from the Air Force and Navy to design and flight test the flight control and navigation system using IISA sensors and the Ada HOL software.

ABICS III REQUIREMENTS AND DESIGN OBJECTIVES

The MFCRS/ABICS III program requirements called for: (1) the design and implementation of modifications to the F-15 aircraft test bed; (2) analysis, development and test of control laws and redundancy management using the IISA sensors; (3) design, development, test and evaluation of all software coded in Ada; (4) planning and performance of laboratory tests including hardware/man-in-the-loop tests, and (5) planning and support of the flight test program at Edwards Air Force Base (EAFB).

The objectives of the MFRCS/ABICS III program were:

- To demonstrate through flight test that navigation quality sensors in strapdown configuration can also be used as fault tolerant flight control references

- To obtain real world experience using Ada in embedded integrated control systems

- To establish confidence in the viability of Ada

- To determine requirements for software engineering environments

- To obtain metrics of Ada usage

ABICS III ARCHITECTURE

The ABICS III avionics (Figure 3) consists of the baseline F-15 avionics and instrumentation plus the IISA, DEFCS, IFFC, GPS, and modified CC, but without the F-15 production Inertial Measurement Unit. The avionics systems added for the ABICS III program consist of the avionic equipment described in the following paragraphs.

Integrated Inertial Sensor Assembly (IISA) - The IISA Advanced Development Model is an integrated reference and navigation system using strapped-down ring laser gyros (RLGs) in a skewed configuration that provides navigation, flight control and weapon delivery capabilities with redundancy levels equivalent to a hexad sensor array. The IISA system is composed of the following units:

- Inertial Navigation Assemblies (INAs) - There are two INAs in the IISA system. Each INA contains three RLGs and three accelerometers mounted on a skewed sensor block and software for navigation and flight control dither filtering. The navigation software in one INA was programmed in ADA and the other INA was programmed in Assembly language for comparative purposes. A typical INA is shown in Figure 4.

- Control Display Unit (CDU) - The CDU provides the data display and data entry interface for the following operations: Navigation parameters, misalignments angles, RM thresholds, lever-arms coefficients, simulated sensor faults (programming, selection, initiation and annunciation), RM status, INAs status, FCPs status, and NAV/FCP processors memory inspect/change. The CDU is shown in Figure 5.

- Auxiliary Power Supply (APS) - This unit inputs aircraft power as a primary source and includes an internal battery back-up mechanism to provide uninterrupted power (for primary power loss of 7 seconds max.) to both INAs. Figure 6 is a picture of the APS.

Digital Electronic Flight Control System (DEFCS) - The DEFCS is a programmable flight control system which can be digitally coupled with outer loop devices during any segment of the combat mission. The DEFCS system consists of the following avionic equipment:

- Digital Flight Control Computer (DFCC) - The DFCC has the standard F-15 flight control laws incorporated in software. Modifications for ABICS III were made to the DFCC as follows: Increased processor clock rate to 10 MHz, added a serial digital interface to input rates and accelerations from the INAs, and added IISA redundancy management and flight control software. The DFCC is shown in Figure 7.

- Analog Flight Control Computers (AFCCs) - The pitch and roll/yaw AFCCs were originally modified for the DEFCS program and contain the analog circuitry required for the F-15 actuator interfaces. Figure 8 is a picture of the pitch and roll/yaw AFCCs.

Integrated Flight and Fire Control (IFFC) System - The IFFC system assists the pilot in air-to-air Gunnery, air-to-ground Gunnery, and Bombing. It comprises a modified Central Computer (CC), an Avionics Integration Computer (AIC), a Rolm Hawk/32 computer which issues command signals to the Digital Electronic Flight Control System (DEFCS) to aid in weapon delivery, and an IFFC Control Panel which provides the interface between system and pilot.

- Enhanced Avionics Integration Computer (EAIC) - The EAIC communicates with the Rolm HAWK/32 via the 1553 bus and performs the A-to-D and D-to-A conversions to pass IFFC information to and from the IFFC Control Panel via an analog interface. In addition the EAIC is used during on-ground simulation to provide the HAWK/32 with the necessary flight control feedback signals to perform simulated closed-loop calculations. The EAIC is shown in Figure 9.

- Rolm Hawk/32 Computer - The Rolm Hawk/32 is a general purpose airworthy computer that can communicate as a remote terminal on the MIL-STD-1553 avionics bus. It has a minicomputer styled microcoded CPU and is Ada compatible. The Hawk/32 was programmed in Ada to execute the IFFC control laws and will communicate (via the 1553 mux bus) with the CC, DFCC, and EAIC to perform to IFFC control functions. Figure 10 shows the HAWK/32 computer.

- Genesco Magnetic Tape Transport Assembly (MTT) - The IFFC software is stored in a removable magnetic cartridge which is installed in the Genesco MTT. On power-up the IFFC software is loaded into the Rolm Hawk/32 processor memory through a serial interface. Figure 11 is an illustration of the Genesco MTT.

- IFFC Control Panel (ICP) - The ICP is mounted in the F-15 cockpit and is used by the Pilot to perform the IFFC functions: Power-on, system status annunciation, on-ground and in-flight OBS functions. In addition the ICP has the IISA couple switch and annunciator, the GPS power-on switch and the GPS-ON discrete annunciator. The IFFC Control Panel is shown in Figure 12.

Figure 13 shows the aircraft hardware modifications required for the ABICS III program. The resulting ABICS III aircraft has the following capabilities:

- Flight Control - All F-15 control laws available with either production or IISA motion sensors in the DFCC.

- Navigation - All standard F-15 navigation modes are available using the IISA dual inertial navigation function.

- Fire Control - All functions are available using IISA reference information in the Rolm Hawk/32.

- IFFC - Coupling is available in IFFC weapon delivery modes using production F-15 or IISA control laws in the DFCC.

- GPS - GPS can be used for instrumentation as a reference for navigation position and velocity. This capability was not tested in our program.

ABICS III RESULTS

The control system for ABICS III uses a digital flight control computer and allows total system design. Compensation can now be added in the forward or feedback loops. In order to ensure an accurate aircraft structural model, a structural survey was conducted on the ABICS III aircraft (F-15B, S/N 77-166). An interesting result of this work was that this aircraft did not show the 22 Hz resonance seen on F15A S/N 77-139. No explanation has been found. (Surveys of other F15As, Bs, Cs and Ds would be of interest to characterize F-15 modes.)

Analysis and flight test show that the ABICS III control system has the same stability envelope as the standard F-15 and it achieved the Multifunction Inertial Reference goals of providing equivalent flight control system response as the current dedicated, sensors mounted at nodes and anti-nodes. Figure 14 shows how closely the F15/IISA response is to F-15/standard sensors.

Pilot comments from the flight test program were that the use of IISA sensors and the Ada HOL software for the flight control system in the test aircraft was totally transparent. The test aircraft performed the same with or without IISA or Ada which, of course, was the goal of the control system design.

ABICS III Navigation Performance - The IISA inertial navigation assemblies demonstrated that skewed sensors can have the same navigation performance as those aligned to the aircraft body axes; that the Ada can perform the computations required to process the inertial navigation algorithms and still have the same performance as assembly language software; and that skewed sensors can provide accurate inertial references for weapon delivery.

IISA's navigation performance was better than specified for position error and about as specified for velocity performance. Figure 15 summarizes the IISA navigation results.

ABICS III Integrated Flight and Fire Control (IFFC) - In ABICS III the previously developed IFFC system was coded in Ada and programmed into the Rolm Hawk/32 computer. The IFFC system was designed to help a pilot fly his aircraft, in air-to-air gunnery and air-to-ground bombing maneuvers, by sending steering commands to the flight control system (See Figure 16). These computer generated commands provide an attack solution which is blended in with the pilot's steering inputs. Both the AAG and BMG modes also have an onboard simulation (OBS) mode built into them, which generates simulated targets so that a pilot can practice switchology and maneuvering techniques without live targets or ammunition.

- Air-To-Air Gunnery - The IFFC system makes use of a radar director sight which can be flown in the normal manual manner using the standard F-15 flight control system or can be coupled to the IFFC flight control system for automatic tracking. It uses the F-15 radar to obtain angle and range data to the target.

- Air-To-Ground Bombing - The IFFC bombing mode allows both straight-in and maneuvering curvilinear approaches to the target. Both of these modes can be flown either manually or in an automatic (coupled) manner. The maneuvering approach profile, which was of particular interest in IFFC, can be varied by changing parameters such as release range, ingress altitude, ingress airspeed, and ingress climb angle. In the original IFFC program, the INS (ASN/109) was used to provide both angles and range to the target, in ABICS III data from the INAs was used.

Both AAG and BMG were included in ABICS III. Three tactical air-to-air flights against a T-38 and an F-4 were performed as well as several hundred air-to-air on-board-simulation encounters. In the bombing mode, several hundred BDU-33 practice bombs were dropped on the target at EAFB. IFFC bombing accuracy was the same as achieved in the original IFFC program. Pilot comments were that IFFC performed and flew the same using Ada and IISA as it did in the original IFFC program and the software source code and inertial reference source was transparent to them.

ADA IN ABICS III

In the ABICS III system there are three distinct computer systems programmed with Ada Operational Flight Programs (OFP) (see Figures 17 and 18). The IFFC algorithm for the Rolm Hawk/32 milspec computer was programmed entirely in Ada for ABICS III. The ABICS III IFFC OFP consisted of 20,440 lines of Ada source code, and when compiled and linked forms a 843 Kbyte executable object module. Major portions of the navigation computer in the IISA system were also programmed in Ada. The IISA NAV OFP consisted of 2263 lines of Ada source code and 4171 lines of assembly language source code. The total executable object code for the navigation algorithm was over 19 Kbytes. The Digital Flight Control Computer in the DEFCS system has standard F-15, IISA, IFFC, and IISA/IFFC flight control equations programmed in Ada. The flight control OFP was written in 2428 lines of Ada source code and 2644 lines of assembly language source code. These three computer systems are interfaced to each other over a 1553B mux bus forming a highly integrated and complex system (see Figure 19).

ABICS LESSONS LEARNED

One of the goals of the ABICS program has been to identify potential bottlenecks in the use of Ada in real time applications and to find solutions. The requirement, for a 32 bit computer with a hardware floating point implementation was a direct result of our experience with ABICS II. Figure 20 contains data on Ada software memory and timing expansions compared to previously flown assembly language versions of the same software. ABICS III also pointed out some valuable concepts for real time Ada programming.

Hardware Limitations - Throughput and memory limitations are the usual hardware limitations encountered in real time applications. Additional problems arise when a high order language such as Ada is interfaced with a real time piece of hardware.

- DFCC - Flight control applications typically don't require a lot of code, however, they are required to run very fast. In our case the DFCC flight control code runs in a 80 Hertz frame, which means all processing must be completed in less than 12.5 milliseconds.

Throughput was a major problem with the DFCC in ABICS III. ABICS III required twice as many sets of flight control equations as on ABICS II. In addition, due to the inability of the two flight control computers to communicate with one another, the RM software had to be executed in both the pitch and roll/yaw flight control computers. While only one set of flight control laws is used at a time, all modes (Standard F-15, IISA, IFFC, IFFC/IISA) were calculated every 80 Hz frame. Figure 21 compares the execution time requirements for the individual flight control software modules written in Ada for number of different computer systems. Although the DFCC clock speed was increased from 6 MHz to 10 MHz for ABICS III, there was still not nearly enough throughput to do the floating point Ada version of Redundancy Management in the DFCC. A fixed point assembly language version of RM was ultimately used.

The amount of available memory in the DFCC was not an issue in ABICS III. Even the large, all Ada, version of RM fitted into both the pitch and roll/yaw channels with room to spare.

- INA - The amount of available memory for the INA navigation software was also not an issue in ABICS III, even with a worst case 3 to 1 expansion factor of Ada code size over assembly.

Because the decision to use fixed point arithmetic was made at the outset, the Ada OFP execution time, which had an average expansion factor of almost 4 to 1 over assembly code, did not become a problem either.

- Rolm/Hawk/32 - In terms of available memory and throughput, the Rolm Hawk/32 was more than adequate for the IFFC OFP. The IFFC algorithm (in Ada) executes twice as fast, on the Hawk, as the original assembly language version did on it's 16 bit computer. The Ada IFFC OFP requires about eleven times the memory of the original assembly language OFP however. With two MBytes of memory, this was not a problem, even with the Hawk's operating system added in. A real bottleneck was, however, encountered when doing I/O. The scaling, conversions, and storing of data required to interface floating point Ada to the fixed point 1553B Hawk MUX I/O interface required as much execution time as the entire IFFC OFP. After running into timing problems initially, more efficient scaling/conversion routines were written. This reduced the total execution time below the allowed 50 msec. While acceptable for ABICS III, a better solution is being researched for future, follow on projects.

Ada Compiler Limitations - Extensive post compilation processing was required for Ada code targeted to the DFCC. This was also the case, to a certain extent, with the Ada software in the INA computer. Special code, "hooks", had to be written in order to run code in the INA and DFCC hardware. The Rolm Hawk/32's high level operating system freed the Ada programmers from these sorts of hardware related concerns and the object code generated by the compiler could be run under the operating system without any special processing.

- DFCC - One problem encountered with the version of the compiler used with the DFCC, was its inability to handle multidimensional arrays and fixed point arithmetic. Although RM code was rewritten to accommodate these limitations the DFCC with its compiler (using software floating point arithmetic) did not have nearly enough throughput to do RM in the time allowed.

- INA - Because of strict program speed and size requirements due to available memory size and CPU speed, it was necessary to examine the code generated by the compiler. Moreover, early compiler errors lowered compiler credibility and forced frequent examinations of compiler output.

Many of the problems encountered with the compiler were with the implementation of fixed point arithmetic. Some of the errors encountered were:

- The compiler would not allow a fixed point delta (precision) of less than 2**(-31).

- Output scaling of the product was incorrect. The compiler produced the incorrect answer for fixed point multiplications.

- The lack of the capability to implement data more than 32 bits long forced consideration of alternative data representations, including integers. That is, a fixed point type with a large range and a small delta (precision) could not be declared.

- A product could not be shifted to the right by more than 30 bits to rescale and divide by 2**(32) to save the upper half.

- The compiler rescaled to the right using a long divide, which is the slowest instruction on the Z8002.

- The compiler unnecessarily rescaled the multiplier to match the scaling of the multiplicand before a multiplication operation.

A significant amount of time was spent identifying compiler problems and developing solutions to circumvent these limitations.

An analysis of compiler speed comparing the time that it took to assemble the original navigation software with the time required to compile the equivalent Ada software was conducted. This analysis indicated that the Ada compiler, running on a Zilog system 8000, took as much as four times as long to compile Ada modules. The Alsys Ada compiler, running on an IBM PC-AT, was about twice as fast.

- Rolm Hawk/32 - The Data General Compiler is a fully validated Ada compiler. Although one minor bug in the compiler was found during testing, there were essentially none of the limitations encountered with the other Ada compilers used in ABICS III.

To determine the effect of compilation speed on overall software productivity, the IFFC OFP was compiled on two different systems. On the Data General, compilation took 95.45 minutes with only one user on the system. By comparison, with 12 users, a VAX 11/785 took only 46.89 minutes.

Support Equipment Limitations - One requirement of embedded computer systems, on a research project like ABICS III, is that they are supportable in the field. The ability to make changes to an OFP, to adjust constants or restructure 1553 mux I/O messages for example, is absolutely necessary.

- DFCC - Portable test support equipment was not available at EAFB. This hindered expedient testing and trouble-shooting of the flight control system. Portable support equipment with the following capabilities would have been advantageous for both the laboratory and the field:

- Ability to load/dump Non-Volatile Memory (NVM) data from/to data files. This would provide an expedient method to update fault sequences and/or filter coefficients between flights or to re-establish scrambled NVM.

- Ability to monitor and record data real-time during ground testing and/or trouble-shooting. This includes multi-channel programmable Digital to Analog Computers (DACs) to provide analog inputs to chart recorders and digital signal analyzers.

- Ability to load or dump the OFP (or portions of it) to aid software patching.

- INA's - The existing support equipment for the IISA INAs provided satisfactory support for both the laboratory effort at MCAIR and field support at EAFB. The field support was complemented with vendor support facilities in close proximity to EAFB. Improvements could have been made to expedite the laboratory development/testing by providing support equipment and incentives to allow software modifications (patching) at MCAIR.

- ROLM HAWK/32 - The lack of a real time monitoring capability in the Hawk was a major hindrance both in the lab and in the field. A capability to do debugging was also not available and special ground test software had to be written. Our Rolm Hawk system was designed to boot off of a tape drive unit. Tape booting our OFP took over two minutes in the ground test mode and made ground OFP testing much less productive than it might otherwise have been.

An EPROM based OFP, built into the Hawk would have saved much time and effort. The tape could be retained as a method of loading a new test OFP. This EPROM might also include the Hawk's real time operating system. This approach would allow file transfers from a PC, so that different OFP's could be stored on a PC format disk and thus eliminate the more expensive and less reliable tape drive.

CONCLUSIONS

The ABICS III test program can be considered a complete success, with primary objectives of demonstrating the MFCRS/IISA concept as a viable system for future advanced aircraft and expanding Ada programming in flight mission-critical software totally fulfilled. Five specific flight test program objectives were successfully accomplished.

(1) It was verified that ABICS III flying qualities utilizing IISA sensors were comparable to those of ABICS III F-15 when standard CAS feedback sensors were utilized and did not degrade the stability and control of the aircraft.

ABICS III airworthiness and flying qualities were tested by utilizing a build-up approach, flown first with standard CAS feedback sensors and then with IISA sensors to allow the pilot to obtain a direct comparison.

A complete IISA flight control sensor buildup was completed up to Mach 1.2 and 30,000 feet. Entire flights were completed with the IISA sensors engaged with few noticeable differences in flight control characteristics and handling qualities except an initial roll ratcheting problem that was resolved. No switching transients were felt by the pilot when switching between standard flight control sensors and IISA sensors.

Switching was tested by the pilot at loadings of more than 6g where static fuselage bending is very high for forward nose barrel mounted flight control sensors.

(2) It was verified that the ABICS III control system was not susceptible to false alarms as a result of changing flight conditions or sensor configurations. Changing flight conditions or IISA sensor triad combinations did not cause IISA disengagements or flight control transients.

(3) Redundancy Management (RM) operation was also demonstrated. RM was able to identify simulated failed IISA sensors and remove them from the system. When flight testing simulated hardover IISA sensor failures, the pilots did not experience any objectionable switching or reconfiguration transients. During simulated ramp failures of IISA sensors, where a specific sensor's performance would gradually decrease, the pilots were able to distinguish switching transients evident as slight bumps or slight aircraft movements during ramp type failure RM testing. The flight control transients were never

objectionable or uncontrollable during RM testing. RM operation was tested on a total of five missions and was performed during benign straight and level flight, as well as maneuvering flight, to demonstrate effectiveness. All sensor faults were programmed prior to the flight and inserted by the system operator during the mission.

(4) ABICS III IISA sensor navigation performance was evaluated on every test mission. Initial position data and final position and velocity data were gathered for each of the two separate INAs after each mission. The INAs proved to be excellent navigators operating within their performance specifications. A failure in the auxiliary power supply caused the loss of INA No. 2 on three flights, but the problem was resolved during subsequent test missions. Fly-over marks were obtained over known geographical points at low level whenever possible to evaluate navigation performance, and a ground Schuler period was performed on external aircraft power after nearly every one of the first 16 test missions.

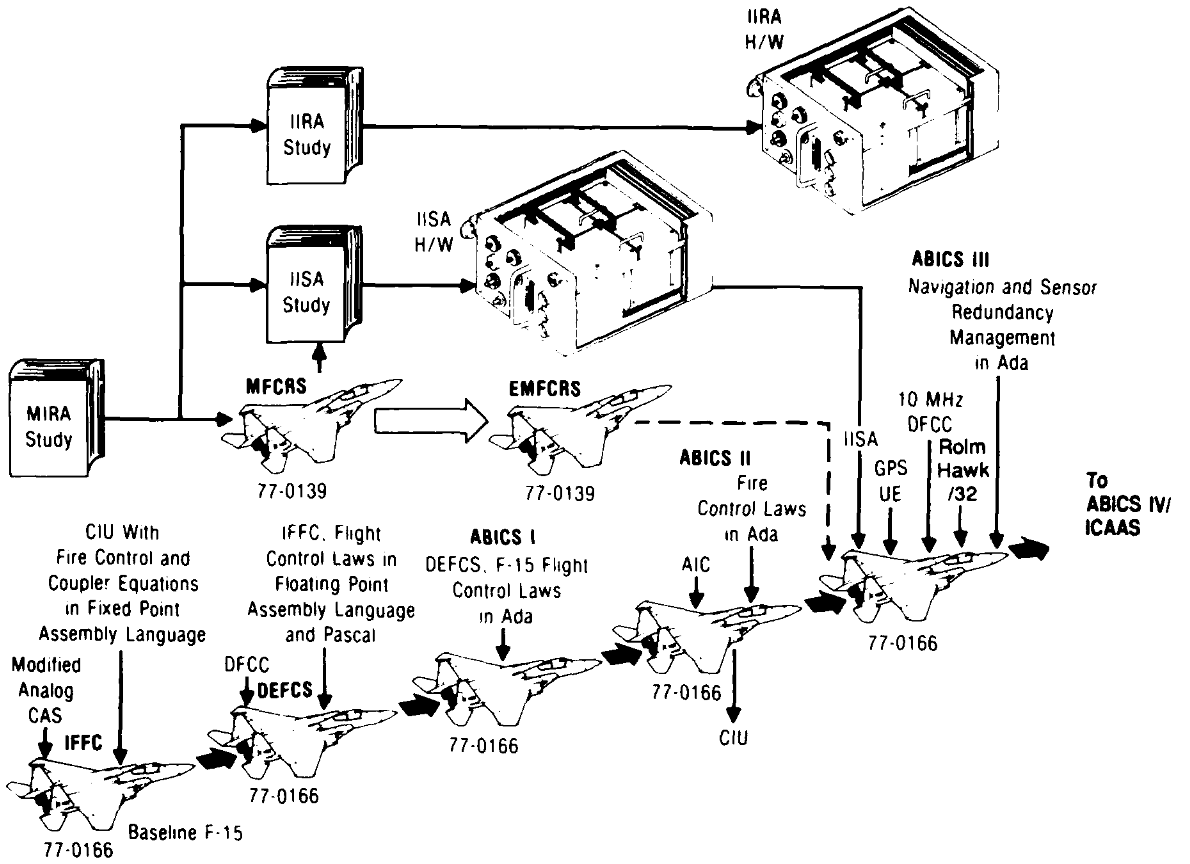
(5) All goals for demonstrating IFFC Ada software were accomplished. IFFC testing was also accomplished using a build-up approach. Manual tracking IFFC Air-to-Air Gunnery (AAG) was first tested using OnBoard Simulation (OBS), where the pilot inputted the steering command against a simulated target. IFFC coupled AAG tracking worked well also against simulated targets once gain settings were optimized. The pilot had no trouble getting simulated hits on 3g reversing and 3g and 5g targets of low, medium, and high aspect angles. Nine AAG OBS encounters were tested using manual and IFFC coupled tracking. IFFC AAG OBS was evaluated using both standard F-15 and IISA flight control reference sensors. No noticeable difference in IFFC AAG performance was noted between standard and IISA sensors.

IFFC AAG was also tested against an actual co-altitude F-4 target. All nine encounters that were tested against OBS targets were evaluated versus the F-4 utilizing manual as well as coupled tracking of standard F-15 and IISA sensors.

IFFC bombing (BMG) testing was very successful. IFFC BMG OBS was first tested with a simulated target at an altitude of 13K ft using manual and coupled IFFC and standard and IISA flight control reference sensors to verify operation. Eleven bombing encounters were performed. The aircrew expressed that IFFC coupled BMG seemed smoother with IISA flight control sensors than with standard F-15 sensors.

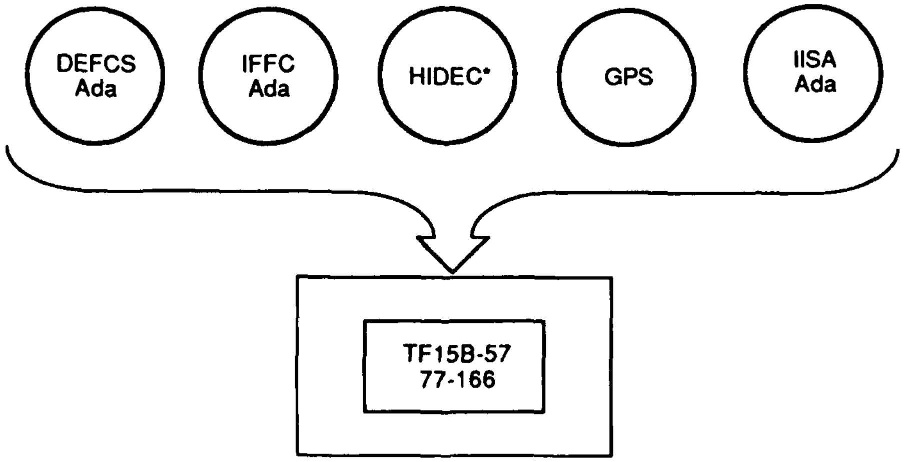
IFFC tactical bombing was performed on range with BDU-33 practice bombs against the PB-10 range target. All IFFC BMG deliveries were performed with 12K ft offset range from the INS designated target. Curvilinear level, curvilinear diving, and curvilinear climbing deliveries were attempted with manual and coupled IFFC. No bomb releases were obtained on four attempts using manual IFFC because of a failure to obtain the correct bombing solution. Seven bombs were dropped using IFFC coupled with standard F-15 or IISA sensors. Performance was about equal to what was seen during the IFFC program. No performance difference between standard and IISA sensors was noted.

In summary, the ABICS III program was 100% successful in meeting all of its goals and has shown by flight test the viability of using multifunction inertial sensors and the Ada HOL for embedded flight critical real time aircraft systems.



GP53-0874-5

Figure 1. ABICS III Program History



*Configuration studies only

GP83-0339-21-D

Figure 2. ABICS III Aircraft Configuration

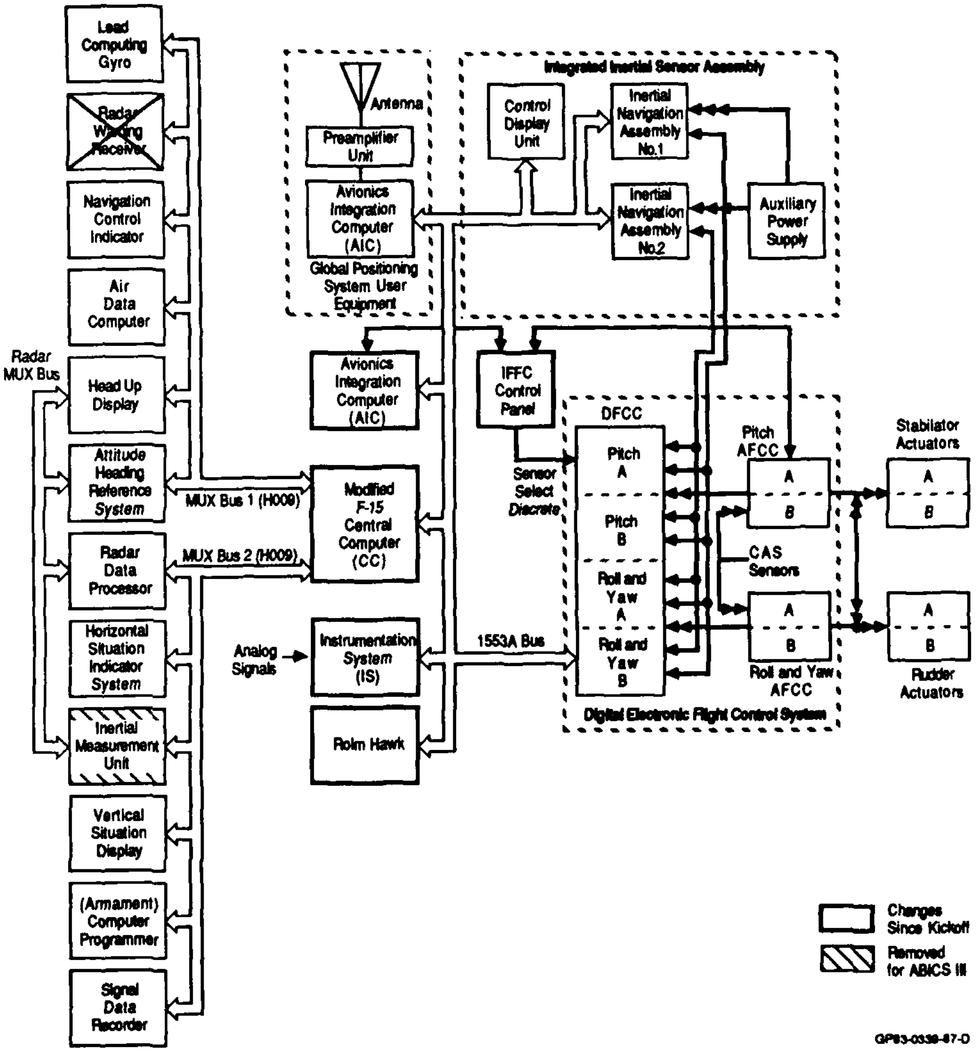
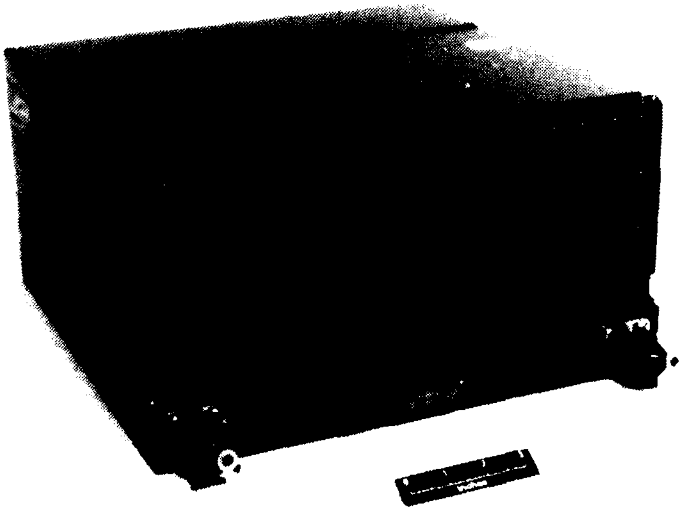
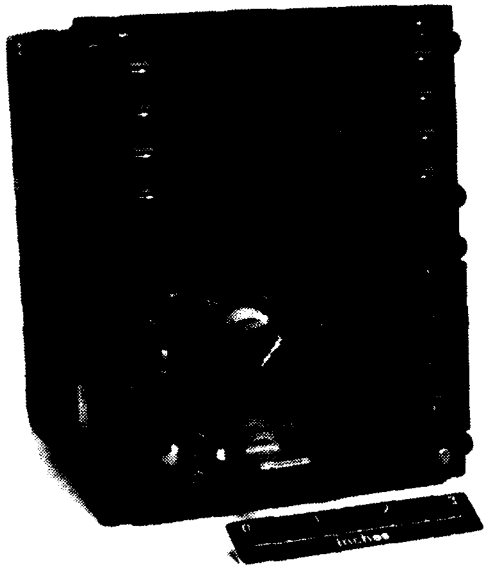

Lead Computing Gyro
Radar Warning Receiver
Navigation Control Indicator
Air Data Computer
Radar MUX Bus
Head Up Display
Attitude Heading Reference System
Radar Data Processor
Horizontal Situation Indicator System
Inertial Measurement Unit
Vertical Situation Display
(Armament) Computer Programmer
Signal Data Recorder
MUX Bus 1 (H009)
MUX Bus 2 (H009)
Antenna
Preamplifier Unit
Avionics Integration Computer (AIC)
Global Positioning System User Equipment
Integrated Inertial Sensor Assembly
Control Display Unit
Inertial Navigation Assembly No.1
Inertial Navigation Assembly No.2
Auxiliary Power Supply
Avionics Integration Computer (AIC)
IFFC Control Panel
Modified F-15 Central Computer (CC)
Analog Signals
Instrumentation System (IS)
1553A Bus
Rolm Hawk
Sensor Select Discrete
DFCC
Pitch A
Pitch B
Roll and Yaw A
Roll and Yaw B
Pitch AFCC
A
B
CAS Sensors
A
B
Roll and Yaw AFCC
Digital Electronic Flight Control System
Stabilator Actuators
A
B
A
B
Rudder Actuators
Changes Since Kickoff
Removed for ABICS III
GP83-0339-47-D

Figure 3. ABICS III System Architecture



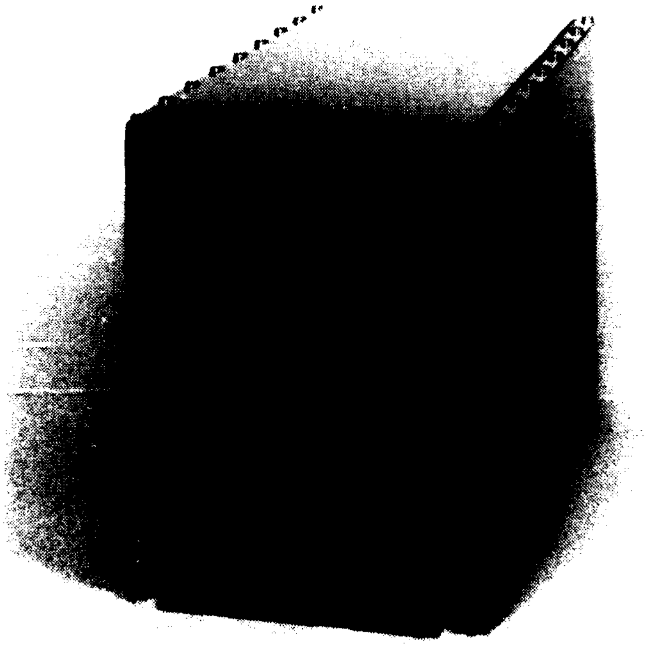
GP83-0339-47

Figure 4. Inertial Navigation Assembly



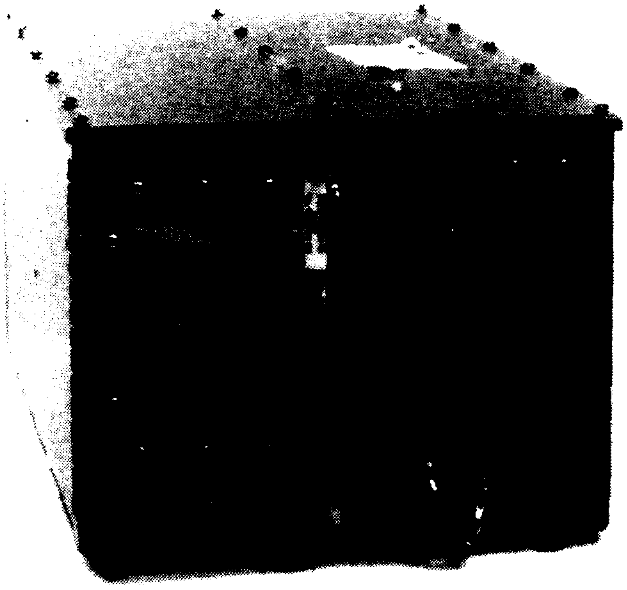
GP83-0339-48

Figure 5. Control Display Unit



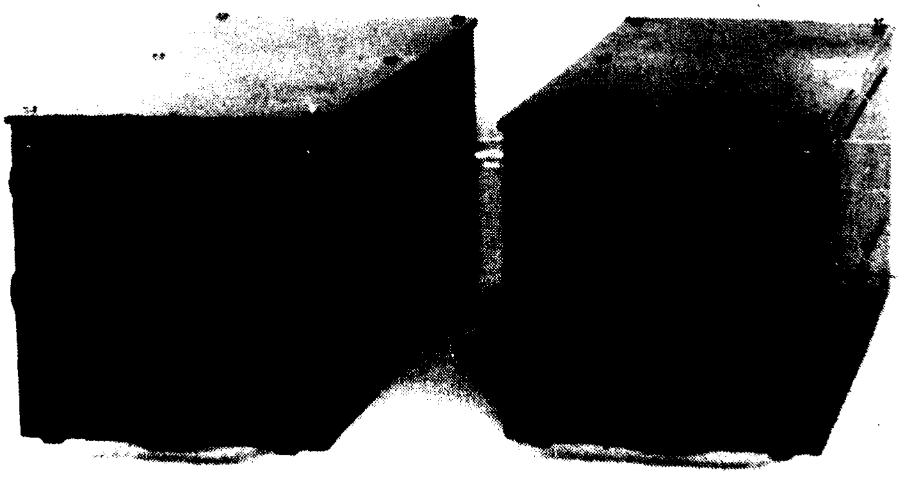
GP83-0339-78

Figure 6. Auxiliary Power Supply



GP83-0339-71

Figure 7. Digital Flight Control Computer



GP83-0339-72

Figure 8. Analog Flight Control Computers (Modified)

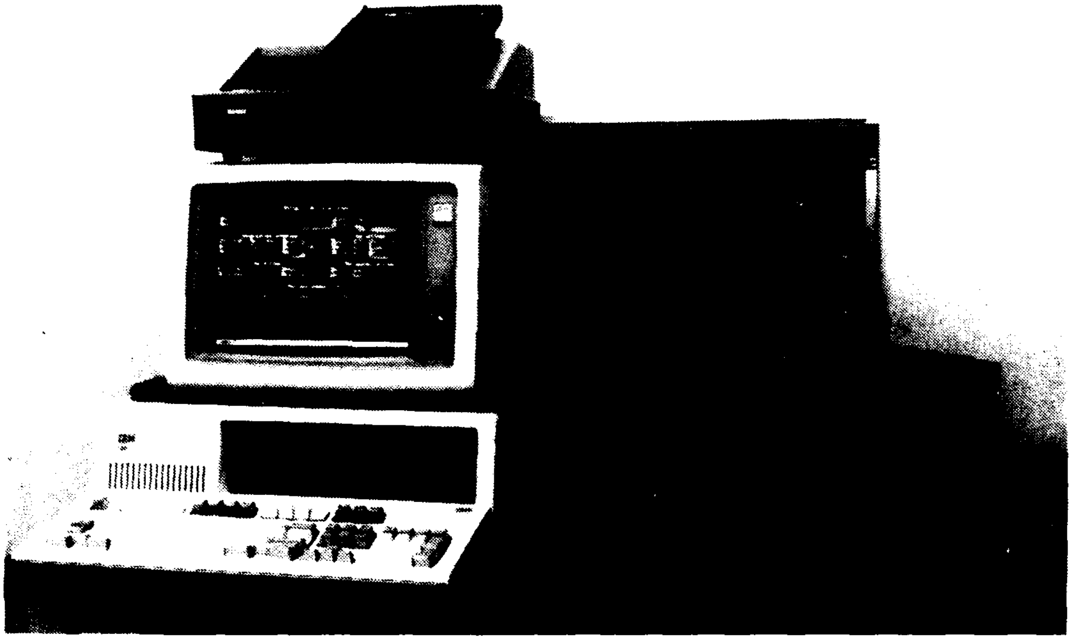

GP83-0339-50

Figure 9. Enhanced Avionics Integration Computer

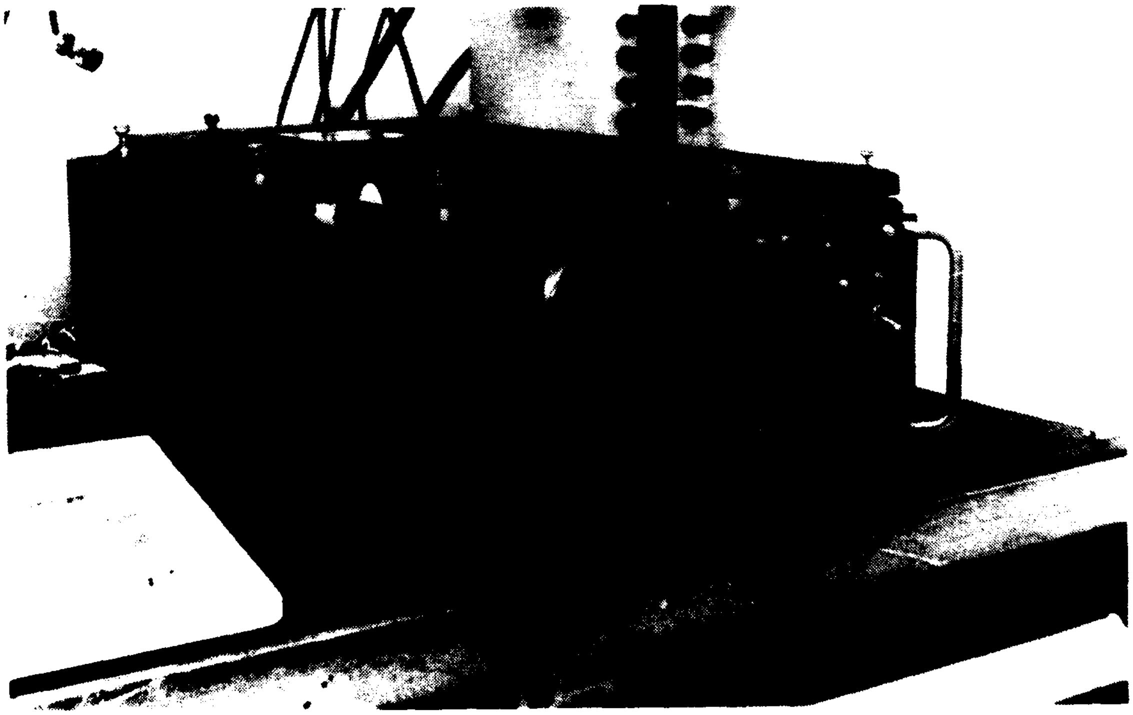
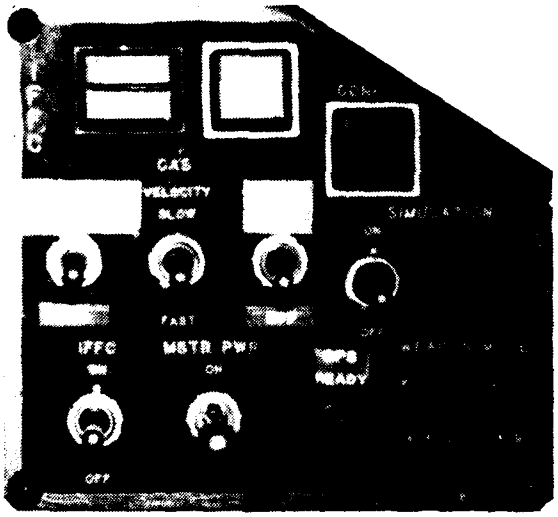

GP83-0339-73

Figure 10. Rolm Hawk/32 Computer

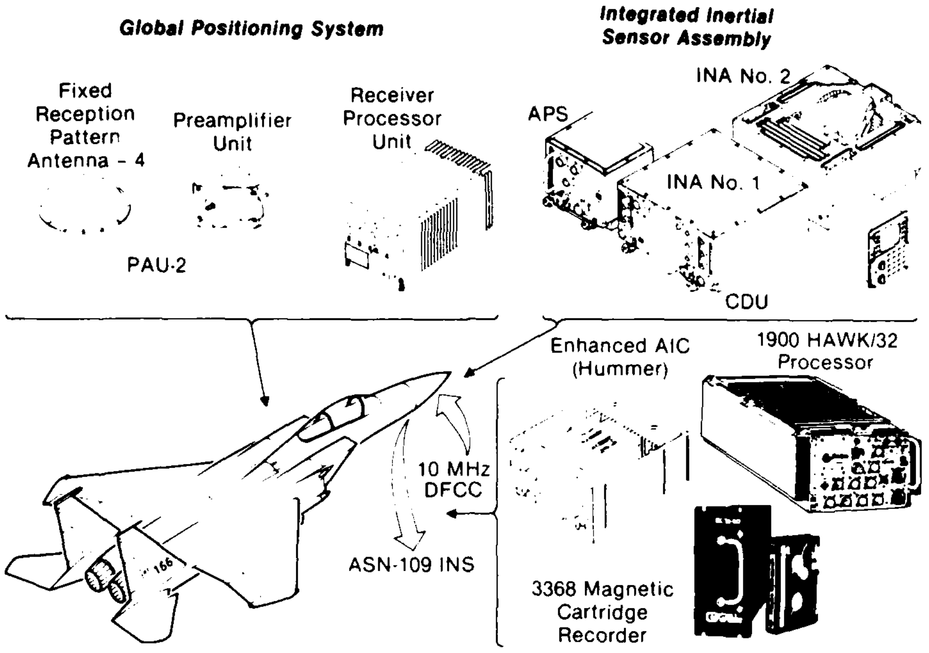
GP83-0339-79

Figure 11. Genesco Magnetic Tape Transport Assembly



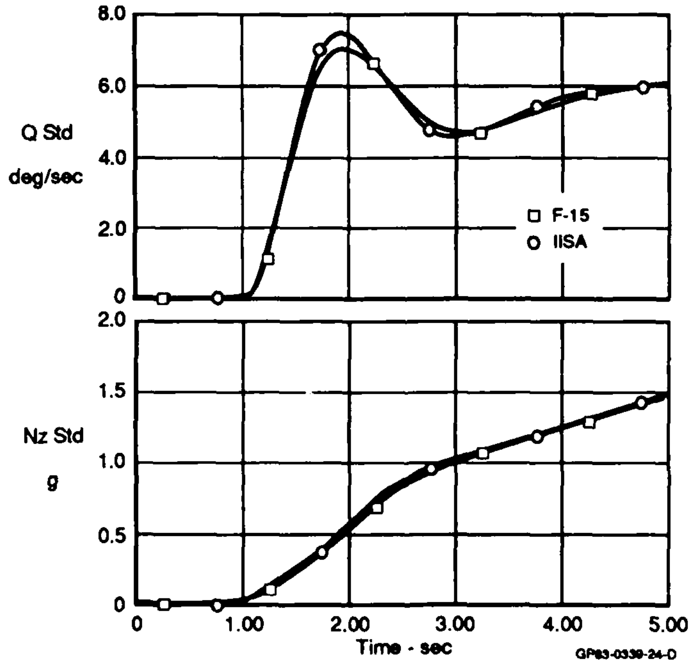
GP83-0339-107

Figure 12. IFFC Control Panel (Modified)



GP83-0339-14

Figure 13. ABICS III Hardware Modifications



GP83-0339-24-D

Figure 14. Response to 10 lb Stick Mach 0.80 30,000 ft

Parameter	INA PAN 03	INA PAN 04	Specification
CEP (NM/hr)	0.66	0.53	1
Velocity (ft/sec RMS)			
$V_{N\text{-}S}$	2.90	1.78	3
$V_{E\text{-}W}$	1.69	2.12	3

GP83-0339-147-T

Figure 15. IISA Performance by INA Serial Number

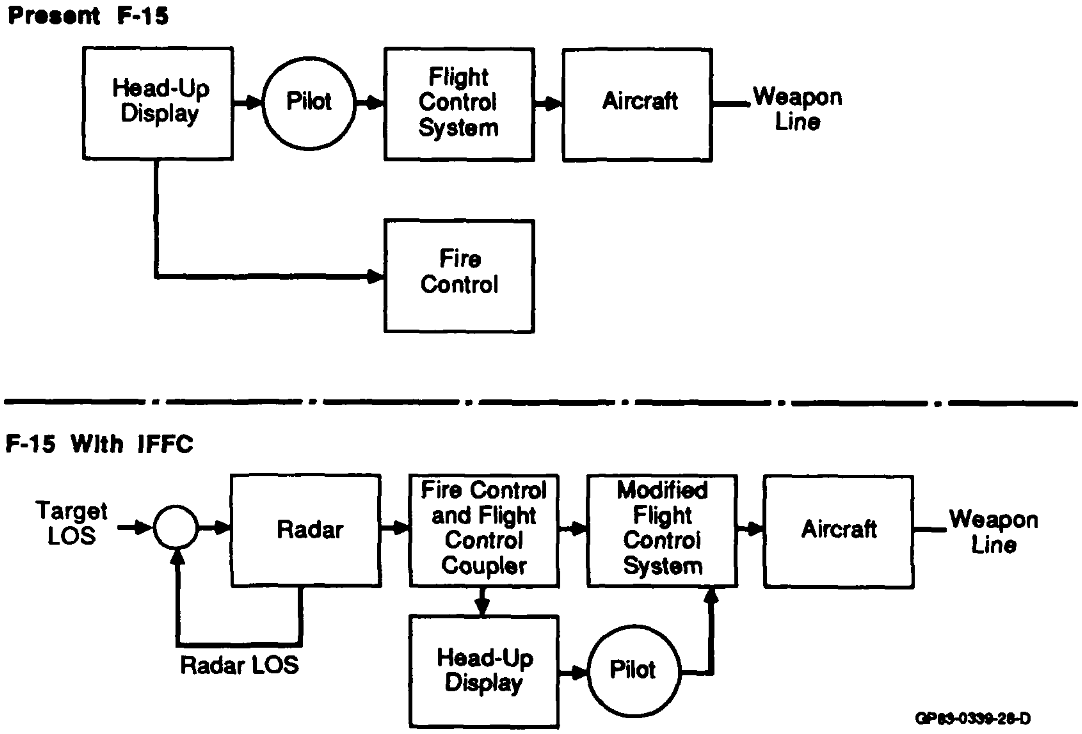


Figure 16. Block Diagram - Present F-15 and F-15 With IFFC

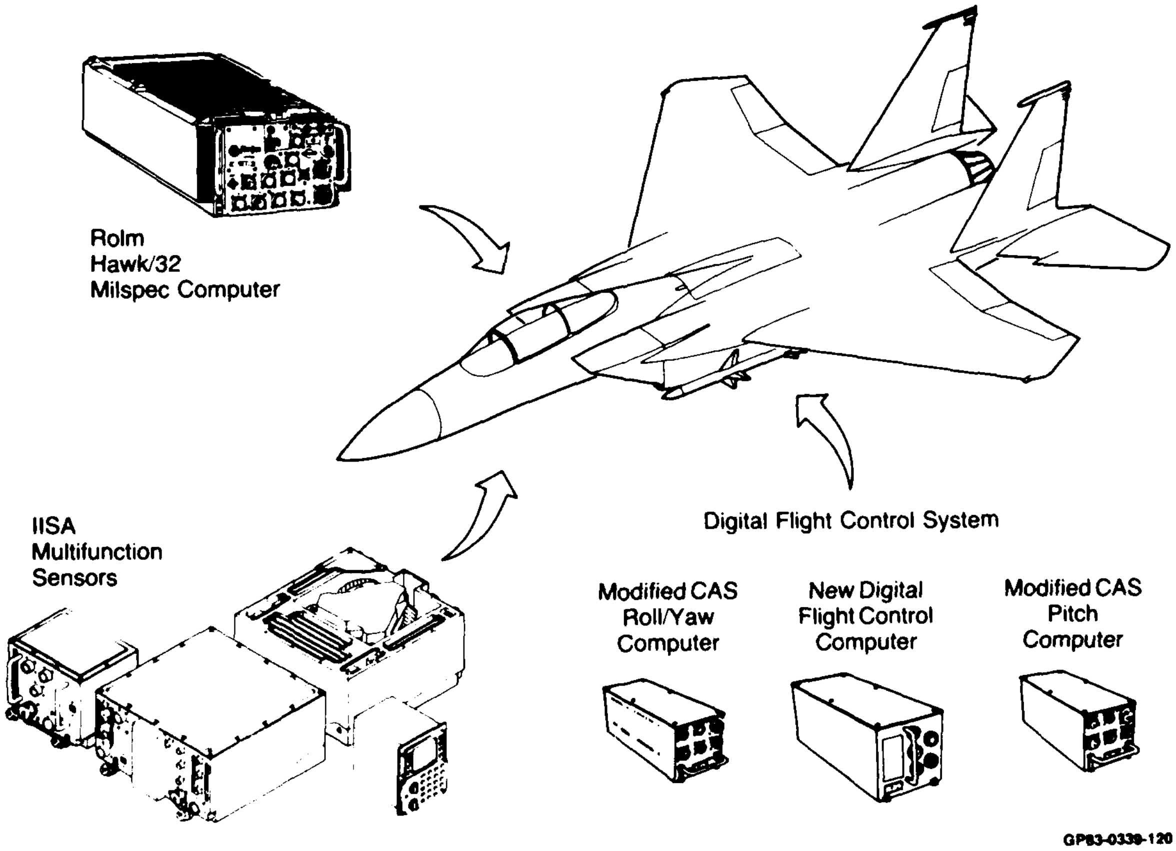


Figure 17. ABICS III Hardware

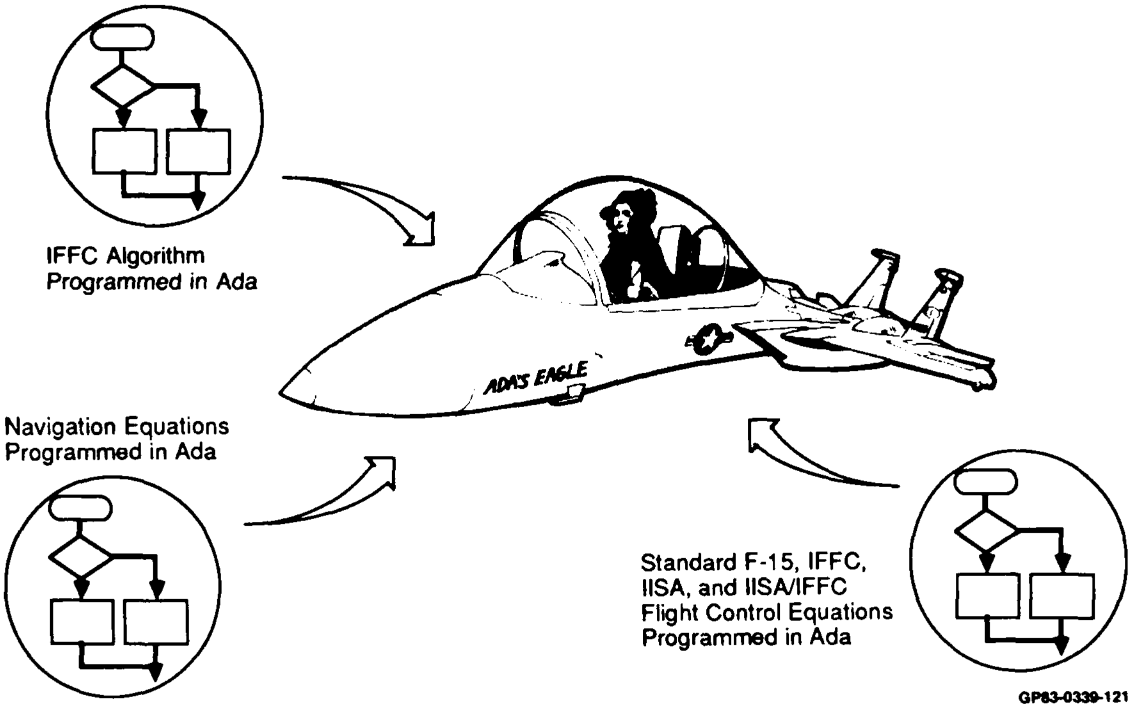


Figure 18. ABICS III Ada Software

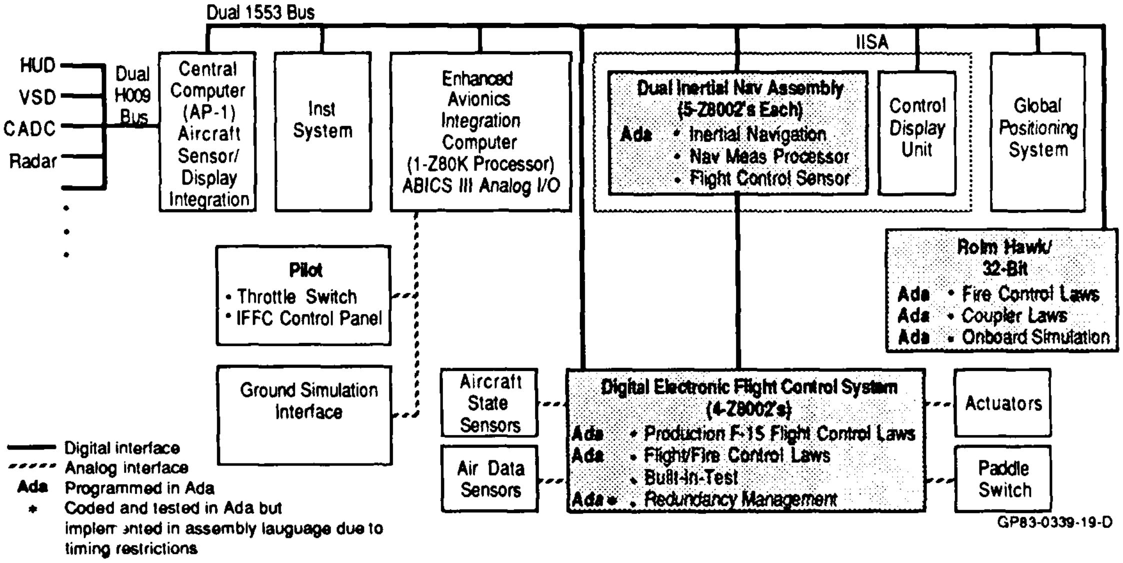

Dual 1553 Bus
HUD
VSD
CADC
Radar
Dual H009 Bus
Central Computer (AP-1) Aircraft Sensor/ Display Integration
Inst System
Enhanced Avionics Integration Computer (1-Z80K Processor) ABICS III Analog I/O
IISA
Dual Inertial Nav Assembly (5-Z8002's Each)
Ada • Inertial Navigation
• Nav Meas Processor
• Flight Control Sensor
Control Display Unit
Global Positioning System
Rolm Hawk/ 32-Bit
Ada • Fire Control Laws
Ada • Coupler Laws
Ada • Onboard Simulation
Pilot
• Throttle Switch
• IFFC Control Panel
Ground Simulation Interface
Aircraft State Sensors
Air Data Sensors
Digital Electronic Flight Control System (4-Z8002's)
Ada • Production F-15 Flight Control Laws
Ada • Flight/Fire Control Laws
• Built-In-Test
Ada * • Redundancy Management
Actuators
Paddle Switch
Digital interface
Analog interface
Ada Programmed in Ada
* Coded and tested in Ada but implem ented in assembly lauguage due to timing restrictions
GP83-0339-19-D

Figure 19. ABICS III Avionic Configuration

	Software Measures		Ada (percent)
	Timing Expansion	Memory Expansion	
Flight Control	1.1	1.5	100
Navigation	4.0	3.0	60
Redundancy Management	12.0	2.0	None
32-Bit IFFC	0.8	11.0	100

Note: Fast 16-bit and 32-bit architectures overcome limitations

GP93-0169-1-T

Figure 20. ABICS Software Measures

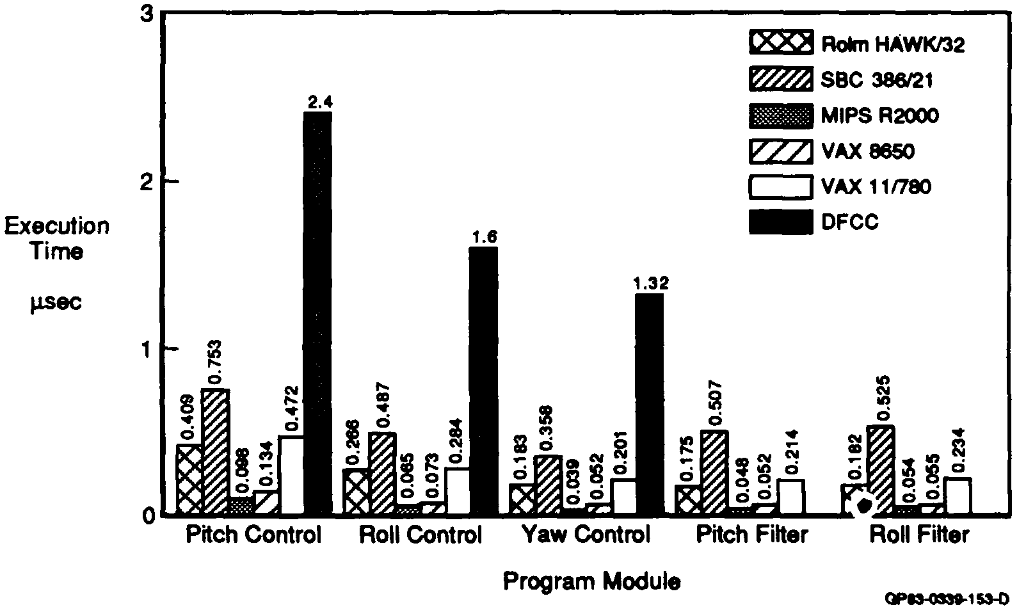


Figure 21. Flight Control Ada Execution Time Comparison

RECONNAISSANCE D'AMERS INTELLIGENTE DANS UNE IMAGE SATELLITE

***Olivier* REICHERT , *Dominique* BERTON**

SAGEM
Division Missiles/Espace
Centre d'études d'Eragny
B.P. 51 95612 CERGY-PONTOISE Cedex
Tel : (1) 34 64 92 00

Résumé: la visée d'amers connus sur des images terrestres permet en particulier de recaler un filtre de navigation embarqué sur satellite. Cet article présente une méthode robuste de reconnaissance d'amers utilisant des techniques de corrélation d'images supervisées par système expert temps réél. Le principe consiste à piloter une bibliothèque d'algorithmes: le système choisit dynamiquement les algorithmes optimaux en fonction des caractéristiques de l'amer, des perturbations éventuelles sur l'image de recherche et des contraintes de temps de réponse, puis interprète leurs résultats par analyse multi-critère. L'enchaînement de plusieurs algorithmes et la fusion de leurs résultats permet de renforcer la confiance sur la décision de recalage. Les performances sur images SPOT multispectrales démontrent la supériorité de cette approche sur des techniques plus classiques de reconnaissance d'amers, en terme de robustesse et d'autonomie.

Mots-clés: reconnaissance d'amers, corrélation, images satellite, système expert temps réel, supervision d'algorithmes, fusion d'information, recalage de navigation

Abstract: the determination of the position of a known landmark within a ground image allows, among other things, to update an on-board satellite navigation filter. In the application described in this paper, the landmark recognition process uses image correlation techniques and is supervised by a real-time expert system. The system controls a library of algorithms by dynamically selecting the most appropriate algorithms considering the features of the landmark, the likely perturbations in the search area and the time constraints, then by interpreting the results through a multicriteria analysis. Several processings are successively triggered and the fusion of their results provide a better confidence on the final update decision. The performances on SPOT multispectral images show the superiority of this approach with respect to more classical recognition techniques, especially in terms of robustness and autonomy.

Key-words: landmark recognition, correlation, satellite images, real-time expert system, expert çontrol, data fusion, navigation updating

Les travaux décrits ici ont été réalisés dans le cadre du marché DRET 87/464 en collaboration avec le laboratoire de l'IRISA (INRIA Rennes) et la société EURISTIC Systèmes (Clamart)

INTRODUCTION

Dans un proche avenir, les systèmes de navigation inertielle seront dotés d'équipements auxiliaires de vision. Cette hybridation permettra aux centrales inertielles des véhicules (missiles, satellites d'observation, robots terrestres autonomes, avions) de disposer de mesures externes de recalage [1]. Ces observations, issues de visées sur des points de référence terrestres, contribueront à recaler un filtre de navigation embarqué ou à contrôler le système de référence d'attitude. Cette technique, dite recalage par visée d'amers, a déjà fait l'objet d'applications militaires (navigation de missiles de croisière, guidage terminal des missiles Pershing). Cet article décrit la maquette logicielle d'un système de reconnaissance automatique d'amers sur images satellite qui présente les caractéristiques d'un système embarqué: fiabilité élevée, autonomie totale.

Les amers, dont les caractéristiques sont connues a priori, sont détectés sur des images 2D par des techniques de corrélation d'images, connues pour leur robustesse et leur simplicité. Cependant, le défaut majeur de nombreux systèmes de vision est la limitation du champ d'applications des algorithmes de traitement d'images. Or un spécialiste du traitement d'images est à même, à partir de l'observation des images de référence et de recherche, d'identifier les algorithmes optimaux et d'associer aux résultats une vraisemblance et une erreur typiques. Une structure de système expert se prête donc bien à la supervision des différentes étapes du processus de recalage. Dans notre application, le choix du générateur de système expert s'est porté sur un outil temps réel, CHRONOS (réalisé conjointement par EURISTIC Systèmes et SAGEM).

La connaissance porte sur les propriétés intrinsèques et le comportement des algorithmes disponibles: la nature de cette expertise traduit une approche différente de celles généralement utilisées en interprétation intelligente d'images. L'objectif consiste ici à construire un système de recalage autonome, donc à effectuer une reconnaissance robuste. La bibliothèque d'amers, constituée au sol, comprend donc une représentation matricielle de l'amer, mais la base de faits peut s'enrichir de connaissances symboliques sur l'objet et son contexte. Les performances d'un tel système se mesurent alors en taux de détection lorsque l'amer figure explicitement sur l'image de recherche, et en taux d'échec dans le cas contraire. Si une position est validée, le système fournit également une mesure de confiance dans le recalage.

L'application choisie concerne le recalage d'un satellite d'observation navigant en haute altitude. Le traitement d'images s'effectue sur des images SPOT multispectrales, de résolution 20 mètres. Le principe du recalage et les algorithmes de base sont décrits dans la partie I. La méthodologie nécessaire pour obtenir un fonctionnement robuste du système de reconnaissance d'amers est exposée en partie II. Le processus de raisonnement, illustré par des scénarios de recalage, et les performances du système expert constituent la troisième partie de l'article.

I. RECALAGE PAR CORRELATION D'IMAGES

1. Principe du recalage par corrélation d'amers

On appelle amer une image de référence acquise dans des conditions connues et représentant un objet terrestre de caractéristiques (structure, radiométrie) uniques dans son environnement. L'amer effectivement stocké est constitué d'une matrice de $N_A x N_A$ pixels et de données structurelles, texturelles, statistiques ou symboliques sur l'objet et son contexte.

Le but du recalage par corrélation d'amers consiste à retrouver la position (X_A, Y_A) de l'origine d'un amer A dans une image de recherche de taille $N_B x N_B$ (avec $N_B > N_A$). On procède par comparaison de A avec toutes les sous-images D_{XY} de taille $N_A x N_A$ incluses dans B. En chaque position (X, Y), on calcule donc un critère de similitude, que nous appelerons coefficient de corrélation par la suite. L'ensemble de ces coefficients donne une matrice de corrélation de taille $(N_B-N_A+1)^2$, souvent représentée sous forme de surface.

Le coefficient de corrélation est théoriquement maximum au point de coïncidence des deux images ($A=D_{X_A Y_A}$). Cependant, si les images sont bruitées, la position de recalage ne correspond pas nécessairement au maximum de la surface de corrélation mais par exemple à un pic secondaire (fig. 1). En outre, si l'amer ne figure pas dans l'image de recherche, l'interprétation de la surface de corrélation devient complexe (fig. 2).

Il est donc nécessaire de construire un système d'analyse des surfaces de corrélation susceptible de fournir un diagnostic fiable sur l'existence éventuelle et la position de l'amer dans l'image de recherche. D'autre part, l'objectif de robustesse ne sera atteint que si l'on adapte le choix des algorithmes au type de l'amer: on n'utilisera pas les mêmes traitements d'images pour détecter un croisement de routes ou pour identifier un lac. Pour cela, il est nécessaire d'élargir le terme de corrélation classiquement utilisé en traitement du signal.

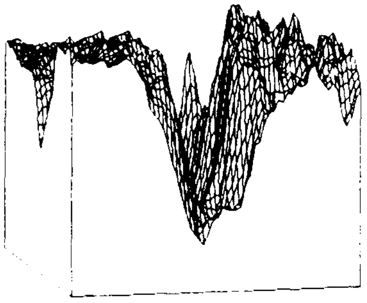


fig.1

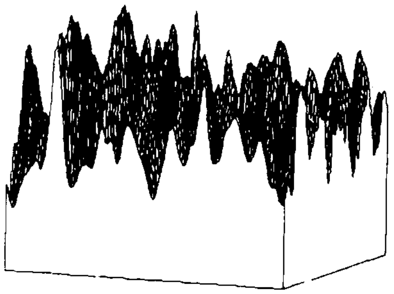


fig.2

2. Extension de la notion de corrélation

Dans cette étude, le terme coefficient de corrélation désigne par extension tout indice de ressemblance (p.ex. les moindres carrés) entre deux matrices de même taille, cet indice étant maximal en cas d'identité. Le principe de corrélation consiste alors à balayer exhaustivement les différentes positions de l'amer dans la zone de recherche à l'aide d'un indice optimal, c'est-à-dire adapté à la nature de l'amer et aux conditions de la recherche (couverture nuageuse, temps de recalage alloué, ...).

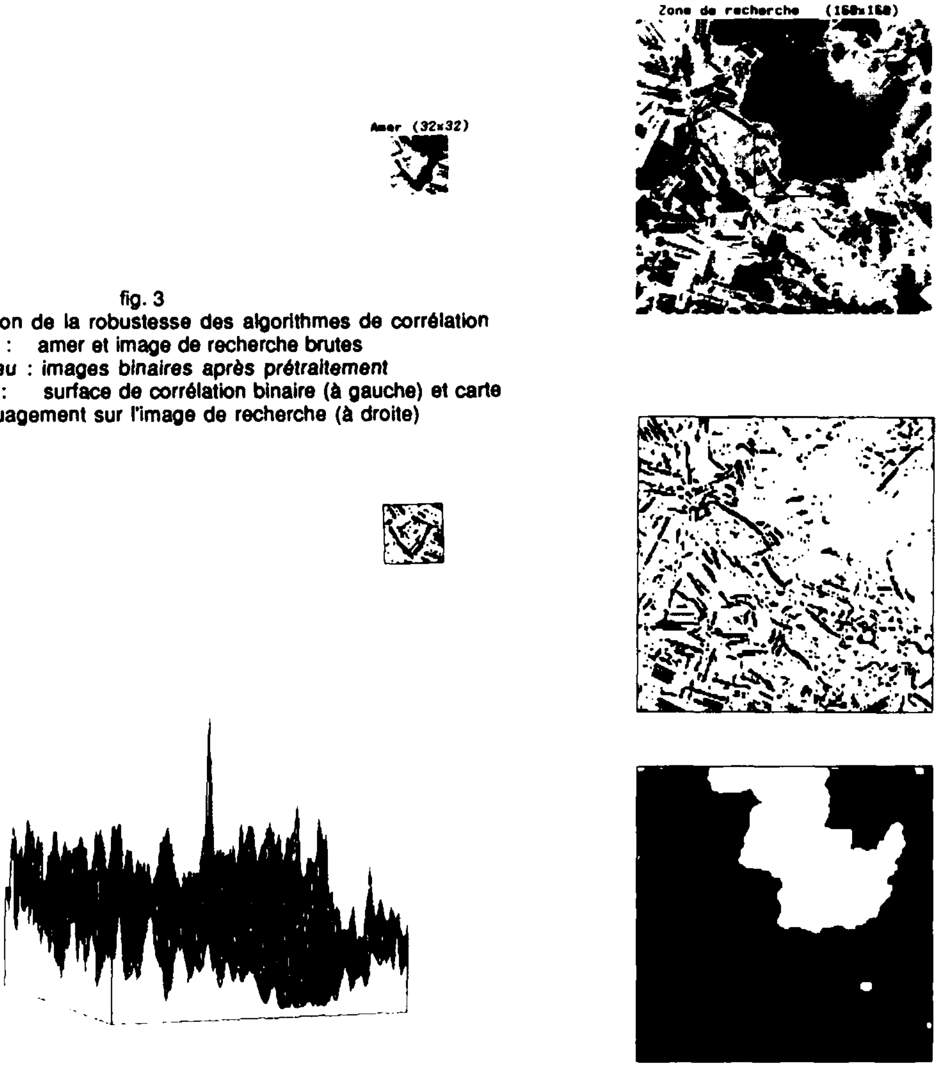


fig. 3
Illustration de la robustesse des algorithmes de corrélation
En haut : amer et image de recherche brutes
Au milieu : images binaires après prétraitement
En bas : surface de corrélation binaire (à gauche) et carte d'ennuagement sur l'image de recherche (à droite)

Cette approche semble coûteuse en temps de calcul, mais elle se prête aisément à un parallélisme massif de type SIMD (Single Instruction Multiple Data). Elle offre surtout des performances très avantageuses en termes de robustesse et d'insensibilité aux perturbations,ainsi que l'illustre la figure 3.

Dans cet exemple, l'objectif consiste à reconnaître un amer (une structure de marais salants) dans une image de recherche où la surface de l'amer est recouverte à près de 50% par une couverture nuageuse opaque (au N.E.). Après un prétraitement de l'amer et de l'image destiné à extraire les structures fines (transformée morphologique "top-hat"), suivi d'un seuillage avec binarisation, l'application d'une corrélation binaire optimale fournit une surface de corrélation qui présente un pic pointu à l'endroit de la position de l'amer. Malgré une mauvaise qualité de segmentation due à des perturbations imprévues sur la scène, le recours à une méthode bas-niveau (corrélation) permet de détecter facilement l'amer, ce qui n'aurait vraisemblablement pas été possible par raisonnement symbolique sur des primitives incomplètes.

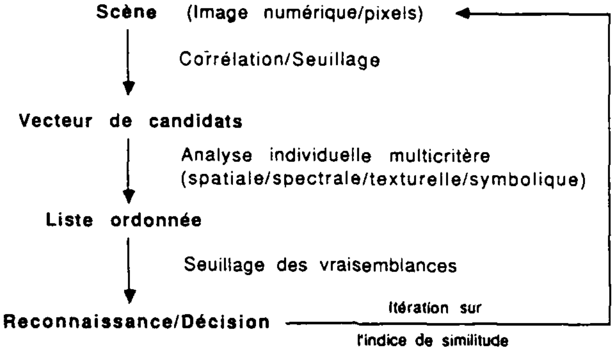

Scène (Image numérique/pixels)
Corrélation/Seuillage
Vecteur de candidats
Analyse individuelle multicritère (spatiale/spectrale/texturelle/symbolique)
Liste ordonnée
Seuillage des vraisemblances
Reconnaissance/Décision
Itération sur
l'indice de similitude

fig. 4
Paradigme de corrélation étendue

Dans cette étude, les performances des algorithmes sont renforcées par une méthodologie de reconnaissance résumée sous forme de paradigme (fig. 4): un premier indice de similitude balaye toutes les positions candidates et sélectionne quelques régions ponctuelles (les pics de corrélation). On peut alors procéder à une analyse individuelle détaillée en chacun de ces endroits, à l'aide de critères de nature très variée : étude locale de la surface de corrélation, comparaison spatiale, spectrale ou statistique avec l'amer, étude texturelle du contexte, comparaison d'attributs symboliques, ...

Le classement hiérarchique des positions candidates selon une fonction d'évaluation relative (du type coefficient de vraisemblance) permet ensuite, soit de choisir une nouvelle fonction de corrélation pour affiner l'hypothèse de recalage, soit de prendre une décision sur le recalage (validation d'une position, abandon de la recherche).

Cette approche diffère de celles généralement utilisées en interprétation d'images: le processus de mise en correspondance de primitives avec un modèle est remplacé par une corrélation d'images 2D, la recherche exhaustive sur toute l'image de recherche facilite l'interprétation des résultats et l'itération sur l'indice de similitude renforce la confiance sur la décision finale.

3. Bibliothèque d'algorithmes

De nombreux algorithmes de corrélation sont disponibles [2]. On peut les regrouper en cinq classes:

- les corrélations linéaires, auxquelles on peut rattacher les corrélations statistiques ([3])
- les corrélations par comptage dont le principe se fonde sur le calcul d'une distance radiométrique moyenne entre images
- les corrélations binaires
- les corrélations dérivées de la transformée de Fourier (corrélation FFT, corrélation de phase avec ou sans préfiltrage)
- les corrélations de formes (corrélation de vecteurs, comparaison de chaînes de Freeman, transformée de Hough, ...)

La plupart de ces algorithmes nécessitent un prétraitement des images, qui consiste à mettre en forme les images avant corrélation (normalisation, filtrage, seuillage) ou à extraire des structures privilégiées (détection de contours, de structures fines, ...) par des outils classiques de traitement d'images.
Parmi la liste ci-dessus, plusieurs algorithmes n'ont pas été retenus, pour leurs faibles performances (corrélation FFT), leur complexité de mise en œuvre ou la limitation de leur champ d'applications (corrélations de formes). Quelques algorithmes, développés au cours de cette étude ou intéressants par leurs propriétés ou leurs performances, sont cités ci-après. Une méthode d'extraction des nuages est également proposée.

a. Corrélations linéaires

La méthode classique pour comparer un signal avec une référence est le calcul du coefficient de corrélation linéaire normalisé (ou intercorrélation) :

$$C_{XY} = \frac{\Sigma_i \Sigma_j \ (a_{ij} - \bar{a}) \ (d_{ij} - \bar{d})}{\sqrt{\Sigma_i \Sigma_j \ (a_{ij} - \bar{a})^2 \ \ \Sigma_i \Sigma_j \ (d_{ij} - \bar{d})^2}}$$

(les coordonnées (X, Y) du coefficient de corrélation correspondent à la position dans B de l'origine de la sous-image D_{XY})

Par ses performances, cet algorithme reste la référence. La normalisation par les écarts-types des images le rend insensible aux différences globales de contraste entre l'amer et l'image. De plus, la valeur du coefficient d'intercorrélation constitue en soi une mesure précise de la confiance sur la coïncidence du pic avec l'amer.

Cependant, plus la dynamique de l'amer ou de l'image est faible (c'est-à-dire un dénominateur tendant vers 0), plus le coefficient C_{XY} est élevé. La corrélation linéaire normalisée est donc un indice de ressemblance sensible aux régions uniformes de l'image (critère de surface).
Différentes variantes simplifiées ont été étudiées, en particulier un coefficient non-centré et non-normalisé, dit coefficient de corrélation linéaire :

$$C'_{XY} = \Sigma_i \Sigma_j \ a'_{ij} d_{ij} \text{ avec } a'_{ij} = a_{ij} - \bar{a}$$

Cet algorithme rapide est sensible aux régions de forte luminance (critère d'intensité), mais il peut être perturbé par une présence de nuages fortement réflectants sur l'image. On peut réduire cette sensibilité radiométrique en normalisant le coefficient par la moyenne de l'image de recherche.

b. Corrélations par comptage

Le principe de calcul d'une distance entre images est à la base de nombreux algorithmes. On peut citer deux familles principales :

* Corrélations par comptage des différences seuillées
Aprés prétraitement des images (démoyennage, normalisation ou modélisation des variations de radiométrie), on calcule la valeur absolue des écarts entre images pixel à pixel :

$$e_{ij} = |a_{ij} - d_{ij}|$$

Ces écarts sont comparés à un seuil S. Le nombre de pixels pour lesquels $e_{ij}<S$, divisé par le nombre total de pixels de A, est conservé comme mesure de la distance entre les deux images.

Malgré quelques difficultés d'implémentation liées au réglage du seuil de comptage, cet algorithme permet d'identifier de nombreux pics (effet de "rateau") parmi lesquels figure très souvent le point de coïncidence.
Une variante de cet algorithme, la corrélation par accumulation des différences avec calibration logarithmique, a été proposée ([6]): elle permet de s'affranchir d'un seuil de comptage, et fournit des performances supérieures à la corrélation par comptage des différences seuillées (fig.5).

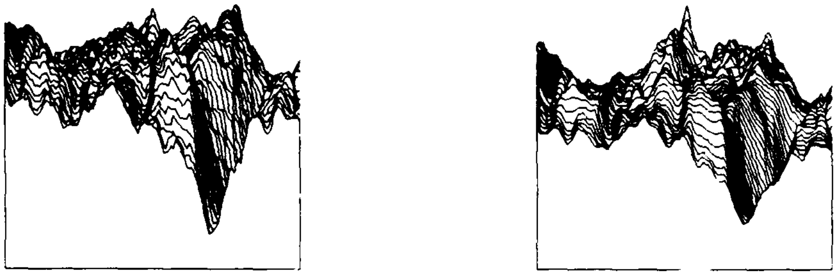


fig.5
Comparaison de performances de corrélations par compage:
à gauche : comptage des différences seuillées
à droite : accumulation des différences (calibration log.)

* Corrélations SSDA (Sequential Similarity Detection Algorithm)
Cette classe d'algorithmes très rapides reprend le principe de calcul de distance entre images utilisé en comptage des différences seuillées [4]. Ici, on somme les résultats des comparaisons pixel à pixel tant que l'on n'atteint pas un seuil. Si les écarts cumulés sont grands dès le début du balayage, on n'aura donc pas à parcourir toute l'image, ce qui limite de façon importante le nombre d'opérations.
Cependant, le seuil est modifié chaque fois que le balayage ne s'arrête que quand l'image est totalement décrite. En raison de cet ajustement, qui accélère la convergence, l'algorithme ne fournit pas une surface de corrélation mais un seul pic maximum. Le calcul de l'indice de qualité du recalage, qui dépend en partie d'une analyse de la surface de la corrélation, est donc imprécis. D'autre part, la position de recalage ne correspond pas nécessairement au maximum de corrélation, d'où un risque de "rater" l'amer.
Un algorithme de corrélation SSDA avec seuillage par hystérésis a donc été proposé ([6]). Il permet d'obtenir plusieurs pics de corrélation ainsi que leur voisinage, sans augmenter fortement le temps de corrélation.

c. Corrélations binaires

Cette famille d'algorithmes présente de nombreux avantages (rapidité, insensibilité aux variations saisonnières de luminance, applicabilité à la plupart des classes d'amers). La représentation binaire des amers permet de combiner et de sélectionner l'information utile.

Le corrélateur binaire utilisé classiquement est un simple ET logique entre les deux images. Ce critère, susceptible d'implémentations rapides [5], n'est pas optimal : il est en particulier très sensible à des zones à forte densité locale de pixels non-nuls sur l'image de recherche. Cette limitation nous a conduit à proposer un opérateur binaire appelé coefficient quadratique [6], dont les performances sont très largement supérieures à celles de la corrélation binaire simple et qui présente l'avantage d'être insensible à une présence nuageuse sur l'image (fig.3).

d. Algorithme d'extraction des nuages

Les nuages constituent le principal handicap d'une méthode de recalage par reconnaissance d'amers sur une image terrestre de type CCD. Il est donc important de disposer d'indicateurs fiables de présence nuageuse (p.ex. critères de radiométrie moyenne locale), mais également d'un algorithme d'extraction capable de fournir une carte précise de l'ennuagement sur l'image de recherche (fig 3).

Les méthodes d'extraction dépendent de la texture des nuages (brume filtrante, couverture opaque). L'algorithme utilisé dans cette étude s'appuie sur un modèle réaliste des nuages à forte énergie sur les images SPOT : une analyse radiométrique montre en effet que la luminance des nuages présente une structure pyramidale, dont la hauteur croît avec la taille du nuage. On peut donc identifier les régions nuageuses par une signature commune aux spectres visible et proche Infra-Rouge: une forte luminance et un gradient élevé. L'algorithme d'extraction combine ces deux caractéristiques:

- seuillage et binarisation de l'image de luminance, pour extraire des composantes connexes très réflectantes
- reconstruction morphologique de cette image binaire, à l'aide d'une image de "marqueurs" qui consiste en un gradient de l'image (p.ex. de type Sobel) également seuillé et binarisé.
- dilatation morphologique et remplissage des trous de l'image binaire résultante

Le seuil sur la luminance est déterminé par prédiction de la radiométrie sur l'image de recherche, à l'aide d'un modèle de transformation des luminances tenant compte de l'incidence solaire, de l'angle de prise de vue et de corrections inter-capteurs.

II. METHODE DE RECALAGE ROBUSTE

1. Choix des algorithmes optimaux

Un système de recalage robuste doit pouvoir s'adapter aux caractéristiques de l'amer (taille, forme, contexte, ...) ainsi qu'aux conditions de prise de vue (heure, date, angle de prise de vue) et aux perturbations affectant l'image de recherche (distorsions géométriques, occlusion nuageuse, modifications locales ou globales de la luminance et du contraste de la scène en fonction des saisons et de l'incidence lumineuse, bruit de quantification, ...). Or, chaque algorithme réagit différemment à chacun de ces paramètres. Par exemple, les algorithmes de comptage des différences seuillées sont sensibles aux écarts d'orientation géographique entre l'amer et l'image (causés par un mouvement de lacet du satellite p.ex.). L'influence des perturbations sur le fonctionnement des prétraitements est également très variable. Par exemple une extraction de réseaux fins (ex. transformée "top-hat") est insensible à une présence nuageuse, contrairement à un filtrage de type gradient de Sobel.

D'autre part, les performances des algorithmes de corrélation sont directement liées à la nature et la qualité des amers choisis. Par exemple les lacs se caractérisent par une surface homogène dans un spectre proche-infra-rouge et par des contours contrastés par rapport à l'environnement. Pour reconnaître cette famille d'amers on cherchera donc à restituer l'intérieur de la surface par une détection optimale de contours (du type filtre de Canny) suivie d'un seuillage avec hystérésis (pour améliorer la connexité) et d'une corrélation binaire optimale.

Le choix des algorithmes optimaux doit donc être guidé par des lois heuristiques générales, qui dépendent à la fois des formes des objets et du comportement (difficilement quantifiable) des traitements disponibles.

2. Analyse optimale des surfaces de corrélation

L'analyse des résultats va s'ordonner autour d'un constat simple: une surface de corrélation comporte une grande richesse d'informations, souvent suffisantes pour conclure à un éventuel recalage. On peut raisonner sur de nombreux critères d'interprétation, qui se regroupent en trois familles:

- critères globaux d'analyse de la surface de corrélation: nombre de pics, Peak-to-Background Ratio [7], Signal-to-Noise Ratio, ...
- critères d'analyse locale autour de chaque pic: hauteur et forme du pic de corrélation (test de plateau, finesse, pente,...)
- critères de similitude entre l'amer et la sous-image de même taille extraite de l'image de recherche à l'endroit du pic: intercorrélation, écart des paramètres statistiques, indices de texture pour vérifier a posteriori l'appartenance de l'amer supposé à son contexte (végétation, littoral, agglomération,...)

La combinaison de ces critères produit un effet de convergence, lorsque l'amer se trouve effectivement dans l'image, ou de divergence dans le cas contraire. L'efficacité de chaque critère est cependant liée à l'algorithme de corrélation employé. La hauteur d'un pic de corrélation linéaire normalisée est par exemple un critère absolu très fiable. Au contraire, dans une corrélation par comptage, la valeur du coefficient maximum n'est pas significative (elle dépend d'un seuil de comptage) alors que la finesse des pics ou le rapport signal-à-bruit constituent de bons critères. Cette expertise est fondamentale, car elle permet de créer un système d'interprétation très fiable, et donc d'envisager l'automatisation du processus de décision pendant un recalage.

Le principe de l'analyse de surface va consister à faire collaborer ces différents critères, afin de mesurer la confiance (coefficient de vraisemblance) accordée au pic de corrélation considéré. Chaque critère émet un avis sur le ou les pics significatifs (extraits de la surface par seuillage adaptatif ou recherche d'extrema locaux). La représentation adoptée pour ces valeurs de décision suit une logique à trois états [8] :

Validation	:	$D_i = +1$
Indécision	:	$D_i = 0$
Rejet	:	$D_i = -1$

parfois étendue à une logique à valeurs continues sur [-1,+1]. Ces valeurs de décision sont ensuite pondérées par l'efficacité relative des critères pour l'algorithme considéré. Leur combinaison fournit alors un coefficient de vraisemblance normalisé, qui représente un indice de qualité de la position étudiée et non une probabilité de recalage.

La souplesse de cette formulation autorise l'intégration de comportements très différents: par exemple certains critères fonctionnent uniquement comme alarmes (ex. un écart important des paramètres statistiques), c'est-à-dire qu'ils ne rentrent jamais dans la zone de validation. De plus, l'étalonnage par coefficients de vraisemblance permet une classification des pics différente de celle fournie par la surface de corrélation. On peut ainsi rejeter un pic principal de faible vraisemblance ou déclarer candidat au recalage un pic secondaire de forte vraisemblance.

3. Fusion multitest

Le raisonnement utilisé dans la procédure de détection d'amers est construit sur un principe de redondance: pour correspondre au point de recalage recherché, une même position, c'est-à-dire un même pic de corrélation, doit avoir été retrouvé par deux algorithmes différents, ou par le même algorithme dans deux bandes spectrales disjointes. La confiance finale sur le recalage dépend alors des vraisemblances issues de chaque test.

La justification de ce principe provient des propriétés intrinsèques des algorithmes. Pour cela, on peut se représenter les corrélations comme étant des projections sur des axes de similitude. La corrélation linéaire normalisée, par exemple, correspond à une mesure de cosinus d'angle entre les vecteurs amer et image. Lorsque ces axes de projection sont orthogonaux, les différents indices de ressemblance ne convergent qu'à l'endroit de l'amer. Cet effet se vérifie expérimentalement: les pics de corrélation obtenus par deux algorithmes différents sont très souvent distincts, sauf si l'amer se trouve effectivement dans l'image.

Dans le système décrit ici, cet effet de synergie impose de combiner plusieurs corrélations avant de conclure à la présence ou non de l'amer dans l'image. Le système choisit par exemple en priorité un algorithme sensible aux surfaces (corrélations linéaires), puis raisonne sur les contours (corrélations binaires). Après chaque corrélation, un processus dit de traitement croisé compare chaque pic de la surface avec la position correspondante dans une surface de corrélation précédemment calculée. Les vraisemblances des pics obtenus aux tests précédents sont réévaluées en fonction des nouveaux résultats, ce qui permet de valider ou de rejeter certains pics.

Cette méthode de recherche converge en deux ou trois tests, à l'issue desquels une décision est prise. La complexité du diagnostic (départage de plusieurs pics candidats, validation forcée de positions non confirmées,...) nécessite l'usage de mécanismes d'inférence capables de raisonner sur des données incomplètes ou contradictoires.

Pour que la méthode soit optimale, il faut cependant adapter chaque nouveau choix d'algorithmes aux résultats des précédents. Cet aiguillage en ligne s'avère notamment indispensable lorsque certains critères d'interprétation ont indiqué une présence nuageuse sur l'image: l'algorithme d'extraction fournit alors une carte d'ennuagement de l'image, ce qui permet de rejeter les pics situés dans des régions ennuagées et de sélectionner des algorithmes robustes en présence de nuages.

4. Contrôle temps réel

La maîtrise des temps de réponse est inhérente à tout système embarqué. Dans l'application concernée, le principe de balayage des positions de l'amer dans l'image de recherche permet de quantifier précisément les temps de corrélation en fonction de l'algorithme et des tailles de l'amer et de l'image. Si les perturbations exercées sur la navigation obligent à accélérer l'acquisition de mesures de recalage ou entraînent un élargissement significatif de la zone de recherche, une méthode de recalage à deux niveaux de résolution est adoptée [9] :

- Corrélation grossière : l'amer et l'image sont lissés et sous-échantillonnés avant corrélation. Le seuillage de la surface de corrélation résultante détermine des régions d'étude privilégiées.
- Corrélation fine : on découpe une sous-image centrée sur le pic de plus forte vraisemblance, sur laquelle on applique une corrélation fine exhaustive, telle qu'elle a été décrite précédemment. En cas d'échec, le système effectue une recherche itérative autour de chaque pic de corrélation grossière, selon l'ordre de vraisemblance décroissante, jusqu'à validation d'une position.

Ce processus de recherche hiérarchique présente des limitations, par exemple lorsque la description de l'amer contient des structures fines (ex. routes), mais il offre d'énormes gains de temps par rapport à une méthode directe (de l'ordre de 25 fois plus rapide).

Le couplage physique d'un système de navigation inertielle avec un système de recalage par visée d'amers impose d'autre part de pouvoir contrôler la séquence asynchrone des demandes de recalage en fonction des scènes survolées (densité d'amers non-uniforme). Dans le cas d'un satellite, certaines perturbations (frottements,. chocs,...) peuvent également amener le contrôleur de navigation à accélérer la fréquence des recalages. Ces contraintes nécessitent un traitement typiquement temps réel (limitation du temps de réponse, possibilité d'interruption du mécanisme de recherche) qui est rendu possible par l'outil d'inférence temps réel CHRONOS.

5. Choix de la structure de contrôle

La faisabilité opérationnelle d'un système de reconnaissance d'amers passe par un contrôle rigoureux des différentes étapes du mécanisme de recherche: choix des algorithmes et de leurs paramètres en fonction des caractéristiques structurelles et sémantiques des objets recherchés, adaptation aux perturbations sur l'image (bruit, distorsions, occlusions), gestion non-déterministe de l'enchaînement des différents traitements, analyse des résultats, capacité de prise de décision. Ces différentes spécifications nous ont conduit à choisir une structure de contrôle du type système expert.

Le choix du générateur de système expert temps réel CHRONOS [10], développé en Ada, découle d'une réflexion sur le concept de système expert embarqué. La supervision dynamique d'une bibliothèque d'algorithmes est facilitée par la puissance du langage déclaratif d'ordre 1 et par la richesse de la syntaxe (langage procédural en partie action des règles). L'enchaînement des algorithmes et le contrôle de leur fonctionnement (choix des paramètres pertinents, analyse des résultats) est guidé soit par les faits initiaux, soit par les données déduites des tests précédents ("data driven processing"). Les spécificités temps réel du moteur d'inférence (acquisition asynchrone de données externes, raisonnement temporel, interruptibilité, déclenchement sur transitions) permettent d'élaborer un véritable mode de programmation, souple et adapté au contrôle continu du déroulement des traitements. L'interprétation symbolique des résultats garantit la fiabilité du système, par l'intermédiaire de règles de fusion et de décision.

III. PERFORMANCES DU SYSTEME DE RECALAGE

1. Implémentation

Le générateur CHRONOS se compose de quatre modules (éditeur de règles, compilateur de règles, moteur d'inférence et interface d'éxécution); il fonctionne sur Compaq 386. Les algorithmes de corrélations sont codés en langage propre sur une carte accélératrice d'applications TAAC installée sur SUN 4/260. Le moniteur de traitement d'images est développé en langage C.

La base de connaissance, développée à l'intérieur de l'environnement CHRONOS, appelle des procédures externes sur le SUN 4 via une communication série RS-232 et reçoit en retour des données d'interprétation qui activent les règles. La base de règles contient 67 règles d'ordre 1 écrites en syntaxe CHRONOS. Les différents aspects liés à l'architecture du système expert et à son implémentation sont développés dans [11].

Les amers sont extraits d'une image SPOT de référence ne comportant aucun nuage. Les deux images SPOT d'où sont tirées les zones de recherche ont été acquises à des saisons différentes; des nuages de textures variées (brume filtrante, modèle opaque) figurent en différents endroits des deux images.

Le choix préalable des amers et de leur taille constitue une étape critique dans la préparation de mission. Une étude approfondie, dont les résultats sortent du cadre de cet article, a conduit à sélectionner quelques grandes catégories d'amers présentant des propriétés d'unicité et de non-périodicité: bords de côtes (crique, chenal, île, contour rocheux), croisements de routes ou de canaux, lacs, structures géométriques artificielles (aéroports, barrages), bras de rivière, ponts, réseaux fluviaux ou ferroviaires,... Ce regroupement sémantique se double de classifications croisées, par exemple par structures voisines (structures fines ou bidimensionnelles, objets contrastés, amers de brillance, de frontière,...), ce qui permet de prédire le comportement de certaines classes en présence de perturbations spécifiques (p.ex. rotation entre images).

2. Présentation de sessions de recalage

Deux scénarios de recalage typiques illustrent en annexe le fonctionnement du système expert. Les sessions consistent dans le listing simplifié des messages fournis par le système expert au cours de la recherche.

La première session est caractéristique d'une recherche d'un amer présent dans l'image de recherche. L'amer (une intersection de route et de canal dans l'agglomération de La Rochelle) est de type croisement. Il se trouve au centre de l'image de recherche, sur laquelle on distingue un croisement similaire au Nord (fig.6). L'algorithme optimal extrait les structures fines de l'amer et de l'image par une transformation morphologique "chapeau haut-de-forme" (fig.7) dans le spectre Proche Infra-Rouge. La corrélation binaire effectuée sur les images ainsi prétraitées isole alors un seul pic (fig. 8) qui correspond à la position attendue. Il obtient un coefficient de vraisemblance de 0.832 qui le range comme candidat prioritaire. Le système déclenche ensuite une phase de redondance et retrouve la même position par un indice de surface (corrélation linéaire normalisée). Le processus de traitement croisé confirme donc la proposition de recalage et lui attribue, par combinaison des coefficients partiels, une vraisemblance renforcée de 0.979 et une incertitude sur la mesure inférieure au pixel. Le temps nécessaire au recalage est ici inférieur à trois minutes.

La seconde session démontre le comportement sélectif du système. On cherche en effet à retrouver un amer (de type lac) dans une image où il ne figure pas (fig.9). L'image de recherche, une zone portuaire contenant plusieurs bassins de forme proche de l'amer, est perturbée par une couverture nuageuse sur les bords Est et Ouest de l'image. La première corrélation s'attache à retrouver les contours du lac (fig.10) et fait apparaître un seul pic principal (fig.11), qui est éliminé du fait d'un écart radiométrique important avec l'amer. Cette différence suggère une présence de nuages dans la région correspondante et déclenche une extraction de la carte globale d'ennuagement. Le taux de couverture n'étant pas décisif (10 %), le système expert passe donc en phase suboptimale et en mode nuages (choix d'algorithmes robustes). Au deuxième test, il lance une corrélation par comptage, qui fournit quatre pics (fig.11). Trois pics sont rejetés en raison de leur faible coefficient d'intercorrélation, un seul est conservé comme pic secondaire afin que les tests suivants puissent éventuellement le confirmer. Cependant, une étude locale montre un fort taux de couverture nuageuse sur la sous-image située à l'endroit du pic; celui-ci est donc éliminé. Un troisième test est déclenché, afin de faire apparaître de nouvelles positions susceptibles de correspondre à l'amer. Un seul pic est retenu (fig.11), mais sa vraisemblance est trop faible (0.1) et il est rejeté. Au cours de la recherche, aucune position n'a offert suffisamment de garanties de correspondre à celle de l'amer. Le système expert délivre donc une conclusion mixte: l'amer est soit hors de l'image soit dissimulé par une couverture nuageuse limitée à certaines zones de l'image.

3. Synthèse des performances

Les simulations ont porté sur une bibliothèque de 30 amers et 60 images de recherche. Chaque amer, généralement de taille 32x32 pixels, devait théoriquement être retrouvé dans deux images, à l'exception des quelques cas où l'amer est dissimulé par une couverture nuageuse opaque.
A l'intérieur d'une combinatoire élevée, plusieurs centaines de sessions souvent caractéristiques ont été testées. Elles conduisent aux performances suivantes:

- Taux de détection lorsque l'amer figure dans l'image de recherche et n'est pas totalement recouvert par des nûages: 100 %
  Coefficient de vraisemblance moyen: 0.91 (à comparer au seuil de validation fixé à 0.5, sur une échelle [-1, 1]).
- Taux de rejet (non-validation d'une position par le système expert) lorsque l'amer se trouve hors de la zone de recherche ou sous des nuages: 99 %.
  .Dans les rares cas d'erreur, les faux pics validés ont en moyenne une vraisemblance finale de 0.58 qui permettrait de les éliminer par simple seuillage.

La fiabilité et la sélectivité du système sont donc quasi-parfaites. La précision du recalage est de l'ordre du pixel, soit 20 mètres, et pourrait être améliorée par interpolation. La dégradation des performances n'est pas perceptible sur des images sous-échantillonnées de degré 2, y compris pour des structures fines du type aéroports.
Le caractère stationnaire des coefficients de vraisemblance sur plusieurs saisons permet d'établir un classement préférentiel des familles d'amers utile en préparation de mission.
Les règles de détection de nuages offrent une bonne fiabilité. Les performances du module d'extraction de la carte d'ennuagement sont essentielles au fonctionnement correct d'un système embarqué, en raison du taux moyen de couverture sur le globe (40%).

CONCLUSION

Cet article démontre la faisabilité d'un système autonome de recalage de navigation par visée d'amers : le système reconnaît l'amer s'il existe dans l'image, et ne renvoit pas de fausses mesures au filtre de navigation. L'originalité de l'approche consiste à exploiter une base de connaissance dont l'expertise porte sur les algorithmes appliqués, non sur les images. L'apport d'une approche système expert a été fondamentale dans la phase de développement: la simplicité de la formulation déclarative a permis de réduire les temps de maquettage du système expert à quelques semaines, et ainsi de concentrer l'étude sur l'acquisition et la validation de l'expertise. De plus, le concept de système expert est ici un moyen de fusionner des connaissances de nature différente (algorithmique/symbolique) à l'intérieur d'une structure ouverte et flexible; ainsi, pour définir une nouvelle classe d'amers il suffit d'insérer une règle et quelques faits initiaux dans la base existante.

L'expertise liée au comportement des algorithmes est indépendante des caractéristiques du système optique (largeur des bandes spectrales, dynamique des luminances, bruit de mesure). La nature de cette connaissance et les performances du système laissent espérer de larges possibilités d'application à d'autres systèmes d'imagerie, moyennant une hypothèse préliminaire: le cap et l'attitude du véhicule doivent être connus avec précision. Dans le cas d'images CCD, il est utile de disposer d'un modèle radiométrique global des images.

De nombreuses extensions peuvent s'appuyer sur ce concept de supervision d'algorithmes ou de processus. Cette approche constitue en particulier un banc d'essai très souple pour développer, tester et combiner des algorithmes de traitement d'images. Parmi les applications envisagées dans le prolongement de cette étude, ou peut citer la reconnaissance d'amers par corrélation de vecteurs d'attribut, la navigation par suivi de routes ou l'extension de la reconnaissance d'objets au cas tridimensionnel pour l'application au recalage sur amers proches (guidage terminal).

Remerciements:

Nous remercions MM. Camillerapp et Labit (IRISA) pour leur contribution bénéfique tout au long de l'étude, et M. Madar (EURISTIC Systèmes) pour sa participation au développement du système expert.

BIBLIOGRAPHIE

[1] NASR, BHANU, SCHAFFER
"Guiding an Autonomous Land Vehicle using Knowledge-based Landmark Recognition"
Image Understanding Workshop, Proc. pp. 432-439, 1987

[2] GODART,LABASQUE,SELLA,CHAMBRETTE
"Positionnement d'un satellite en utilisant des amers"
MARI/COGNITIVA 87, PARIS May 1987, Proceedings

[3] PRATT
"Correlation Techniques of Image registration"
IEEE Transactions on Aerospace and Electronic Systems, Vol.AES-10, May 1974

[4] GILBERT,MAHAJAN
"On-board landmark and attitude reference parallel processor system"
Martin Marietta Aerospace and NASA Goddard Space Center, N7914129,
Mechanics/Estimation Theory Symp. 1979

[5] CAMILLERAPP
"Utilisation du contexte en reconnaissance des formes",
Thèse de Doctorat d'Etat, Institut de Programmation, Paris VI, 1976

[6] Rapport final "Etude SEKTI", Marché DRET 87/464, Document SAGEM AS89-009

[7] MURAKAMI, KUMAR
"Correlation of Binarized Images"
EEE Transactions on Aerospace and Electronic Systems,
Vol AES 19, N°2, March 1983

[8] KIM, PAYTON, OLIN, TSENG
"A context-dependent Automatic Recognition System"
SPIE vol. 485 Applications of Artificial Intelligence, 1984

[9] ROSENFELD, VANDERBRUG
"Coarse-fine Template Matching"
IEEE Transactions on Systems, Man and Cybernetics, Feb. 1977

[10] Manuel de Référence de CHRONOS

[11] REICHERT, MADAR, BERTON
"Supervision d'un recalage de navigation par visée d'amers"
9èmes Journées "Systèmes Experts et leurs Applications,Session "IA & Défense", Avignon,
France, 29 Mai-3 Juin 1989

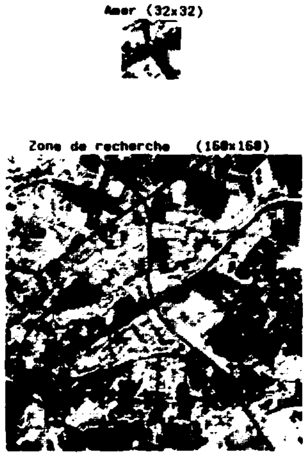


fig.6
amer_73 (taille 32*32 pixels) et
zone_373 (taille 160*160 pixels)

```
    Image de recherche : zone_373
    Amer de reference : amer_73
    ---------------------------
date de prise de vue de l'image : 08/10/87
date de prise de vue de l'amer : 16/02/87
temps imparti au recalage : illimite

l'amer etudie est de type croisement
  caracteristiques: structure fine
  contexte: agglomération

la phase de recherche est : initiale
Choix des algorithmes :
      pretraitement : top_hat
      correlation : binaire_quadratique
      canal SPOT : XS3

la surface de correlation comporte 1 pics
    coordonnees du pic_1 : (64,64)

Le coefficient de vraisemblance du pic_1 vaut 0.832
=> pic_1 : prioritaire
prochaine phase de recherche : redondance

Choix des algorithmes :
      pretraitement : aucun
      correlation : lineaire_normalisee
      canal SPOT : XS3

la surface de correlation comporte 1 pics
    coordonnees du pic_1 : (64,64)

Le coefficient de vraisemblance du pic_1 vaut 0.878

traitement croise au point (64,64)
=> le pic_1 est candidat au recalage, avec un
   nouveau coefficient de vraisemblance egal a 0.979

Validation du recalage
    position de l'amer : (64,64)
    vraisemblance : 0.979
```

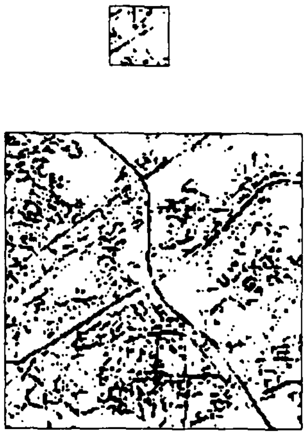


fig.7
amer_73 et zone_373 après
prétraitement : top_hat transform
+ binarisation

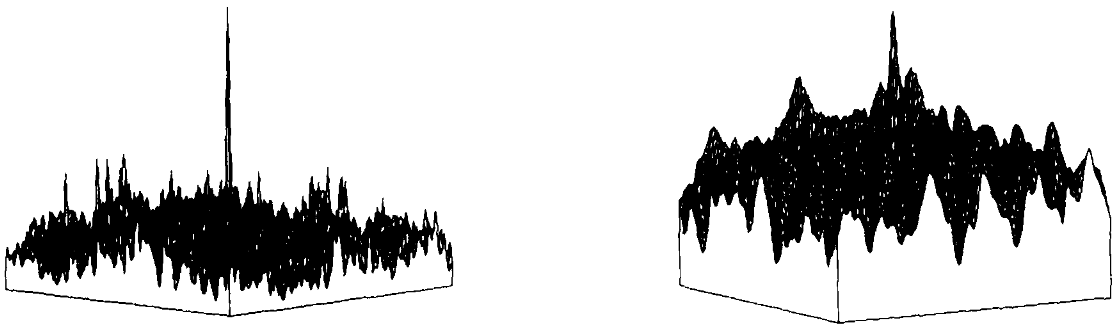


fig.8
Surfaces de corrélation associées au test initial
(à gauche) et de redondance (à droite)

SESSION 2

```
   Image de recherche : zone_334
   Amer de reference : amer_20
   --------------------------
date de prise de vue de l'image : 08/10/87
date de prise de vue de l'amer : 16/02/87
temps imparti au recalage : illimite

l'amer etudie est de type lac
  contexte: littoral

la phase de recherche est : initiale
Choix des algorithmes :
      pretraitement : canny_seuil_hysteresis
      correlation : binaire_quadratique
      canal SPOT : XS3

la surface de correlation comporte 1 pics
    coordonnees du pic_1 : (0,117)

ecart des variances reelle et estimee au pic_1 = 130.3 pc
=> rejet du pic_1 car cet ecart est trop important
=> hypothese de presence de nuages dans cette region

Une couverture nuageuse a ete detectee sur l'image
Taux de couverture nuageuse global sur l'image : 9.89 pc

prochaine phase de recherche : sub_optimale

Choix des algorithmes :
      pretraitement : aucun
      correlation : comptage_differences_seuillees
      canal SPOT : XS3

la surface de correlation comporte 4 pics
    coordonnees du pic_1 : (90,29)
    coordonnees du pic_2 : (113,[illegible])
    coordonnees du pic_3 : [illegible]
    coordonnees du pic_4 : ([illegible],95)

le pic_1 est rejete du fait de sa faible intercorrelation : 155
le pic_3 est rejete du fait de sa faible intercorrelation : 139
le pic_4 est rejete du fait de sa faible intercorrelation : 172

Le coefficient de vraisemblance du pic_2 vaut 0.250
=> pic_2 : secondaire

Taux de couverture nuageuse autour du candidat_secondaire_1 : 15.53 pc
=> rejet du candidat_secondaire_1 car il se trouve dans une zone nuageuse

aucun pic n'ayant une vraisemblance suffisante, un nouveau test va
etre lance sur l'image. Prochaine phase de recherche : sub_optimale

Choix des algorithmes :
      pretraitement : aucun
      correlation : lineaire_normalisee
      canal SPOT : XS3

la surface de correlation comporte 1 pics
    coordonnees du pic_1 : (114,64)

Le coefficient de vraisemblance du pic_1 vaut 0.100
Le pic_1 est rejete en raison de son faible CV : 0.100

Le candidat_secondaire_1 n'a pas ete confirme

Aucun pic suffisamment vraisemblable n'etant apparu au cours des trois
premiers tests, et la couverture nuageuse sur l'image etant non
negligeable, il est possible que l'amer se trouve sous les nuages, ou
hors de l'image
```

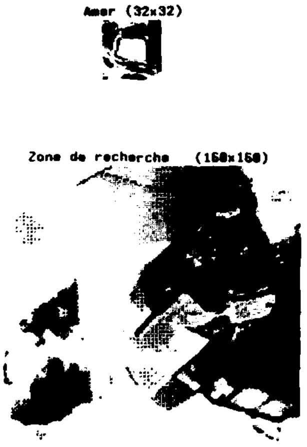


fig.9
amer_20 et zone_334

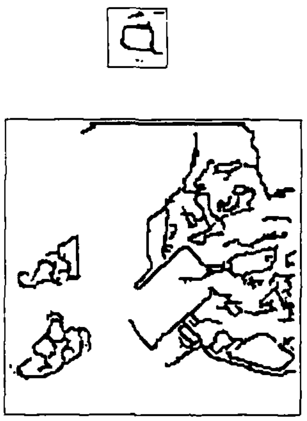


fig.10
amer_20 et zone_334 après prétraitement
(canny + seuillage avec hystérésis)

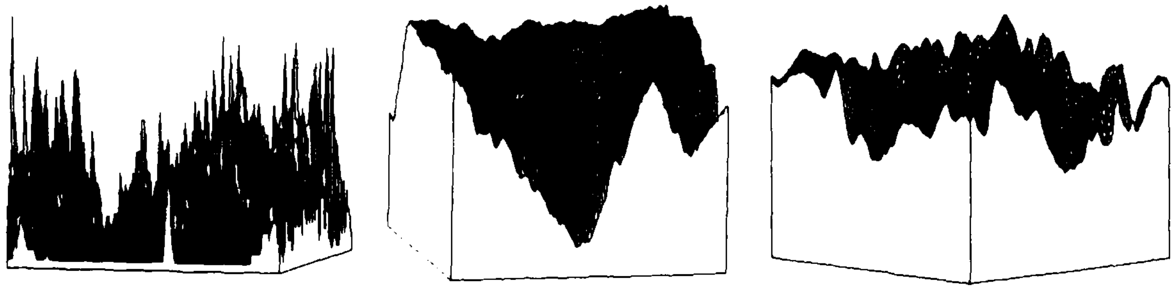


fig.11
Surfaces de corrélation associées respectivement
aux premier, deuxième et troisième tests(de g. à d)

REUSABLE NAVIGATION MODULE

by
C. W. Hall
K. A. Kohout
Naval Weapons Center
China Lake, California
93555-6001
U.S.A.

SUMMARY

The principal objective of the reusable software task is to determine if missile software that is mission specific can be made reusable, and if so, how best to identify and create the reusable parts. A baseline was established from an Independent Exploratory Development (IED) project. The baseline consisted of many packages, procedures, subprograms, and functions, but rather than attempt to make the entire package reusable, attention was focused on the navigation portion. From this navigation portion evolved the common metaparts into a reusable navigation module that could be used either intact or modified in other missile systems. This reusable navigation module was implemented in the MIL-STD-1815A (Ada) language, mandated by Directives 3405.1 and 3405.2 for computers integral to weapon systems.

A team of engineers at the Naval Weapons Center explored methodologies for identifying, developing, and verifying a reusable navigation module. Currently, several navigational metaparts have been identified and developed. The remainder of the project will be devoted to porting the software to a multiprocessor system, upon which it will be further tested and evaluated for reusability on another typical missile application.

PREFACE

The reason reusability is of such importance to developing software rests in what has been termed a software crisis. The complexity of software projects has caused spiraling costs, long development schedules, a product that fails to meet the requirements, and is unmaintainable or unable to be modified easily for new requirements. Software developments have experienced this software crisis by exhibiting the following problems:[1]

1. Cost. Software costs are seldom predictable and often perceived as excessive.

2. Modifiability. Software maintenance is complex, costly, and error prone.

3. Timeliness. Software is often late and frequently delivered with less-than-promised capability (does not meet the requirements).

4. Transportability. Software from one system is seldom reused in other systems, even when the same functions are needed.

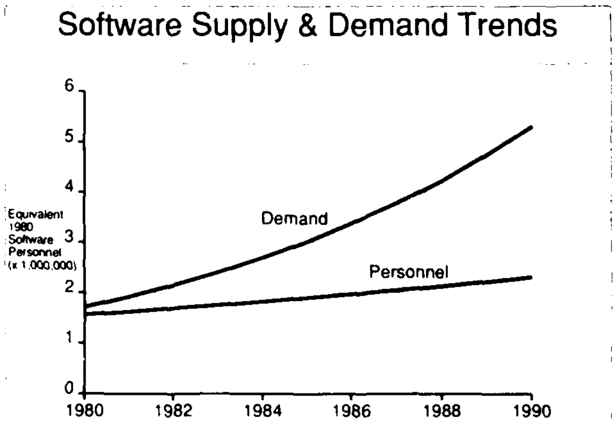
A leading expert in industry has indicated that the underlying causes of this crisis are:[1]

1. A failure to understand life-cycle implications.

2. A shortage of trained software engineers (see Figure 1).

3. Machine architectures that discourage modern software development practices.

4. Development entrenched in the use of archaic programming languages and practices.



source Grady Booch Software Components with Ada

Figure !.

With hardware becoming cheaper, software has become and will continue to be the principal cost factor of most modern computer systems (see Figure 2). Management has identified the software development phase which accounts for most of this cost and is trying to improve this phase. As studies have shown (see Figure 3), maintenance accounts for more than half of the software life-cycle of most computer systems. Hence, a highly (or even somewhat) maintainable system would lower the cost of the system. With most of the maintenance consisting of changes or new requirements, a highly maintainable system would probably have a longer life.

Approved for public release; distribution is unlimited.

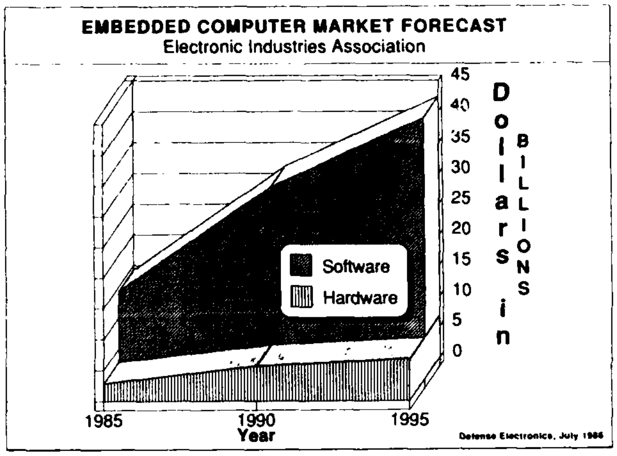


Figure 2.

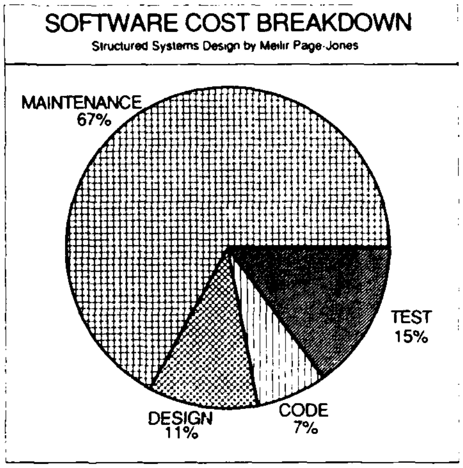


Figure 3.

Ada was designed as a tool for controlling costs for tactical embedded computer systems in military weapon systems. The Department of Defense has standardized Ada for Military systems, and the North Atlantic Treaty Organization has approved the use of Ada for implementation of its information systems that support the exercise of command, control, and communication within the alliance. This standardization was adopted to reduce costs of new systems. Common Ada Missile Packages have demonstrated that missile software can be made reusable and costs can be reduced.[2]

There has been a problem with Ada in real-time mission critical software. In particular, because of the high frequency loops in missiles, there has been a sharp focus on performance criteria. Ada has been called "too inefficient" for mission-critical software in demanding domains like missiles. Yet Ada has only just become a teenager. TeleSoft has announced that its latest Ada compiler is faster than many C compilers (see Figures 4 and 5). As Ada matures, projections of computer throughput and compiler performance will increase. Some of the remaining inefficiencies will be offset by faster hardware.

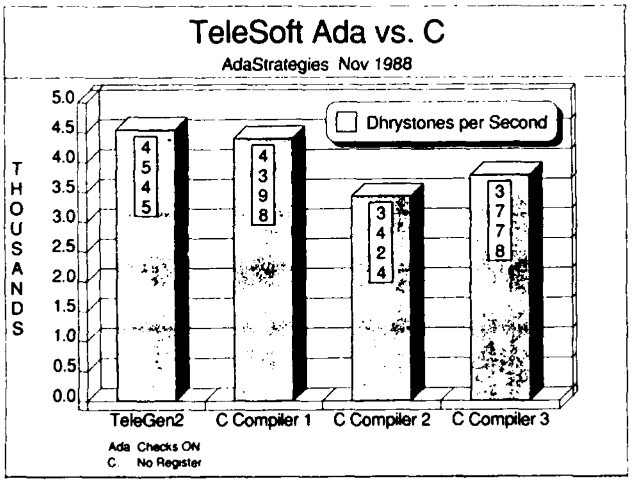


Figure 4.

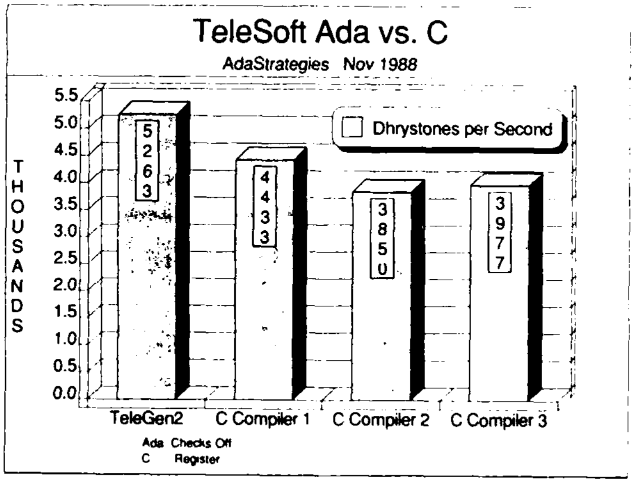


Figure 5.

INTRODUCTION

As previously stated, the purpose of the project is to demonstrate that computer software development costs can be reduced if components can be made reusable. If a project can build from previously developed and verified components, then the cost would be lowered. With the lack of software people to develop new software, projects are not able to easily reduce project time by hiring more software engineers nor is this necessarily recommended. One solution is to use already developed and verified software to meet the needs at hand.

An IED project's missile system was used as a baseline from which reusable operations were extracted. This missile system was designed using Ada as the program design language, and then implemented and verified in the Ada language. The lead engineer, having intimate knowledge of this and other navigation systems, considered the navigation module to be an excellent candidate for reusability. The navigation module was examined from both a functional and component (unit) basis.

REUSABILITY CONCEPTS

The Ada language is portable, but software written in Ada is not necessarily reusable. Portable software is software that is machine independent. Reusable software is software that is application independent. The Ada software, being portable, means that the code can be compiled on any machine with a validated Ada compiler. For software to be reusable, it must be designed with reusability in mind. Software is developed for one system, but is reusable if more than one application can use it with few or no modifications.

Since reusable software is created with reusability in mind, the design methodology can directly affect the quality of the reusable software created. The traditional design methodology is structured design.[3,4] Structured design creates software by developing data flow diagrams, a data dictionary, and mapping this to the architecture using structure charts. A design methodology that targets reusability as a product is object-oriented design. An object-oriented design creates software by analyzing and defining the objects and operations of the software. The advantage object-oriented design has is that it supports modern software engineering principles which assist in creating reusable software and can produce top-level packages of Ada modules. These methodologies are not mutually exclusive but seem to be complementary when designing software.

Ada SUPPORTING REUSABILITY

The concepts that Ada supports reflect modern software engineering principles. These principles facilitate creating reusable software. The goals, concepts, and features of Ada that support reusability are:[5,6]

1. Generic Units. Often the logic of part of a program is independent of the types of the values being manipulated. A mechanism is therefore necessary for the creation of related pieces of programs from a single template. This is particularly useful for the creation of libraries.

2. Data Abstraction. Extra portability, maintainability, and reusability can be obtained if the details of the representation of data can be kept separate from the specifications of the logical operations on the data.

3. Locality. Ada allows for collecting logically related computational resources. This creates highly cohesive software that if loosely coupled will facilitate development of reusable software.

4. Code Portability. Ada makes it easier to reuse source code between different computers with a minimum of effort.

5. People Portability. With language standardization, software personnel will be more able to move from project to project with minimal startup times.

Ada is a very stable language. No subsets or supersets of the Ada language can be called Ada. With the rigorous compiler validation suite, the differences among implementations are minimized. This allows software to be ported from machine to machine. Reusable software can then be developed by any group using a validated Ada compiler. What makes Ada a reusable language is that it embodies and facilitates the use of modern software engineering principles of abstraction, information hiding, modularity, and locality.

REUSABLE MISSILE NAVIGATION MODULE

A simulation of the target missile and its environment was needed to test the navigation package. The missile module was a single 6-degree-of-freedom simulation that provided the stimulus to the sensor data for the package. Algorithms and specifications for a bona fide missile navigation module were used to construct a working package in Ada for the demonstration. Once this module was operational, parts were examined and the design altered while examining how viable the reusable concept could be made. The missile simulation helped verify that the navigation module could navigate a missile and was operating correctly in all of its various modes.

Two different design methodologies were applied. The first was a traditional top-down structured analysis technique and then an object-oriented design methodology using the literature provided by EVB Software Engineering, Inc. consultants. Using structured analysis and design, a navigation module was created that contained common interfaces for a missile domain. The interface of data was only allowed to be passed in through parameter calls and not via common global data blocks. After examining different missile navigation systems, the interface appeared to be common.

An object-oriented design was used to define and create submodules within the navigation system. These submodules were then evaluated for replacement in rebuilding the navigation module. This makes the navigation module not only functionally reusable but reusable at a component level. These submodules (components) can then be used to build other functionally reusable components (i.e., autopilot).

Problems of Using Reusable Software

Reusable software has problems that must be overcome for reusable software libraries to flourish. Some major problems that need to be examined are:

1. Documentation. Should a reusable library mandate a documentation style? And if so, what should that be and how would it be enforced?

2. Verifications. Does the library give a warranty or is it as-is? What level of verification is necessary for reusable software to be placed in a library?

3. Support. Will the software have any support if the software needs modifications or explanation? If the person supporting the software leaves, will it continue to be supported?

4. Identification and Retrieval. How do you locate and retrieve reusable software and its documentation? What types of key words are necessary?

5. Financial. How is the library supported and staffed? Are users of the library charged for each piece of software?

6. Creation Syndrome. A certain mind set must be attained. Many software engineers won't or can't use other people's software because of an attitude problem. Being a creative sort, the typical software engineer will convince themselves that their task must be done from scratch.

Front-End Interface GENS

Locating and identifying reusable software, once it has been placed in a library, is a difficult task. To make this easier for the individual using the reusable navigation parts, a user-friendly interface was created. This interface, Generic Expert Navigation System (GENS), would allow the individual to choose parts of different missiles that could be used on a navigation system. Initially, these parts would be: inertial measurement units, integration schemes, navigation schemes, and alignment schemes. Once, these parts have been chosen, the navigation software components would then be located, integrated, and written to a file. GENS would allow a navigation routine to be built quickly and with little effort.

CONCLUSION

Effective, maintainable Ada code is dependent upon the compiler and the expertise of programmer/analysts. An Ada reusable software team of very knowledgeable programmer/analysts was formed. Each engineer has knowledge in structured design, object-oriented design, modern software engineering principles, and missile applications. An excellent compiler cannot compensate for an inadequate software engineering discipline. The bottom line is well-trained software engineers can produce excellent Ada code that can be reusable. This experience and those of R. Holibaugh at the Software Engineering Institute and D. McNicoll at McDonnell Douglas indicate that weapons software can be made reusable. This reusable software can save time, money, and manpower for other projects that use it.[2]

As reusable software components become available, there will be a definite need for reusable libraries. These libraries would consist of staffs of people knowledgeable in the parts within the library along with a system of locating reusable components given specific and nonspecific requests. When a given project begins, a software engineer would go to the library and request components that relate to the project at hand. For a missile, navigation is required. This navigation component has requirements that specify precision, time and memory constraints, and functionality. A navigation component would be chosen from a list of navigation components that satisfy the requirements.

Conceptually, a reusable software library could be thought of as a library of software chips. They would be adequately described, identified, and verified so that an engineer can put them into their system and get expected results. In a similar fashion, digital engineers look up hardware chips in a transistor transistor logic (TTL) manual and select the ones required for their system design. There are numerous instances of widely varying applications using TTL chips.

If this state of software development is realized through reusable software chips, the software discipline will have emerged from its infancy. Sorting chips, Kalman filter chips, flight control chips, and navigation chips would be used to develop future missile systems. Software engineers may no longer be creating the "wheel" for the n^{th} time.

REFERENCES

1. Grady Booch. *Software Engineering With Ada.* Menlo Park, Calif., Benjamin/Cummings Publishing Co., 1983, pp. 5-10.

2. D. McNicoll. *Volume II. Common Ada Missile Packages.* St. Louis, Mo., McDonnell Douglas Astronautics Co. AFATL-TR-88-66, pp. 25-32 and p. 8.

3. Meilir Page-Jones. *The Practical Guide to Structural Systems Design.* New York, Yourdon, Inc., 1980.

4. Tom DeMarco. *Structured Design and System Specification.* New York, Yourdon, Inc., 1978.

5. J. G. P. Barnes. "Programming in Ada," *International Computer Science Series,* 2nd ed. Menlo Park, Calif., Addison-Wesley Publishing Co., 1984, p. 7.

6. John Foreman and John Goodenough. *Ada Adoption Handbook: A Program Manager's Guide.* Pittsburg, Pa., Carnegie-Mellon University, Software Engineering Institute, 1987. CMU/SEI-87-TR-9, p. 2.

PARALLEL PROCESSING IMPLEMENTATION OF A FLIGHT CONTROLLER

by

P.J.Fleming, F.Garcia Nocetti, C.M.Jones and **H.A.Thompson**
University of Wales, Bangor
and
G.Ingle and **E.Balley**
Royal Aerospace Establishment, Bedford MK41 6AE
United Kingdom

1.0 Summary

This paper investigates the feasibility of using a parallel processing transputer-based network, programmed in occam, for the implementation of an aircraft flight control law. Three techniques to generate the concurrent realization of this control law are described and illustrated together with some indications of their strengths and weaknesses. Software tools have been used and developed to automate the mapping of the control law on the transputer system. Integration of the existing control hardware on-board the aircraft and the parallel processing hardware has also been addressed.

2.0 Introduction

2.1 Why use parallel processing in flight control ?

Digital controllers, such as those required for present and future aircraft, demand a powerful computing capability to achieve the required performance. For example, new airframe designs and extended aircraft performance envelopes require complex control laws to be computed at high sampling rates. In recent years, in many instances digital control requirements have begun to outstrip the performance of a general purpose microprocessor. Control engineers are trying to implement more complex algorithms over shorter computational intervals. A major research programme at Bangor has been to evaluate a range of parallel architectures on a variety of control application areas.

2.2 Parallel processing computer architectures

One objective of parallel processing is to speed up the execution time for a task. This is achieved by dividing the problem into several sub-tasks and allocating multiple processors to execute multiple sub-tasks simultaneously. Before dealing with the specific architecture used in this Demonstrator Project, a generalised representation of a parallel processing system is presented in Fig.1 , where PE represents a Processing Element. The PEs execute concurrently, communicating with each other when necessary.

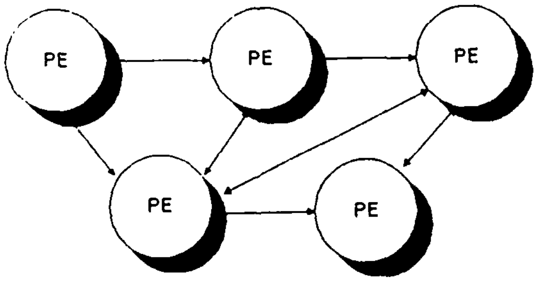


Fig.1 Generalised parallel processing system

Parallel architectures differ in respect of the flexibility of each individual PE and the degree of interconnectivity between PEs. Granularity is a measure of the size of an individual task to be executed on a parallel machine. Fine-grain architectures have simple PEs (often a 1-bit ALU and tens to hundreds of bits of local RAM) with restricted functionality and a wide bandwidth for local data communication. While PEs of medium-grain architectures are more general-purpose in nature, inter-processor communication bandwidths tend to be narrower and, perhaps, more globally oriented in these systems.

Parallel architectures are further distinguished by being either programmable or fixed-function systems. When contemplating these architectures for real-time control implementation, the control engineer must pay.close attention to issues such as algorithm structure and complexity, potential for parallelism and interprocessor communication requirements.

In this Project then we have concentrated on medium-grain architectures and, indeed, on one specific type. A medium-grain architecture is typically a MIMD (Multiple Instruction Multiple Data) system, constituting a parallel computer system composed of multiple independent processors.

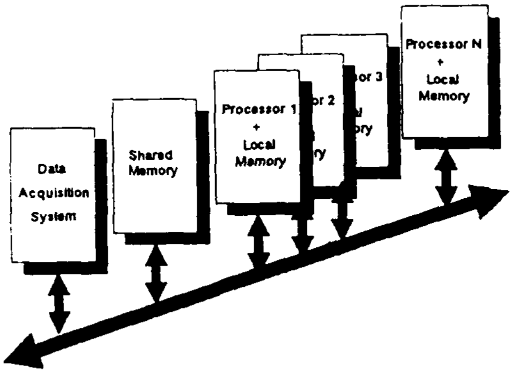


Fig.2 Single-bus multiprocessor configuration

Bus-contention problems arise when using a single-bus multiprocessor configuration (Fig.2) for MIMD machines and limit system performance. As processors are added to this system, both the interprocessor and processor-memory bandwidths are degraded, resulting in extended "idle" times for each processor as they wait to gain access to the bus. The "network" configuration supported by the Inmos Transputer device has been extensively investigated for real-time control applications (Fleming 1988). Furthermore, it is an architecture which supports a special parallel programming language, OCCAM, which is intended to significantly simplify the parallel programming task.

2.3 Demonstrator project

This paper describes software and hardware aspects of a Demonstrator Project undertaken by University of Wales, Bangor and Royal Aerospace Establishment, Bedford, to map an experimental automatic flight control law onto a parallel processing system, specifically to investigate the potential of the INMOS transputer and the OCCAM language for this task. The control law had previously been flown on the Civil Avionics Section's BAC 1-11 at RAE, Bedford, implemented via the Versatile Auto-Pilot (VAP) onboard that aircraft. The VAP control law was chosen because it is representative of a class of flight control laws. However, it is not particularly demanding for implementation on a "fast" transputer array. In this project it is simply used as a "Demonstrator vehicle".

3.0 The Transputer and OCCAM

3.1 The INMOS Transputer

The Transputer is a family of single-chip computers which incorporates features to support the occam model of parallelism. The T800 version (Fig.3a) consists of a 32-bit 10 MIPS RISC processor, 4Kbytes of local RAM, an external memory interface and four link interface units. Avoiding the potential communication bottleneck of a single-bus system, the Transputer is intended to function in a network configuration such as that illustrated in Fig.3b. Bit-serial communication rates of 20Mbit/s are achieved via the four bidirectional interprocessor links. Serial/parallel data conversion is performed by the on-chip link interface hardware.

The simple machine structure of the Transputer allows the complex instructions necessary for occam's process handling and message passing to be implemented in microcode. It uses different procedures for external channel communication (via the links) and internal channel communication (on-chip interprocess communication). Provided that frequently accessed data can be held in local RAM, a high level of performance is maintained by the stack-oriented six-register architecture. It is worth noting at this point that it is possible to run a parallel set of tasks on a single transputer. Operating in this way, the transputer simply time-slices the various parallel tasks, thus simulating parallel implementation. The software is easily restructured to run on varying numbers of processors.

While the Transputer was designed specifically to support OCCAM, it is possible to run such languages as Parallel-C, Pascal and FORTRAN.

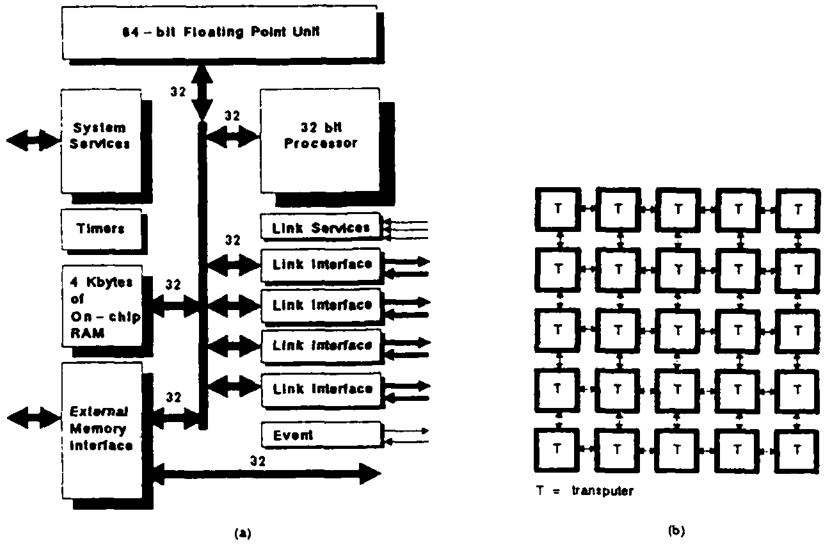


Fig.3 (a) T800 Transputer architecture; (b) Network configuration

3.2 OCCAM

A sequential programming language is characterised by its actions occurring in a strict, single execution sequence. The behaviour of such a program thus depends only on the effects of the individual actions and their order. The time taken to perform the individual actions is not of consequence.

A parallel program, however, may consist of a number of tasks or processes which themselves are purely sequential, but which are executed concurrently. Two important issues which must be addressed by a parallel programming language are those of synchronisation and the handling of shared variables. When several tasks are executing concurrently, they do so asynchronously, i.e. each task proceeds at its own speed. If the activities of these tasks are dependent on one another then the programming language must provide a means of synchronisation to co-ordinate their activities. Further, when two or more tasks are accessing the same variable, control must be exercised to prevent data corruption arising from conflicting operations on that variable.

A number of mechanisms has been devised to handle these two issues - synchronisation and shared variables - such as semaphores, guards, data monitors, rendezvous, etc. However the use of such mechanisms requires careful structuring of the software by the programmer. Through its message-oriented approach, based on "processes" and "channels", occam gives a higher level of support to the programmer and thus reduces the software design overhead associated with concurrent program design.

4.0 VAP Control Law

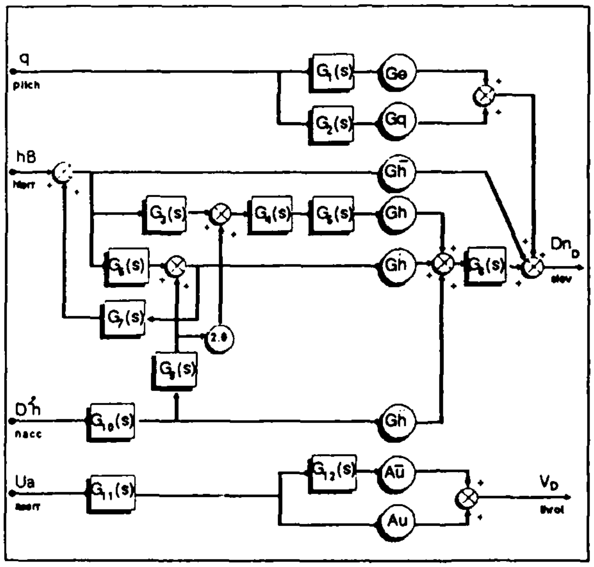


Fig.4 VAP control law - block schematic

The Versatile Auto-Pilot [1], is a 4-input, 2-output control law and it is presented in block diagram form in Fig.4. The VAP control law was chosen because it is representative of a class of flight control laws. It operates in three different modes: Height hold, Glidepath hold and Flare, depending on flight conditions. This project describes methods used to generate the concurrent realization for only one of these modes, the most complex, the Height hold mode.

5.0 Mapping Approaches

5.1 Static and Dynamic Task Allocation

There are two ways in which we can allocate tasks to individual processors: that of static and dynamic task allocation.

In static task allocation, we initially allocate a task to a processor and bind that task to the processor for its lifetime.

Dynamic task allocation is performed by on-line scheduling, that is, allocation of a task to a processor according to certain criteria such as processor availability and intertask dependencies. In dynamic task allocation, therefore, there is no strict association between a task and the processor which will execute it. Tasks are simply allocated to available processors according to certain predefined criteria.

Static task allocation is more appropriate in this specific case. (However, elsewhere [2] the process of dynamic task allocation on a transputer array has been reported.)

The tasks were allocated to the processors in three different ways:

(i) the parallel branches approach,
(ii) the 'heuristic' approach, and
(iii) the 'hybrid' approach.

These approaches are evaluated with respect to execution speed, ease of programming, adaptability and extendability.

5.2 Parallel Branches Approach

In the parallel branches approach, the control law was modified, using block diagram transformations to generate a parallel network of Laplace transfer functions. Then, each of these parallel paths was expanded into partial fractions and converted to the discrete-time domain to reduce the control law to a sum of discrete functions. The result is a network of 37 independent difference equations as shown in Fig.5. These are then combined to form the control signals:

$$Dn_D(k) = U_1(k) + U_2(k) + U_3(k) +...+ U_{35}(k)$$
$$V_D(k) = V_1(k) + V_2(k)$$

where Dn_D and V_D are the elevator and throttle demands and $U_1,U_2,...,U_{35},V_1,V_2$ are simple discrete functions (e.g integrator, gain, first order lag etc.) which represent the software building blocks.

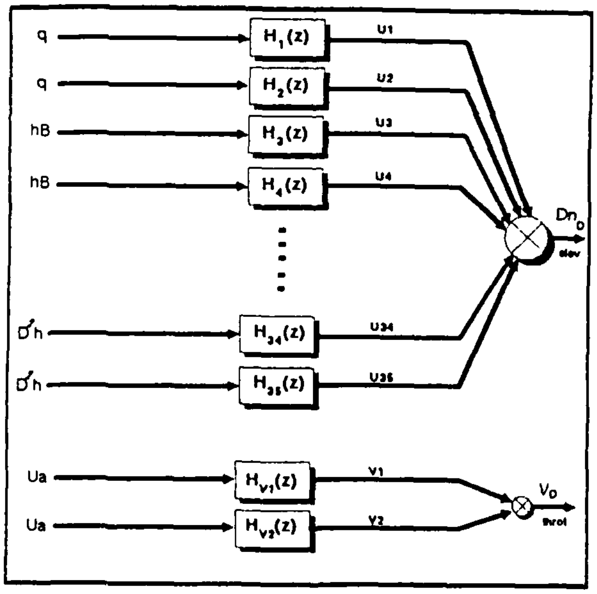


Fig.5 VAP control law: parallel branches approach

5.2.1 Transputer Implementation

For the parallel branches approach the difference equations were programmed as individual tasks. Tasks were then grouped together to give approximate load balancing across the available processors.

The 4-processor implementation is illustrated in Fig.6. The OCCAM software is organised such that process T_0 inputs data from the Monitor process, which controls input and output. Then it sends this data simultaneously to the T_1, T_2 and T_3 processes. T_0 evaluates V_D (throttle demand) and UT_0. Concurrently with this process, T_1, T_2 and T_3 are executed. UT_1, UT_2 and UT_3 are calculated and sent to T_0, which evaluates as a sum of UT_0, UT_1, UT_2 and UT_3. Finally Dn_D and V_D are broadcast to the Monitor process. These processes are repeated over each sample interval.

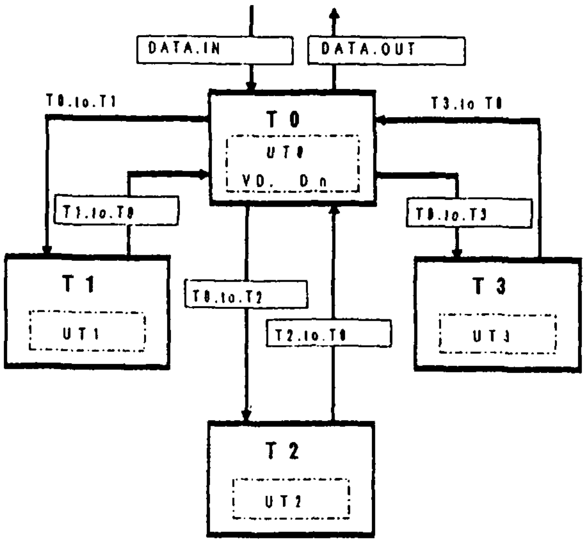


Fig.6 Transputer configuration for VAP control law (4-processor version)

5.2.2 Performance

The performance of this approach for different numbers of processors, (transputers), has been evaluated in real-time implementations. A high priority timer process was included in the programs to provide a clock, "counting" every 1.0 microsecond, which is used to calculate the execution time and the average (%) transputer activity. The program was run on an IMS BOO3 evaluation board in conjunction with an IBM PC and the results are shown in the Table at the end of Section 5.

This Table shows a set of results obtained for different numbers of processors under different mapping strategies. In the Parallel Branches column, the first pair of results corresponds to the allocation of all the processes on one transputer, which, obviously, spends all its time processing the control law. The remaining pairs of results are obtained by mapping the processes onto two up to eight processors showing a significant execution time speed-up as additional processing power is introduced. However, at the same time, a deterioration in the processor average activity can be observed. As a result, the "ideal" linear speed-up is, of course, not realised, nevertheless, the timings do approach this desired goal.

5.3 Heuristic Approach

For the heuristic strategy, the discrete equivalent of each transfer function block in the original VAP control law was calculated. The dependencies of the blocks along the various paths in Fig.4 were inspected. Even though this structure consists of a collection of sequential blocks, through a reasoning process, parallelism in the structure was extracted through observation. A number of independent groups of tasks, with no common factors, was derived as shown in Fig.7. No further partitioning was feasible because of the constraints enforced by sequentialism within the paths. Using this method, the outputs of the control law were reduced to the following sequence of computations:

$$Dn_D(k) = Dn_0(k) + Dn_1(k) + Dn_2(k)$$
$$V_D(k) = V_1(k) + V_2(k)$$

The resulting discrete functions corresponding to the Heuristic design approach of the VAP control law were mapped onto different transputer arrays varying from one up to three transputers running concurrently.

5.3.1 Performance

The average (%) transputer activity and execution times for this strategy are also shown in the Table at the end of this Section. These results show an important reduction in the execution time when varying the number of transputers from one to three processors running concurrently. At the same time there is a deterioration in the processor average activity. Subsequent results for larger numbers of processors show that no further improvement can be obtained through this approach. Average processor activity deteriorates for the other configurations due to the presence of critical sequential paths arising from this mapping.

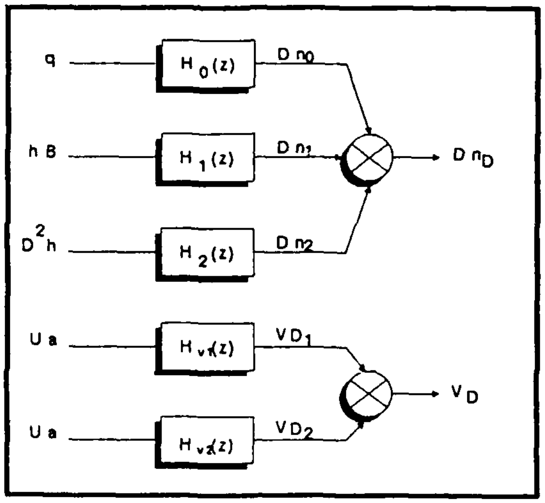


Fig.7 VAP control law - Heuristic approach

5.4 Hybrid Approach

The hybrid method combines the heuristic and parallel branches techniques in the concurrent implementation of control algorithms. According to this strategy, we first apply the heuristic technique to initially decompose the problem. Then, the parallel branches approach is employed to split the critical sequential paths of the control law to execute them concurrently. In the version of the VAP control law shown in Fig.7, the $H_1(z)$ path is critically long for real-time implementation. The parallel branches approach was used on this path to decompose it into a set of shorter parallel paths: the new 'hybrid' representation is shown in Fig.8.

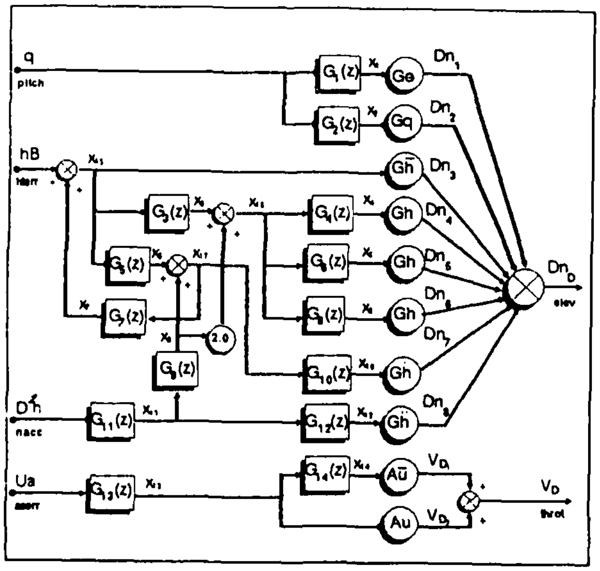


Fig.8 VAP control law - Hybrid approach

5.4.1 Performance

The best time obtained by the hybrid strategy, using a five-transputer array, is 0.525 ms - the lowest time in the case study. The duplication of tasks has been limited to the critical paths only where the parallelism has been extracted through application of the parallel branches approach.

VAP CONTROL LAW			
PROCESSORS	PARALLEL BRANCHES	HEURISTIC	HYBRID
1	2.500 ms 100 % a.p.a	1.055 ms 100 % a.p.a	1.270 ms 100 % a.p.a
2	1.280 ms 99 % a.p.a	0.760 ms 98 % a.p.a	0.865 ms 99 % a.p.a
3	0.910 ms 98 % a.p.a	0.540 ms 79 % a.p.a	0.685 ms 96 % a.p.a
4	0.710 ms 97 % a.p.a	0.540 ms 59 % a.p.a	0.565 ms 95 % a.p.a
5	0.605 ms 94 % a.p.a	0.540 ms 47 % a.p.a	0.525 ms 80 % a.p.a
6	0.560 ms 92 % a.p.a	0.540 ms 39 % a.p.a	0.525 ms 67 % a.p.a
7	0.540 ms 85 % a.p.a	0.540 ms 33 % a.p.a	0.525 ms 59 % a.p.a
8	0.530 ms 81 % a.p.a	0.540 ms 29 % a.p.a	0.525 ms 52 % a.p.a

Table 1. Execution time & average processor activity.

6.0 Automating the Mapping Process

6.1 CAD Software

The purpose of static allocation, involving partitioning of the control law tasks across the available processors, is to minimise the execution time of the tasks allocated to the processors. A clear requirement for the development of CAD tools was identified to assist this process, using block diagram manipulations and generation of the discrete equivalents of the given continuous control law descriptions and their *specific hardware implementation.*

This section presents the development of software to assist the mapping of a control law onto a parallel processing transputer-based system. The tools includes block diagram transformation software, the interactive matrix manipulation package, MATLAB, and the Transputer Development System (TDS) used to develop occam programs for transputer networks. A pictorial representation of the structure proposed for the software is given in Fig.9.

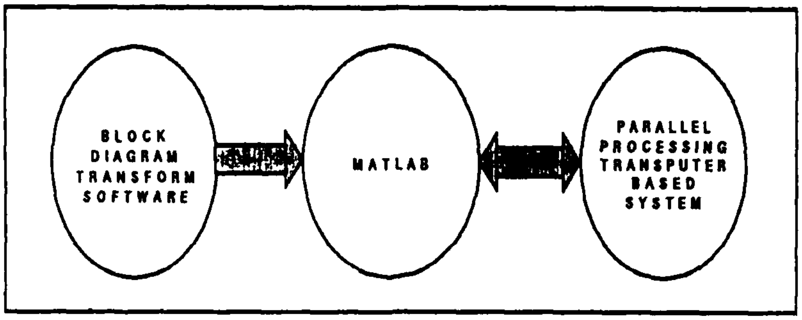

BLOCK DIAGRAM TRANSFORM SOFTWARE
MATLAB
PARALLEL PROCESSING TRANSPUTER BASED SYSTEM

Fig.9 CAD software modules for automating the mapping process

6.1 Block diagram analysis

A block diagram analysis program [3], written in Prolog was developed to perform Signal Flow Graph (SFG) reduction. The reduction procedure is basically aimed at obtaining the overall relationships between each output and input variables in the system. The process of getting a SFG from the block diagram form usually employed to describe control systems can be accomplished through a link between an "in-house" Block Diagram Editor (BDE) and MATLAB. This link permits the automatic generation of a state-space description from the system entered into the BDE, and a matrix that characterizes a SFG for the system can then be obtained from its state-space description.

6.2 Discretisation

The method used to discretise the continuous-time models was the pole-zero mapping method [4]. In the parallel branches approach, this method has been used in a modified version to add one delay in response to the unit step input. Since the resulting discrete-time transfer function has one less power of "z" in the numerator than in the denominator, the execution of the output only requires input from the previous sample time. The purpose of this is to allow one sample period in which to perform the calculation. In the heuristic and hybrid strategies, the modified pole-zero mapping method was only used to discretise the transfer function blocks that are dependent on the control law inputs. Subsequent blocks were discretised using the original version of the method, where the outputs of the discrete-time equations require inputs at the same sample time.

The continuous-time models were discretised using the specially developed MATLAB functions.

6.3 MATLAB Interface

The transfer functions between each pair of input and output nodes of the control law, generated by the block diagram analysis software, described in Section 6.1, are transformed into a set of parallel transfer functions, discretised and represented in state-space form. The linear algebra and matrix computation involved in this process suggested the use of a mathematical tools package such as MATLAB.

The parallel representation, discretisation and state-space equivalent of the control algorithm was calculated using the specially developed MATLAB software. The resulting data files are processed and exported to an OCCAM program which performs the concurrent execution of the control algorithm on a transputer network. An OCCAM output data file can be imported into MATLAB, where a graphics file is created to plot the results of the mapped control law.

(MATLAB software was generated, using PC-MATLAB [5] (the version of MATLAB for IBM-PCs and compatibles) and the MATLAB Control System Toolbox [6].)

7.0 Hardware Integration

An important objective of this work was the integration of the parallel processing hardware and the control hardware on-board the BAe 1-11 test aircraft at the Royal Aerospace Establishment, Bedford. A

VMEbus interface was designed so that the existing MC68000 microprocessor-based system on board the aircraft could exchange data between a VMEbus analog and serial data input/output board and a transputer array. Fig.10 shows a block diagram of the interface interconnecting the data acquisition system and a transputer-based multiprocessor array.

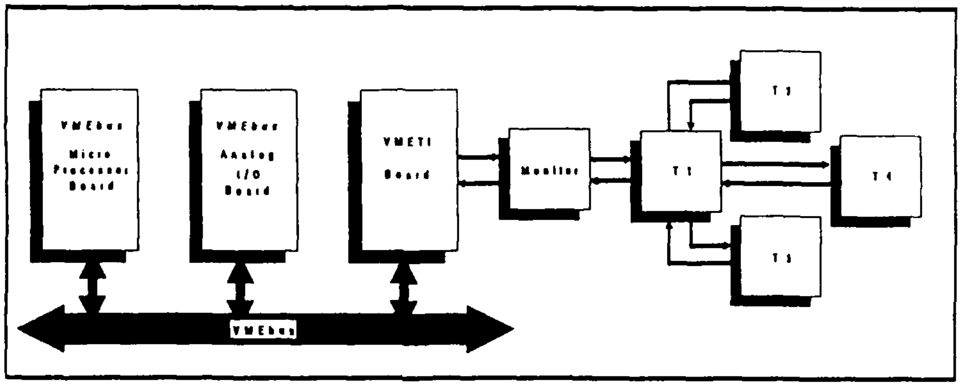


Fig.10 VMEbus hardware interface of transputer-based controller

7.1 VMEbus Transputer Interface (VMETI)

The VMEbus transputer interface uses a dual-port RAM (DPRAM) which is accessible to the transputer array via an IMS C011 link adaptor and to the MC68000 board via the VMEbus. An on-board dedicated 6502 microprocessor is used to coordinate data transfer from the transputer array via the link adaptor to the DPRAM. The DPRAM functions as local memory for the 6502 system. The MC68000 card handles data exchange between the VMETI and the analog input/output board which are both memory-mapped onto the VMEbus. A block diagram of the interface is shown in Fig.11.

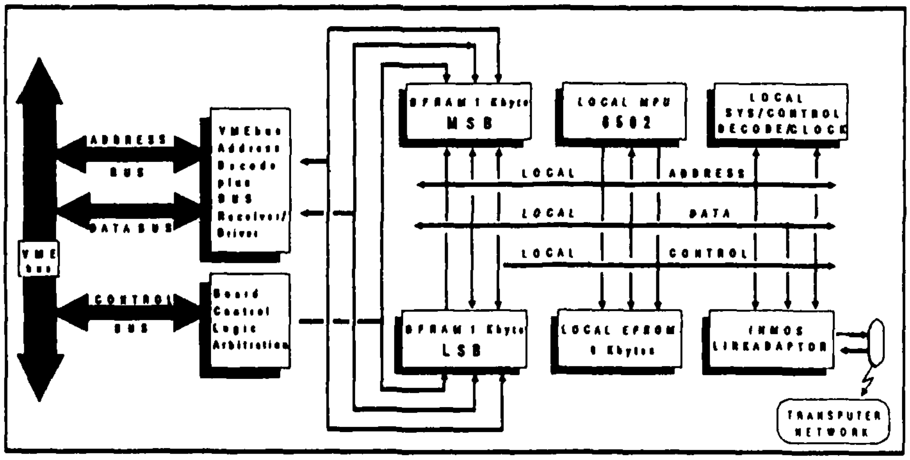


Fig.11 VMEbus Transputer Interface (VMETI)

7.2 Practical results

The complete hardware system - VME data acquisition system, VMETI and 4-transputer network - was tested, plotting the time response of the VAP control law for different input signals. In these tests a number of input signals (step input, bipolar square and unipolar square waves, respectively), have been applied simultaneously to the control law inputs, via the analog input channels of the data acquisition system. The analog output channels were connected to a plotter. Fig.12 shows the relevant output responses to individual step input excitations which closely match previously obtained simulation results.

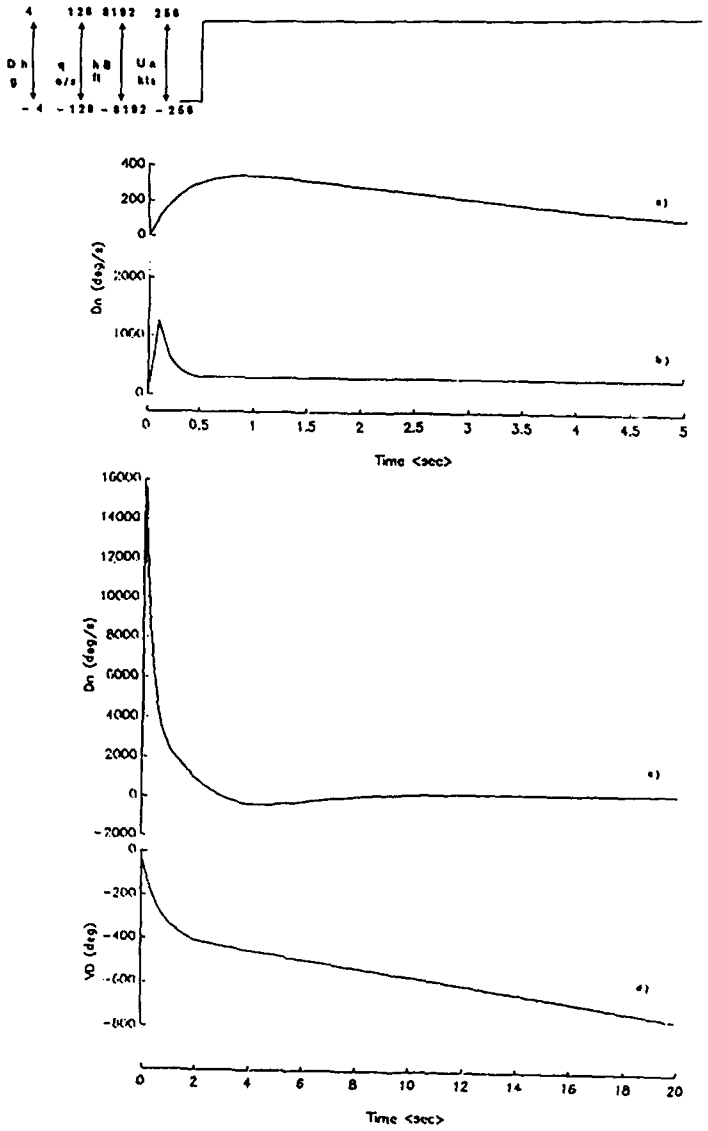


Fig.12 Transputer-based VAP control law output responses to individual step input excitation
(a) Elevator rate response to vertical acceleration perturbation
(b) Elevator rate response to pitch attitude perturbation
(c) Elevator rate response to height perturbation
(d) Throttle response to airspeed perturbation

8.0 Concluding remarks

The demands of realizing complex digital controllers by parallel processing methods has been addressed using the INMOS transputer and OCCAM. We have shown three approaches to generate the concurrent realization for the VAP control law - parallel branches, heuristic and hybrid. These techniques have been evaluated with respect to execution speed, ease of programming, adaptability and extendability when they were allocated to different configurations of transputers. We found that an array of transputers programmed in OCCAM offers a suitable and powerful alternative for reducing the execution time of a real-time control law with a good speed-up factor for a small number of processors. This study has achieved an important improvement in the performance of the controller when static allocation of tasks onto a multi-transputer architecture was used. Thus, the potential of the transputer and the OCCAM language for the implementation of flight controllers has been shown.

This work has generated software tools to automate the implementation of controllers on a parallel processing transputer-based system, reducing development times significantly. Block diagram transformation software has been used to manipulate the original system as a signal flow graph, and convert this system into a set of parallel transfer functions. A number of MATLAB and occam programs has been developed for the mapping of this control law onto a parallel processing transputer-based system.

We have investigated techniques for hardware integration of the existing control hardware on-board the aircraft with the parallel processing hardware and have shown that the complete system is working satisfactorily under laboratory conditions. A flight test of the hardware is planned to be undertaken in the second quarter of 1989.

New techniques are currently being considered to enable each processor to re-organise and assume a greater portion of the processing task in the system when a module fails. These designs, therefore, contemplate fault-tolerance techniques both in software and hardware. These approaches are now being applied to the computation of more complex flight control systems.

Acknowledgements

This work is funded by the UK Department of Trade & Industry. Technical monitoring is performed by RAE, Bedford. Their support is gratefully acknowledged.

REFERENCES

1. Goddard, K.F. and Cooke, N., "Flight trials of an automatic control law for a BAC 1-11 Aircraft", 1980, Technical Report 80003, Royal Aircraft Establishment.

2. Fleming, P.J. and Jones, D.I.: "Parallel Processing in Control", Chapter 13 in Warwick, K. and Rees, D. (eds.): "Industrial Digital Control Systems", Revised Edition, Peter Peregrinus, 1988.

3. D'Oliveira, C., "Block Diagram Analysis Software", 1988, Internal Report, Real-Time Systems Group, SEES, UCNW, Bangor.

4. Franklin, G.F. and Powell, J.D., "Digital Control of Dynamic Systems", Addison-Wesley, 1981.

5. Moler, C., Little, J., Bangert, S. and Kleinman, S., "PC-MATLAB User's Guide", 1987, The MathWorks, Inc., Sherborn, MA 01770, USA.

6. Laub, A. and Little, J., "PC-MATLAB Control Systems Toolbox", 1986, The MathWorks, Inc., Sherborn, MA 01770, USA.

ANNEMOMETRIE BASSE VITESSE : UNE NOUVELLE APPROCHE

par

Yves Paturel
Recherches et Développements/Navigation-Guidage
S.F.I.M.
13 Avenue Marcel Ramolfo Garnier
91344 Massy Cedex
France

Pour les hélicoptères, la vitesse par rapport à l'air est un paramètre important pour la sécurité et la précision des munitions non guidées pour les applications militaires. L'élaboration de ce paramètre dans le domaine des basses vitesses (inférieures à 25 m/s) est un problème délicat puisque les capteurs anémométriques classiques ne sont pas utilisables. La S.F.I.M. a étudié à partir d'enregistrements en vol, un algorithme restituant la vitesse air à partir de la position des différentes positions des commandes de vol de l'hélicoptère. Les résultats de cette étude ont été mis en oeuvre dans un système opérationnel et testés en vol. C'est toute cette démarche et ses résultats que nous présentons ici.

1 - INTRODUCTION :

Les hélicoptères modernes, qu'ils soient civils ou militaires, vont nécessiter la connaissance de la vitesse air dans tout le domaine de vol.

Connaître le vecteur vitesse air dans le domaine des basses vitesses permet d'améliorer la sécurité de l'hélicoptère lors de la tenue des vols stationnaires et des vols à faible vitesse (en avant, en latéral, voire en arrière).

Connaître la vitesse air dans tout le domaine de vitesse permet d'augmenter la précision de tir des armes hélicoptères non guidées puisqu'avec cette information, le système de conduite de tir peut effectuer les corrections nécessaires.

Si les systèmes anémométriques classiques permettent de mesurer la vitesse air dans l'axe longitudinal de l'hélicoptère lorsqu'elle est supérieure à environ 20 à 30 mètres par seconde, la mesure de la vitesse air transverse dans tout le domaine et de la vitesse air longitudinale pour les basses vitesses est beaucoup plus délicate. En effet, le rotor crée autour de l'hélicoptère des perturbations aérodynamiques rendant difficiles les mesures par des méthodes anémométriques. Des solutions astucieuses ont été essayées, telles que des mesures anémométriques effectuées au-dessus du rotor où le flux est moins perturbé ou l'utilisation de capteurs anémométriques tournants. Cependant, toutes ces solutions se sont heurtées à des difficultés de mise en oeuvre ou de fiabilité : mât fixe traversant le rotor, système de rotation des capteurs au bout d'un mât éloignant ces capteurs de la zone aérodynamiquement perturbée.

La solution proposée par la S.F.I.M. consiste à restituer les vitesses air longitudinale et transversale à partir d'informations déjà disponibles à bord de l'hélicoptère : accélérations et positions des organes de commande de vol de l'hélicoptère.

La vitesse air élaborée n'est donc pas une mesure directe mais le résultat d'un calcul. Une application simplifiée de ce principe a été mise en oeuvre dans le système VIMI (Vitesse Indiquée par Moyens Internes). Le principe de ce système avait été dérivé des équations de mécanique du vol de l'hélicoptère. Le brevet de M. DURAND, Ingénieur au C.E.V. décrit ce système simplifié. Cependant, ses performances restaient limitées car la modélisation du comportement de l'hélicoptère était trop éloignée de la réalité.

Pour pallier l'imperfection de la modélisation, S.F.I.M. a effectué une approche "semi-empirique" du problème, pour aboutir à un algorithme de restitution appelé MEDIA (MEthode pour la Détermination des Informations Anémométriques).

Cette méthode complète le brevet de Monsieur DURAND pour lequel l'Etat Français a donné à S.F.I.M. le droit de reproduction (pour les étatiques) et s'appuie sur 2 nouveaux brevets déposés par la S.F.I.M.

2 - APPROCHE DU PROBLEME

2.1 - Données initiales

Des enregistrements de paramètres de vol de l'hélicoptère PUMA du Centre d'Essais en Vol (C.E.V) de Brétigny (France) ont été mis à disposition de la S.F.I.M. par le Service Technique des Télécommunications et Equipements (S.T.T.E) et le C.E.V.

L'hélicoptère utilisé pour ces essais était équipé d'une centrale inertielle à plate-forme fournissant une bonne mesure de la vitesse sol, du roulis, du tangage, du cap et des accélérations.

De plus, l'hélicoptère disposait de capteurs mesurant la position des différentes commandes de vol :

- pas cyclique longitudinal,
- pas cyclique latéral,
- pas collectif,
- pas du rotor anti-couple.

La sortie de ces capteurs était codée et enregistrée sous forme numérique.

Par ailleurs, pour déterminer la vitesse air, une référence de vent de qualité était nécessaire. La référence utilisée était le véhicule "CARAMEL" (CAmion RAdio Mesures ELaborées) développé par le C.E.V. Cette référence de vent originale permet de connaître le vent avec précision (0,5 m/s à 1 sigma) sur chacun des 3 axes sur toute la trajectoire.

Les essais consistaient pour l'hélicoptère, à suivre le véhicule CARAMEL sur un taxiway de la piste de l'aérodrome de Brétigny. De nombreuses passes ont ainsi été effectuées de façon à balayer au mieux le domaine de vol de l'hélicoptère pour les basses vitesses, inférieures à 20 m/s (40 kt).

2.2 - Recherche d'un modèle

L'hypothèse de base est que la vitesse air peut être restituée par une combinaison linéaire de paramètres choisis parmi :

- accélération longitudinale	(a_x)
- accélération latérale	(a_y)
- roulis	(PHI)
- tangage	(THETA)
- pas cyclique longitudinal	(Pc_{yx})
- pas cyclique latéral	(Pc_{yy})
- pas collectif	(P_{col})
- pas du rotor anticouple	(P_{ac})
- une constante	(k)

A partir des différentes passes effectuées au cours d'un même vol et des enregistrements correspondants, plusieurs types de combinaisons linéaires ont été testés afin de déterminer quelle est la loi de restitution qui donne les meilleurs résultats. Le vol choisi est un vol pour lequel le domaine des basses vitesses a été largement exploré.

Ainsi, pour chaque type de combinaison linéaire, les étapes suivantes ont été réalisées (suivant figure 1) :

- identification des coefficients de la combinaison linéaire par méthode des moindres carrés sur vol de référence.

- analyse de l'adéquation du modèle ainsi déterminé avec la réalité par calcul de la moyenne quadratique de l'écart entre la mesure de vitesse air et le modèle.

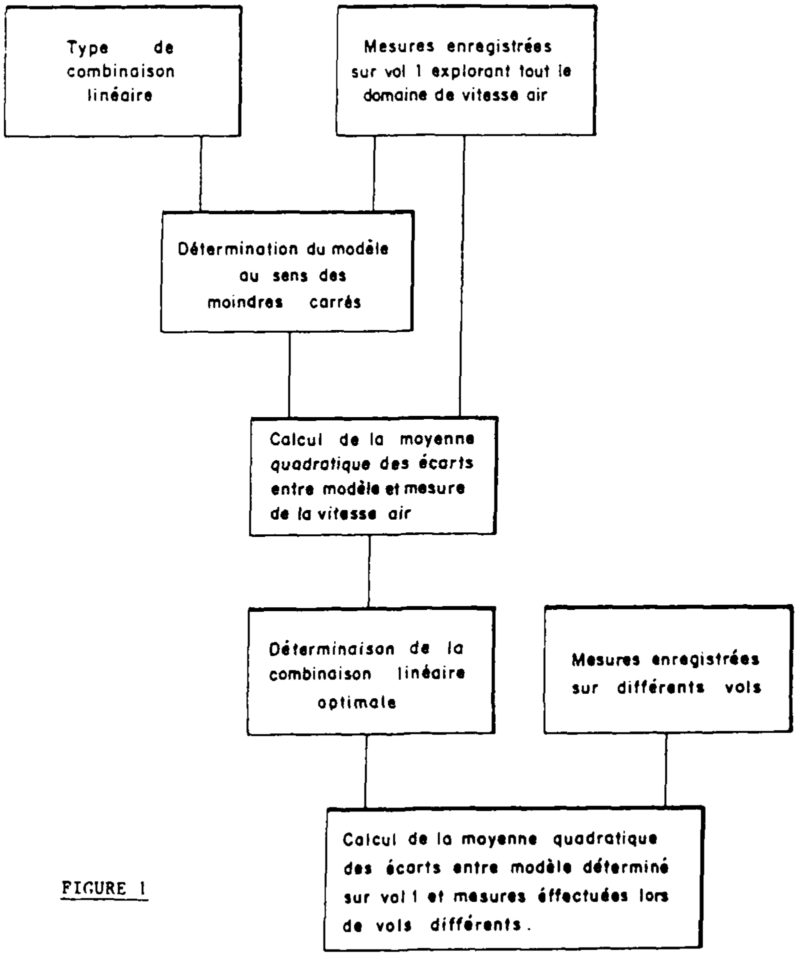

Type de combinaison linéaire
Mesures enregistrées sur vol 1 explorant tout le domaine de vitesse air
Détermination du modèle au sens des moindres carrés
Calcul de la moyenne quadratique des écarts entre modèle et mesure de la vitesse air
Déterminaison de la combinaison linéaire optimale
Mesures enregistrées sur différents vols
Calcul de la moyenne quadratique des écarts entre modèle déterminé sur vol 1 et mesures éffectuées lors de vols différents.

FIGURE 1

A partir de cette analyse, il est alors possible de déterminer la loi qui conduit aux meilleurs résultats sur le vol de référence. L'étape suivante a alors consisté à valider le modèle déterminé à partir du vol de référence, sur des vols effectués par le même hélicoptère dans des conditions de masse et des conditions atmosphériques (force et direction du vent) différentes. Pour ces vols de validation, la moyenne quadratique de l'écart entre la vitesse déterminée par le modèle et celle mesurée expérimentalement a été calculée.

3 - RESULTATS SUR DONNEES ENREGISTREES

3.1 - Détermination de la combinaison linéaire optimale

La figure 2 résume les résultats de la première phase. Il apparaît ainsi que la combinaison n° 7 est celle qui fournit les meilleurs résultats. Les combinaisons linéaires de restitution des vitesses air longitudinale et latérale retenues sont donc :

$$Va_x = k_1.Pc_{yx} + k_2.Pc_{yy} + k_3.a_x + k_4.a_y + k_5.P_{col} + k_6.P_{ac} + k_x$$

$$Va_y = l_1.Pc_{yx} + l_2.Pc_{yy} + l_3.a_x + l_4.a_y + l_5.P_{col} + l_6.P_{ac} + k_y$$

Les valeurs numériques obtenues pour l'hélicoptère utilisé (PUMA) montrent un croisement d'axes important : influence du pas cyclique latéral et de l'accélération latérale sur la vitesse air longitudinale et vice-versa. Ce croisement d'axes est voisin de 45°, à savoir qu'en calculant la vitesse air dans des axes à 45° des axes hélicoptère, le modèle est presque complètement décroisé. Ce phénomène, avec une telle amplitude, n'est pas expliqué théoriquement pour le moment.

Combinaison n°	Nombre de termes	Type de variable	Moyenne quadratique de l'écart en vitesse air longitudinale (m/s)	Moyenne quadratique de l'écart en vitesse air transverse (m/s)
1	3	Pc_{yx}, a_x, k	2,70	4,56
2	3	Pc_{yy}, a_y, k	1,58	1,37
3	4	Pc_{yx}, a_x, P_{col}, k	2,29	4,12
4	4	Pc_{yy}, a_y, P_{col}, k	1,20	1,38
5	5	Pc_{yx}, Pc_{yy}, a_x, a_y, k	0,93	0,98
6	6	Pc_{yx},Pc_{yy},a_x,a_y,P_{col},k	0,73	0,87
7	7	Pc_{yx},Pc_{yy},a_x,a_y,P_{col},P_{ac},k	0,67	0,86
8	7	Pc_{yx},Pc_{yy},PHI,THETA,P_{col},P_{ac},k	0,76	1,12

FIGURE 2

3.2 - Validation du modèle sur des vols différents

Les vols de validation se sont déroulés avec le même hélicoptère, mais il avait alors une masse différente liée a un emport de carburant différent. Par ailleurs, les conditions de vent avaient varié tant en force qu'en direction. Par contre, les moyens d'essais étaient strictement identiques.

Trois vols différents ont été utilisés pour valider le modèle. Les résultats représentent la moyenne quadratique de l'écart entre la vitesse air reconstituée par le modèle et la vitesse air élaborée à partir des mesures de vitesse sol de la centrale inertielle et la mesure de vent fournie par le véhicule CARAMEL.

La figure n° 3 résume les résultats de cette validation. Ces traitements ont montré que dans le domaine des basses vitesses, l'erreur de restitution de la vitesse air est de l'ordre de 1,5 mètres par seconde suivant chaque axe. Le domaine des basses vitesses est considéré comme étant le domaine dans lequel la vitesse longitudinale et la vitesse latérale sont inférieures à 20 m/s.

De plus, il a été ainsi montré que l'utilisation du pas collectif dans la restitution de la vitesse air rend l'algorithme insensible à la masse.

n° de vol	Moyenne quadratique écart vitesse longitudinale (m/s)	Moyenne quadratique écart vitesse latérale (m/s)
Vol de référence	0,67	0,86
Vol n° 1	1,47	2,04
Vol n° 2	1,16	1,28
Vol n° 3	1,58	1,29
Ensemble des vols 1, 2 et 3	1,49	1,69

FIGURE 3

4 - RESTITUTION DE LA VITESSE AIR DANS TOUT LE DOMAINE

4.1 - Transition avec anémométrie classique

Pour le domaine des vitesses longitudinales plus importantes, l'anémométrie classique fournit l'information avec précision. Pour implanter l'algorithme MEDIA dans un matériel opérationnel, la transition entre la vitesse air reconstituée et la vitesse anémométrique doit être effectuée de façon adéquate.

L'algorithme MEDIA tient toutes ses performances dans le domaine des vitesses inférieures à 20 m/s. L'anémométrie classique atteint ces mêmes performances pour des vitesses supérieures à 25 m/s. On a donc choisi de faire une transition linéaire entre 20 et 25 m/s entre la vitesse air longitudinale issue du modèle et celle issue de l'anémométrie classique. Ainsi la formule de restitution de la vitesse air longitudinale est :

$$Va_x = b_1.Va_M + b_2.Va_c$$

où

Va_x est la vitesse air longitudinale calculée

Va_M est la vitesse air longitudinale élaborée par l'algorithme MEDIA

Va_c est la vitesse air longitudinale fournie par le capteur anémométrique

b_1 et b_2 sont deux coefficients dont la valeur est indiquée en figure 4. Cette valeur est fonction de la vitesse air. b_1 et b_2 sont calculés de façon à ce que pour une vitesse air donnée, $b_1 + b_2 = 1$.

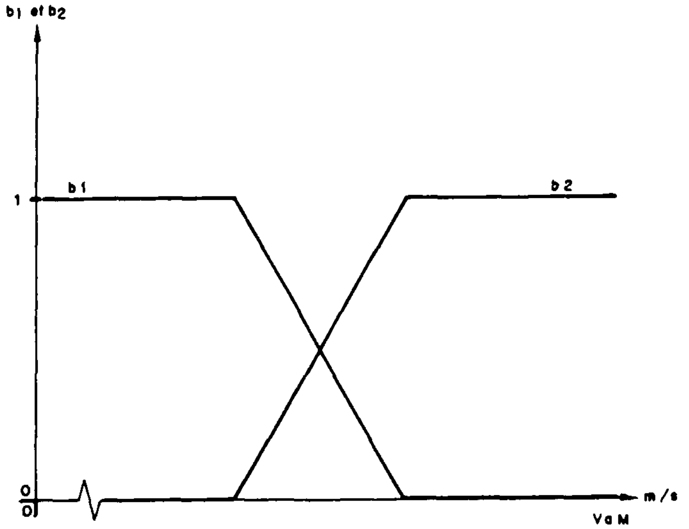


FIGURE 4

4.2 - Algorithme de restitution

L'élaboration de la vitesse air dans le domaine des basses vitesses est très performant lors des phases de vol stabilisées. Par contre, ses performances se dégradent légèrement lors des phases de vol transitoires : vol accéléré, montée, descente, virage.

Pour réduire la sensibilité de la vitesse air restituée, un algorithme de filtrage est mis en oeuvre. Cet algorithme est décrit en figure 5. Il consiste à faire une estimation du vent par comparaison de la vitesse restituée et de la vitesse sol. Ce vent est alors filtré suivant un premier ordre avec une constante de temps de 3 secondes environ. Ce vent filtré est ensuite recomposé avec la vitesse sol pour élaborer une vitesse air optimale.

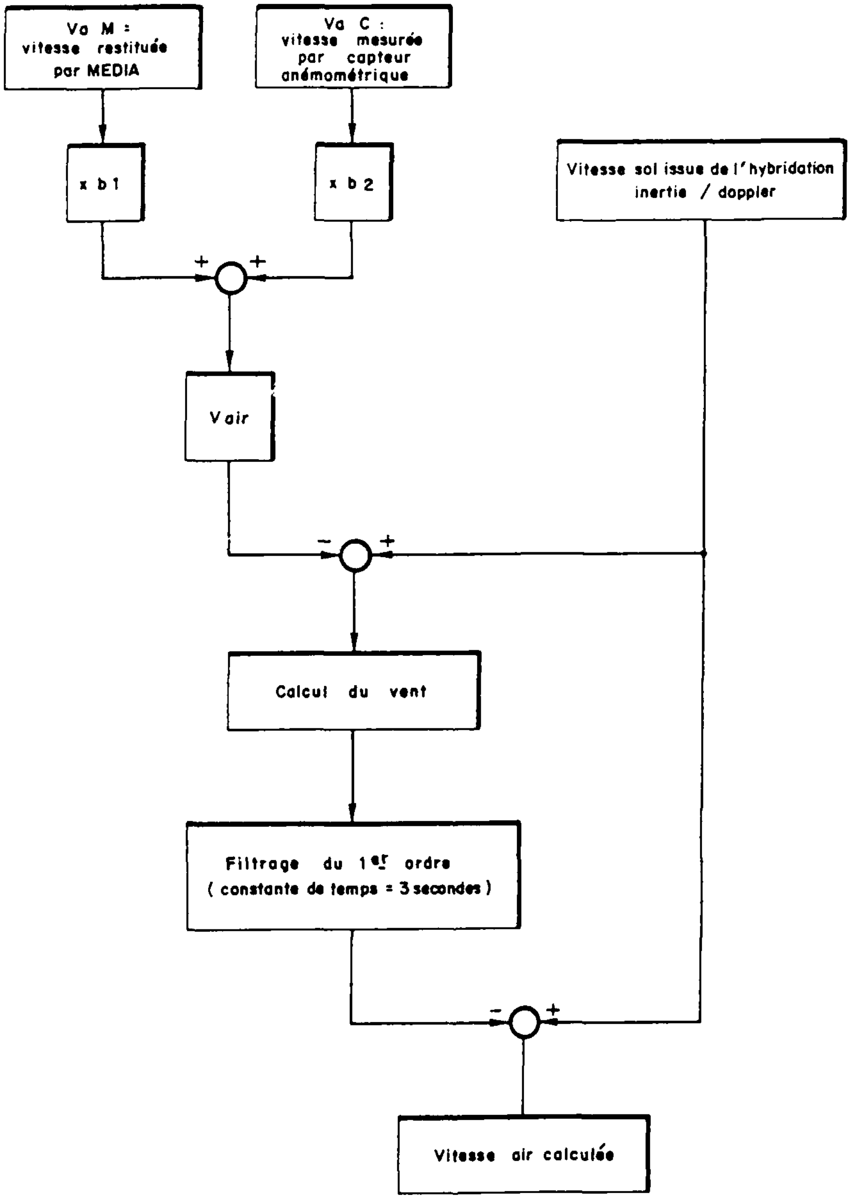

Va M : vitesse restituée par MEDIA
Va C : vitesse mesurée par capteur anémométrique
x b1
x b2
Vitesse sol issue de l'hybridation inertie / doppler
+
+
Vair
-
+
Calcul du vent
Filtrage du 1er ordre (constante de temps = 3 secondes)
-
+
Vitesse air calculée

FIGURE 5

5 - IDENTIFICATION DES COEFFICIENTS DU MODELE DE RESTITUTION DE LA VITESSE AIR

Les coefficients de restitution de la vitesse air sont dépendants :

- de l'hélicoptère lui-même,

- des capteurs de mesure de position des commandes de vol de l'hélicoptère.

Le problème de l'identification de ces coefficients se pose unitairement donc pour chaque hélicoptère.

Il s'agissait donc d'élaborer une méthode d'identification suffisamment simple de mise en oeuvre opérationnelle. La solution développée par la S.F.I..M. permet cette identification sans aucune installation spécifique ni à bord de l'hélicoptère ni au sol. De plus, cette identification se déroule en temps réel et ne nécessite aucun traitement en temps différé ni dans le matériel lui-même ni en centre de calcul.

L'identification des coefficients est effectuée au cours d'un vol spécifique dont la durée est d'environ 20 minutes. Ce vol se décompose en une série de passes à différents caps, à différentes vitesses sol et à différents dérapages. L'ensemble de ces passes est destiné à explorer le domaine des basses vitesses de façon aussi large que possible. Ce vol de calibration a lieu lors de la première mise en oeuvre de l'hélicoptère ou après une intervention importante au niveau du rotor ou d'un des moyens de mesure des commandes des pas. Une fois déterminés, les coefficients sont stockés en mémoire non volatile.

Le processus d'identification s'effectue de façon complètement automatique. A chaque instant, l'algorithme dispose des mesures suivantes :

- accélération longitudinale
- accélération latérale
- pas cyclique longitudinal
- pas cyclique latéral
- pas collectif
- pas du rotor anticouple
- cap
- vitesse sol longitudinale
- vitesse sol latérale.

A partir des mesures, l'algorithme estime par la méthode des moindres carrés les coefficients de la combinaison linéaire restituant la vitesse air et les composantes du vent sur les axes géographiques.

Chacune des composantes du vent est estimée 2 fois :

- par les équations agissant sur l'axe longitudinal
- par les équations agissant sur l'axe latéral.

Cette double estimation permet d'apprécier la qualité de la calibration par comparaison des deux valeurs :

Le principe suppose donc que le vent demeure constant au cours du vol de calibration. L'expérience a montré que les rafales de vent jusqu'à 5 m/s restent cependant acceptables et ne détériorent pas le résultat de la calibration.

La sélection des mesures au cours du vol de calibration est automatique. Cette sélection s'effectue selon deux critères :

- critère de stabilité, afin de ne pas tenir compte des phases transitoires pendant lesquelles l'adéquation du modèle à la réalité est moins performante.

- critère de "conditionnement" du système afin de disposer de mesures suffisamment réparties dans le domaine de vol. Ainsi une mesure n'est prise en compte qu'à la condition de procurer au système de l'information complémentaire pour la détermination des coefficients.

Le principe de l'algorithme de calibration est dessiné en figure 6.

Lorsque le nombre de mesures est suffisant et que le critère de conditionnement a atteint le seuil prédéfini, le traitement des mesures est effectué afin de déterminer la série de coefficients nécessaires à la restitution de la vitesse air.

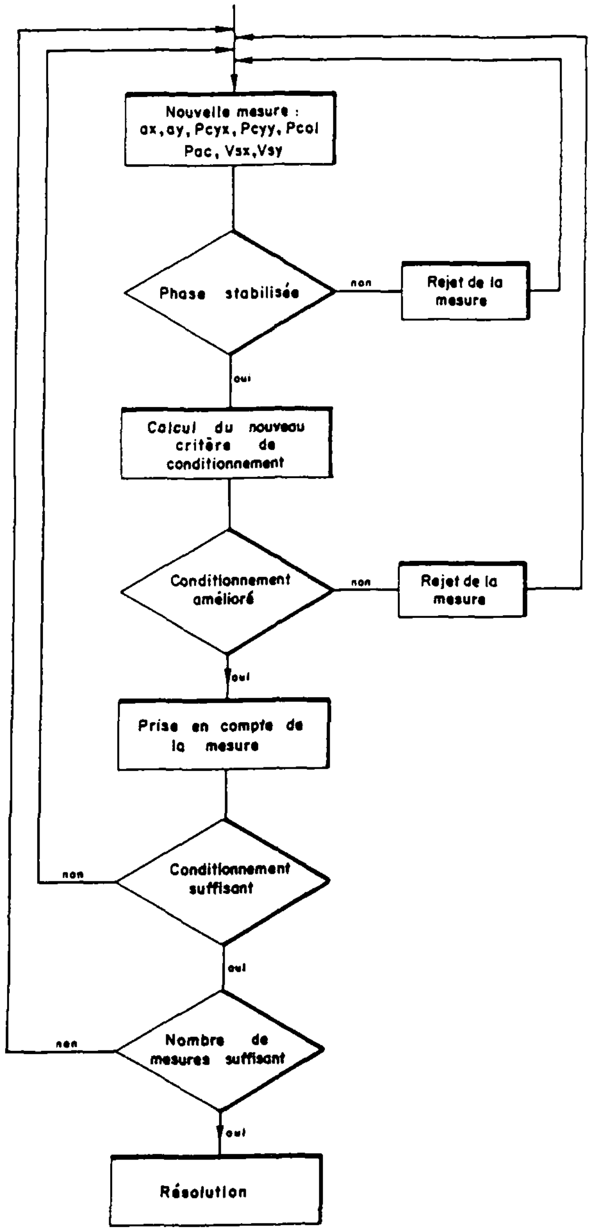


FIGURE 6

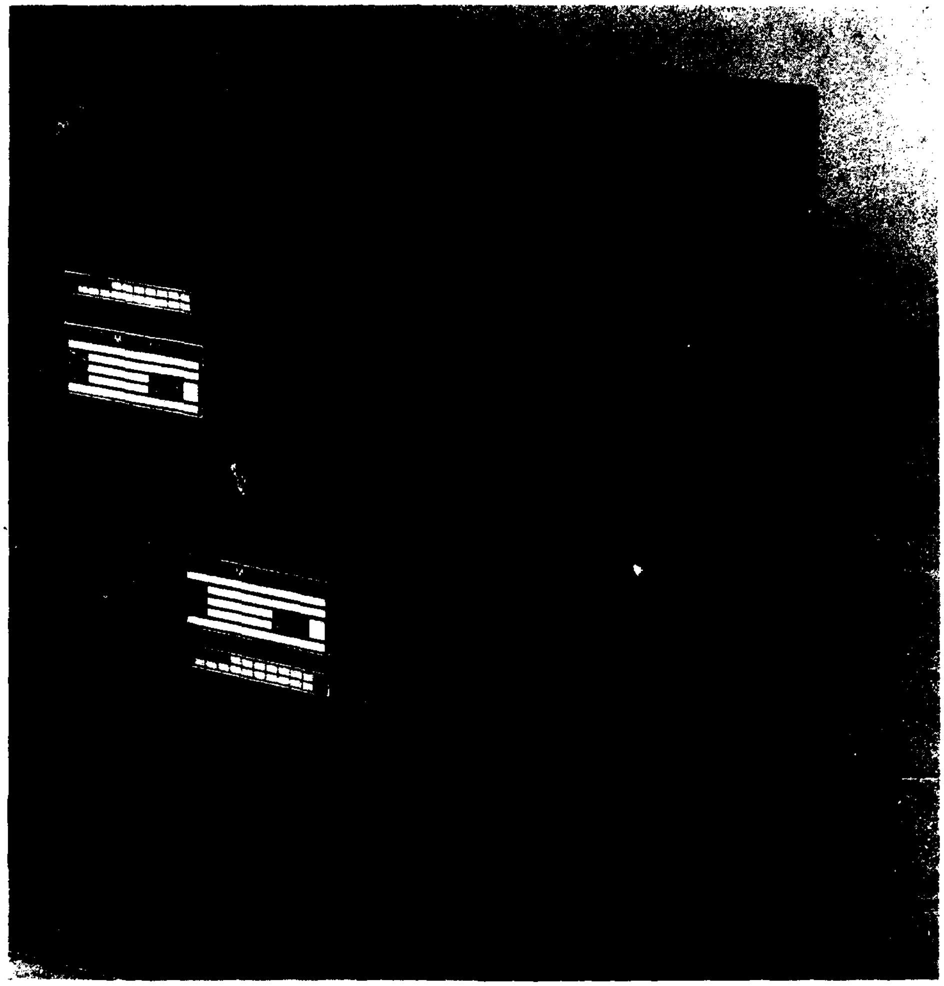
6 - MISE EN OEUVRE DE CES PRINCIPES DANS UNE CENTRALE STIRS 28SH DE SFIM

L'étude précédemment décrite a permis de déterminer le type d'algorithme de restitution de la vitesse air ainsi que le principe de la calibration des coefficients spécifiques à l'hélicoptère. Pour être pleinement opérationnels, ces algorithmes doivent être implantés dans un système disposant déjà des données nécessaires aux algorithmes.

Ces données sont :

- accélérations longitudinale et latérale,
- positions des différentes commandes de pas de l'hélicoptère,
- cap,
- vitesse sol de l'hélicoptère,
- mesure des capteurs anémométriques.

La S.F.I.M. fabrique des centrales de navigation pour hélicoptère, en particulier la STIRS (Strap-down Inertial Reference System) 28SH. Ce système, de par sa conception dispose, de façon interne, des accélérations, du cap et de la vitesse sol issue d'une hybridation entre vitesse inertielle et vitesse Doppler. De plus, les informations des capteurs anémométriques sont acquises par la STIRS 28SH soit sous forme analogique, soit sous forme numérique ; les positions des différentes commandes de pas sont mesurées par des potentiomètres alimentés et codés en numérique par la STIRS. Disposant ainsi de toutes les données nécessaires, les algorithmes de calibration et de restitution de la vitesse air ont été implantés et la STIRS ainsi équipée a été testée en vol au C.E.V. de Brétigny pour évaluer les performances en temps réel des principes retenus.



Centrale STIRS 28SH

7 - RESULTATS D'ESSAIS EN VOL

Le Centre d'Essais en Vol de Brétigny a évalué les algorithmes de restitution de la vitesse air dans la STIRS 28SH lors de 9 vols effectués entre novembre 1987 et janvier 1988.

Deux vols ont été consacrés à la calibration.

Sept vols ont été consacrés à l'évaluation des performances vitesse air. Seules ont été conservées les positions de vol pendant lesquelles la référence de mesure du vent restait dans le domaine de précision de 0,5 m/s en module et de 0,8 degré en direction.

Ces positions de vol représentent 41 passes stabilisées explorant le domaine des basses vitesses (inférieures à 25 m/s) de façon a peu près équirépartie.

Les résultats sont résumés dans le tableau suivant :

	Moyenne de l'erreur (m/s)	Ecart-type de l'erreur (m/s)
Vitesse longitudinale	0,3	1,3
Vitesse latérale	0,5	1,4

Il apparaît que ces résultats sont conformes aux prévisions effectuées grâce aux enregistrements en vol préliminaires. Ainsi aucune différence notable n'a été remarquée entre le comportement en temps différé et en temps réel des algorithmes d'identification et de restitution.

8 - CONCLUSION

L'étude, la mise en oeuvre dans un système opérationnel et les essais en vol de l'algorithme MEDIA ont montré que :

- l'approche du problème donne des résultats satisfaisants,

- l'implantation de cet algorithme dans une centrale de navigation est une solution plaisante dans la mesure où elle ne nécessite pas de liaisons spécifiques entre les différents équipements de l'hélicoptère.

- l'identification en vol des coefficients de l'algorithme sans installation spécifique au sol ou à bord est opérationnellement très attrayante.

- l'insensibilité de l'algorithme de restitution de la vitesse air à la masse de l'hélicoptère par l'utilisation du pas collectif de l'hélicoptère .

Cependant, la susceptibilité de l'algorithme à des conditions de vol différentes n'a pas été étudiée en vol :

- susceptibilité au type d'hélicoptère,
- susceptibilité au centrage de l'hélicoptère,
- susceptibilité de l'algorithme aux phases non stabilisées.

Ces points sont en cours d'étude à la S.F.I.M.

THE DEVELOPMENT OF AN AIRBORNE SYNTHETIC APERTURE RADAR MOTION COMPENSATION SYSTEM

D.F. Liang and D.J. DiFilippo
Defence Research Establishment Ottawa
Shirley Bay
Ottawa, Ontario
Canada K1A 0Z4

ABSTRACT

The Canadian Department of National Defence has developed an airborne Synthetic Aperture Radar Motion Compensation System (SARMCS) for the AN/APS-506 Maritime Search Radar. To attain high quality airborne SAR imagery, the SARMCS must provide accurate information on spurious motion of the antenna phase centre during a SAR scene. This information is used to compensate the radar returns, thereby enhancing the SAR image quality.

The SARMCS is effectively an integrated navigation system consisting of a ring laser gyro inertial navigation system (LTN-91), a Decca doppler radar, a baro-altimeter, and a motion compensation inertial measurement subsystem (MCIMS). This paper describes the hardware and software configurations of the SARMCS implemented on board a Convair 580 research aircraft. In addition, the flight testing approach and some of the simulation and flight test results are presented.

1.0 INTRODUCTION

The Canadian Department of National Defence is developing a high resolution airborne Synthetic Aperture Radar (SAR) mode for the AN/APS-506 radar. To attain high quality airborne SAR imagery, it is essential that very accurate motion compensation be applied to the radar returns to account for deviations of the radar phase center from a smooth reference path, chosen a priori. The main task of the SARMC system is to determine spurious high frequency deviations from the desired motion along the reference path; these displacements are used by the SAR processor to adjust the relative alignment and phase of the radar returns, after which it removes low frequency (quadratic) errors by autofocusing, which amounts to fitting a quadratic phase adjustment to the radar returns across the synthetic aperture to maximize a specified measure of image contrast.

This paper briefly describes the hardware instrumentation and software algorithm design of the SARMCS implemented on board a Convair 580 research aircraft. Some simulation and flight test results are also presented.

2.0 MOTION COMPENSATION REQUIREMENTS

There are two main modes of SAR operation. In spotlight mode, the antenna is aimed at the designated target (Figure 1) or specified coordinates, and wide bandwidth radar pulses are emitted. In the strip-mapping mode, the orientation of the radar boresight is held constant approximately at right angles (Figure 2) to the nominal flight path, thus illuminating a swath to the side of the aircraft.

For airborne SAR processing, the ideal situation is that the radar antenna, mounted on the aircraft, moves along a straight line in space, transmitting and receiving pulses at equally spaced intervals along this path, which forms the synthetic aperture. However, in general, the actual path of the antenna will deviate from the nominal path due to aircraft turbulence, autopilot inaccuracies, etc. These spurious motions, if uncompensated, can severely degrade the SAR image. The function of the motion compensation system is to sense the antenna motion, and compute the deviations between the actual path and the nominal path. This information is then used to correct the phase of the radar returns so that, ultimately, as far as the radar processor is concerned, the pulses look like they were emitted at these ideal points along the nominal track.

2.1 Performance Accuracy Requirements

Since this project is in support of the development of a SAR radar capability, the performance requirement for motion compensation has been specified in terms of power spectral density (PSD) of the tolerable error in measurement of the displacement of the antenna phase center (Figure 3). The displacement error spectrum has been divided into two components. The portion of the PSD below A Hz lies in the "Don't-Care Region". This contains the components of the displacement error which have characteristic times longer than the maximum aperture time of T seconds (A Hz = 1/T seconds). Displacement error components having frequencies above A Hz must be controlled by the motion compensation system. An estimate of the RMS magnitude of the allowable motion compensation error is obtained as:

$\delta R_r = 0.33$ mm (RMS).

This is the residual displacement error along the line-of-sight (LOS) to the target not including contributions from constant and linear components of range error, which have no effect on image quality, or quadratic components which are removed by the autofocusing.

The displacement error components above A Hz affect the contrast of the SAR image while those below A Hz degrade resolution and produce geometric distortions of the image. Although the lower frequency components within the "Don't-Care" region may affect the SAR image quality, this error component is to be controlled through the autofocusing algorithm. Therefore the SARMC project only considers error sources above the A Hz region. It should be noted that during the early phase of the design study, it was felt that the performance requirement of the order of a millimeter would not likely be achieved in a practical situation as there are a large number of practical design variables which would easily overwhelm this level of accuracy requirement.

2.2 Analysis of Error Sources

In view of the extremely stringent accuracy requirement, primary design consideration was concentrated on estimating the magnitude of the contribution of major system error sources to the residual range error. The following error sources were considered:

- uncompensated phase center motion due to antenna pointing errors
- computational and related errors arising from the imperfect solution of the strapdown motion compensation equations
- gyro and accelerometer errors of the strapdown IMU
- attitude errors of the strapdown navigator due, in part, to imperfect transfer alignment from the master INS

The analysis indicated that the principal sources of error in computing the LOS change in range are the mislevels of the strapdown navigator, the error in determining the initial depression angle of the target LOS, computational errors in navigation and targetting algorithms, error in the measured relative azimuth of the strapdown IMU and the radar antenna boresight, and errors in the accelerometers and gyros. From the results obtained for the required aperture under conditions of medium turbulence, the error budget of Table 1 was drawn up.

Table 1: Error Budget

Error Source	Max. Contribution to	
	RMS (mm)	%
1. Sensor errors		
-accelerometer	0.02	0
-gyro	0.09	8
RSS TOTAL (sensors)	0.09	8
2. Computational errors	0.10	9
3. S/D heading error	0.02	0
4. S/D mislevels	0.18	30
5. S/D azimuth alignment	0.05	2
6. Azimuth angle encoder	0.05	2
7. Interpolation errors	0.10	9
8. Initial depression angle	0.15	20
9. Contingency	0.15	20
RSS TOTAL (all)	0.33	100

3.0 SARMC SYSTEM CONFIGURATION

The SARMC system instrumentation that was designed and installed on board the National Aeronautical Establishment (NAE) Convair 580 aircraft is shown in Figure 4.

3.1 Sensor Description

The principal motion compensation sensor is the MCIMS (Motion Compensation Inertial Measurement Subsystem), which is a Strapdown (S/D) Inertial Measurement Unit (IMU) located in the nose of the aircraft with the SAR antenna. It provides raw measurements of the angular rates and linear accelerations of the SAR antenna at a 50 Hz rate in the form of angular and velocity increments. Developed by Lear Siegler, the MCIMS is actually a two box configuration consisting of an instrument package and an electronics package. The instrument package contains two dry tuned-rotor gyroscopes and three accelerometers whose performance specifications are indicated in Table 2. The package is 8.6 cm wide by 8.9 cm high by 12.7 cm deep, and together with its mounting tray, weighs only 1.9 Kg. It is mounted on the antenna ring gear housing (Figure 5) which, along with the rest of the antenna, is nominally stabilized in roll and pitch by servo loops. This location places the sensor package as close as possible to the antenna phase centre (about 50 cm away) and ensures that there is minimal relative motion between the phase centre and sensor package. The second box of the MCIMS, the electronics unit, contains all the electronics required to operate the sensor package, including power supplies and sensor rebalance loops. The electronics package also performs digitization of the analog sensor signals and compensation of the sensor outputs for factory-calibrated errors. This box is installed at a site located aft of the aircraft nose bulkhead.

The master INS is located about 6 metres away from the MCIMS, near the centre of gravity of the aircraft. It is an off-the-shelf LTN-91, employing ring laser gyroscopes and providing full three-axis inertial navigation with nominal 2 nm/hr (95%) accuracy. Standard outputs from the system, which include 3-D position, velocity, attitude and heading, are provided on an ARINC 429 bus at rates varying from 8 Hz to 64Hz.

The Decca 72 Doppler velocity sensor used in the SARMCS employs a strapdown lambda 3-beam configuration to measure aircraft velocity in aircraft body coordinates relative to the terrain surface. The primary error in the measurement of the forward velocity component is a scale factor type error in the order of 1-2% depending upon the terrain. The error in measuring the lateral velocity component principally results from an antenna boresight azimuth misalignment which is typically less than 2^{o}. For overwater operation, the ability of the Doppler sensor to measure velocity with respect to the earth is degraded because of shifts in Doppler frequency due to sea currents and waves.

The SARMCS also utilizes information from air data sensors, which includes a Digiquartz pressure transducer to measure static air pressure and a Rosemount air temperature probe mounted on the wingtip to measure outside air temperature. This information is needed to compute baroaltitude.

3.2 Signal Processor Overview

The simplified block diagram for the SARMCS Phase II configuration is shown in Figure 6.

Velocity and angular increments from the MCIMS are processed in a S/D navigator algorithm which mechanizes the navigation equations in a wander azimuth frame. Since the MCIMS is mounted on the antenna ring gear housing, the algorithm fundamentally computes antenna attitude and heading, and velocity and position of a particular point at the MCIMS location. A lever arm correction is then applied within the navigator to obtain estimates of the velocity and position of the antenna phase centre. Static pressure and outside air temperature from the air data sensors are inputs to a Blanchard algorithm for a baroaltitude calculation. [1] Baroaltitude is used in a classical third-order damping loop to stabilize the S/D vertical channel.

The S/D navigator outputs are then provided to a targetting algorithm. This algorithm uses the S/D information, in conjunction with scene initialization information from the radar subsystem, to calculate the LOS displacement of the antenna phase centre relative to the LOS displacement for a nominal smooth path trajectory. From these relative LOS displacements, the targetting algorithm then generates appropriate phase corrections that can be used to compensate the phase of the radar returns for the spurious phase centre motion. Additional motion corrections computed by the targetting algorithm include range gate corrections, which are implemented to keep the target energy in the same range cells over the scene, and cross-range corrections, which are used to resample the radar data in order to maintain constant spacing between radar samples.

The outputs of the S/D navigator, along with information from the master INS and Doppler velocity sensor, are also supplied to a Kalman filter which estimates various system and instrument errors associated with these devices. The filter-estimated S/D navigation and MCIMS instrument errors are fed back to the S/D navigator algorithm and used there to correct the relevant quantities in a closed loop fashion. The net result of this Kalman filter integration is that alignment of a Doppler-damped master INS is continually transferred to the S/D navigator. Filter diagnostic data and corrected master navigation data are provided as secondary SARMCS outputs.

4.0 SARMCS Simulation Software

The SARMCS simulation software is divided into 3 separate packages:

a. The Data Synthesis Package generates realistic synthetic data from the system error models which include

- strapdown IMU model
- Master INS model
- Doppler radar model
- Atmospheric pressure and temperature models

b. The Data Processing Package can process both synthetic and real sensor data. It implements the motion compensation processor of Figure 6, which includes

- strapdown navigation and sensor compensation algorithms
- barometric altitude algorithm
- Kalman filter
- targetting algorithms

c. The Evaluation Package evaluates the performance of the processing package by comparing computed master and strapdown navigator positions, velocities and attitudes with corresponding accurate reference data generated by the Synthesis Package.

5.0 KALMAN FILTER DESIGN

A baseline 35 state Kalman filter was developed to indicate the best level of achievable performance. The filter is mechanized using Bierman's U-D factorized formulation 2 . The filter structure is implemented in such a way that arbitrary subsets of the full error state vector may be selected for a particular run by specifying values for input parameter tables. Similarly any subset of measurements may be selected. This results in a very flexible design tool. An error control routine uses position, velocity and misalignment error state values to correct position, velocity and attitude estimates of the strapdown navigator after each filter update, unless the update occurs during a SAR window. The corresponding error states are zeroed after executing error control.

The complete set of error states is specified in Table 3. The measurements used to obtain the baseline filter simulation results presented in the next section are given in Table 4. Additional measurements were mechanized, namely: master/strapdown velocity matching, as well as 2-dimensional versions of all the position and velocity matching measurements, constructed in the level components of the wander azimuth coordinates.

5.1 Sub-Optimal Kalman Filter Design

For practical implementation, a suboptimal 21 state Kalman filter was designed. The error state vector is described in Table 5.

The error dynamics of the system states in subvectors x_m and x_s are modelled in the Kalman filter using the general propagation equations for a ϕ angle (true frame) inertial error formulation [3] resolved in a wander azimuth frame. The augumenting states in x_{mi}, x_d, and x_{si}, which represent time-correlated errors in the various instruments, are modelled as first-order Markov processes.

The modelling of relative S/D system errors instead of absolute S/D system errors in the state vector is a design decision that is motivated by several considerations. First, from a theoretical viewpoint, this is an appropriate choice because measurements constructed by comparing information from two systems with the same error dynamics only allow observability of the relative error between the two systems. A practical motivation for modelling relative S/D errors is that for this case where S/D instrument errors are expected to be much larger than master INS instrument errors, it can be shown that the estimation of x_s and x_{si} is essentially decoupled from the estimation of x_m, x_{mi} and x_d, in the sense that no significant correlation develops between these two sets of subvectors during Kalman filter operation. This behaviour is mathematically equivalent to having two independent Kalman filters, one of which accomplishes transfer-of-alignment from the master to the S/D platform while the other performs Doppler-damping of master errors. This is a robust configuration in that the effects of slight mismodelling of the lower quality S/D IMU in the Kalman filter cannot feed back through the velocity matching measurements to corrupt the estimation of master system errors.

One feature of the SARMCS Kalman filter that noticeably distinguishes it from a navigation-type Kalman filter is the absence of measurements that bound the inertial position error. This is a direct consequence of tailoring the Kalman filter for the specific task of performing accurate S/D platform alignment. The presence of S/D position errors have only a relatively weak effect on the buildup of platform misalignments and velocity errors, so there is no need to accurately estimate them in the SARMCS. In fact, S/D position states are dropped altogether; the use of the angle error formulation for the Kalman filter inertial error models conveniently allows this to be done without impacting on the filter's ability to estimate S/D velocity errors and platform misalignments from the velocity matching measurements.

It is worthy to note that the choice of augmenting states in the SARMCS Kalman filter is based primarily on the criterion of observability. The states in x_{mi}, x_d and x_{si} represent only those significant instrument errors that are separately observable with the given measurements.

5.2 Measurement Modelling

The Kalman filter processes a set of four measurements every 10 seconds to update the state vector and associated error covariance matrix. Two master-Doppler velocity matching measurements are formed from the x and y components of the following vector velocity difference, coordinatized in the aircraft body frame:

$$\Delta V_{md} = V_m - V_d + \omega x L_{md} \qquad (1)$$

where V_m is master indicated velocity, V_d is Doppler indicated velocity, ω is aircraft angular rate and L_{md} is the lever arm from the master INS to the Doppler antenna. The other two measurements are master-S/D velocity matching measurements calculated as the x and y components of the following vector velocity difference, coordinatized in the wander azimuth frame:

$$\Delta V_{ms} = V_m - V_s + \omega x L_{ms} \qquad (2)$$

where V_s is S/D indicated velocity, L_{ms} is the lever arm from the master INS to the antenna-mounted IMU and V_m and ω are as defined for (1). The Kalman filter measurement model is derived by perturbing (1) and (2) and expressing the results in terms of the modelled error states.

6.0 SARMC SIMULATION AND FLIGHT TEST VERIFICATION

The mission profile for the simulation is shown in Figure 7. It is consistent with the flight tolerance limits of the CV 580 research aircraft. It involves an initial climb to an altitude of 1000 metres, followed by a racetrack manoeuvre after 10 minutes and an s-turn 20 minutes after takeoff. This is followed by a period of nominally straight and level flight during which 23 SAR apertures are simulated. About 1 hour after takeoff, a second s-turn is carried out to control the strapdown navigator heading error. This is followed by another section of straight and level flight during which 9 more SAR apertures are simulated. All simulation results are obtained using the baseline Kalman filter design unless otherwise stated.

6.1 Simulation Results

Figures 8 to 10 show the north velocity, roll and heading errors of the strapdown navigator, together with RMS values computed from the Kalman filter error covariance. The pitch error has similar characteristics to those of the roll error. Notice that the 1σ bound of the roll error conforms well with the single run error trace. In Figure 9, the heading error RMS is reduced to the level of master heading error after the second S-turn.

Figures 11 and 12 show the LOS displacement error before and after autofocusing. The performance is well within the stated requirements.

Simulation results indicating the performance of the suboptimal 21 state filter are shown in Figure 13. This plot depicts the roll errors of the strapdown navigator, as well as the filter predicted 1σ value. From comparison of Figure 13 to Figure 9, the performance of the two filters is quite similar except for brief periods during aircraft manoeuvres.

6.2 Flight Testing Philosophy

The philosophy for flight testing the SARMCS involves validating the correct operation of subsystem configurations which increase in complexity until the complete configuration is attained. This type of approach provides a systematic method for detecting and isolating unexpected error sources in the hardware and/or software functions. There are five sequential steps in this flight test plan:

1) evaluate the performance of the Master/baro subsystem.

2) evaluate the performance of the MCIMS/baro subsystem,

3) evaluate the performance of the Master/Doppler/baro/ subsystem,

4) evaluate the performance of the Master/Doppler/baro/MCIMS subsystem,

5) evaluate the performance of the full SARMCS with all SARMCS sensors utilized.

In the first four steps, the subsystems are tested to verify that their velocity accuracies are consistent with expected values predicted by the earlier simulations. A flight reference system (FRS) developed by DREO is used to provide the "truth" data for evaluation of these subsystems [4]. The FRS employs an extended Kalman filter which optimally integrates precision microwave ranges from a Del Norte Trisponder system with information from an LTN-91 inertial navigation system. For the FRS, three ground transponders are positioned at surveyed locations. They provide a rectangular coverage area of width 45 kilometres and length 130 kilometres within which at least two ranges with good geometry are received by the aircraft. Under these conditions, the FRS provides continuous aircraft velocity information accurate to 0.1 metres/second and position information accurate to 10 metres. This accuracy is sufficient for evaluating these subsystems since velocity errors in the order of 1 metre/second are expected.

After it has been verified that the various subconfigurations are operating properly, the final step involves testing the full SARMCS configuration by applying motion corrections computed by the SARMCS system to spotlight SAR data. The extent to which the SAR image is enhanced is the ultimate indication of the performance of the SARMCS.

6.3 Flight Test Results - System Performance

The flight test results presented here were obtained from processing recorded data acquired on a representative flight for which reference flight trajectory data was also available. Since the long term accuracy of the FRS is determined by the precise ranges, the reference data is considered to be independent of the SARMCS Kalman filter outputs. The flight profile roughly consisted of seven parallel straight legs with length varying from 75 Km to 125 Km. Aircraft speed was 100 m/s and altitude was 1200 m.

An important set of variables that provide a good diagnosis of the Kalman filter behaviour are the Kalman filter residuals associated with the measurements constructed at each update time. In Figures 14(a) and 14(b), the solid traces indicate the sequence of measurement residuals for the X master-Doppler velocity matching measurement and X master-S/D velocity matching measurement respectively while the dashed lines show the filter-predicted rms for these residuals. If a measurement residual exceeds its predicted rms by more than a factor of three, the measurement is assumed to be bad, and it is rejected. Gaps in the plot of Figure 14(a) occur during turns when master-Doppler measurements are disabled because of degraded Doppler accuracy. Also, during turns, the predicted rms for the master-S/D measurement residuals increases to account for the effects of dynamically induced errors in the S/D IMU. It is observed from these plots that the actual rms of each residual set is generally consistent with the filter-predicted value within a factor of two. This is an indication that the Kalman filter's error models are valid representations of "real-world" error behaviour. The equivalent plots for the Y components of the two measurement types are similar to the X component plots, except that both the predicted and actual rms levels of the Y master-Doppler residuals are higher, as expected from Doppler radar operation.

The ability of the SARMCS Kalman filter to Doppler-damp platform tilts in the master INS was directly evaluated. In Figure 15, the solid line indicates the total platform tilt of the Doppler-damped master INS over the course of the flight, while the dashed line shows the total platform tilt of the undamped master for comparison. The traces consist of data points plotted at 10 second intervals where each data point is the root sum square (rss) of the roll and pitch errors determined at that point in time by differencing the roll and pitch outputs of the damped or undamped master INS with the reference roll and pitch provided by the FRS to an accuracy of 0.1 arc minutes rms. For nominally zero aircraft pitch, these roll and pitch errors are essentially equal in magnitude to the platform tilt components. If the data up to t=2000 seconds are ignored to allow for the Kalman filter settling time, the rms of the platform tilt calculated over the remaining data points is 0.19 arc minutes for the Doppler-damped case and 0.40 arc minutes for the undamped case. The filter-predicted rms of the damped total platform tilt for steady-state filter conditions after t=2000 seconds is 0.22 arc minutes which is in close agreement with the calculated rms of the actual damped tilt. These results indicate that the Kalman filter exhibits well-tuned behaviour and is successful in using the Doppler information to reduce master platform tilts. It should be noted that this was an overland flight; for an overwater flight, Doppler-damping performance can be degraded because of added Doppler errors.

A secondary function of the Kalman filter is to transfer azimuth alignment from the master to S/D platform. Figure 16 is a plot of S/D indicated heading differenced with master indicated heading. The large "spikes" in the plot are actual heading differences between the master INS and the antenna-mounted S/D IMU that result when large impulsive torques are applied to the slightly compliant antenna structure by the antenna servo loops while the radar is in a sector sweep mode. The slower trends in the heading difference result from S/D platform azimuth mi:alignment with respect to the master platform. The plot shows that such misalignments are mostly less than 10 arc min during the flight. This is a good indication that the Kalman filter can adequately transfer azimuth alignment to the S/D platform.

6.4 Flight Test Results - SAR Imagary

Figures 17 to 20 indicate the performance of the SARMCS using the suboptimal Kalman filter. Figure 17 shows t.ie motion compensated SAR image of essentially a point scatterer. The target is a satellite receiving station's 10 meter reflector-type antenna, equipped with a dual frequency S/X-band feed. The X-band feed was short circuited in order to provide a strong reflection from the antenna. In this type of display, the vertical scale is signal amplitude, one horizontal scale is distance along the radar line-of-sight, and the other horizontal scale is distance perpendicular to the radar line-of-sight. Theoretically, the SAR image for a point target should be one sharp peak, which is fairly close to what is being achieved in this image.

Figure 18 shows the same radar data but processed without motion compensation. Here, it is apparent that the effect of spurious uncompensated aircraft motion is to cause energy from the main peak to spill out into side lobes. The implication of this for a more typical image containing many point targets is that weaker targets in the vicinity of stronger ones might be completely obscured by this sidelobe energy; in optical terms, the resulting image would be described as having poor contrast.

Figure 19 shows a motion-compensated SAR strip map image taken of the Sudbury area in northern Ontario. The strip is about 650 metres wide and 1500 metres long. The area is near the nickel smelter in Sudbury that refines the ore from the surrounding mines. The dim lines are roads, and the brighter ones are railway tracks. The very bright lines are pipes that lead into a cluster of five circular storage tanks; the radar reflection off the circular edge of one of the tanks can be clearly seen in the image.

Again, Figure 20 shows the same piece of radar data but processed without motion corrections. The image is initially focused but then starts to smear out as the aircraft deviates from the nominal straight line track. In this case, the deviation is a result of a 2 degree change in aircraft heading. The pipelines can barely be distinguished in this image.

7.0 CONCLUSION

The future prospects for the SARMC system look very promising. The SARMCS development, as well as the entire spotlight SAR development is on schedule to be completed in time for the CP-140 radar update in the mid-1990's. Also, the SAR motion compensation technology has significant potential for commercial SAR applications such as ice surve:llance, and geological and hydrological mapping.

REFERENCES

1. Blanchard, R.L., "A New Algorithm for Computing Inertial Altitude and Vertical Velocity", IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-7, No. 6, November 1971, pp. 1143-1146.

2. G.J. Bierman, Factorization Methods for Discrete Sequential Estimation, New York: Academic Press, 1977.

3. D.O. Benson, "A Comparison of Two Approaches to Pure Inertial and Doppler-Inertial Error Analysis," IEEE Trans. on Aerospace and Electronics Systems Vol. AES-11, No 4, July 1975, pp. 447-455.

4. DiFilippo, D.J. and B. Leach, "A Precise Flight Reference System for Evaluating Airborne Navigation Aids", to be published

MCIMS Gyro Performance Characteristics	
Characteristic	**Performance**
Range	150 °/s
Scale Factor Repeatability	150 ppm (1σ)
Scale Factor Linearity	150 ppm (1σ)
Bias Repeatability	0.2 °/hr (1σ)
G-sensitive Drift Repeatability	0.2 °/hr/g (1σ)
Axes Nonorthogonality	20 arc sec (1σ)
Bandwidth (3 db)	65 Hz
Output Quantization	3.3 arc sec

MCIMS Accelerometer Performance Characteristics	
Characteristic	**Performance**
Range	10 g
Scale Factor Repeatability	200 ppm
Bias Repeatability	100 μg (1σ)
Axes Nonorthogonality	20 arc sec (1σ)
Bandwidth (3 db)	250 Hz
Output Quantization	2.5 mm/s

TABLE 2. PERFORMANCE CHARACTERISTICS OF MCIMS INSTRUMENTS

Type of Error State	Description
master inertial	x,y,z position errors[1] x,y,z velocity errors[1] x,y,z platform misalignment angles[1]
S/D inertial	x,y,z position errors[2] x,y,z velocity errors[2] x,y,z platform misalignment angles[2]
master exponentially correlated	z accelerometer bias[1] x,y,z gyro biases[1]
S/D exponentially correlated	x,y,z accelerometer biases[3] x,y,z gyro biases[3]
baro & Doppler exponentially correlated	baro altimeter bias[1] x,y sea bias states[1] x Doppler scale factor error[4] y,z Doppler boresight errors[4] Doppler carrier drift

[1] master w. a. coords.

[2] strapdown w. a. coords.

[3] strapdown body coords.

[4] aircraft body coords.

TABLE 3. KALMAN FILTER ERROR STATES

Master-Doppler velocity matching (Doppler beam coordinates)

$$C_b^d \left[C_w^b V_m^w - V_d^b + \omega_b^b \times L_{md} \right]$$

V_m^w	computed master velocity (w. a. coords.)
V_d^b	Doppler velocity (aircraft body coords.)
ω_b^b	aircraft angular rate
L_{md}^b	lever arm from master INS to Doppler radar
C_w^b	master w.a.-to-body DCM
C_b^d	body-to-beam DCM

Master-strapdown position matching (body coordinates)

$$C_g^b \left[R_s^g - R_m^g \right] - L_{ms}^b$$

R_m^g	master geographic position
R_s^g	strapdown geographic position
L_{ms}^b	lever arm from master to strapdown
C_g^b	geographic-to-body DCM

Master-baro position matching (master w.a. coordinates)

$$h_m - h_b$$

h_m	master altitude
h_b	barometric altitude

TABLE 4. ERROR MEASUREMENTS

Sub-vector	Description	Frame
x_m	x,y master position errors x,y master velocity errors x,y,z master platform misalignments	W.A.[1] W.A. W.A.
x_{mi}	x,y master accelerometer biases	A/C[2]
x_d	Doppler forward scale factor error Doppler azimuth boresight error x,y sea biases	A/C A/C W.A.
x_s	x,y S/D velocity errors (relative to master) x,y,z S/D platform misalignments (relative to master)	W.A. W.A.
x_{si}	x,y,z S/D gyro biases	S/D[3]

[1] Wander azimuth frame:x,y-level,z-vertical
[2] Aircraft frame:x-forward,y-right wing,z-down
[3] S/D IMU body frame:x-forward,y-transverse,z-down

TABLE 5. SUBOPTIMAL FILTER ERROR STATES

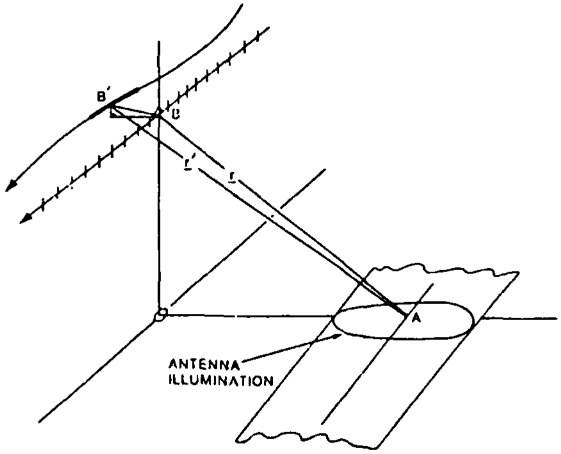


FIGURE 1: SPOTLIGHT SAR GEOMETRY

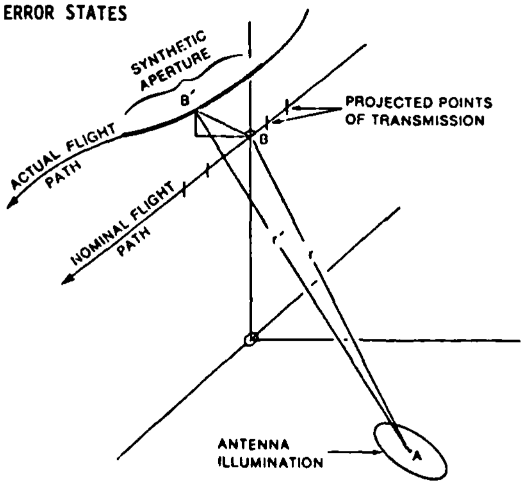


FIGURE 2: STRIPMAPPING SAR GEOMETRY

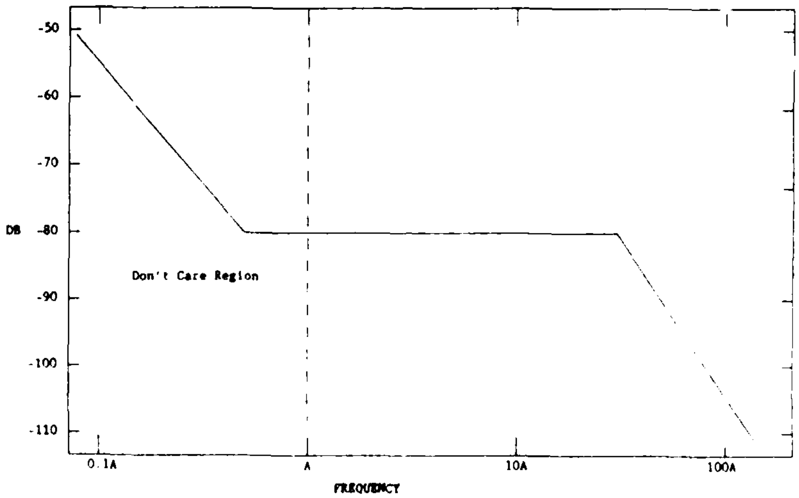


FIGURE 3: NORMALIZED POWER SPECTRAL DENSITY OF TOLERABLE ANTENNA PHASE CENTRE DISPLACEMENT ERROR

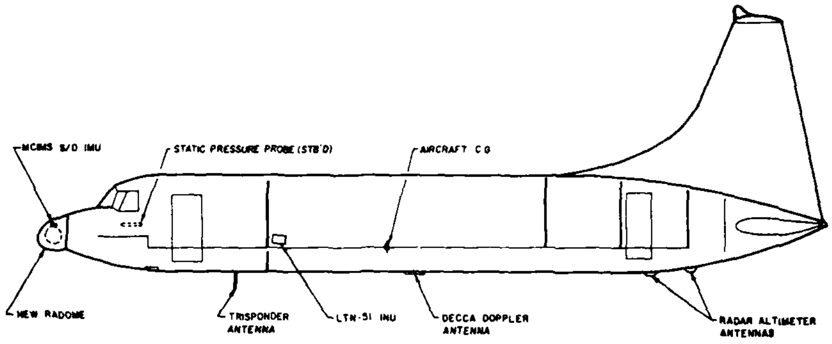


FIGURE 4: LOCATION OF SARMC INSTRUMENTATION ON THE NAE CONVAIR 580

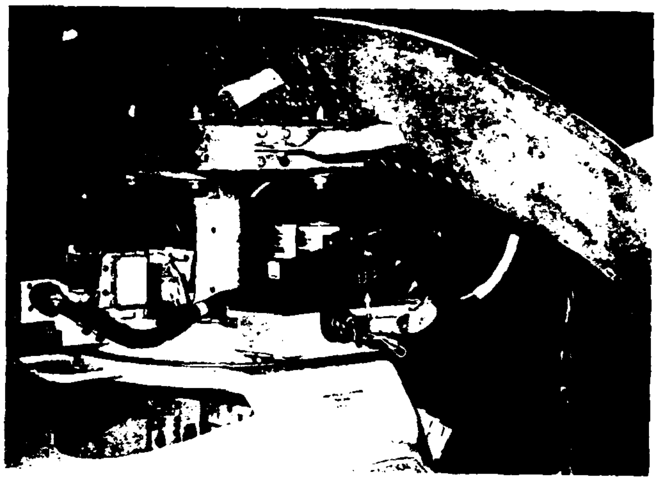


FIGURE 5: MCIMS MOUNTING ON SAR·ANTENNA.RING GEAR

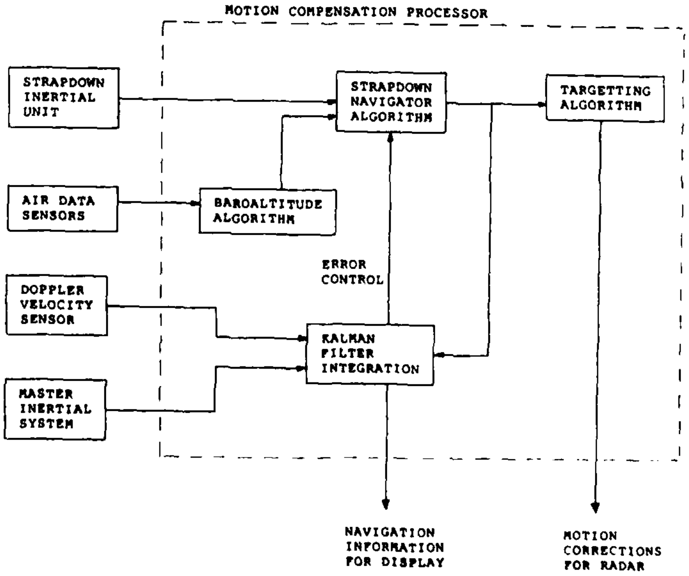

MOTION COMPENSATION PROCESSOR
STRAPDOWN INERTIAL UNIT
AIR DATA SENSORS
DOPPLER VELOCITY SENSOR
MASTER INERTIAL SYSTEM
STRAPDOWN NAVIGATOR ALGORITHM
BAROALTITUDE ALGORITHM
ERROR CONTROL
KALMAN FILTER INTEGRATION
TARGETTING ALGORITHM
NAVIGATION INFORMATION FOR DISPLAY
MOTION CORRECTIONS FOR RADAR

FIGURE 6: BLOCK DIAGRAM OF SARMC SYSTEM

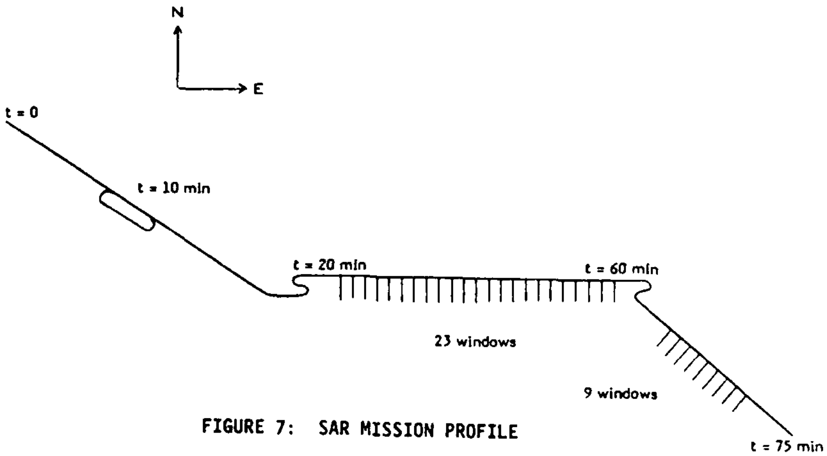


FIGURE 7: SAR MISSION PROFILE

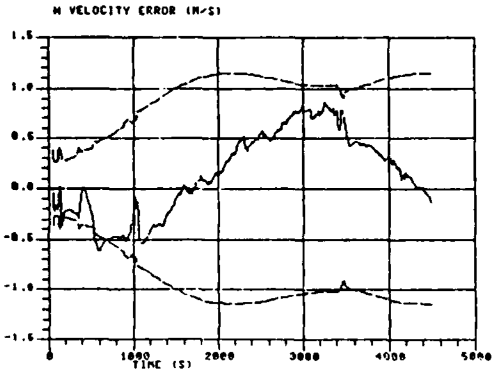


FIGURE 8: NORTH VELOCITY ERROR & RMS FOR STRAPDOWN NAVIGATOR

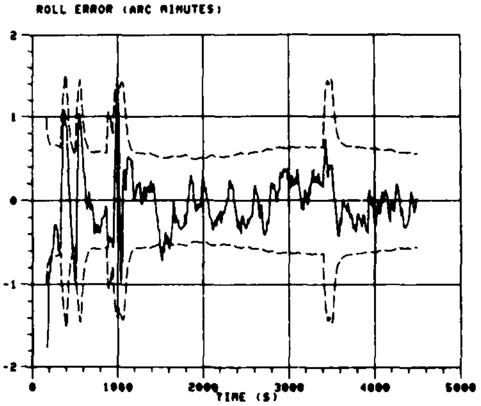


FIGURE 9: ROLL ERROR & RMS FOR STRAPDOWN NAVIGATOR

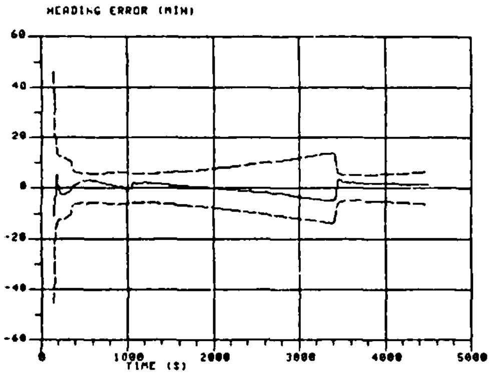


FIGURE 10: HEADING ERROR & RMS FOR STRAPDOWN NAVIGATOR

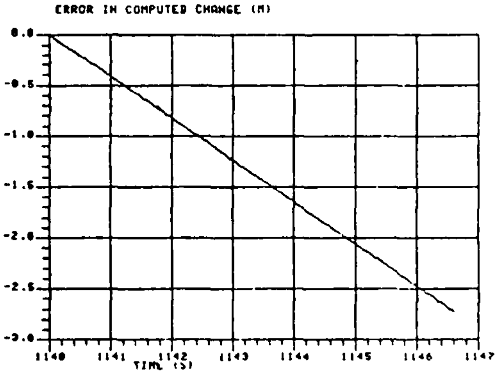


FIGURE 11: LOS DISPLACEMENT ERROR BEFORE AUTOFOCUSING

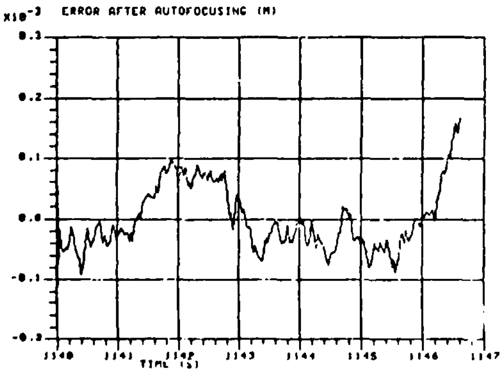


FIGURE 12: LOS DISPLACEMENT ERROR AFTER AUTOFOCUSING

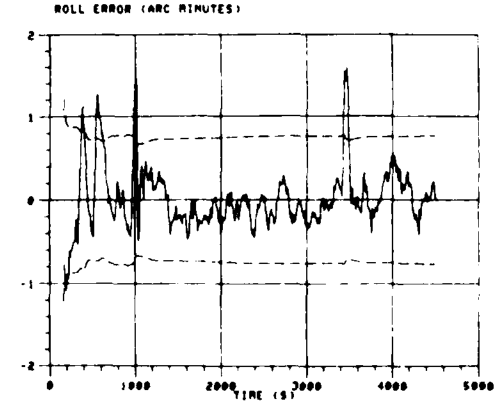
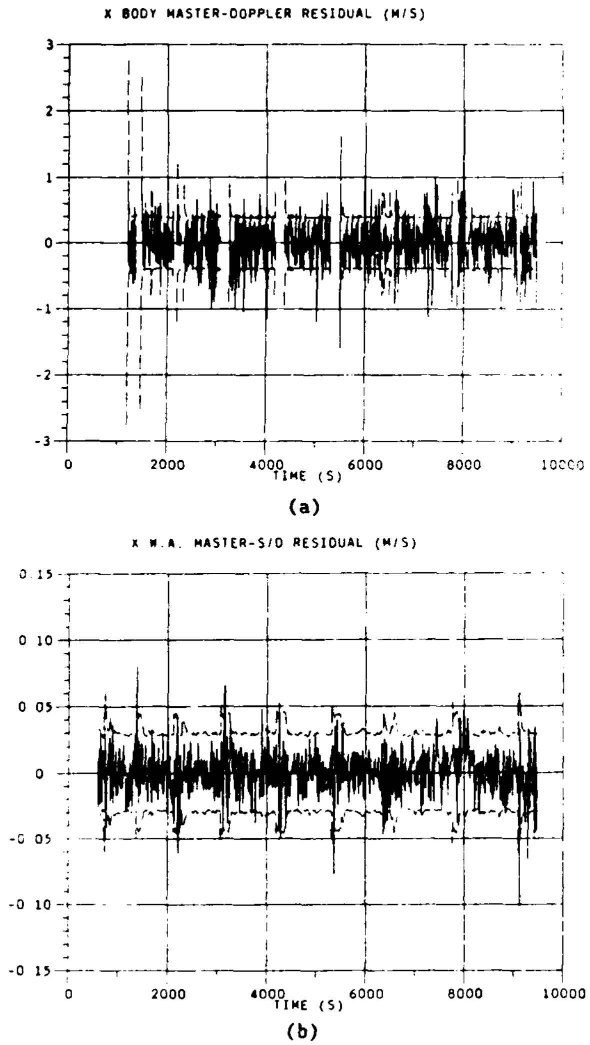


FIGURE 13: STRAPDOWN NAVIGATOR ROLL ERROR AND RMS FOR 21 STATE KALMAN FILTER



(a)

(b)

FIGURE 14: KALMAN FILTER MEASUREMENT RESIDUALS

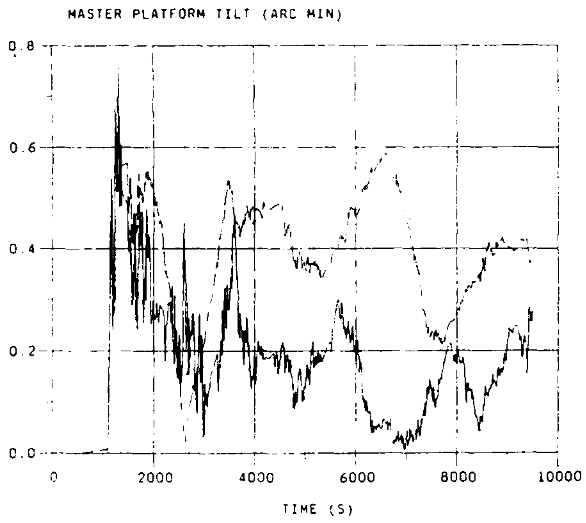


FIGURE 15: ABSOLUTE MASTER PLATFORM TILT FOR UNDAMPED AND DOPPLER-DAMPED OPERATION (DASHED LINE: UNDAMPED)

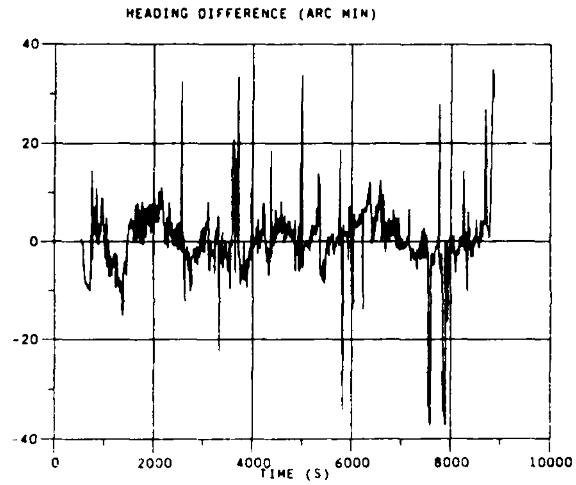


FIGURE 16: DIFFERENCE BETWEEN MASTER AND STRAPDOWN HEADING

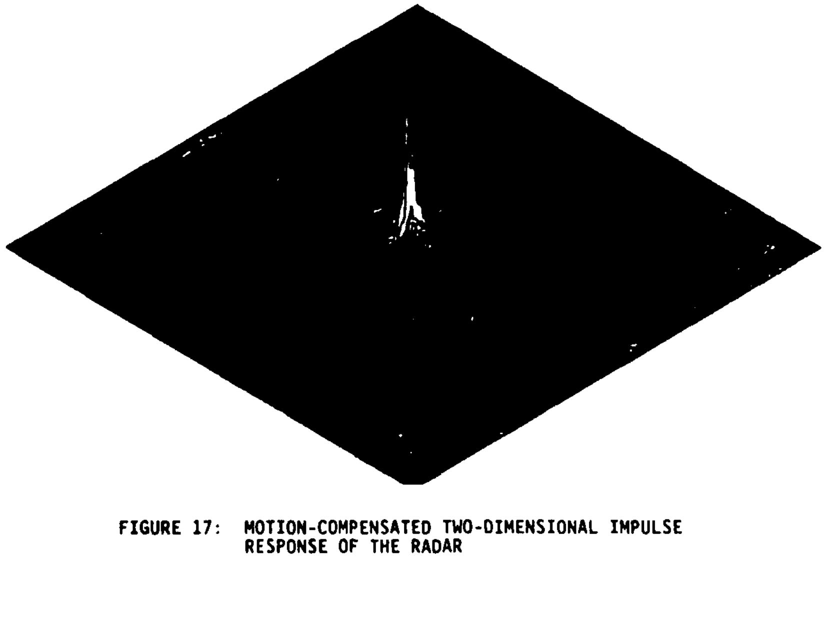


FIGURE 17: MOTION-COMPENSATED TWO-DIMENSIONAL IMPULSE RESPONSE OF THE RADAR

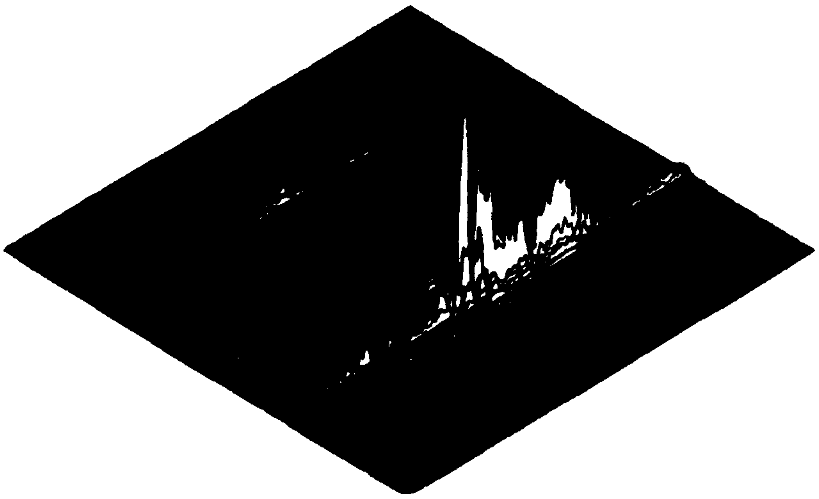


FIGURE 18: UNCOMPENSATED TWO-DIMENSIONAL IMPULSE RESPONSE OF THE RADAR

FIGURE 19: MOTION-COMPENSATED SAR STRIP-MAP IMAGE

FIGURE 20: UNCOMPENSATED SAR STRIP-MAP IMAGE

GUIDANCE AND CONTROL CONSIDERATIONS FOR A MANEUVERING AERIAL TARGET SYSTEM

Alex B. Markov and Robert W. Herring
Defence Research Establishment Suffield, Ralston, Alberta, Canada TOJ 2NO
and
Eric N. Solowka
Atlantis Flight Research Inc., 1 Kenview Boulevard, Brampton, Ontario, Canada L6T 5E6

ABSTRACT

In 1981 the Defence Research Establishment Suffield (DRES) initiated proof-of-concept (POC) development of a high subsonic, maneuvering fire-and-forget aerial target system named ROBOT-X. This effort was in response to a Canadian Navy requirement for aerial targets simulating low flying aircraft or anti-ship missiles, but has resulted in a target system that is also useful in many army low level air defence training scenarios. The POC development, recently completed, included seventeen POC test flights conducted at DRES.

Of critical importance for ROBOT-X to meet its performance objectives is a programmable, cost-effective flight control system. The system that was developed, which we call MicroPilot, incorporates a number of software and hardware features that allow it to fulfill ROBOT-X guidance and control mission requirements, permit convenient re-configuration of the guidance and control sensor suite, and facilitate implementation of algorithmic changes. This modularity has resulted in a flight control system that can also be used in other unmanned flight vehicles.

The paper includes a description of the ROBOT-X aerial target system. Guidance and control aspects are emphasized, and the methodology adopted in the flight control system design and prototyping is reviewed. MicroPilot algorithmic, software and hardware features are described as they evolved in POC ROBOT-X development. Highlights of results from simulation, testbed aircraft trials and POC ROBOT-X flight trials are given.

1. BACKGROUND

The Defence Research Establishment Suffield (DRES) has been involved in R&D dir'cted at aerial target technologies since 1979. Since 1981, this target research has beer focussed on ROBOT-X (Figure 1), a maneuvering rocket-boosted flight vehicle which has been developed to enable technologies for providing naval point defence missile systems with a target which simulates subsonic invader aircraft and anti-ship missiles coming from beyond the radar horizon (Figure 2). ROBOT-X also has potential for air-to-air and low level air defence applications.

The proof-of-concept (POC) development had the primary objectives of demonstrating tha ROBOT-X could meet its basic performance objectives as an aerial target system, and of ddressing the key technical areas and developments required to establish the system cost-effectively. A detailed description of the ROBOT-X system and the POC development his1ory are given in Reference 1. The system's basic performance parameters are as follows:

1. Maximum speed - Mach 0.85;
2. Altitude hold (pressure altimeter) - ±30 m;
3. Altitude hold (radar altimeter) - ±15 m;
4. Homing capability to a nondirectional beacon (NDB);
5. Sea level range ≥37 km (20 nm); and
6. Maneuvering capability (n_z) - ±8 g.

ROBOT-X utilizes 19 CRV-7 (C14) rocket motors for its propulsion system. These Canadian developed solid propellant rocket motors were developed principally as unguided air-to-surface 70 mm rocket weapons. They were selected for ROBOT-X use due to their availability and low cost. In the ROBOT-X application they are fired sequentially in up to 9 stages under the command of the flight control system (FCS). The guidance and control problems associated with ROBOT-X are demanding, in large measure due to the use of these motors. The short burn time (2.3 seconds) and high peak thrust (1550 lbs/6900 N per motor) result in the flight vehicle undergoing multiple boost/glide phases throughout its flight. During the 2.3 second boost phases the vehicle can accelerate at up to 15 g through airspeed changes as high as 205 m/s. The FCS must continue to actively function and control the vehicle throughout all phases of the flight including the boost phases.

The POC R&D has resulted in an ongoing Advanced Engineering Development (AED) program that has as its primary objectives to adapt the POC technology base for routine ROBOT-X aerial target operations for the Canadian Navy, and to value-engineer the system for cost-effective production. The AED effort is being managed by the Department of

National Defence Directorate of Maritime Combat Systems (DMCS), and the prime contractor is Boeing of Canada Winnipeg (BOCW). BOCW has been licensed by the Crown for the ROBOT-X technology. Atlantis Flight Research Inc. (AFRI) has been licensed for the guidance and control technologies and is pursuing potential markets for these technologies separate from that of the ROBOT-X application. The technological scope of this paper is confined to the POC R&D effort.

2. ROBOT-X FLIGHT CONTROL SYSTEM

2.1 MicroPilot Background

DRES involvement in aerial target R&D has led naturally towards the development of cost-effective flight control technologies. These efforts have resulted in the development of two distinct autopilot systems. One, named Pegasus, was developed in-house at DRES in response to requirements for a micro-processor based system suitable for mini-RPV applications (Reference 2). The second, now commonly referred to as MicroPilot, has been developed in response to more demanding ROBOT-X requirements, and provides a modular, multiprocessor autopilot system capable of both remotely piloted and autonomous operation in a high-g environment.

The ROBOT-X flight control system (FCS) is a fly-by-wire system centered around MicroPilot. MicroPilot was developed **ab initio** in support of ROBOT-X requirements, but as it is modular, fully programmable and configurable to different sensor and radio subsets, it has potential for use in a variety of unmanned aerial vehicle applications. Atlantis Flight Research Inc. is pursuing these markets independently of the ROBOT-X aerial target applications.

The MicroPilot technology base was developed in the course of a "proof-of-concept" effort that had three phases:

1. Phase 1 - MicroPilot concept/configuration development;
2. Phase 2 - Detailed design; and
3. Phase 3 - Prototype development and testing.

Five MicroPilot prototypes were designed and built by Atlantis Flight Research. The prototypes were built based on specifications, flight control algorithms and attitude and heading reference system (AHRS) algorithms defined by DRES, after preliminary control law synthesis work had been conducted by the University of Toronto Institute for Aerospace Studies. Most of the MicroPilot system integration work has been conducted at DRES, and has included extensive testing in a captive flight mode on a manned testbed aircraft (Figure 3) and on a motion facility (Figure 4).

2.2 MicroPilot Development Philosophy

MicroPilot development was undertaken when it was recognized that no "off-the-shelf" system was available that had both the required performance for the ROBOT-X application and the required low cost. The development philosophy that was adopted included the following main features:

1. Modularity

   The autopilot design strived to achieve the highest degree of modularity possible. This applied to software as well as hardware.

   Modular hardware facilitated the changing of components in the field for maintenance purposes as well as for configuration changes should a requirement arise for different sensor configurations. This happened several times during ROBOT-X POC and AED development.

   Modular software facilitated program development, modification and maintenance as well as configuration changes.

2. Maintainability

   The autopilot was designed to be maintainable by non-dedicated technical staff. Further, it was designed to have components available from multiple sources wherever possible.

3. Cost Effectiveness

   The autopilot was designed to achieve the lowest cost/performance ratio. Fundamentally, this was accomplished by using low cost electronics and sensors married to relatively sophisticated algorithmic/software subsets, to yield a cost effective system with high performance.

4. Application Insensitivity

   As well as being able to fulfill the specific needs of the ROBOT-X target requirements, from the outset MicroPilot was designed to have the ability to serve as an autopilot system for other vehicle or controller applications.

2.3 Design Requirements

Wherever possible, and without compromising aspects associated with flight safety (e.g., rocket motor ignition and flight termination), the development has been made to modest requirements as summarized in Tables 1 and 2. These requirements are suitable for the ROBOT-X application but were sufficiently demanding to provide a good baseline for the development of a multipurpose autopilot technology.

TABLE 1 ENVIRONMENTAL REQUIREMENTS

OPERATING TEMPERATURE	-40 to +55 deg C
AMBIENT PRESSURE	Sea Level to 9150 mASL
HUMIDITY	5 to 95% non-condensing
ACCELERATIONS (OPERATING)	X-axis -3 to +25 g's Y-axis ±4 g's Z-axis ±8 g's
ACCELERATIONS (RECOVERY)	-15 g's in 30 deg cone about ROBOT-X longitudinal axis
GROUND IMPACT (SURVIVABILITY)	±40 g's for 40 ms in all axes

TABLE 2 STATE ESTIMATION AND SENSOR REQUIREMENTS*

PITCH ATTITUDE	+70 to -60 deg min ±50 deg/sec pitch rate
ROLL ATTITUDE	±85 deg min ±100 deg/sec roll rate
HEADING	0 to 360 deg min ±30 deg/sec yaw rate
AIRSPEED	70 m/s to 340 m/s, ±5%
PRESSURE ALTITUDE	0 to 1500 m, ±15 m 1500 to 9150 m, ±1%

* AHRS performance is strongly influenced by the quality of the inertial sensors. Improved performance can be achieved with upgraded inertial sensors.

The multiple boost/glide phases that occur during a ROBOT-X flight presented some unusual problems in developing long-term pitch and roll references for the strapdown AHRS algorithms. These will be further discussed in a later section. It is noted that overall AHRS performance is strongly influenced by the quality of the inertial sensors. Improved performance can be achieved with upgraded inertial sensors, with a portion of ongoing second generation R&D effort currently focussing on achieving modest strapdown INS capability.

2.4 MicroPilot Functional Block Description

MicroPilot represents an integration of multi-processor based hardware, a complement of sensors and extensive software used to execute estimation, control and state monitoring of the flight vehicle. A general overview of MicroPilot and its interfaces to the ancillary airborne avionics is illustrated in Figure 5.

The autopilot, via specific cards of the autopilot computer, interfaces directly with the following ancillary airborne avionics:

1. Command Recovery and Integrated Staging Ignition System (CRISIS);
2. Pulse Code Modulated (PCM) telemetry downlink; and
3. Control surface servo actuators (canard, port aileron, starboard aileron).

In addition to the airborne avionics components, the autopilot is also linked to a ground based launch station. The launch station to autopilot interface allows an operator to control and monitor autopilot operation while the vehicle is on the launch pad. As well, the launch station provides a variety of vehicle pre-flight check-out, arming and countdown capabilities.

2.5 Hardware Configuration

From a hardware perspective, MicroPilot consists of three main subsets:

1. The autopilot computer.
2. The sensor/radio subset.

3. The Power Management Unit (PMU).

The autopilot computer consists of the 8088 microprocessor based CPU board and the peripheral electronics cards used to interface to the sensors and other ancillary airborne avionics. A more detailed description of the hardware configuration is given in Table 3.

TABLE 3 MICROPILOT ELECTRONICS CONFIGURATION

CPU PCB -	2 K RAM
	UP TO 32 K PROM
	2K EAROM
	WATCH DOG TIMER
	8 HARDWARE INTERRUPTS
COMMUNICATIONS PCB -	RS232/RS422
	16 BIT PARALLEL HIGH SPEED INTERFACE
	DISCRETE BIT I/O
ANALOG OUTPUT -	UNIPOLAR
	12 BIT RESOLUTION
	4 CHANNEL
PWM OUTPUT -	8 CHANNEL
	PROGRAMMABLE PULSE WIDTHS AND REP RATES
ANALOG INPUT -	GAIN AND POLARITY SELECTABLE
	8 CHANNELS PER PCB
	UP TO 5 PCB'S
	16 BIT RESOLUTION
TM INTERFACE -	CAPABILITY FOR INTERFACING
	WITH TM PCM ENCODER

The sensor subset comprises the sensors used directly by the autopilot to compute vehicle state estimation and provide vehicle control. The ROBOT-X sensor configuration is described in Table 4.

TABLE 4 MICROPILOT ROBOT-X SENSOR CONFIGURATION

ACCELEROMETERS (3 AXIS)
ANGULAR RATE TRANSDUCERS (3 AXIS)
ABSOLUTE PRESSURE TRANSDUCER
DIFFERENTIAL PRESSURE TRANSDUCER
MAGNETOMETER (3 AXIS)
RADAR ALTIMETER
HOMING RECEIVER
TOTAL TEMPERATURE PROBE (OPTIONAL)

The PMU conditions the 28 Volt DC nominal power provided from the battery pack into +5 V, +12 V, and ±15 V for use by the autopilot computer and sensors. The PMU also contains logic to control up to three heater channels. These channels can be set to the desired temperature, and the logic will turn the heaters on and off as required to maintain the autopilot thermal environment.

It is noted that there is a high level of integration involved between the autopilot computer and its sensors. Each autopilot computer is tailored to a particular sensor subset in order to optimize the quality of sensor data and minimize sensor specific non-idealities affecting overall vehicle performance. This level of integration, to a great degree implemented through software based sensor calibration data and through modular analog input cards, makes feasible the use of low cost electronics with relatively low cost sensors yielding an autopilot with performance typically found only in higher cost systems.

A ROBOT-X avionics beam complete with MicroPilot is shown in Figures 6 and 7. A generalized view of the arrangement of this equipment in the ROBOT-X flight vehicle is shown in Figure 8. It is noted that the air data pressure transducers and a 3-axis magnetometer are located remotely. Also, configurations are available in which the angular rate sensors and the accelerometers are packaged in one module as an Inertial Measurement Un't (IMU).

2.6 Algorithmic and Software Configuration

The development of MicroPilot/ROBOT-X state estimation, mission profile (MiP), flight control and state monitoring algorithms was greatly influenced by a number of factors, as follows:

1. The large Mach number, dynamic pressure, altitude and g-envelopes of the ROBOT-X flight vehicle necessitated the development of flight control algorithms that were robust to these effects. This was largely implemented in the form of control law scheduling as a function of dynamic pressure and Mach number.
2. The ROBOT-X flight vehicle's dynamic duty cycle, consisting of multiple large acceleration (up to 15 g) boost phases of 2.3 seconds duration followed by longer periods (typically 20 to 30 seconds) of gliding deceleration presented some unusual problems in developing long-term pitch and roll references for the strapdown attitude and neading reference system (AHRS) algorithms and in the control of the vehicle during the high acceleration boost phases.
3. The requirement for a "fire-and-forget" capability governed the development of the MiP definition and interpretation algorithms as well as the state monitoring software.
4. The use of relatively low cost sensors necessitated the development of estimation and flight control algorithms that were robust to sensor nonidealities, and that, in the case of some sensors, could correct for nonidealities such as temperature, acceleration, scale and bias effects.

These algorithms were developed by DRES after preliminary control law synthesis work had been conducted by the University of Toronto Institute for Aerospace Studies under R&D contract to DRES. Extensive use has been made of a six degree-of-freedom simulation package both in the course of algorithm development and for support of POC and AED flight testing.

The main features of these algorithms are summarized in the following. An algorithm functional breakdown appropriate for the ROBOT-X application is given in Figure 9.

2.6.1 State Estimation

A full set of air data estimation algorithms has been implemented. These provide estimates of dynamic pressure q_D, airspeed V, Mach number M_a, rate of climb $\dot{h}_p$ and flight path angle γ.

The air data is compensated for static port position errors. It is noted that in the ROBOT-X application direct sensing of total air temperature and the aerodynamic angles was not required, although MicroPilot can readily be configured for these sensor inputs. In ROBOT-X the air temperature is estimated through a knowledge of the launch site temperature and the assumption of standard lapse rate conditions, while the aerodynamic angles are estimated through a knowledge of the flight vehicle mass (which is available throughout the flight), the lift slope coefficient $C_{L_\alpha}(M_a)$ and the side force slope coefficient $C_{y_\beta}(M_a)$ characteristics.

All sensor data is partially conditioned prior to digitization through the use of analog filters on the sensor input cards. The characteristics of these filters are tailored to the characteristics and data requirements associated with each particular sensor. After digitization, the data is compensated for scaling, bias and temperature effects, as appropriate, in analog input card software and then is passed to the CPU. At the CPU, additional compensation (e.g., for g-effects) and digital filtering may be implemented, again as appropriate for the particular type of sensor data.

The Euler bank angle Φ_B, pitch angle Θ_B and heading angle Ψ_B are estimated through the use of a strapdown AHRS implementation. This implementation is illustrated in Figure 10. It uses a second order complementary filtering algorithm to mix long-term and short-term estimates of the attitude and heading angles to provide "best" estimates of these angles. These are then used in vehicle control and navigation algorithms. A second order implementation is used to eliminate bias effects. An alternative approach to the AHRS implementation is that of Kalman filtering.

The combination of lower cost sensors and the multiple boost/glide environment of ROBOT-X presented some unusual problems in the development of the AHRS algorithms. Special problems included development of a suitable formulation of the long-term estimates (particularly in the pitch axis) and establishing complementary filter parameter values suitable for use with the ROBOT-X sensors.

2.6.2 Control Command Algorithms

In the case of ROBOT-X, all longitudinal and lateral/directional control modes are implemented by generating appropriate flight path angle γ, angle of attack α_r and bank angle Φ_B commands. The resulting control laws consist of proportional, integral and initial condition terms. The control commands generated by these control laws are summed to corresponding commands generated by stability augmentation control laws. The associated control law gains are appropriately scheduled as a function of dynamic pressure and/or Mach number.

Load limiting and control demand limiting algorithms have also been implemented.

2.6.3 Flight Management Algorithms

The POC ROBOT-X flight management algorithms were developed under the concept of relating all flight events to flight segments, with a given flight segment beginning at the initiation of a rocket motor burn for a stage and ending at either the start of the next stage burn or at recovery system deployment.

During POC flight testing, rocket motor stages were fired to manage the vehicle's energy by defining a minimum segment dynamic pressure, while control mode transitions occurred based on segment time. Alternative event-dependent strategies have been developed during the course of AED and for ongoing use by ROBOT-X in flying laboratory R&D roles.

Control modes available on POC ROBOT-X vehicles included both radar and pressure altitude hold, angle-of-attack hold, flight path angle hold, heading hold, rate of turn hold, bank angle hold and a homing mode. More advanced modes have been implemented during AED and include maneuver-g modes.

State monitoring algorithms have also been developed and implemented. These focus on checking the "reasonableness" of the vehicle state in order to enable subsequent events (e.g., rocket motor firings). If an "unreasonable" state is maintained, the flight is terminated through automatic deployment of the recovery system.

2.6.4 Software Implementation

ROBOT-X software capabilities as implemented on MicroPilot are summarized in Table 5.

TABLE 5 MICROPILOT ROBOT-X SOFTWARE CONFIGURATION

BENCH TEST AND BUILT IN TEST CAPABILITY
100 HZ UPDATE RATE
MISSION GUIDANCE: PRE-LAUNCH MONITORING COUNTDOWN AND LAUNCH INITIATION PRELAUNCH AND IN-FLIGHT STATE MONITORING WAYPOINT GUIDANCE & CONTROL MODE DETERMINATION RECOVERY MONITORING
VEHICLE CONTROL AND NAVIGATION: SENSOR SAMPLING AND SIGNAL CONDITIONING STATE ESTIMATION: STRAPDOWN AHRS AND AIR DATA CONTROL LAW IMPLEMENTATION CAPABILITY FOR STRAPDOWN INS WITH UPGRADED SENSOR SUBSET

The software was written in assembly language but was extensively modularized through the use of compact subroutines and macros. The latter, in particular, give the software features that are normally associated only with higher level languages. This considerably enhances readability as well as facilitating software modifications.

CPU software consists of over 20 kbytes of code. Approximately 1 kbyte of additional software is resident in each of the 8742-based analog input cards.

3. TEST RESULTS

3.1 Flight Qualification Testing

Prior to flight testing the complete ROBOT-X POC airborne system, numerous preflight tests were performed on a number of critical subsystems. This testing included the following:

1. Temperature and g-tests of selected avionics subsystems;
2. Bench testing of control servo actuators;
3. Flight testing of the two stage parachute system on a water ballasted version of a ballistic test airframe;
4. Drop tests of the frangible nose structure;
5. Environmental tests of the airframe at the Aerospace Engineering Test Establishment; and
6. Captive flight testing of the autopilot, autopilot sensors and airborne telemetry on a manned testbed aircraft and on a motion facility.

These tests are reviewed in more detail in References 1, 3 and 4.

3.2 Overview of ROBOT-X POC Flight Test Results

Seventeen ROBOT-X POC flights were conducted in the period 13 March 1986 to

9 June 1988, as summarized in Table 6. All seventeen flights were conducted at the Suffield Military Range, primarily within the DRES Experimental Proving Ground (EPG). In addition to telemetry and radar data, all flights have had extensive video and high speed cine (16 mm and 35 mm) coverage. The seventeen flights have accumulated more than 34.4 minutes of flight time, not including airborne time after drogue parachute deployment.

TABLE 6 ROBOT-X PROOF-OF-CONCEPT FLIGHT SUMMARY

FLT	DATE	LAUNCH TIME (UCT)	FLIGHT TIME (to drogue) (secs)	$n_{x_{max}}$ (g)	$n_{z_{max}}$ (g)	MAX M_a	MAX ALT (mAGL)	FLIGHT PATH LENGTH (km)	STAGES PLANNED /ACTUAL	LAUNCH MASS (kg)	LAUNCH C of M (FS, cm)
1	13/03/86	22:08:39	3.4	14.4	9.5/-4.0	0.55	660	1	3/1	250.8	197.1
2	04/07/86	19:13:52	5	14.9	8.0/-4.	0.6	700	1	3/1	247.6	195.5
3	29/08/86	16:33:00	30	14.6	2.6/-3.2	0.7	1800	5	3/3	252.1	196.9
4	04/11/86	21:34:00	71	14.3	±2	0.65	1800	12	4/4	257.1	195.8
5	27/11/86	17:44:00	89	14.3	±2	0.65	1500	14.7	4/4	257.4	194.0
6	09/12/86	21:18:00	84	14.8	±2	0.65	1500	12	4/4	249.5	195.8
7	05/03/87	16:59:00	137	14.3	±2	0.65	1500	23	5/5	257.7	193.5
8	03/04/87	18:25:12	48	13.7	+3/-2	0.6	1500	7.8	6/3	268.7	193.7
9	13/05/87	18:40:00	169	13.5	+3/-2	0.65	1600	24	6/6	272.6	193.3
10	16/06/87	17:33:00	32	13.6	+3/-1	0.6	1500	4	8/3	269.0	190.9
11	08/07/87	16:37:00	217	14.3	+3/-2	0.65	1800	31	7/7	260.0	189.7
12	26/08/87	19:05:00	123	13.8	+3/-2	0.88	1500	21	8/7	267.0	191.4
13	11/09/87	17:21:00	245	13.7	±3	0.6	1400	35	8/8	268.6	191.2
14	27/10/87	18:31:59	246	14.0	+2/-0.6	0.6	1540	35	8/8	266.0	191.1
15	16/12/87	19:54:00	204	13.0	+3.2/-0.6	0.86	1770	32	8/8	269.0	191.5
16	28/01/88	20:28:00	110	13.0	+2.7/-0.7	0.79	1550	20	8/7	265.0	191.7
17	09/06/88	18:35:00	252	13.0	+3.0/-0.7	0.58	1050	35.5	8/8	268.0	192.2

NOTE: Max n_x, n_z given for flight conditions prior to chute deployment.

The system has been launched at gross weights between 247.6 kg (545.9 lb) and 272.6 kg (601.0 lb), and launch centre of gravity locations between fuselage station 189.7 cm (74.7 in) and 197.1 cm (77.6 in). The basic operational configuration of the system is well below the 250 kg (550 lb) objective defined in 1982, since approximately 25 kg (55 lb) of the POC system's weight was due to test instrumentation and telemetry. The system has been shown to be quite robust in the high-g rocket boost and maneuvering environments and repeatedly survived and operated through rocket boost acceleration up to 14.9 g and maneuvering accelerations up to 9.5 g. Significant loads can also be generated during recovery system deployment. The system flew in launch site ambient temperatures between -10°C and +25°C and encountered no significant temperature related problems.

3.3 ROBOT-X Stability and Controllability

After adjustment of longitudinal and lateral control law gains during the course of the first six flights, the system's stability and controllability were excellent, with both ailerons and canards demonstrating exceptional control power. This was been found to be the case for the canards despite the relatively forward centre of gravity conditions at which the system was flown, and the correspondingly greater demand placed on the canards.

Flight dynamic characteristics associated with launch rail clearance and staging were uneventful for all flights. Asymmetric thrust concerns, particularly during the launch boost, proved to be unfounded, and some relaxation of the lateral centre of gravity envelope occurred during the course of the POC flights. Stringent crosswind launch limits that were enforced during the early flights were gradually relaxed, as the system's crosswind characteristics revealed themselves to be quite satisfactory.

On some early flights the vehicle had a marked tendency to turn to the left during the first 10 seconds or so of the flight. The sources of this turning tendency were traced to several factors related to airframe asymmetries. The known asymmetries were either corrected or were compensated for by the control system; this problem did not arise in the last seven POC flights.

3.4 ROBOT-X Airframe Integrity and Recovery Reliability

The POC airframe proved to be quite robust in the demanding ROBOT-X environment, and functioned and survived at thrust loads up to 15 g and maneuvering loads over ±8 g. A partial failure of the canards that occurred on Flight 12 due to flutter at Mach 0.8 was addressed through a canard structural redesign. The impact on POC test flights was negligible, as the failure occurred at the edge of the flutter envelope of the system.

The recovery method, i.e., nose down recovery utilizing a frangible nose structure, is similar to that used on the British Falconet aerial target system, and proved to be quite effective (Figure 11), avoiding the requirement for complex (and costly) cushioning systems.

Despite an extensive effort at flight qualifying the recovery system prior to flight on ROBOT-X, Flights 1, 5, 6 and 16 encountered recovery system problems that prevented main canopy deployment. Consequently, system recovery on these flights was on the drogue alone, with the result that these airframes were substantially destroyed. Nevertheless, significant portions of the onboard electronics and aft fuselages were refurbishable. It is noted that the drogue decelerates the system to a 120 knot (60 m/s) airspeed. The main recovery system reliability during AED test flights has been considerably higher.

Investigation of the recovery problems caused some delay to the POC flight trials. In the case of Flight 1, the problem was identified as an electronic system design problem that was subsequently rectified. In the case of Flight 6, the problem was identified as being caused by the unexpectedly strong adhesive properties of the tape used to hold down the parachute lines, which prevented main canopy pull out (the tape held over 400 lbs of load). This problem was rectified by replacing the tape wit'. rubber extrusions. This configuration functioned properly on all subsequent flights. In the case of Flight 16, the problem was identified as the g-induced failure of an electromechanical relay during a high-g, high speed parachute deployment. Modifications to the relay mode of operation eliminated this problem.

Largely because of the problem encountered on Flight 6, Flights 7 and on were flown with a backward looking video camera that was intended to provide video coverage of the chute deployment sequence. This video signal was successfully down-linked on all these flights and provided spectacular coverage, as well as visually showing clean parachute deployments with the use of the rubber extrusions. Both B&W and colour video cameras have been utilized; on POC Flight 16 there were two onboard video cameras.

It is worth noting that for all seventeen POC flights there were no failures of the two stage parachute system, the drogue deployment or the radio link for ground commanded parachute deployment. Drogue deployment is particularly important from the range safety point of view, and a high degree of reliability has been demonstrated.

3.5 Ground Support

All ground support facilities associated with a ROBOT-X operational system functioned well and evolved into a comprehensive preflight checkout and ground support test set.

3.6 MicroPilot Flight Performance

MicroPilot has demonstrated the ability to conduct fire and forget mission profiles of substantial complexity, including firings of eight rocket motor stages (nine during some AED flights) and multiple longitudinal and lateral waypoints. As an example, the x-y radar position track associated with POC ROBOT-X Flight 13 is shown in Figure 12. This eight stage firing had a mission profile that had several heading and altitude changes.

In general, MicroPilot and its associated sensors performed very well, and met or exceeded all of the performance requirements listed in Tables 1 and 2. The major software related problems encountered in the first two flights were resolved in Flight 3, with software changes since Flight 3 being relatively minor and consisting largely of refinements to AHRS software and control law gains.

3.6.1 Altitude Hold Performance

Pressure altitude hold performance of the system was very good (±20 m) relative to sensed pressure altitude. The latter is still subject to static port position errors, and accounts for most of the altitude deviation that was observed. The static port position error correction is done in autopilot software and will continue to be refined in the course of AED test flights. In conjunction with minor adjustments of the control law gains, this is projected to allow altitude hold to ±5 m.

The seventeenth POC test flight was conducted utilizing data from a radar altimeter for altitude guidance for a large portion of the flight. During this flight altitude hold to better than ±10 m was achieved relative to sensed radar altitude.

3.6.2 Roll Stability

Figure 13 shows the progressive reduction in roll rate oscillation that resulted in Flights 4 to 6 as aileron control law gains were gradually turned down. This oscillation was produced by a combination of the lags introduced by the characteristics of the low pass sensor data filters that were used initially and by ailerons that had considerably more rolling effectiveness than predicted. Flights 6 and on showed no unusual roll oscillations.

3.6.3 Hardware and Sensor Problems

The only significant hardware related problems encountered within MicroPilot were a threshold voltage problem associated with a battery back-up RAM chip (which was resolved after Flight 3 by replacing the chip with an EAROM chip), and possible altitude lock problems associated with the use of a general aviation radar altimeter in the particularly demanding ROBOT-X environment. Neither problem was flight critical. The radar altimeter problem was resolved through the use of a different type of radar altimeter.

In addition, some antenna pattern problems arose with the general aviation type ADF homing used in POC flight testing, probably aggravated by the limited and distorted ground planes available on ROBOT-X.

3.6.4 Algorithm Performance

The AHRS algorithms are unique and proved to be satisfactory in the dynamic ROBOT-X environment.

Autopilot commanded rocket ignition was found to be quite reliable, and there were no stage or individual motor failures in the course of 87 POC airborne and/or launch stage firing commands involving the firing of 225 CRV-7 (C14) rocket motors.

3.6.5 Canard and Aileron Servos

The ROBOT-X canards and the left and right ailerons are each independently controlled by one of three electrical servos. The servos that were used for fifteen of the seventeen POC test flights were Simmonds Precision "off-the-shelf" high performance units, selected because of their convenient availability. They were, however, too costly for production targets, and thus DRES sponsored the development of a more cost-effective system. The result has been a more compact and considerably less costly pulse width command system (Figure 14). These new servos were flown on POC Flights 14 and 16 and functioned nominally. They have also been used on all 28 AED flights conducted to date by BOCW.

3.6.6 MicroPilot Captive Flight Testing

A technique that was found to be very useful in flight qualifying ROBOT-X avionics hardware and software was that of captive flight testing in a manned testbed aircraft. Initial tests on a Cessna 172 aircraft rented from a local fixed-based operator on a day-to-day basis proved to be sufficiently useful to warrant the longer term lease of a dedicated aircraft (Figure 3) that was specially configured and instrumented to support ROBOT-X and other unmanned vehicle avionics tests. DRES has recently accepted delivery of a second generation testbed aircraft (a leased Seneca II) to be utilized in future R&D work.

Captive flight testing proved to be especially valuable in flight qualifying the ROBOT-X autopilot, autopilot sensor and airborne telemetry system hardware and software. In particular it was used extensively in the development and refinement of the attitude and heading reference system (AHRS) algorithms, in checking for gross autopilot flight control software errors, in checking the ground command recovery uplink and telemetry downlink and for general "burning-in" of ROBOT-X avionics in a moderately harsh, vibration filled environment. The speed difference between the test aircraft and ROBOT-X has not proven to be a significant limitation, and was one of the more surprising results of this approach to flight qualification.

The aircraft had provision for the entire avionics beam as flown in ROBOT-X as well as a reference vertical gyro whose data was also telemetered to the ground. This allowed direct comparison with estimates of pitch and bank angle generated by the autopilot AHRS algorithms. Other features of the aircraft included auxiliary 28 VDC and 110 V, 60 Hz power outlets, a permanently installed flight test boom acting as a primary air data source and all antennas for transmitters/receivers associated with DRES unmanned aerial system research.

3.6.7 System Simulation

In conjunction with autopilot algorithm development and mission profile definition, DRES has developed an extensive six degree-of-freedom (DOF) capability for simulating the close-loop dynamics of ROBOT-X. Based on the results from the seventeen POC flights, the aerodynamic model associated with the six DOF simulation has been updated and provides good correspondence between predicted and actual stability, control and performance characteristics. This model was used successfully in resolving a number of control algorithm/control gain problems, as well as in defining the mission profiles associated with all POC and AED flights.

A real-time capability for processor-in-the-loop simulation was not available during ROBOT-X POC development. This deficiency is now being rectified with the commissioning of a real-time simulation facility (SIMFAC). This capability will be available for all future DRES guidance and control R&D.

4. SUMMARY AND FUTURE WORK

The ROBOT-X POC development provided a framework through which a number of technologies critical to cost-effective aerial targets were investigated. The design work, ground and captive flight tests, and seventeen POC flight trials demonstrated the feasibility of the ROBOT-X concept and have enabled the necessary technologies within Canadian industry. This facilitated the rapid and successful conduct of the ROBOT-X advanced engineering development work by prime contractor Boeing of Canada Winnipeg and its subcontractor Atlantis Flight Research Inc., who have taken the system into preproduction as a Canadian naval aerial target system.

The flight control technologies that were developed in association with the POC effort are centered around MicroPilot and have applicability to a broad class of unmanned flight vehicles. This technology has been licensed separately to Atlantis Flight Research Inc.

Current and planned future guidance and control R&D at DRES is being addressed by building on the expertise and facilities made available through the ROBOT-X and MicroPilot efforts. These facilities include video and telemetry data acquisition and analysis capabilities, a real-time processor-in-the-loop facility, a motion facility and a captive test research aircraft. Most of these facilities were justified through lessons learned during the ROBOT-X POC and MicroPilot R&D efforts. This R&D will explore a number of guidance and control areas including advanced AHRS and strapdown INS algorithms, advanced inertial sensor technologies, and advanced flight control architectures.

From a longer-term perspective, these efforts have generated a technical base that complements Department of National Defence activities in weapon system guidance and control, and provide an opportunity to address all aspects of such R&D. ROBOT-X and its derivatives, in addition to fulfilling a number of Canadian Forces' aerial target roles, provide an advanced flying laboratory with which a number of topical sensor, hardware and software guidance and control problems can be investigated.

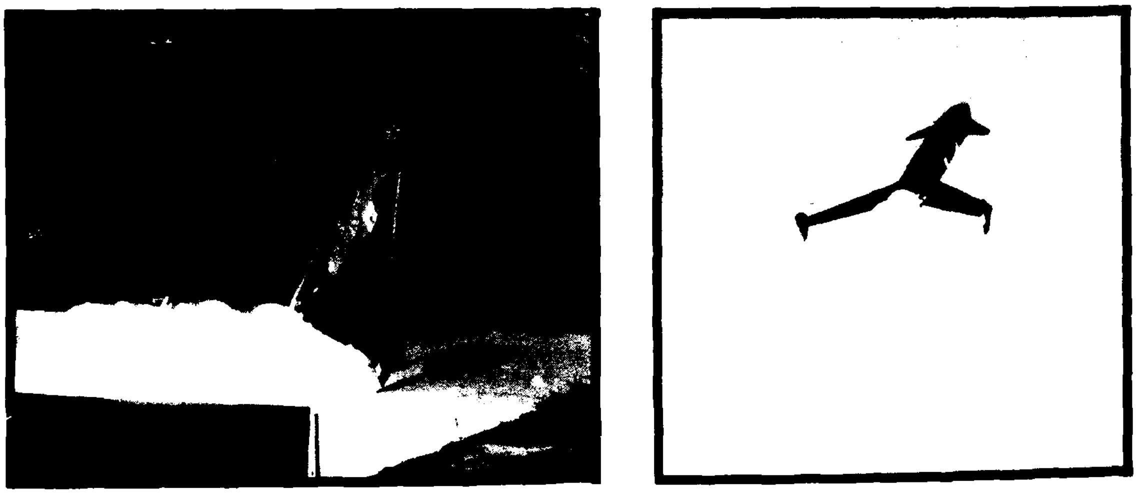
5. REFERENCES

1. A.B. Markov, C.G. Coffey, H.P. Hudema, D.N. Benson, S.G. Penzes, T.V. Meidinger and R.A. Eirich, "ROBOT-X Proof-of-Concept Flight Trials: A System Review", **CASI Flight Test Symposium**, 18-19 March, 1987.

2. S.K. Burton and D.R. Weiler, **The Pegasus Autopilot for Remotely Piloted Vehicles (U)**, Suffield Memorandum No. 1167, June 1986. UNCLASSIFIED.

3. S.G. Penzes, C.G. Coffey, H.P. Hudema, B.G. Boulter and A.B. Markov, "ROBOT-X Recovery System Parachute Trials", **CASI Flight Test Symposium**, 26-27 March, 1985.

4. S.G. Penzes, A.B. Markov, A.E. Turner and B.G. Boulter, **Environmental Tests of the ROBOT-X Airframe** (U), Suffield Memorandum No. 1192, March 1987. UNCLASSIFIED.



t = 0 seconds

t = 0.65 seconds

Figure 1

ROBOT-X FLIGHT 3 LAUNCH
(Taken With 70 mm Hulche Camera Running at 40 Frames Per Second)

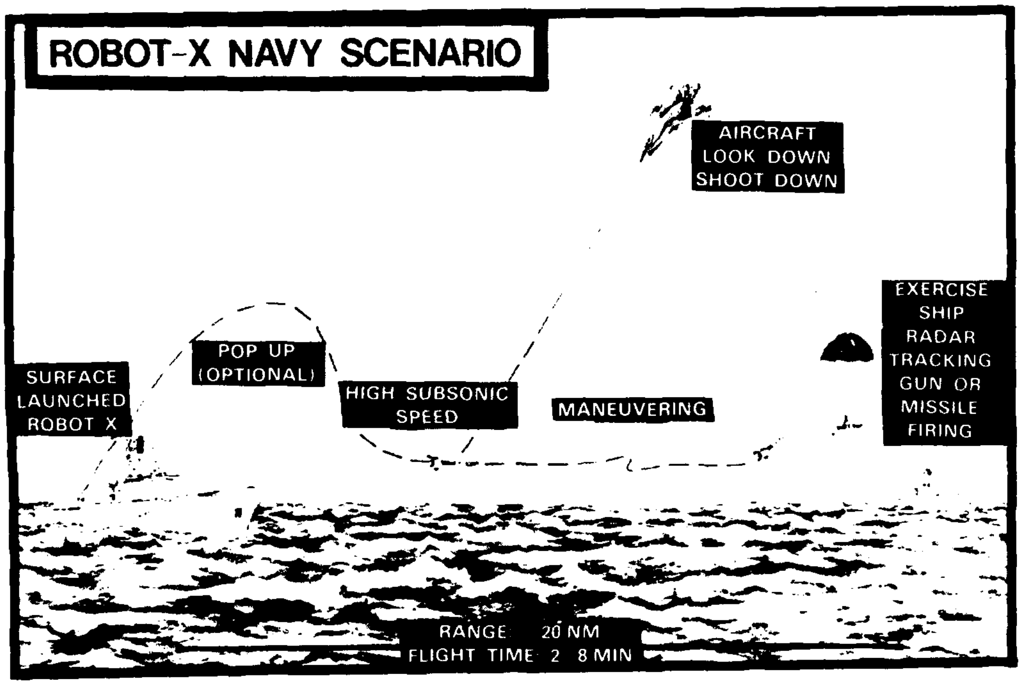


Figure 2

ROBOT-X OPERATIONAL SCENARIO

Figure 3

DRES TESTBED AIRCRAFT

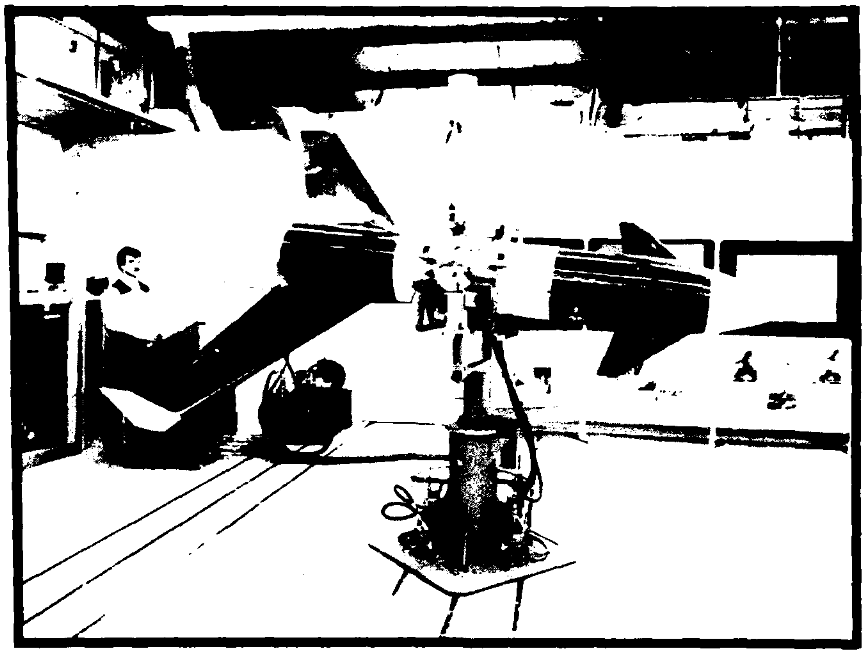


Figure 4

ROBOT-X MOTION FACILITY (MOFAC)

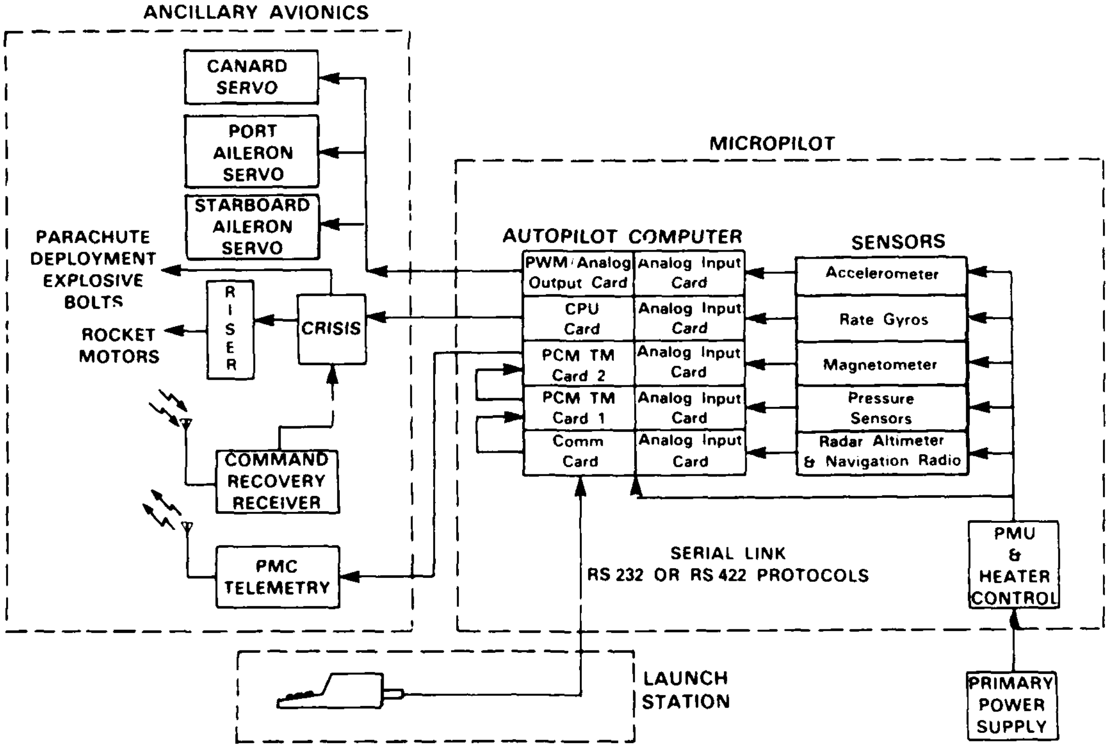


Figure 5

MICROPILOT FUNCTIONAL BLOCK DIAGRAM (ROBOT-X CONFIGURATION)

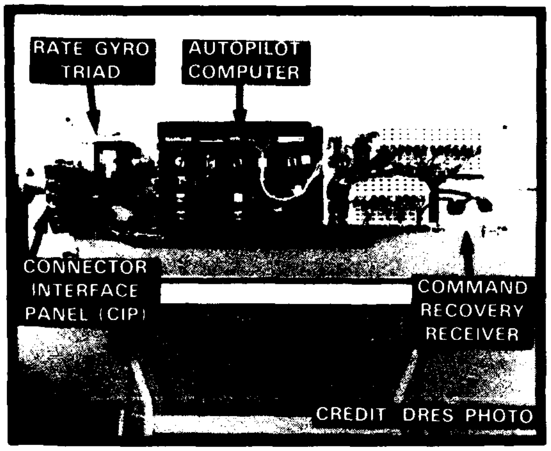


Figure 6

ROBOT-X AVIONICS BEAM COMPONENT LAYOUT VIEW FROM STARBOARD

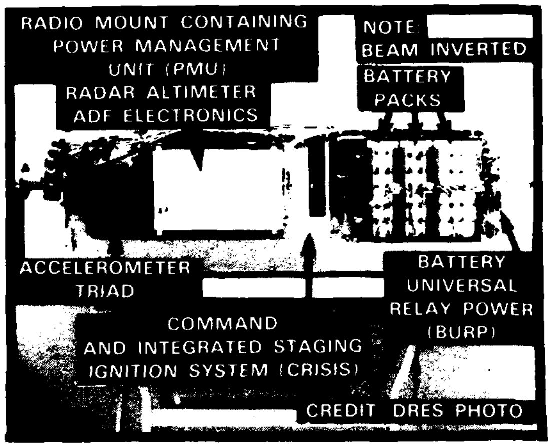

RADIO MOUNT CONTAINING POWER MANAGEMENT UNIT (PMU) RADAR ALTIMETER ADF ELECTRONICS
NOTE: BEAM INVERTED
BATTERY PACKS
ACCELEROMETER TRIAD
BATTERY UNIVERSAL RELAY POWER (BURP)
COMMAND AND INTEGRATED STAGING IGNITION SYSTEM (CRISIS)
CREDIT DRES PHOTO

Figure 7

ROBOT-X AVIONICS BEAM COMPONENT LAYOUT VIEW FROM PORT

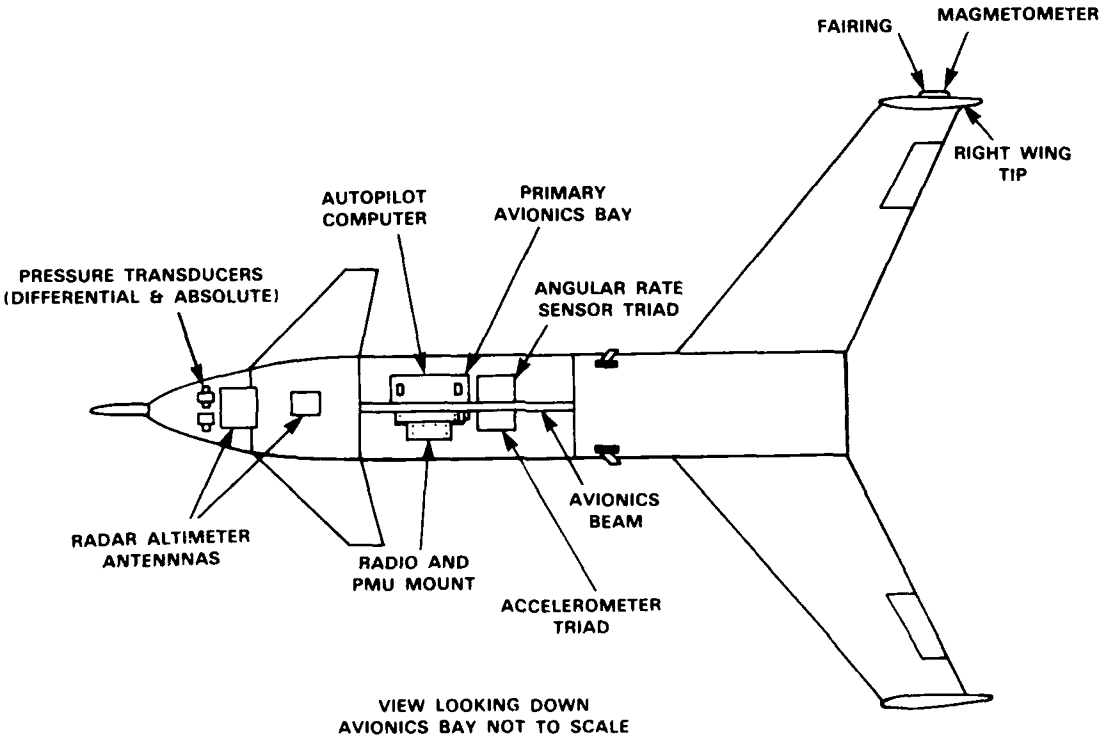

FAIRING
MAGMETOMETER
RIGHT WING TIP
AUTOPILOT COMPUTER
PRIMARY AVIONICS BAY
PRESSURE TRANSDUCERS (DIFFERENTIAL & ABSOLUTE)
ANGULAR RATE SENSOR TRIAD
AVIONICS BEAM
RADAR ALTIMETER ANTENNNAS
RADIO AND PMU MOUNT
ACCELEROMETER TRIAD
VIEW LOOKING DOWN AVIONICS BAY NOT TO SCALE

Figure 8

AUTOPILOT COMPONENT LOCATIONS IN ROBOT-X

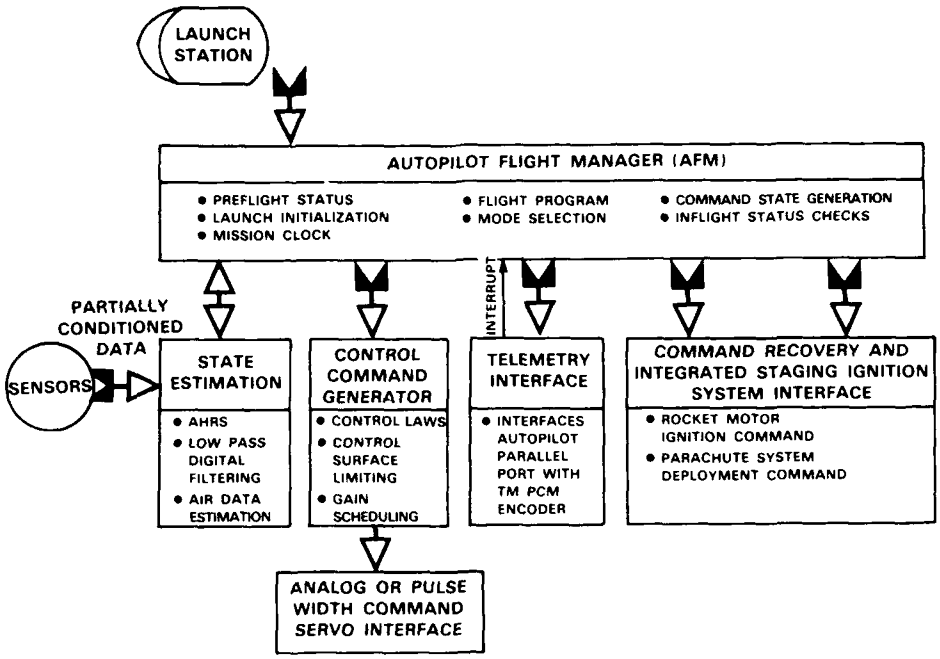
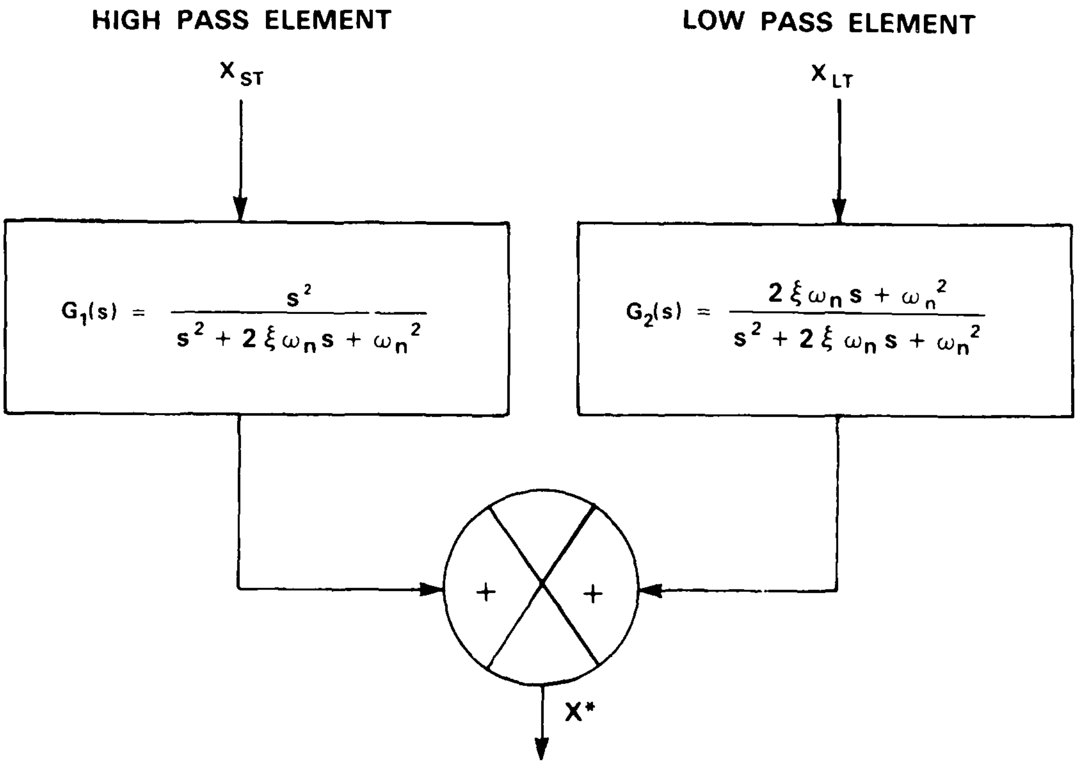

LAUNCH STATION
AUTOPILOT FLIGHT MANAGER (AFM)
• PREFLIGHT STATUS
• LAUNCH INITIALIZATION
• MISSION CLOCK
• FLIGHT PROGRAM
• MODE SELECTION
• COMMAND STATE GENERATION
• INFLIGHT STATUS CHECKS
PARTIALLY CONDITIONED DATA
SENSORS
STATE ESTIMATION
• AHRS
• LOW PASS DIGITAL FILTERING
• AIR DATA ESTIMATION
CONTROL COMMAND GENERATOR
• CONTROL LAWS
• CONTROL SURFACE LIMITING
• GAIN SCHEDULING
INTERRUPT
TELEMETRY INTERFACE
• INTERFACES AUTOPILOT PARALLEL PORT WITH TM PCM ENCODER
COMMAND RECOVERY AND INTEGRATED STAGING IGNITION SYSTEM INTERFACE
• ROCKET MOTOR IGNITION COMMAND
• PARACHUTE SYSTEM DEPLOYMENT COMMAND
ANALOG OR PULSE WIDTH COMMAND SERVO INTERFACE

Figure 9

MICROPILOT SOFTWARE FUNCTIONAL BREAKDOWN
(ROBOT-X CONFIGURATION)


HIGH PASS ELEMENT
LOW PASS ELEMENT
X_{ST}
X_{LT}
$G_1(s) = \frac{s^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$
$G_2(s) = \frac{2\xi\omega_n s + \omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$
+
+
X*

X_{ST} = SHORT-TERM ESTIMATE
X_{LT} = LONG-TERM
X* = COMPLEMENTARY FILTER "BEST" ESTIMATE

Figure 10

SYMMETRIC SECOND ORDER COMPLEMENTARY FILTERING
AHRS FORMULATION

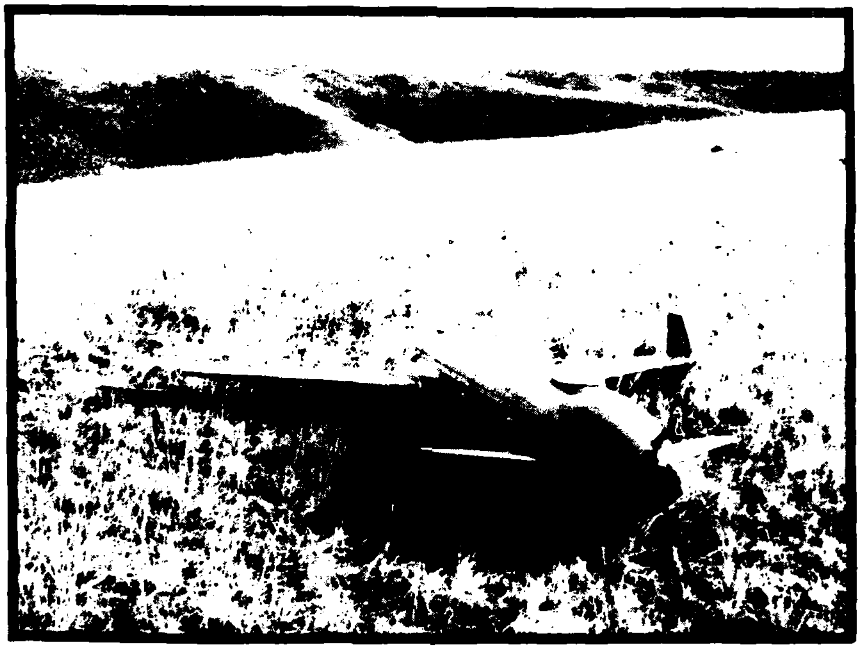


Figure 11

ROBOT-X AFTER FLIGHT 7 RECOVERY
(5 MARCH, 1987)

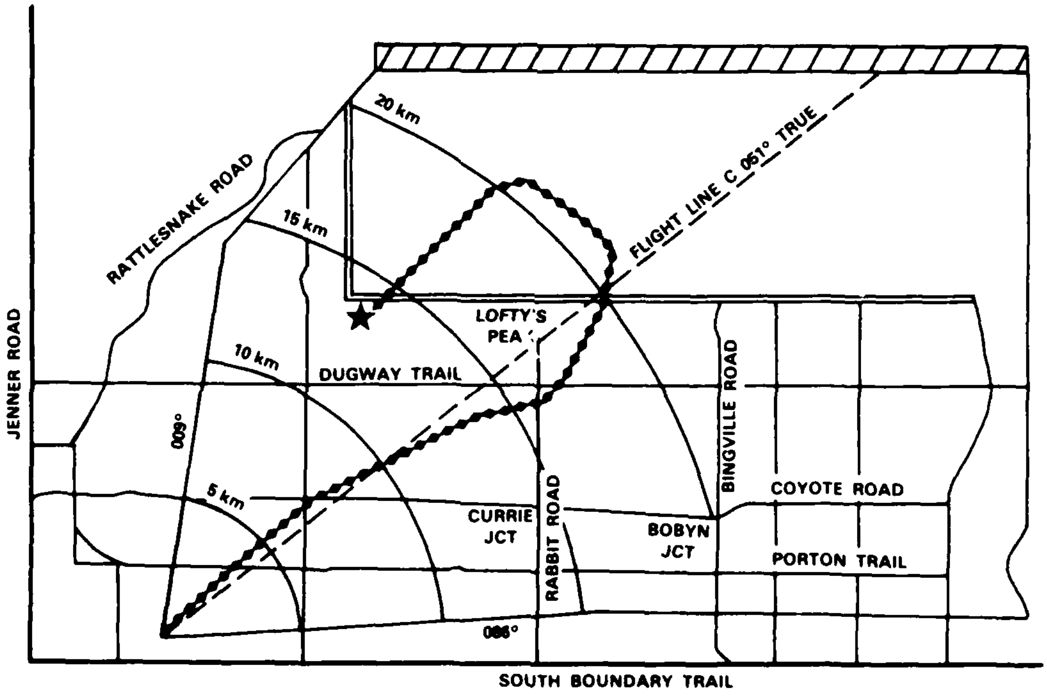


Figure 12

ROBOT-X POC FLIGHT 13 X—Y RADAR TRACK
(11 SEPTEMBER, 1987)

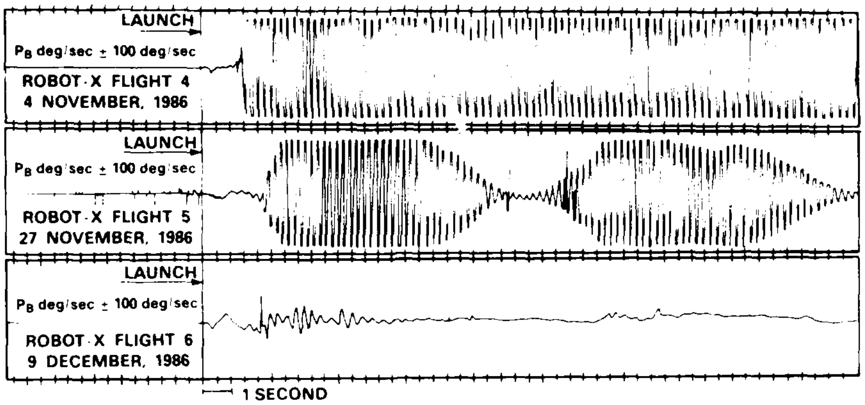


Figure 13

ROBOT-X ROLL OSCILLATION DECREASE AS CONTROL LAWS GAINS WERE ADJUSTED IN FLIGHTS 4 TO 6

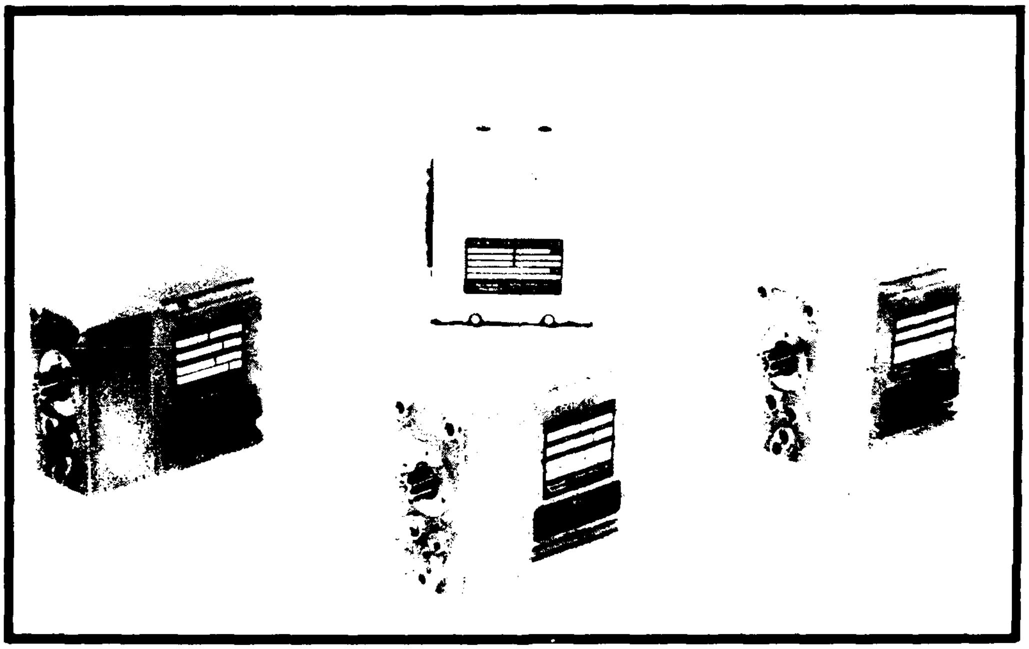


Figure 14

SIMMONDS PRECISION PWC SERVO ACTUATORS DR 1730 M 2 (PHOTOGRAPH PROVIDED BY SIMMONDS PRECISION)

TERRAIN-AIDED NAVIGATION AND TARGET ACQUISITION ON THE AFTI/F-16

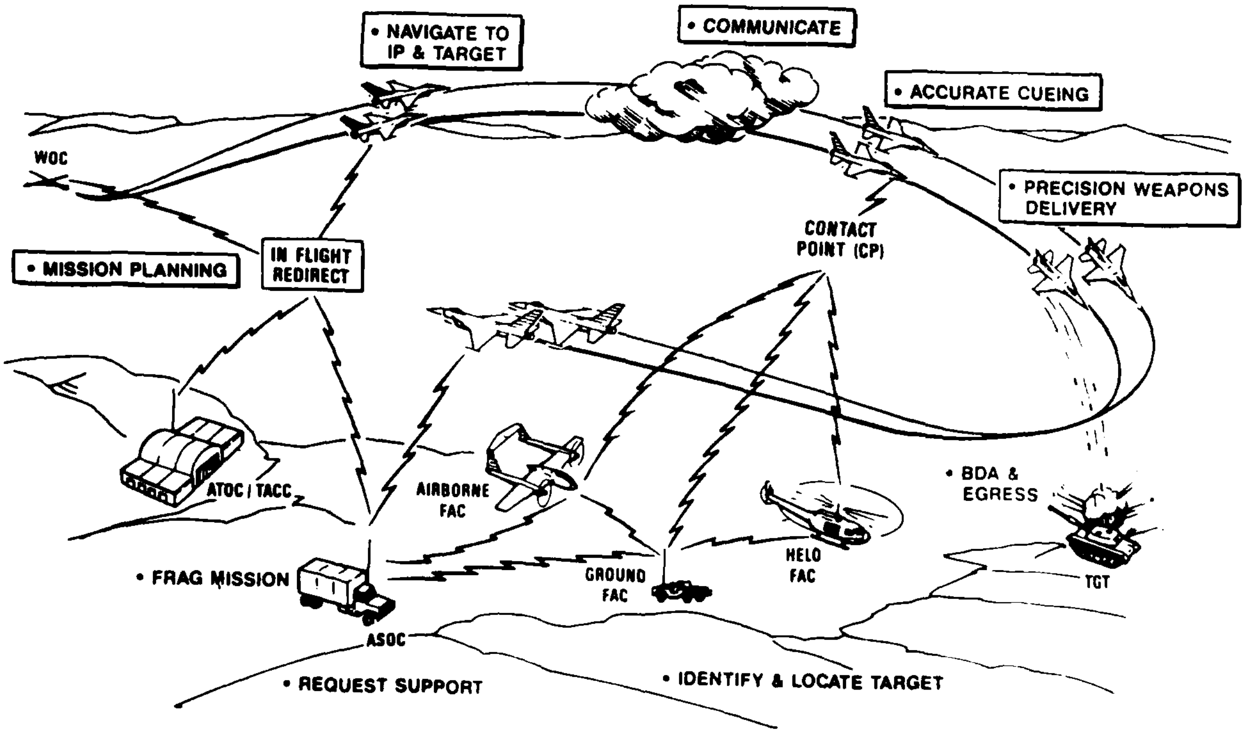
Dr. Charles A. Baird and Noel Collins
Harris Corporation, Government Aerospace Systems Division
Melbourne, Florida, 32901
United States of America

and

Major Myres Drew
Wright Research and Development Center
Wright Patterson Air Force Base, Ohio 45433
United States of America

SUMMARY

The Advanced Fighter Technology Integration/F-16 (AFTI/F-16) aircraft is currently involved in flight testing aimed at developing technologies and capabilities to support the close air support (CAS) and battlefield air interdiction (BAI) missions. Figure 1 illustrates the major features of the CAS mission. Such missions require joint operations between air and ground forces. For the air forces, aircraft survivability in a high threat environment during night and adverse weather are major concerns. For the ground forces, response times and weapon deliveries close to friendly ground forces are also major concerns. Key to the success of these missions will be the accurate navigation and target acquisition techniques which are presented. Also critical to these missions are the joint interoperability requirements between the air and ground forces for communication of targetting information and the ability to accurately deliver weapons on these targets.



The AFTI/F-16 is well suited to examine the requirements of the CAS/BAI missions. Previous work on this aircraft has led to the development of both a triply redundant digital flight control system (DFCS) and the Automated Maneuvering Attack System (AMAS), which has accumulated 349 hours of testing over 238 sorties. These systems provide fully coupled curvilinear bombing using 5-g maneuvering at 200 feet AGL and 500 knots airspeed, with equivalent accuracy of an F-16 in conventional wings-level attack. To provide this capability, a conformal sensor tracker system (STS) containing a FLIR and Laser Ranger, fire control radar, helmet mounted sight, and HUD, have been fully integrated. Safety-of-flight considerations have lead to the development of system wide integrity management (SWIM), including automated ground collision avoidance (GCAS) and g-loss of consciousness (GLOC) protection. Also integrated into this aircraft is the Digital Terrain Management and Display System (DTMDS) (References 1 and 2) which provides digital terrain elevation data and associated tactical situation displays. The DTMDS provides the terrain referenced navigation and passive ranging functions described in this paper, as well as a number of other capabilities to be reviewed below, which are critical to the CAS missions.

The functional check flights for the terrain referenced navigation system on the AFTI/F-16 have been recently completed at Edwards AFB in California. These flights confirmed the ability of the Sandia Inertial Terrain-aided Navigation (SITAN) to operate effectively over operational quality elevation data and also indicated several improvements in the algorithm. Also, during the past summer, demonstration flights were conducted at Fort Irwin (also in California). For these flights the U. S. Army's Automatic Target Handoff System (ATHS) was integrated into the aircraft and represented the first demonstration of joint U. S. Army/Air Force voiceless CAS operation. This system allows voiceless transmission of target coordinates to be passed to the fast moving F-16 from a scout helicopter or forward air controller (FAC). This data, coupled with the accurate navigation from SITAN, provided the pilot with the improved situational awareness required to consistently demonstrate first pass target acquisition and attack. Future CAS work on the AFTI/F-16 will involve the integration of the passive ranging techniques described in this paper into the target acquisition and weapon delivery systems, as well as a number of other terrain data based techniques, including automated terrain following, terrain avoidance, threat avoidance and route optimization.

This paper is organized in the following manner. In the next section the CAS/BAI problem is described from a pilot's mission performance perspective. This section is followed by a description of how the aircraft systems are integrated to utilize the terrain-aided navigation and passive ranging functions provided by the DTMDS to support the target acquisition and weapon delivery aspects of the CAS missions. Detailed presentation of the terrain-aided navigation flight test results and the analytical work supporting the passive ranging functions are then presented. This is followed by the final two sections describing the three phase CAS flight test program (Reference 3). First the recently completed phase one is described, culminating in the flight demonstrations featured in the video presentation. The last section in the paper describes the future program goals to be addressed in phases two and three of the CAS program.

THE CLOSE AIR SUPPORT MISSION

The battlefield of any future, major conventional conflict is expected to be a very dynamic and somewhat confused series of armed encounters. The Air Force is tasked with providing concentrated firepower (i.e. CAS) to assist friendly ground forces in defending their positions or initiating counterattacks. This mission must be carried out in an arena containing a high concentration of enemy surface-to-air weapons designed to thwart any CAS effort.

A typical CAS mission begins with an Army battalion commander having some activity which he cannot suppress with his battalion's organic weapons. He contacts the tactical air control party (TACP), an Air Force element colocated with his battalion, and requests air support through the air liaison officer (ALO) assigned to the TACP. The ALO, who is also a qualified forward air controller, gathers the necessary target information (location, type, and strength) and makes a request for close air support over a voice radio net to the Air Support Operations Center (ASOC) at the Army corps level. TACPs located at intermediate Army echelons also monitor the CAS request net and coordinate on the request if appropriate. If the request is approved by the senior ground force headquarters, the ASOC in coordination with the Tactical Air Control Center (TACC), the manager for tactical air assets, checks the availability of aircraft and configurations. If airborne aircraft cannot be diverted to the target area, the ASOC contacts the TACC to scramble a flight of fighters on ground alert.

Once airborne, the flight receives radar control to a designated contact point initially from control and reporting centers (CRCs) and then control and reporting posts (CRPs) or airborne warning and control system (AWACS) aircraft. As the flight approaches the contact point the controlling agency will tell the flight to contact either a tactical air coordinator-airborne (TAC-A) or forward air controller (FAC). Contact points (CPs) are normally chosen 20-40 nautical miles behind the forward edge of the battle area (FEBA) to provide some safety against enemy communications jamming and surface-to-air missiles. The flight lead initiates contact with the FAC or TAC-A at the CP over voice radio and provides them with mission identification, number and type aircraft, ordinance, and the amount of time the flight can remain on station. The FAC or TAC-A will then brief the flight by voice radio on the specifics of the CAS mission, beginning with the designation of an initial point (IP) from which the flight will initiate the attack. IPs, like CPs, are normally topographical features which can be easily identified from the air, however IPs are typically chosen to allow a 10-15 nautical mile (2-3 minute) attack run for the flight. After designation of the IP, the FAC or TAC-A will provide the flight with the following, as a minimum: the straight line heading and distance from the IP to the target in degrees magnetic and to the nearest tenth of a nautical mile, a brief target description, the location of the nearest ground friendlies relative to the target, and attack clearance. If time and conditions permit, the FAC or TAC-A can also provide additional information, such as, target coordinates, target elevation, known threats, flight hazards, altimeter and winds, and if the target will be identified or designated by some device. After acknowledgement of the FAC/TAC-A briefing, the flight is then cleared to depart the CP for the IP, and handed off to the FAC providing the final control for the mission. Often the FAC providing the final control is the ALO who originally initiated the request for close air support.

The FAC providing the final control is responsible for insuring that the fighters find and attack the correct target. For targets that are in close proximity to friendly ground forces, the FAC must use positive control to ensure that the fighters do not cause collateral damage to the friendlies. The FAC must also coordinate friendly ground artillery to ensure that the fighters providing the close air support do not get hit. These and other considerations may make it necessary for the FAC to transmit additional information and direction to the flight over voice radio, often in the last 30-40 seconds of the attack run.

The fighter pilot experiences his highest workload once he departs the CP for the attack. He must control his aircraft while being constantly aware of his position relative to other flight members and to the target area. As the aircraft approaches the FEBA, the threats imposed by enemy surface-to-air and air-to-air weapon systems dramatically increase and demand constant attention. In addition to acquiring the correct target and managing the systems necessary to deliver the weapon, the fighter pilot must constantly monitor his radio for information from the FAC or other flight members.

Problems often arise during the CAS mission because all of the information is passed by voice over a radio. Critical attack and target information is often confused or misunderstood requiring additional transmissions for clarification. Secondly, voice data is not often in a form that the fighter pilot can readily use. For instance, if the fighter pilot wants to enter the CAS target coordinates into his aircraft's inertial navigation system, he must manually enter the data during flight. Thirdly, the voice transmissions are either long enough or numerous enough to make them susceptible to communications interception and jamming.

The ultimate success of the CAS mission depends on the fighter pilot's ability to find and attack the target. Target acquisition and attack are coupled to survivability. A high speed, low altitude ingress to the target is most survivable, however such an ingress makes target acquisition most difficult. A slower airspeed and higher altitude ingress allows the pilot to acquire targets easier, but exposes his aircraft more to threats. The most survivable attack is one in which the pilot only attacks once and does not have to overfly the target, however, this demands the use of costly and scarce guided munitions or air-to-surface gunnery. If the pilot uses freefall munitions, he must currently overfly the target if he wants a high probability of target kill, at the risk of increased exposure to surface-to-air threats.

SYSTEM DESCRIPTION

The integrated digital avionics suite being developed on the AFTI/F-16 is currently being configured to support the CAS mission described in the last section. A key component in this design is the terrain data base and its associated functions. The block diagram of the terrain referenced navigation and passive ranging techniques, as they are mechanized for target acquisition and weapons delivery, is shown in Figure 2. The major components of this system are outlined in dashed lines and are the DTMDS, fire control computer (FCC) and the aircraft displays, as well as the Automatic Target Handoff System (ATHS). The DTMDS provides the terrain elevation data required for both the SITAN terrain referenced navigation and the terrain-aided passive estimation (TAPE) which provides the passive target ranging and position location functions. Figure 3 is a block diagram of the CAS-configured DTMDS, showing the addition of two 1750-based processor cards to host the new CAS algorithms (Reference 4). These new cards are inserted in spare processor slots provided for in the original DTMDS design, and will host real-time threat intervisibility computations and several terrain data based guidance functions. Referring back to Figure 2, the SITAN algorithm requires sensor inputs of position and velocity from the inertial navigation system (INS) and the AGL altitude from the radar altimeter (RALT), while the TAPE algorithm requires line of sight (LOS) angles from the helmet mounted sight (HMS) and/or the sensor tracker system (STS). SITAN outputs accurate terrain-referenced aircraft position data and TAPE provides both target range and location data to the FCC. The ATHS also provides voiceless target location data from U. S. Army Scout helicopters and Air Force Forward Air Controllers (FACs) using existing voice radio equipment.

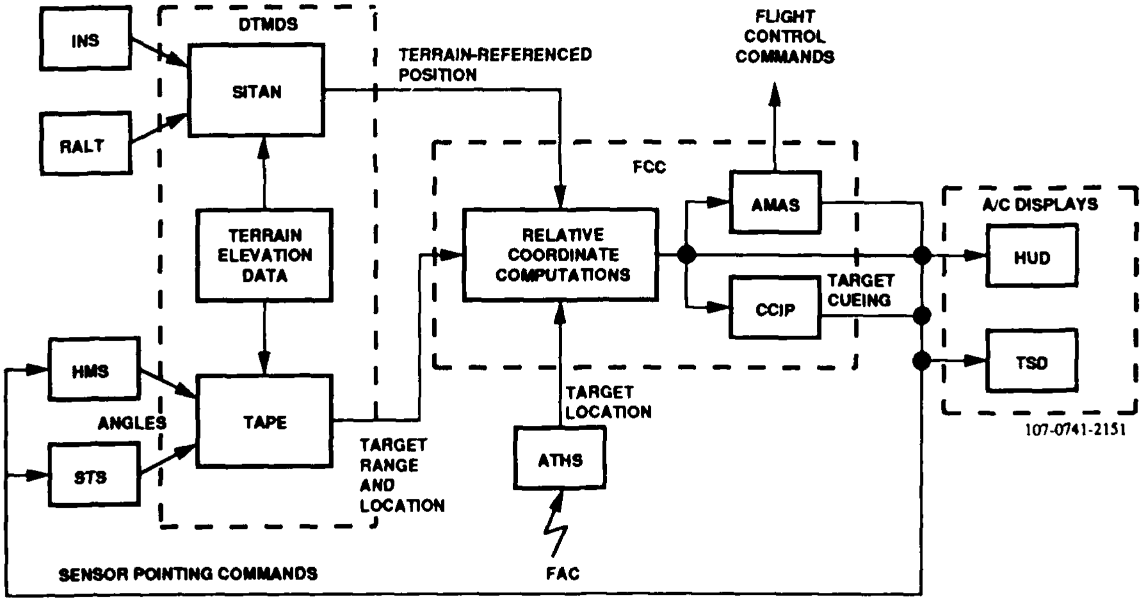


Figure 2. Terrain-aided Navigation and Passive Ranging in the AFTI-F-16.

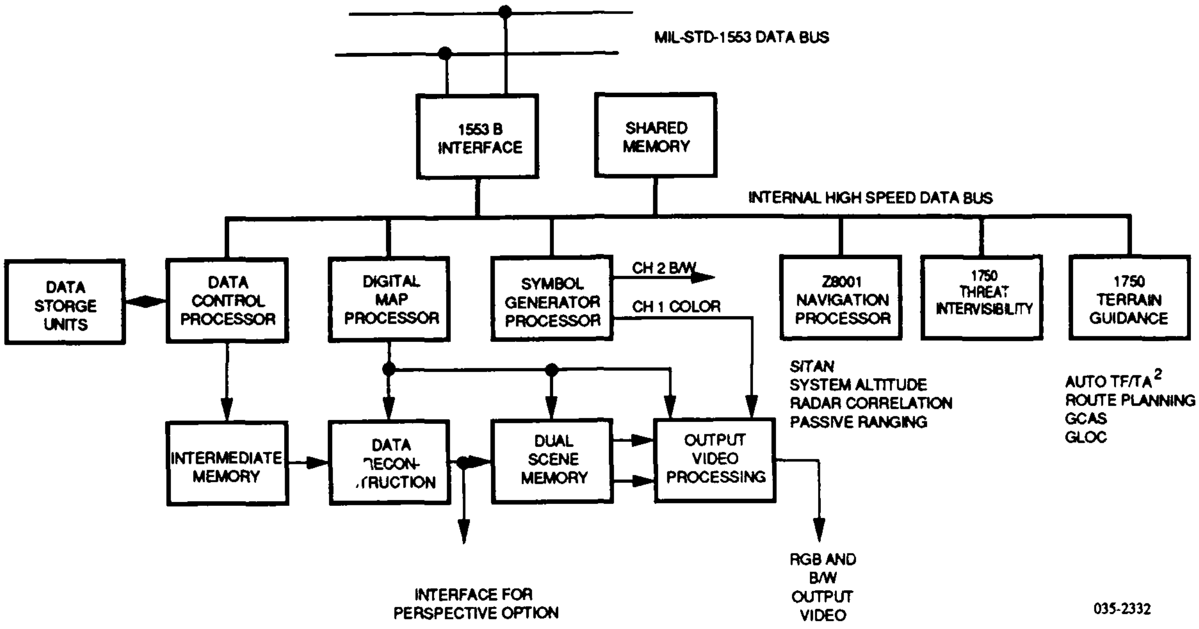


Figure 3. CAS-Configured DTMDS.

The fire control computer is the central computing element in this configuration, accepting the targetting data described above and producing the required flight control commands and target cues. It is important to note that this targetting data must all be referenced to the same elevation data map to insure that any bias errors in the map will not degrade the system performance. That is, both the aircraft and FAC generated target data must be referenced to the same coordinate system, not to an absolute earth centered coordinate frame such as GPS data. This targetting data is converted by the FCC to aircraft relative coordinates to be used by the weapon delivery computations. The automated maneuvering attack system (AMAS) computes flight control commands for fully coupled weapons delivery. The continuously computed impact point (CCIP) algorithm provides target cueing to the pilot for manual weapons release.

The target cueing commands are presented to the pilot via the aircraft display systems. Target cues are presented on both the heads-up display (HUD) and the helmet mounted sight (HMS). Target locations are also placed on the map-based tactical situation display (TSD). The TSD background is provided by the DTMDS and consists of a variety of display formats (Reference 2) generated from Defense Mapping Agency (DMA) Digital Landmass (DLMS) data. This data base consists of digital terrain elevation data (DTED) and digital feature analysis data (DFAD). This data is augmented by various DTMDS symbol generator overlays denoting the aircraft, target and friendlies positions, flight path and waypoints, and both the expected aircraft and weapon paths from the AMAS calculations. This allows the pilot to preview the computed paths, provide consent commands and gives him a warning of the aggressive (5-g) flight regimes he will be experiencing in the AMAS deliveries. For the future CAS demonstrations the TSD will also display real time threat locations and lethality regions computed by the DTMDS as a function of terrain masking, threat envelope and aircraft altitude.

TERRAIN-AIDED NAVIGATION - SITAN

The first auxiliary application for the DTMDS-provided onboard digital terrain elevation data on the AFTI/F-16 has been for autonomous navigation. A variety of terrain-aided navigation algorithms have been developed, where comparative simulations of TERCOM, SITAN and a hybrid algorithm combining the features of both of these algorithms indicates that these algorithms, if properly implemented, all provide similar tracking performance. The major performance difference is in the acquisition behavior, which can be improved by judicious design of the acquisition logic. (References 5 and 6.) Figure 4 is a representation of the performance to be expected from such procedures over various types of terrain with varying degrees of system noise. The terrain types vary from smooth to mountainous, with the terrain characteristics of roughness (sigma-T), slope variance (sigma-S) and terrain elevation correlation distance (Xc) as shown in the boxes. The horizontal axis is SNR, defined as the ratio of terrain roughness (sigma-T) to the system noise (sigma-n - combined map and altimetry errors, with plotted values of 4, 8, and 12 meters). The left vertical axis shows the resultant track accuracy (CEP) in meters, while the right vertical axis indicates the approximate acquisition distance in kilometers. While these plots were obtained using simulations over synthetic terrain, the flight test results verify their accuracy. Post flight analysis of data obtained from helicopter flights at Fort A. P. Hill in Virginia (Reference 7) substantiates the smooth terrain results at 0 dB. Flight tests conducted by Sandia Laboratories in the South Central United States (Reference 8) verify the moderate terrain value at 7 dB. The present AFTI/SITAN flight tests at Edwards AFB (Reference 9) confirm the high SNR region for the hilly and mountainous terrain.

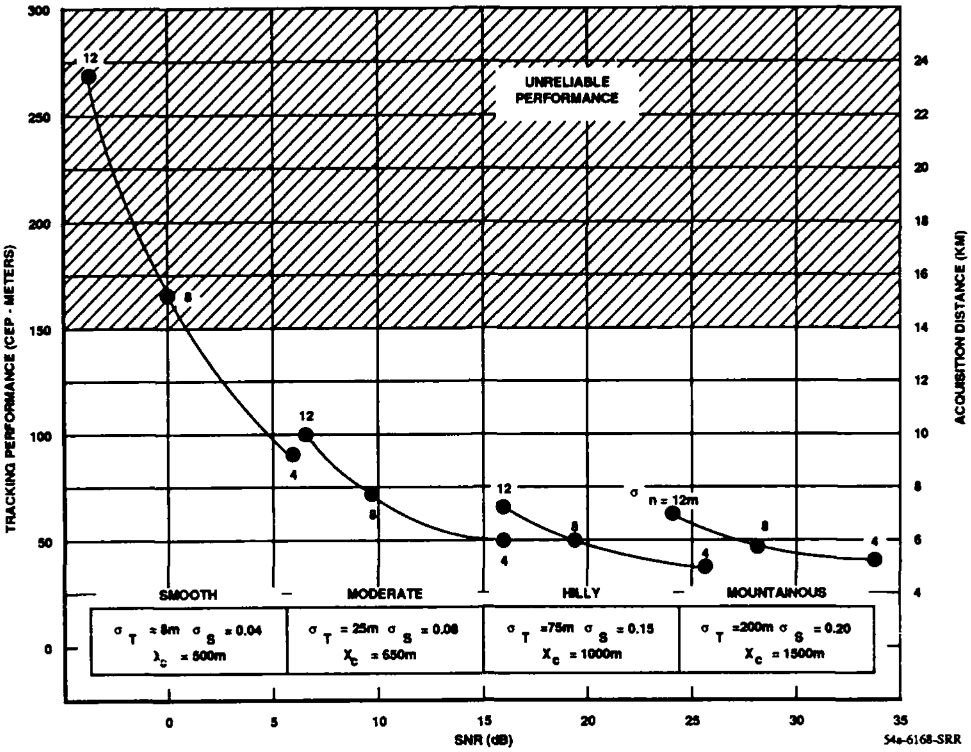


Figure 4. Terrain-Aided Navigation Performance.

The terrain-aided navigation algorithm implemented in the DTMDS is the SITAN algorithm which is illustrated in Figure 5. This extended Kalman filter based algorithm has been under extensive flight testing on the AFTI/F-16 at Edwards AFB since September 1986 with analysis, simulation, and previous flight testing by Sandia Labs going back to the late 1970's. As shown in the figure, this algorithm takes as inputs the radar altimeter indicated ground clearance, the inertial navigation system (INS) estimated position, a combined INS-air data computer (CADC) indicated sea-level referenced altitude (system altitude), and the terrain data base elevations. The algorithm outputs improved position and velocity estimates. Also computed by this algorithm are estimates of the errors in these outputs and SITAN mode status. Perhaps the most challenging aspect of this algorithm is the automatic mode control logic which determines when to switch between the acquisition and track modes. For acquisition, this implementation requires initial INS position inputs to be within an approximately 4.2 kilometer circular area since 57 three state linearized Kalman filters, spaced on 525 meter centers, are used in this mode. Note that a larger acquisition region could be accommodated using more parallel filters. The 57 filter outputs are monitored by the mode control logic, and when the average weighted residuals of a particular filter become small enough, it is selected as the track filter. At that point, the horizontal velocity states are added to make a five state filter. The track filter residuals are continually monitored to insure that the track performance remains accurate enough, a function of the roughness of the terrain being flown over.

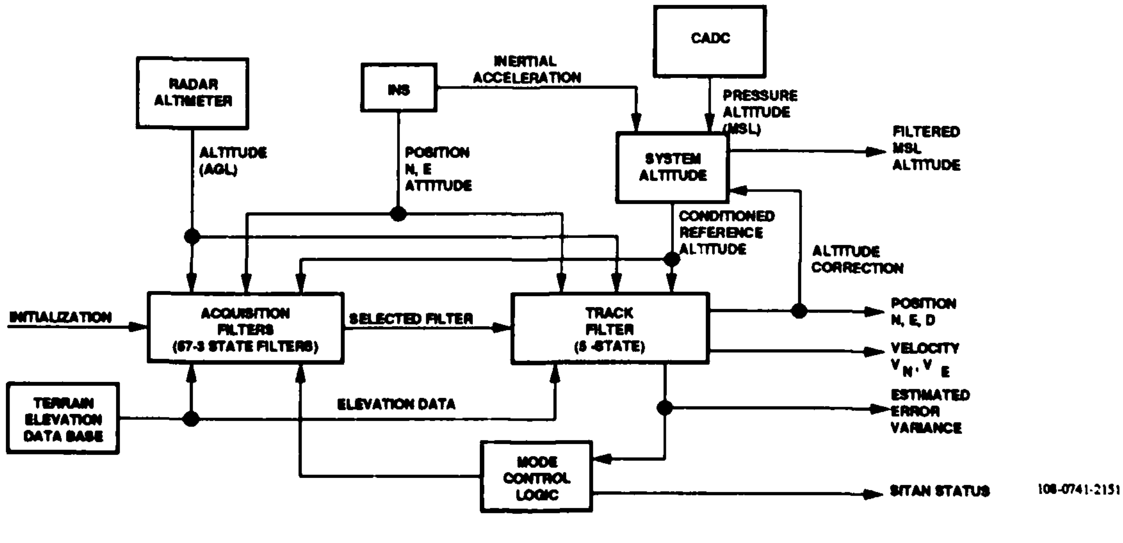


Figure 5. SITAN Block Diagram.

Table 1 is a compilation of the acquisition and track performance for the first five functional test flights (FCFs) (Reference 9). The acquisition performance indicates an average acquisition distance over all the flights of 18.7 km. Note that this is somewhat longer than would be expected from the right-hand vertical axis of Figure 4, but these results were dominated by the extremely long (320 second/48 km) acquisition sequence on flight FCF 1. This was over very smooth terrain, where it is difficult to accurately predict the algorithm's behavior. Simulations using variable filter gains described below significantly reduce this particular acquisition distance. This improved performance is shown in parenthesis in the table. The track performance shown in the table indicates an average track accuracy of 70 meters CEP. Also shown are the results for a sequence of flight tests on a Twin Otter aircraft, the so-called SAINT/SITAN system. Although this was for a short flight sequence, it indicates the robustness of the AFTI/SITAN algorithm, as it is able to operate effectively in an aircraft with significantly different performance characteristics from the F-16.

Table 1. Flight Test Performance of AFTI / SITAN (Reference 9).

FCF #	FLIGHT TIME	TRACK ACCURACY	ACQ. TIME/ DISTANCE	TERRAIN ROUGHNESS
1	2430 s	75 m (58 m)	320 s/48 Km (85 s/12.8 Km)	Smoothest
			127 s/17.4 Km	Moderate
2	2105 s	66 m	60 s/8.2 Km	Moderate
3	390 s	40 m	-	Moderate
4	1030 s	-	71 s/9.7 Km	Smooth/Moderate
5	820 s	86 m	103 s/12.8 Km	Smoothest
TOTALS	6755 s/113 m	70 m (64 m)	136 s/18.7 Km (89 s/12.2 Km)	
SAINT/SITAN	2000 s	75 m	60 s/3.2 Km	Roughest
			193 s/9.6 Km	Rough

111-0741-2151

This initial sequence of flight tests of the SITAN algorithm revealed several implementation errors which were manifested in the form of fixed biases over the test flights and were removed from the track accuracy results presented above. For example, software addressing errors for the SITAN database within the DTMDS resulted in a fixed bias which has since been removed, with the remaining bias errors well within the error bounds specified by DMA for the DTED data (Reference 10). Such database errors indicate the importance of referencing all targetting and navigation data within the system to the same map database. For the weapon delivery scenarios being examined during these first CAS-related flights this implies that both the FAC and the aircraft must use the same reference data. This becomes even more significant for future CAS flights where the database will be used for low level flights aimed at improved terrain following and terrain masking.

During these flight tests several algorithm modifications were identified which will result in significantly improved performance. In particular, the mode control switch logic only iterates every 32 updates (about every 10 seconds at the 3 Hz update rate), and if this were continually monitored, slightly faster acquisition performance would be obtained. More significantly, it was observed that over the rough terrain in the mountains, the combined map and altimetry errors were larger than those for which the filter was tuned (15 meters), while over the smoother regions, these errors are somewhat smaller. This can be explained by the fact that it is easier to accurately map the smoother areas, and in the rougher terrain the radar altimeter does not necessarily range to the point directly below the aircraft. Sandia has conducted simulations of this variable gain filter (Reference 9) using the flight data from FCF 1, indicating track accuracy improvement over a 1000 second portion of this flight from 100 meters to 60 meters (reducing the overall CEP figure for FCF # 1 from 75 meters to 58 meters), and acquisition time/distance improvements from 320sec/48km to 103sec/12.8km. The effect these improvements would have on the flight test results are shown by the parenthetical figures in Table 1. These improvements will be mechanized for the future CAS flight tests.

TERRAIN-AIDED PASSIVE ESTIMATION FOR TARGET ACQUISITION

The terrain elevation data base provided by the DTMDS will be used in the CAS target acquisition and weapon delivery applications with line-of-sight (LOS) angles provided by passive sensors to calculate range-to-target and target geodetic location. The avoidance of active radiating sensors, at least during the initial target acquisition process provides the aircraft with increased survivability. Also, use of the data processing described here allows less accurate, wider field-of-view (FOV) devices, such as the helmet mounted sight, to provide initial approximate ranging and target location which can then be more effectively handed off to the more accurate narrow FOV sensor tracker system, which suffers from the "soda straw" effect limiting pointing ability. When the target has been more accurately located, the active laser ranger can then be used to further reduce the errors for the final stage of weapon delivery. This allows the STS to more accurately acquire the target, reducing search areas and eliminating the need for spurious radiation dangerous to aircraft survivability.

The terrain-aided passive estimation (TAPE) function (Reference 11) projects the LOS vector into the terrain data base as shown in Figure 6, with multiple measurements being combined in a three-state extended Kalman filter. Use of the Kalman filter will minimize the mean-square error provided the aircraft dynamics and the measurement errors are accurately modeled. Figure 7 is a top-level block diagram of this Kalman filter implementation, with Figure 8 defining the reference coordinates, filter states, measurements and measurement transformation matrix. This solution is a unique application of Kalman filtering to passive ranging in that it uses target range, found from the intersection of the LOS vector with the digital terrain data, as a new measurement quantity for the filter. In addition to minimizing the mean-square error, the Kalman filter has a number of other properties including:

1. Fusion of multiple measurements from additional sensors can readily be accommodated.
2. The Kalman filter provides a convenient real time measure of the estimation accuracy through the covariance matrix.
3. The filter will automatically perform time sequenced bearing intersection (triangulation) as the aircraft maneuvers.
4. The filter gains can easily be adapted to reflect terrain conditions (measured by the "terrain sniffer" shown in Figure 7).
5. Similar to SITAN, the filter gains are a function of the terrain slopes (see Figure 8) in the vicinity of the target.

A three state, position only, filter has been developed for stationary targets, however, velocity states can be added to track moving targets, using a six state filter with a constant velocity target model. Maneuvering targets can be tracked using a multiple model Kalman filter scheme.

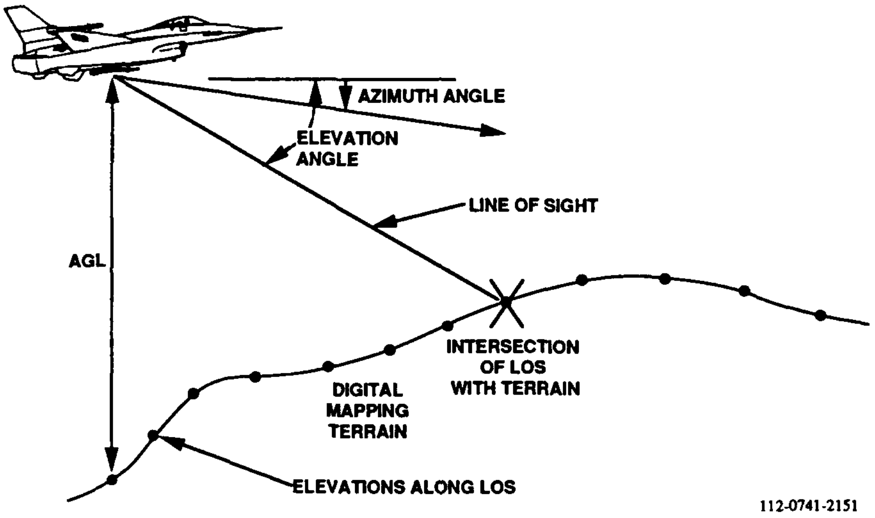


Figure 6. Geometry for Terrain-aided Passive Estimation.

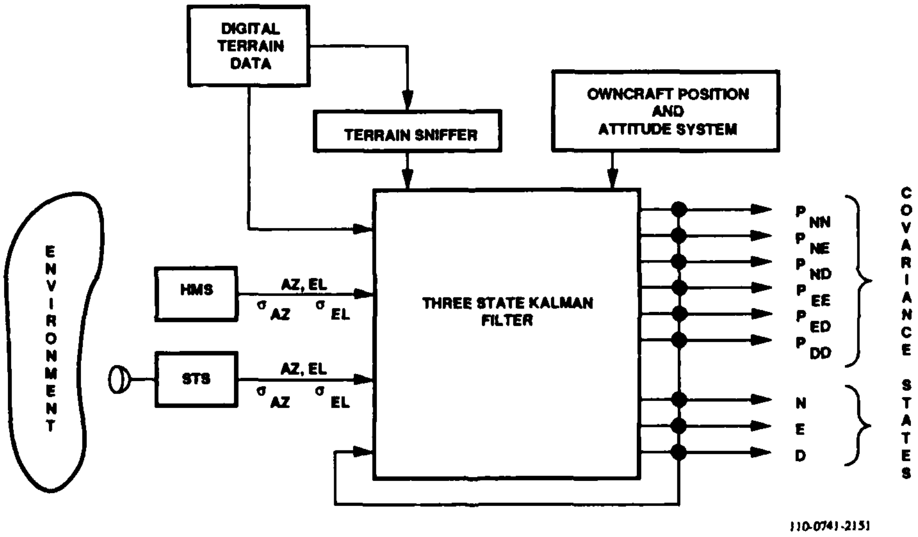


Figure 7. The TAPE Kalman Filter.

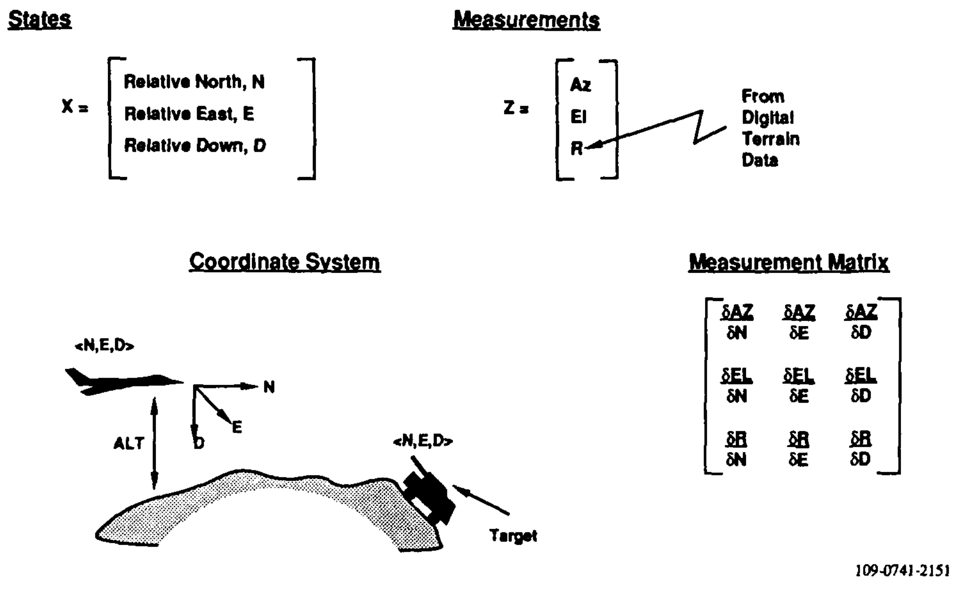

States
$X = \begin{bmatrix} \text{Relative North, N} \\ \text{Relative East, E} \\ \text{Relative Down, } D \end{bmatrix}$
Measurements
$Z = \begin{bmatrix} Az \\ El \\ R \end{bmatrix}$
From Digital Terrain Data
Coordinate System
<N,E,D>
N
E
D
ALT
<N,E,D>
Target
Measurement Matrix
$\begin{bmatrix} \frac{\delta AZ}{\delta N} & \frac{\delta AZ}{\delta E} & \frac{\delta AZ}{\delta D} \\ \frac{\delta EL}{\delta N} & \frac{\delta EL}{\delta E} & \frac{\delta EL}{\delta D} \\ \frac{\delta R}{\delta N} & \frac{\delta R}{\delta E} & \frac{\delta R}{\delta D} \end{bmatrix}$
109-0741-2151

Figure 8. TAPE Kalman Filter Parameters.

Both closed form analysis and detailed Monte Carlo averaging simulations of this algorithm have been developed to predict performance (Reference 11). For this analysis an important derived parameter, the effective grazing angle, has been defined. This angle is the angle made by the LOS to the earth's surface at the target's location, and is simply the sum of the sensor elevation (or depression) angle and the terrain slope at the target location. When this effective grazing angle becomes small, these errors become large. The Monte Carlo simulation was exercised placing the target at six different target locations within a terrain database of Edwards AFB using three different sensor configurations. For the analysis to be presented, the helmet mounted sight and sensor tracker system have error characteristics shown in Table 2. This is typical of the target acquisition system being developed for the AFTI/F-16. Figure 9 is a matrix of the results derived from a single measurement from the respective sensor. Specific sensor configurations for these results included: a 1 mrad sensor tracker at 10 nm and 3000 feet above the target; a 10 mrad helmet mounted sight at 3 nm and 1000 feet above the target; and a 1 mrad sensor tracker at 2 nm and 500 feet above the target.

Table 2. Simulation Parameters.

Passive Sensor Description

Sensor	Angular Accuracies σ_{Az}, σ_{El}	Rate	Target Detection Range
Helmet Mounted Sight (HMS)	10 mrad	10 Hz	3 nm
Sensor Tracker System	1 mrad	10 Hz	10 nm

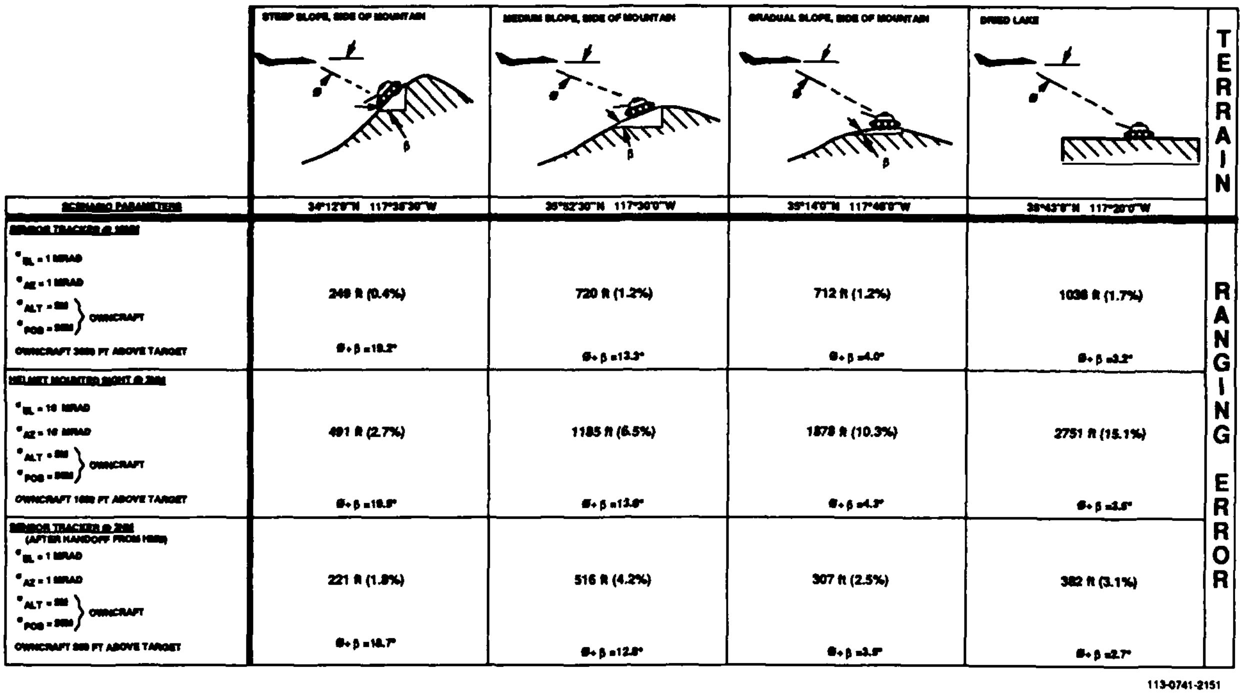
Owncraft Location Accuracies

Altitude	=	5 m	(1 Sigma)
Position	=	50 m	(1 Sigma)

Digital Terrain Data

Post Spacing = 100 m

114-0741-2151



SCENARIO PARAMETERS	STEEP SLOPE, SIDE OF MOUNTAIN 34°12'0"N 117°38'30"W	MEDIUM SLOPE, SIDE OF MOUNTAIN 35°52'30"N 117°30'0"W	GRADUAL SLOPE, SIDE OF MOUNTAIN 35°14'0"N 117°46'0"W	DRIED LAKE 35°43'0"N 117°20'0"W	TERRAIN
SENSOR TRACKER @ 10NM σ_EL = 1 MRAD σ_AZ = 1 MRAD σ_ALT = 5M, σ_POS = 50M } OWNCRAFT OWNCRAFT 3000 FT ABOVE TARGET	248 ft (0.4%) θ + β = 19.2°	720 ft (1.2%) θ + β = 13.3°	712 ft (1.2%) θ + β = 4.0°	1036 ft (1.7%) θ + β = 3.2°	RANGING ERROR
HELMET MOUNTED SIGHT @ 3NM σ_EL = 10 MRAD σ_AZ = 10 MRAD σ_ALT = 5M, σ_POS = 50M } OWNCRAFT OWNCRAFT 1000 FT ABOVE TARGET	491 ft (2.7%) θ + β = 19.5°	1185 ft (6.5%) θ + β = 13.6°	1878 ft (10.3%) θ + β = 4.3°	2751 ft (15.1%) θ + β = 3.5°	
SENSOR TRACKER @ 2NM (AFTER HANDOFF FROM HMS) σ_EL = 1 MRAD σ_AZ = 1 MRAD σ_ALT = 5M, σ_POS = 50M } OWNCRAFT OWNCRAFT 500 FT ABOVE TARGET	221 ft (1.8%) θ + β = 18.7°	516 ft (4.2%) θ + β = 12.8°	307 ft (2.5%) θ + β = 3.5°	382 ft (3.1%) θ + β = 2.7°	

Figure 9. Monte Carlo SImulation Results for Various Locations.

The simulation results shown in Figure 9 point out the effect of target location on algorithm performance. Depending on target location within the terrain, ranging accuracies can vary from a few hundred feet to several thousand feet. Exercising the simulation establishes the fact that a single measurement will produce good results in behaved terrain, but degraded results in flat terrain where the effective grazing angle is small. For instance, on the side of a mountain with a steep slope the target can be ranged within 250 feet by a Sensor Tracker at 10 nm, whereas if the target were on a flat, dry lake the ranging accuracy is reduced to 1000 feet for the same sensor accuracies.

A simulation of the Kalman filter processing a set of time sequenced measurements for a target acquisition and sensor hand-off scenario was also developed. Results generated from this Monte Carlo simulation of this algorithm are shown in Figure 10. The scenario for these simulation runs as the owncraft starting at 3 nautical miles from the targets and flying directly at the targets at 300 m/s (580 knots) at an altitude of 2000 feet above the targets. One target is located on the side of a steeply sloped mountain and the other target is located on the gradual slope of a hill. The locations are the same as presented in Figure 9 for the single measurement analysis. The targets are initially seen by the 10 mrad helmet mounted sight (HMS), and two seconds later detected by the 1 mrad Sensor Tracker System (STS). Both sensors are operating at 10 hertz. The owncraft position inaccuracy is 50 meters, with a 5 meter error in altitude. Examination of Figure 10 shows the immediate advantage of multiple measurement processing using the Kalman filter. The filter quickly refines the position estimates from the HMS for both terrain cases. Then, at two seconds, the STS data is fused to the processed HMS data to further improve estimation accuracy in both terrain conditions. In this scenario there was no owncraft maneuvering, since the owncraft was flying straight at the target. With a maneuver, or a flight path somewhat tangential to target, observability would be gained and triangulation would help the estimation. Note that the limiting errors here (approximately 150-200 feet) are due to the owncraft position uncertainities. At this stage in the target acquisition sequence the sensors can be accurately pointed, reducing the threat to covertness and survivability of turning on the active, narrow beam laser ranger. These accurate range measurements can also be accommodated by the TAPE Kalman filter mechanization, further reducing the target location errors for weapon delivery.

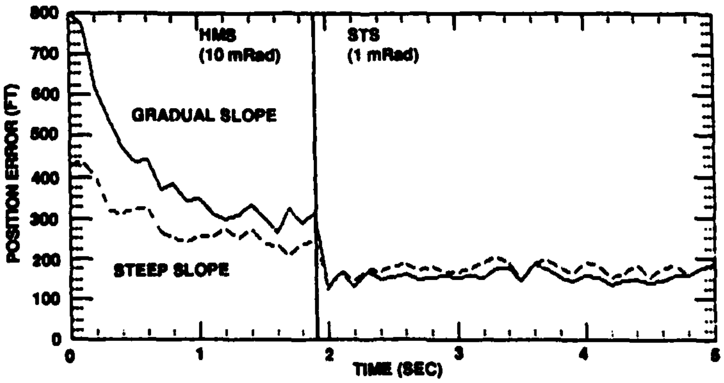


Figure 10. Results for Simulated Target Hand-Off Scenario.

FLIGHT DEMONSTRATION

The goal of the CAS demonstration program's first phase was to show that digital datalink and digital terrain systems could be integrated onboard a CAS aircraft to accurately cue the pilot and onboard sensors to a CAS target location, thereby improving first pass target acquisition. Figure 11 illustrates the essential components of the CAS mission which were demonstrated. At the beginning of this phase the Automatic Target Handoff System (ATHS) was installed in the aircraft to permit databurst transmissions with ground forward air controllers and Army OH-58D helicopters. The ATHS was used to pass target location, target type, and target strength to the AFTI/F-16. The AFTI/F-16 fire control computer used this information along with precise aircraft position data from SITAN to cue the pilot's eyes to the target using the target designation box in the head-up display (HUD) or the helmet mounted sight. Figure 12 illustrates this accurate target cueing resulting from the precise terrain-aided navigation. The fire control computer also cued the sensor tracker set to the target upon pilot consent. Upon receipt of the ATHS messages the AFTI/F-16 color digital map would automatically display IP, target, and friendly force location information, increasing pilot situational awareness. Target attack was accomplished using either conventional F-16 weapon delivery systems or the AMAS. Enhancements to the sensor tracker set were made to permit the pilot to acquire targets at greater distances and to improve the performance of the AMAS, giving a greater probability of a successful first pass attack.

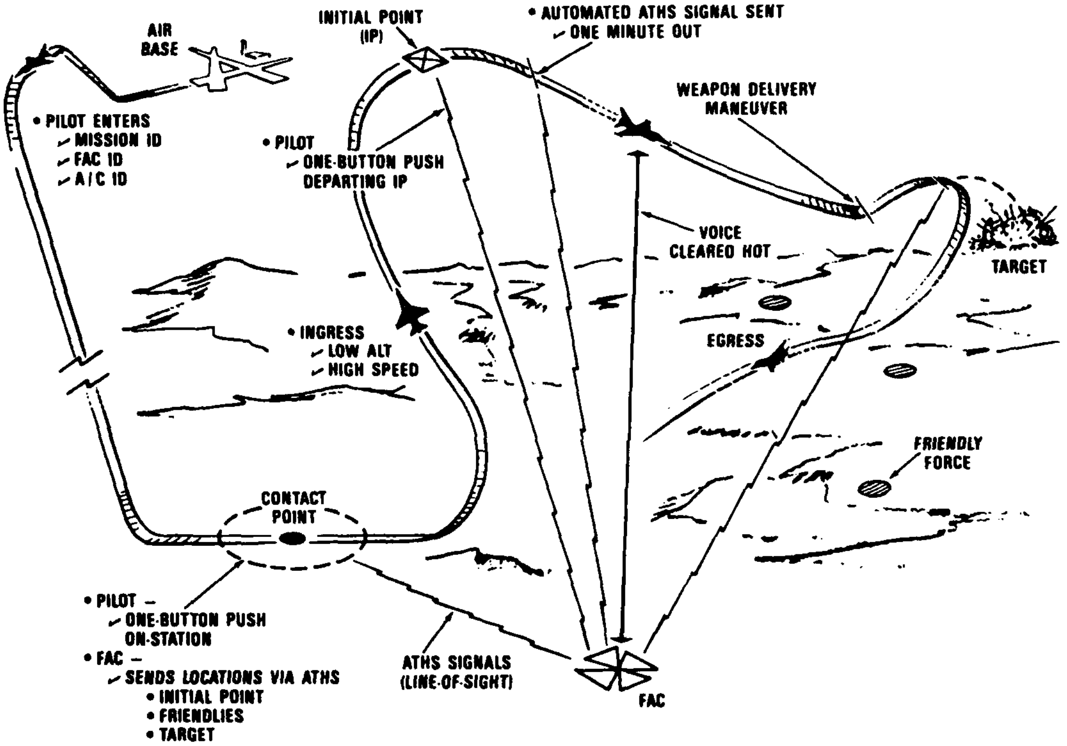
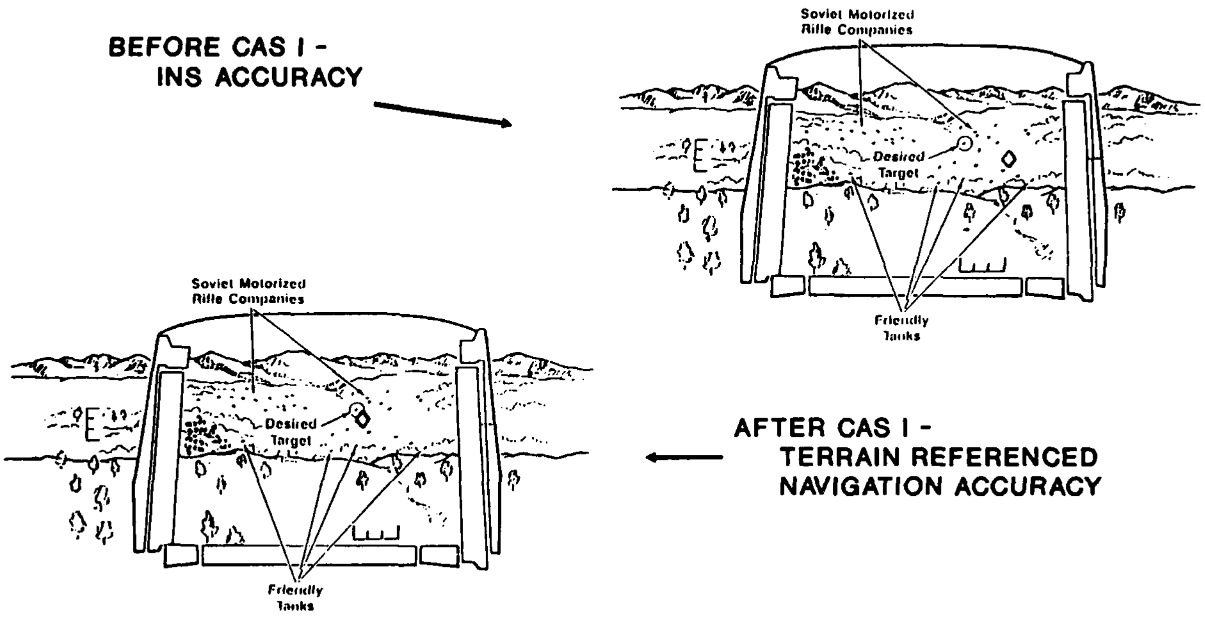


Figure 11. The Demonstration Flight Elements.

The current sequence of CAS flight tests of the AFTI/F-16 began 15 January 1988. A total of 40 test flights were flown. CAS Phase I flight testing ended on 14 July 1988 with two successful joint Army/Air Force CAS demonstration flights. These flights were conducted at the Superior Valley Tactical Bombing Range CA and used an OH-58 helicopter from Ft. Hood Texas, a USAF Ground Forward Air Controller (GFAC) from George AFB CA, and the AFTI/F-16 in a tactical scenario. The use of the Automatic Target Handoff System (ATHS) eliminated voice command and control communications. Accurate target information from the OH-58D, combined with data from the terrain reference navigation system on the AFTI/F-16, permitted the pilot to acquire targets on the first pass in all 12 target runs. The Head Up Display (HUD) and Helmet Mounted Sight (HMS) were used to cue the pilot's eye to the correct target. Mission data datalinked from the OH-58D and GFAC was correctly and automatically presented in the cockpit on the HUD, digital map, and other displays, reducing pilot workload and greatly increasing pilot situational awareness. Representatives for the Tactical Air Command (TAC), and the other service components attended these demonstrations. Data collected from the flight test program is currently being used by all services to identify future capabilities and equipment necessary to better perform the ground attack mission.

The July 1988 CAS demonstration was the first time a CAS mission employing Army helicopter and Air Force ground and air assets was performed using the ATHS datalink instead of voice. These datalink operations reduced the time to target from the initial fighter on-station call by 33 percent. This was accomplished through the elimination of voice communications and the need for the pilot to copy and manually enter CAS mission data into the aircraft's fire control and navigation systems. The OH-58D/AFTI/F-16 operations demonstrated joint battlefield interoperability, and added a new dimension to the Air Land Battle concept using existing, field equipment. The integration of datalink and digital terrain systems demonstrated a markedly improved ability for the fighter pilot to tactics without these technologies historically have a 40 to 50 percent chance of finding similar targets on the first pass. All targets were acquired and attacked on the first attempt during both CAS demonstration flights through accurate target cues and increased pilot situational awareness.



TERRAIN REFERENCED NAVIGATION ACCURACY SUPPORTS AUTOMATIC SENSOR CUEING

Figure 12. Target Cueing for the Demonstration.

FUTURE PHASES OF THE CAS DEMONSTRATION PROGRAM

The primary goal of Phase II of the CAS program is to further exploit the capabilities of digital terrain systems. Aircraft sensor lines of sight and ground threat intervisibilities will be calculated and displayed on the color moving map to inform the pilot when he can "see" a target or if he can be "seen" or engaged by ground threats. The addition of this information will make the digital map a mission planning and management aid to the pilot, allowing him to manually plan and execute a more survivable penetration and attack. Additionally, a terrain following algorithm will be added to the digital terrain system to allow the pilot to manually fly a covert terrain following path based on the digital terrain database. The final development in Phase II for the digital terrain system is the mechanization of a DTS based automated ground collision avoidance system.

Phase II also seeks to improve pilot tactical situational awareness and effectiveness during formation flight operations. A series of flight tests will use two aircraft equipped with the ATHS and complimentary digital terrain systems. Target information collected from aircraft sensors will be transmitted over the datalink for "hunter/killer" type missions and for mutual support. The AFTI/F16 pilot will use the ATHS to pass information on targets of opportunity or "pop-up" threats from the aircraft to any other ATHS equipped vehicle. Look angles from the aircrafts to such a target or threat can be calculated using the STS, helmet mounted sight, or HUD target designator box. Target coordinates will then be calculated using the digital terrain system since aircraft position is known within the digital database. An ATHS message will then be developed for transmission to the other aircraft in the formation. Formation aircraft positions transmitted ia the datalink and displayed on digital maps will allow intraflight attack coordination and station-keeping Better battlefield coordination will result from more extensive internetting with ground forces during joint service demonstrations.

A third goal is to further improve first pass target acquisition and kill through improvements in AFTI/F-16 sensor capabilities allowing sensor-aided target acquisition. Thermal cueing and target screening will be new STS capabilities to help the pilot acquire ground targets. A laser spot tracker receiver will also be installed on the aircraft to locate targets that are laser designated by ground forces. Installation of an APG-68 radar will improve ground attack capability through use of its improved air-to-ground ranging and ground moving target indicator. Previous work in sensor fusion by the AFTI/F-16 program will continue to exploit the synergistic effects resulting from integrating multiple targeting sensors. Finally, the steps necessary to perform ground attack missions at night will be implemented. Installation of a navigation FLIR, raster capable HUD, night cockpit lighting, and an integrated, ejection-safe night vision helmet system will give the AFTI/F-16 a basic night-under-the-weather penetration capability when used with the DTS manual terrain following system.

A primary goal of Phase III of the program is survivable, covert, day/night penetration using a 5G maneuvering terrain following/terrain avoidance/threat avoidance (TF/TA/ThA) system based on digital terrain and sensor data fusion. Sensor look into the turn capability will give critical obstacle and ground collision avoidance. Automated route planning provided by the digital terrain system will give a measure of automatic terrain avoidance and avoidance of known ground threats. Use of passive or accurately cued limited emissions sensors will reduce enemy detection while permitting safe under-the-weather flight. The combination of integrated night vision helmet, terrain perspective displays, and sensor displays will allow the pilot to perform high-G, low-altitude, maneuvering, day/night CAS operations with confidence.

Additional goals of this final phase are an all-terrain automated maneuvering attack system and the development of the sensor capabilities and avionics suite required to support multiple target engagements per pass. Like the automated TF/TA/ThA system, this system will rely on the digital terrain system supported by the other aircraft sensors. Accurate standoff deliveries will insure first pass target kill while increasing survivability. Passive ranging using the STS and digital terrain system will permit covert attack. A new tracker for the FLIR STS will permit the tracking, classification, and prioritization of up to six targets. Other AMAS enhancements will include provisions for precision weapon systems such as the hypervelocity missile (HVM), low-level laser guided bomb (LLLGB), and infrared Maverick missile. The aircraft will then be postured to integrate these advanced avionic systems with advanced weapons technologies such as Rapid Fire Maverick, Sensor-Fuzed Weapons, and Hypervelocity Missiles as a possible follow-on phase.

ACKNOWLEDGEMENTS

The AFTI/F-16 flight test program is a multi-year, multi-service and multi-contractor program, and as such, a large number of people and organizations have made contributions to the work described in this paper. The authors would like to acknowledge the following support. The service organizations involved include the U. S. Air Force, Navy and Army, NASA, and SANDIA National Laboratories. The contractor team is lead by General Dynamics and includes Harris Corporation and Rockwell/Collins. Individual contributors from the Air Force Wright Research and Development Center include Pat Boone, Bill Young and Lt. Paul K. Rose. Mike Dante, Jim Harris, Capt. Greg Hoglan, and Lt. Col. Mark Stubben of the Flight Test Center at Edwards assisted in the measurement and analysis of the flight data. Dr. Drayton Boozer and Rick Fellerhoff of SANDIA Labs developed and provided AFTI/SITAN data analysis. Mike Griswold, Stan McClellan, and Bob Psencik of General Dynamics have provided analytical support, system definition and technical leadersihp. Finally, Jim Gracia, Gordon Morrison, and Vicky Vickers of Harris provided support in the TAPE algorithm development, and Barb Sella in the preparation of the manuscript.

REFERENCES

1. Cantrel, G. W., Application of Digital Terrain Data in Flight Operations, NATO AGARD Symposium, The Hague, Netherlands, May 1985.
2. Baird, C. A. and F. B. Snyder, Terrain Elevation Based Sensor Fusion for Attack Aircraft, NATO AGARD Symposium, London, Oct. 1986.
3. Drew, M., Military Flight Test Programs - An Overview, SAE Aerotech 88, Anahiem, CA, Oct. 1988.
4. Baird, C.A. , N. Collins and O. Doty, Onboard Digital Terrain Data: AFTI/F-16 Flight Test Results and Close Air Support Enhancements, SAE Aerotech 88, Anahiem, CA, Oct. 1988.
5. Baird, C. A. and M. R. Abramson, A Comparison of Several Digital Map-aided Navigation Techniques, IEEE Position Location and Navigation Symposium, San Diego, CA., Nov. 1984.
6. Baird, C. A., Design Techniques for Improved Map-aided Navigation, NAECON-85, Dayton, OH., May 1985
7. Abramson, M. R. and C. A. Baird, Trajectory Estimation from Flight Test Data, Memorandum Report #56, Advanced Technology Dept., Harris Government Systems, Melbourne, Fla., Oct. 1983.
8. Boozer, D. D., et. al., SITAN Design for Low-level Attack Aircraft, SAND84-2059, Sandia Labs., Albuquerque, NM, May 1985.
9. Fellerhoff, J. R., AFTI/SITAN Final Report, SAND88-7325, Sandia Labs, Albuquerque, NM, Nov. 1988.
10. Defense Mapping Agency, Product Specifications for Digital Landmass System (DLMS) Data Base, DMA Aerospace Center, St. Louis AFS, MO, Various Editions.
11. Collins, N. and C. Baird, Terrain Aided Passive Estimation, NAECON-89, Dayton, OH, May 1989.

REPORT DOCUMENTATION PAGE

1. Recipient's Reference	2. Originator's Reference	3. Further Reference	4. Security Classification of Document
	AGARD-CP-455	ISBN 92-835-0535-2	UNCLASSIFIED

5. Originator

Advisory Group for Aerospace Research and Development
North Atlantic Treaty Organization
7 rue Ancelle, 92200 Neuilly sur Seine, France

6. Title

ADVANCES IN TECHNIQUES AND TECHNOLOGIES FOR AIR VEHICLE NAVIGATION AND GUIDANCE

7. Presented at

the Guidance and Control Panel 48th Symposium held at the Instituto da Defesa Nacional in Lisbon, Portugal from 9th to 12th May, 1989.

8. Author(s)/Editor(s)	9. Date
Various	December 1989
10. Author's/Editor's Address	11. Pages
Various	206

12. Distribution Statement

This document is distributed in accordance with AGARD policies and regulations, which are outlined on the Outside Back Covers of all AGARD publications.

13. Keywords/Descriptors

Navigation
Guidance
Terrain reference navigation
Image processing
GPS
Sensor management
Sensor fusion
Expert systems
Inertial sensors
MMW/IR seeker

14. Abstract

This volume contains 16 unclassified papers out of the 31 presented at the Guidance and Control Panel Symposium held in Lisbon from 9 to 12 May 1989. The remainder of the papers (except papers Nos 31, 41, 64, not available at time of printing) are included in the classified supplement (CP-455(S)).

The papers were presented covering the following headings: Terrain reference navigation methods; Positioning by image processing or GPS; Mission and sensor management; New techniques and algorithms; Sensor technology; Systems applications.

AGARD Conference Proceedings No.455
Advisory Group for Aerospace Research and Development, NATO
ADVANCES IN TECHNIQUES AND TECHNOLOGIES FOR AIR VEHICLE NAVIGATION AND GUIDANCE
Published December 1989
206 pages

This volume contains 16 unclassified papers out of the 31 presented at the Guidance and Control Panel Symposium held in Lisbon from 9 to 12 May 1989. The remainder of the papers (except papers Nos 31, 41, 64, not available at time of printing) are included in the classified supplement (CP-455(S)).

P.T.O.

AGARD-CP-455

Navigation
Guidance
Terrain reference navigation
Image processing
GPS
Sensor management
Sensor fusion
Expert systems
Inertial sensors
MMW/IR seeker

AGARD Conference Proceedings No.455
Advisory Group for Aerospace Research and Development, NATO
ADVANCES IN TECHNIQUES AND TECHNOLOGIES FOR AIR VEHICLE NAVIGATION AND GUIDANCE
Published December 1989
206 pages

This volume contains 16 unclassified papers out of the 31 presented at the Guidance and Control Panel Symposium held in Lisbon from 9 to 12 May 1989. The remainder of the papers (except papers Nos 31, 41, 64, not available at time of printing) are included in the classified supplement (CP-455(S)).

P.T.O.

AGARD-CP-455

Navigation
Guidance
Terrain reference navigation
Image processing
GPS
Sensor management
Sensor fusion
Expert systems
Inertial sensors
MMW/IR seeker

AGARD Conference Proceedings No.455
Advisory Group for Aerospace Research and Development, NATO
ADVANCES IN TECHNIQUES AND TECHNOLOGIES FOR AIR VEHICLE NAVIGATION AND GUIDANCE
Published December 1989
206 pages

This volume contains 16 unclassified papers out of the 31 presented at the Guidance and Control Panel Symposium held in Lisbon from 9 to 12 May 1989. The remainder of the papers (except papers Nos 31, 41, 64, not available at time of printing) are included in the classified supplement (CP-455(S)).

P.T.O.

AGARD-CP-455

Navigation
Guidance
Terrain reference navigation
Image processing
GPS
Sensor management
Sensor fusion
Expert systems
Inertial sensors
MMW/IR seeker

AGARD Conference Proceedings No.455
Advisory Group for Aerospace Research and Development, NATO
ADVANCES IN TECHNIQUES AND TECHNOLOGIES FOR AIR VEHICLE NAVIGATION AND GUIDANCE
Published December 1989
206 pages

This volume contains 16 unclassified papers out of the 31 presented at the Guidance and Control Panel Symposium held in Lisbon from 9 to 12 May 1989. The remainder of the papers (except papers Nos 31, 41, 64, not available at time of printing) are included in the classified supplement (CP-455(S)).

P.T.O.

AGARD-CP-455

Navigation
Guidance
Terrain reference navigation
Image processing
GPS
Sensor management
Sensor fusion
Expert systems
Inertial sensors
MMW/IR seeker

The papers were presented covering the following headings: Terrain reference navigation methods; Positioning by image processing or GPS; Mission and sensor management; New techniques and algorithms; Sensor technology; Systems applications.

ISBN 92-835-0535-2

The papers were presented covering the following headings: Terrain reference navigation methods; Positioning by image processing or GPS; Mission and sensor management; New techniques and algorithms; Sensor technology; Systems applications.

ISBN 92-835-0535-2

The papers were presented covering the following headings: Terrain reference navigation methods; Positioning by image processing or GPS; Mission and sensor management; New techniques and algorithms; Sensor technology; Systems applications.

ISBN 92-835-0535-2

The papers were presented covering the following headings: Terrain reference navigation methods; Positioning by image processing or GPS; Mission and sensor management; New techniques and algorithms; Sensor technology; Systems applications.

ISBN 92-835-0535-2

AGARD

NATO OTAN

7 rue Ancelle · 92200 NEUILLY-SUR-SEINE

FRANCE

Telephone (1)47.38.57.00 · Telex 610 176

DISTRIBUTION OF UNCLASSIFIED AGARD PUBLICATIONS

AGARD does NOT hold stocks of AGARD publications at the above address for general distribution. Initial distribution of AGARD publications is made to AGARD Member Nations through the following National Distribution Centres.Further copies are sometimes available from these Centres, but if not may be purchased in Microfiche or Photocopy form from the Sales Agencies listed below.

NATIONAL DISTRIBUTION CENTRES

BELGIUM
Coordonnateur AGARD — VSL
Etat-Major de la Force Aérienne
Quartier Reine Elisabeth
Rue d'Evere, 1140 Bruxelles

CANADA
Director
Dept of :
Ottawa,

DENMARK
Danish I
Ved Idra
2100 Co

FRANCE
O.N.E.R
29 Aven
92320 C

GERMANY
Fachinfo
Physik, N
Karlsruh
D-7514 I

GREECE
Hellenic .
Aircraft S
Departme.
Holargos, Athens, TGA 1010

ICELAND
Director of Aviation
c/o Flugrad
Reyjavik

ITALY
Aeronautica Militare
Ufficio del Delegato Nazionale all'AGARD
3 Piazzale Adenauer
00144 Roma/EUR

LUXEMBOURG
See Belgium

NETHERLANDS
Netherlands Delegation to AGARD
National Aerospace Laboratory, NLR

nent

ARD

Ankara

UNITED KINGDOM
Defence Research Information Centre
Kentigern House
65 Brown Street
Glasgow G2 8EX

UNITED STATES
National Aeronautics and Space Administration (NASA)
Langley Research Center
M/S 180
Hampton, Virginia 23665

THE UNITED STATES NATIONAL DISTRIBUTION CENTRE (NASA) DOES NOT HOLD STOCKS OF AGARD PUBLICATIONS, AND APPLICATIONS FOR COPIES SHOULD BE MADE DIRECT TO THE NATIONAL TECHNICAL INFORMATION SERVICE (NTIS) AT THE ADDRESS BELOW.

SALES AGENCIES

National Technical
Information Service (NTIS)
5285 Port Royal Road
Springfield
Virginia 22161, USA

ESA/Information Retrieval Service
European Space Agency
10, rue Mario Nikis
75015 Paris, France

The British Library
Document Supply Centre
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
England

Requests for microfiche or photocopies of AGARD documents should include the AGARD serial number, title, author or editor, and publication date. Requests to NTIS should include the NASA accession report number. Full bibliographical references and abstracts of AGARD publications are given in the following journals:

Scientific and Technical Aerospace Reports (STAR)
published by NASA Scientific and Technical
Information Branch
NASA Headquarters (NIT-40)
Washington D.C. 20546, USA

Government Reports Announcements (GRA)
published by the National Technical
Information Services, Springfield
Virginia 22161, USA

SPS

Printed by Specialised Printing Services Limited
40 Chigwell Lane, Loughton, Essex IG10 3TZ