During the past year three (3) research papers were written and two (2) published conference presentations were given.

Titles of the published research articles are: "A Stochastic Analysis of a Modified Gain Extended Kalman Filter with Applications to Estimation with Bearings only Measurements," "The Modified Gain Extended Kalman Filter and Parameter Identification in Linear Systems"; and "Maximum Information Guidance for Homing Missiles".
Captain John Thomas, Jr.
Air Force Office of Scientific Research
Bolling Air Force Base
Building 410
Washington, D.C. 20332-6448

Dear Captain Thomas:

Enclosed is a summary of the research performed under AFOSR Grant F49620-85-K-0008. When I sent the descriptive portion to Dr. Fox in July, I thought that I had satisfied the contract's requirement for a final technical summary. Our Office of Sponsored Programs informed me that your contracting personnel had not received this document so I am submitting another copy. Sorry for this confusion. Thanks for your continued support of this work.

Sincerely,

John Gannon
Associate Professor

cc: E. Magrum
This research program focuses on improving programming productivity through better methodologies and more powerful programming environments. New environments are being developed and evaluated empirically to see that they meet their goals.

The availability of high performance workstations has led to increased research on their use for enhanced programmer productivity. Toward this end the SUPPORT environment has been implemented to investigate such issues. SUPPORT executes on a VAX 11/780 under UNIX 4.2, on SUN Workstations, and on an IBM PC under PC-DOS. SUPPORT is an environment for developing and testing Pascal programs. (Ada and C versions of SUPPORT are also being considered). Issues under study are (1) extended grammars to convey semantic information, (2) workstations with powerful interactive interpreters, (3) multiple windows, and (4) the effectiveness of syntax-directed editing in code production and modification.

Distributed programs promise improved efficiency (through processors executing concurrently) and reliability (through the use of independent processors). Remote procedure call provides users with transparent service, permitting them to call procedures on processors without shared storage just as they would call procedures on the same processor. We have implemented an atomic remote procedure call mechanism as an extension to the C programming language on ZMOB, a 256-processor system. Concurrent procedure calls are mediated by attaching a call graph path identifier to each call message. We have developed conditions on path identifiers that permit calls to proceed concurrently and still be serialized. Each procedure call is a total operation, with associated states of procedures saved on procedure entry and restored in case of procedure crash.

CleanRoom integrates the use of a mathematically-based design methodology, “right-the-first-time” programming methods, and a statistically-based testing strategy. Developers are not allowed to test their own programs. They focus on review techniques, such as code reading, inspections, and walkthroughs, to assert the correctness of their systems. Independent testers then simulate the operational environment of the product with functional testing, record observed failures, and determine an objective measure of system reliability. Fifteen three-person teams, working separately, built 1200-line message systems to compare CleanRoom software development with a more traditional development approach. The results demonstrate the feasibility of complete off-line development (as in CleanRoom) and suggest that such a development approach is superior to a more traditional approach.

Another study compared the strategies of code reading, functional testing, and structural testing in three aspects of software testing: fault detection effectiveness, fault detection cost, and classes of faults detected. Thirty-two professional programmers and forty-two advanced students applied the three techniques to programs. The professional programmers detected more software faults and had a higher fault detection rate using code reading than with functional or structural testing. In both groups, functional testing was generally superior to structural testing.
AFOSR-Supported Work, 1984


END
DATE
FILMED
DEC.
1987