Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE DÉFENSE

Design of a holonic control architecture for distributed sensor management

A. Benaskeur
H. Irandoust
DRDC Valcartier

P. McGuire
C-core

Defence R&D Canada – Valcartier
Technical Report
DRDC Valcartier TR 2009-056
September 2009

Canada

Design of a holonic control architecture for distributed sensor management

A. Benaskeur
H. Irandoust
Defence R&D Canada – Valcartier

P. McGuire
C-core

Defence R&D Canada – Valcartier

Technical Report

DRDC Valcartier TR 2009-056

September 2009

Principal Author

A. Benaskeur

Approved by

Éloi Bossé
Head/DSS C2 Section

Approved for release by

C. Carrier
Chief Scientist

Abstract

Military operations are typically conducted in demanding, dynamic, semi-structured and large-scale environments. The nature of these operating environments makes it difficult to detect, identify and track all the targets within the Volume Of Interest (VOI). To deal with this problem, sensing resources may have to be distributed across a large area, collecting a wealth of data. Yet, to effectively use that data, the sensing resources need to be properly managed.

This report presents the design of holonic sensor management architecture. It follows two previous documents, which detailed the issues involved in military sensor management and the properties of holonic control, respectively. The holonic control proposed here is a novel approach to sensor management, in that its architecture supports dynamic linkages, thus allowing the achievement of changing objectives.

Résumé

Les opérations militaires sont typiquement conduites dans des environnements exigeants, dynamiques, semi-structurés et à grande échelle. La nature de ces environnements opérationnels rend difficile la détection, l'identification et le pistage de toutes les cibles à l'intérieur d'un volume d'intérêt. Pour s'attaquer ce problème, les ressources de surveillance doivent être distribuées à travers un grand secteur, leur permettant de receuillir beaucoup de données. Cependant, pour exploiter efficacement ces données, les ressources doivent être gérées adéquatement.

Ce rapport présente la conception d'une architecture holonique pour la gestion des capteurs. Il s'inscrit dans le prolongement de deux documents précédents, détaillant respectivement, la problématique de la gestion des capteurs militaires et les propriétés du contrôle holonique. L'architecture holonique proposée ici constitue une approche novatrice au problème de la gestion des capteurs, en cela qu'elle supporte l'établissement de liens dynamiques, permettant ainsi au système d'atteindre des objectifs dynamiques.

This page intentionally left blank.

Executive summary

Design of a holonic control architecture for distributed sensor management

A. Benaskeur, H. Irandoust, P. McGuire; DRDC Valcartier TR 2009-056; Defence R&D Canada – Valcartier; September 2009 .

Historically, interpreting the data and managing the sensors in military operations was done manually; however, this has become difficult, if not impossible, due to the constantly increasing complexity of modern surveillance systems. As sensors increase in complexity and capabilities, their management becomes increasingly challenging and innovative methods will have to be utilized to make effective use of these new surveillance tools.

Sensor management is an automated process that optimizes the utilization of the sensing resources and improves the quality of the acquired data, leading ultimately to better situation awareness. The goal of sensor management is to deliver the appropriate data/information to the appropriate location while balancing the capabilities of the sensor with the quality of the data/information provided.

The classical hierarchical structure for sensor management is effective with systems that operate under relatively stable conditions. However, in situations where dramatic change is the norm, this control and coordination architecture may be too rigid to allow the system to react as quickly as it might with an alternate structure.

Following two previous reports on sensor management issues and the properties of holonic control, this report presents the design of holonic sensor management architecture. The design of sensor management nodes, called holons, and their associated internal components constitute the most significant portion of this document. Three levels of holons and their configurations are considered, *i.e.*, sensor level, platform level and group level.

1. The sensor-level holon is concerned with the control of individual sensors; it is the lowest level of control considered.
2. The platform-level holon forms the heart of the entire system. It is responsible for detecting targets and maintaining target tracks. The platforms are meant to resemble military sensing platforms such as the Halifax Class Canadian frigates.
3. The group-level holon oversees the operation of a number of platforms and coordinates sensing tasks amongst them. This corresponds to a typical configuration of a naval task group or a task force. Due to communication limits between the platforms, the group level does not actively manage target tracks and searches, but acts more as a supervisory control level that delegates tracking and searching tasks to the platforms. It also maintains a central picture of the tactical situation. A major portion of group-level holon functions is concerned with controlling the communications and information exchange, in order to selectively acquire information from the platforms and coordinate them.

The architecture presented here is meant to demonstrate the applicability of the holonic control paradigm to military sensor management and therefore makes a number of simplifying assumptions about military situation analysis, and the underlying surveillance applications. Also, the architecture addresses some of the aspects of military sensor management, but not all of them. These aspects that are addressed include searching and tracking of targets, sensor mode control, sensor allocation, cooperation and control of sensors (handling track cueing and handoff onboard a single platform and between platforms), and communication limitations. Issues such as emission control, platform navigation and various aspects of situation analysis are either overlooked or simplified in order to devise a workable sensor management design.

Sommaire

Design of a holonic control architecture for distributed sensor management

A. Benaskeur, H. Irandoust, P. McGuire ; DRDC Valcartier TR 2009-056 ; Recherche et développement pour la défense Canada – Valcartier ; septembre 2009 .

Historiquement, l'interprétation des données et la gestion des capteurs dans les opérations militaires étaient effectuées manuellement. Or, cela est devenu difficile, voire impossible, en raison de la complexité sans cesse croissante des systèmes modernes de surveillance. À mesure que les capteurs présentent un accroissement de leur complexité et de leurs capacités, leur gestion devient de plus en plus exigeante et des méthodes innovatrices devront être utilisées pour assurer une utilisation efficace de ces nouveaux outils de surveillance.

La gestion des capteurs est un processus automatisé qui optimise l'utilisation des ressources de surveillance et améliore la qualité des données acquises, ce qui résulte au final en une meilleure conscience situationnelle. Le but de la gestion des capteurs est de fournir l'information appropriée à l'endroit approprié, tout en assurant l'équilibre entre les capacités du capteur et la qualité de l'information fournie.

La structure hiérarchique classique pour la gestion des capteurs est efficace avec les systèmes qui fonctionnent dans des conditions relativement stables. Cependant, dans les situations où le changement est la norme, cette structure de contrôle et de coordination peut se réveler trop rigide pour permettre au système de réagir aussi rapidement qu'il le pourrait avec une structure différente.

Suite à deux rapports sur la problématique de la gestion des capteurs et les propriétés du contrôle holonique, ce rapport présente la conception d'une architecture holonique pour la gestion des capteurs. La conception des noeuds de gestion des capteurs, appelés "holons", et leurs composantes internes constitue la majeure partie de ce document. Trois niveaux de holons sont considérés : le niveau capteur, le niveau plateforme et le niveau groupe.

1. Le holon au niveau capteur a trait au contrôle des capteurs individuels ; c'est le niveau le plus bas de la structure de contrôle proposée.
2. Le holon au niveau plateforme constitue le coeur du système dans son ensemble. Il est responsable de la détection des cibles et du maintien des pistes pour les cibles. Les plateformes utilisées ressemblent à des plateformes militaires de surveillance, telles que les frégates canadiennes de classe Halifax.
3. Le holon au niveau groupe supervise les opérations d'un certain nombre de plateformes et coordonne leurs tâches de surveillance. Ceci correspond à une configuration typique d'une force opérationnelle navale. En raison des limites de communication entre les plateformes, le niveau groupe ne contrôle pas directement les opérations de recherche et de pistage des cibles, mais agit plus comme un superviseur, déléguant ces tâches aux plateformes. Il maintient également une image centrale de la situation tactique.

> Une partie importante des fonctions du holon au niveau groupe concerne la gestion des communications et des échanges d'informaiton, afin d'acquérir sélectivement l'information des différentes plateformes et de les coordonner.

L'architecture présentée ici vise à démontrer l'applicabilité du paradigme de contrôle holonique à la gestion des capteurs militaires, et de ce fait, postule un certain nombre d'hypothèses simplificatrices au sujet de l'analyse de la situation dans un cadre militaire et les opérations de surveillance sous-jacentes. En outre, l'architecture porte sur certains aspects de la gestion des capteurs militaires, mais pas tous. Ceux qui sont étudiés incluent : la recherche et le pistage des cibles, le contrôle des modes des capteurs, l'allocation des capteurs, le contrôle et la coopération des capteurs (appel et transfert de piste sur une plateforme individuelle et entre plateformes), et les contraintes de communication. Des questions telles que le contrôle des émissions, la navigation des plateformes et divers aspects de l'analyse de la situation, sont soit délibérement omises, soit simplifiées afin de permettre une conception réalisable.

Table of contents

List of figures

List of tables

This page intentionally left blank.

1 Introduction

The military operations are typically conducted in demanding, dynamic, semi-structured and large-scale environments. The nature of this operating environment makes it difficult to detect, identify and track all the targets within the Volume Of Interest (VOI). To deal with this problem, sensing resources may have to be distributed across a large area. Military platforms can be outfitted with sensing resources, which can potentially provide a wealth of data. Yet, to be effectively used, these sensing resources need to be properly managed [1].

Historically, interpreting the data and managing the sensors was done manually. However, this has become difficult, if not impossible, due to the constantly increasing complexity of modern surveillance systems. As surveillance sensors increase in complexity and capabilities, their management becomes increasingly challenging and innovative methods will have to be utilized to make effective use of these new surveillance tools. *Sensor management* is an automated process that optimizes the utilization of the sensing resources and improves the quality of the acquired data/information, leading ultimately to better situation awareness.

Some of the high-level issues in military sensor management, and in Command and Control (C^2) systems in general, are related to their organizational forms and distributed architectures. These systems are organized along functional lines (*i.e.*, battle management functions) and are typically composed of many geographically distributed decision nodes. System size, heterogeneity, number of inter-relationships, and the volume of data contribute to the complexity of such systems.

In order to meet the required safety, effectiveness and timeliness criteria of decision making, cooperation, coordination and communication among decision nodes, an adequate decomposition of the decision process is critical. The classical hierarchical structure is effective with systems that operate under relatively stable conditions. However, in situations where dramatic change is the norm, this control and coordination architecture may be too rigid to allow the system to react as quickly as it might with an alternate structure.

The objective of the reported work is to present the detailed design of a holonic sensor management capability. The ultimate goal pursed by this on-going effort is to evaluate control architectures and control methods applicable to military sensor management, with a focus on tactical surveillance operations. The current report follows two previous documents detailing the issues involved in military sensor management and the properties of holonic control [1, 2]. In this document, those issues are pulled together to suggest a working combination.

The report is organized as follows. It begins with a brief review of military sensor management and the properties of holonic control (Chapter 2). The holonic sensor management design is presented in Chapters 3, based primarily on Electronically Scanned Antenna (ESA)-type sensors, although other sensor types could easily be included within this framework. First, the general management structure is discussed in Chapter 3. Subsequently, the design details, the roles, and the responsibilities of the three (sensor, platform, group) levels of sensor management are presented in Chapters 4 to 6, respectively. A description of how all the management components function together is given in Chapter 7.

The report is concluded in Chapter 8. Additional information is provided in Annexes A, B, and C.

2 Sensor management and holonic approach

In this chapter we briefly review the sensor management problem and discuss the properties of holonic control. A discussion of the advantages and shortcomings of different system architectures shows the potential that the holonic approach presents for military control problems, such as sensor management. For more details on sensor management and holonic control issues, the reader is referred to [1, 2].

2.1 Sensor management

The objective of any surveillance mission is to gather information about the presence and the activities of all objects within the Volume of Interest (VOI). The information gathered is used to build a representation of the situation of interest. Given that the military typically operate in dynamic, semi-structured and large-scale environments, it becomes difficult to detect and track all targets within VOI, thus increasing the risk of late detection of threatening objects. Yet, military platforms are generally outfitted with a set of sensors that can provide a wealth of data if properly managed.

Efficient sensor management [3, 4, 5, 6] can significantly enhance the process of information gathering by automatically allocating, controlling, and coordinating sensing resources in order to collect the most complete and accurate data from a dynamic scene. Sensor management aids the surveillance process by directing sensing resources as to acquire data that are most relevant to mission objectives.

In the following, we first explain the role of sensor management in the data fusion process and then describe the hierarchical nature of sensor management problems and the control challenges at each level.

2.1.1 Sensor management in the data fusion process

Let us consider the Joint Directors of Laboratories (JDL) [7] model of data fusion. The integration of data fusion and sensor management can be explained as follows: **Level 1** (object assessment) would provide the kinematical descriptions of all of the objects in the environment, **Level 2** (situation assessment) would assess the organizational properties of these objects, while **Level 3** (impact assessment) would indicate which objects are the most important to track (highest priority) and which ones require more or better information. **Level 4** (process refinement) would then consist in assigning and reassigning sensing resources on the basis of this information and the overall mission objectives.

As part of **Level 4** fusion, sensor management acts as an aid to the data fusion process by directing sensing resources in an intelligent fashion. Because the sensors cannot always meet all of the sensing requirements, the management system must decide which priorities to meet and which resources to allocate to those priorities. The idea is to balance long-term objectives with immediate concerns.

The sensing process is often performed following static military operating procedures without feedback from the fusion process. We refer to this as open-loop sensor management. However, by redirecting the sensors on the basis of the real-time data, better measurements can be generated, thereby improving future fused results. This process of data fusion and sensor management can be thought of as a feedback loop, hence the name of closed-loop strategy. In this configuration, decisions regarding the utilization of sensing resources are influenced by the high-level analysis of the situation. Outputs from the fusion process are used by sensor management to make adjustments to the allocation and the operation modes of the sensors, thus closing the loop on the data fusion process.

2.1.2 The hierarchical nature of sensor management problems

The military organization by its very nature has a hierarchical command structure to manage its resources, men, and equipment. Sensor management, a subset of this command structure, is not only hierarchical but also has a fractal property. An example of such a recursive hierarchy is shown in Figure 1 for a typical Naval Task Force configuration.

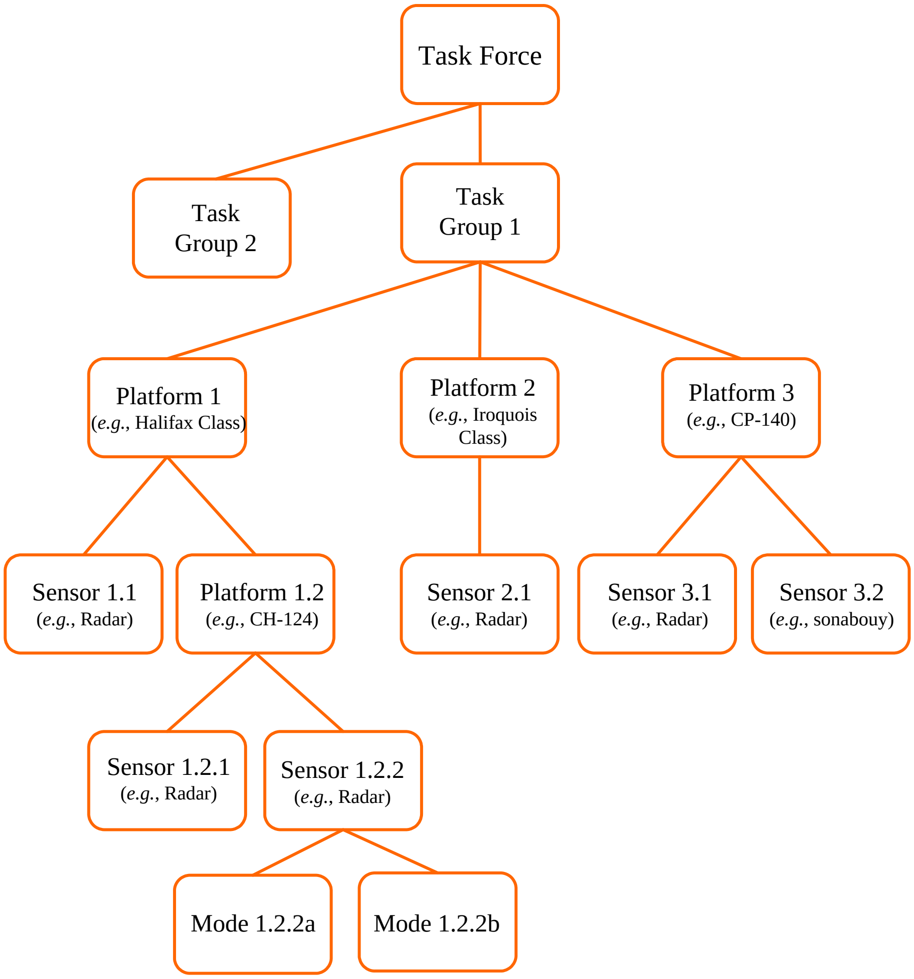

Task Force
Task Group 2
Task Group 1
Platform 1 (*e.g.*, Halifax Class)
Platform 2 (*e.g.*, Iroquois Class)
Platform 3 (*e.g.*, CP-140)
Sensor 1.1 (*e.g.*, Radar)
Platform 1.2 (*e.g.*, CH-124)
Sensor 2.1 (*e.g.*, Radar)
Sensor 3.1 (*e.g.*, Radar)
Sensor 3.2 (*e.g.*, sonabouy)
Sensor 1.2.1 (*e.g.*, Radar)
Sensor 1.2.2 (*e.g.*, Radar)
Mode 1.2.2a
Mode 1.2.2b

Figure 1: Hierarchy of naval sensing resources

The hierarchical relationship between the sensing resource levels follows the command structure. Decision processes occur at each level on the basis of the requirements that flow down from the level above and the information derived from the level below. As one descends the tiers in the hierarchy, the level of responsibility becomes more focused and the volume of data increases. As data move up the hierarchy, they are transformed into information that enables high-level planning and decision making. Sensor management is an element of the decision process that governs the overall behaviour of the sensing resources at all levels.

Figure 2 shows how sensor management high-level objectives are broken down into tasks and sub-tasks. Some of sensor management tasks are briefly described below.

2.1.2.1 Allocation

This is concerned with determining the sensing resource(s) to use to achieve the sensing objectives. Sensor management needs to determine the most suitable resource to allocate to each task. Depending on the task level, the resource can be a sensor, a platform, or a group of platforms.

2.1.2.2 Coordination/Cooperation

If a sensing resource in operation is in conflict with other resources, then sensor management must determine which resource is more important and prevent the others from operating or set up some schedule to allow one resource to operate for a period of time and then some other resource for another period. This defines the coordination or conflict resolution problem. Dual to this is the cooperation problem, where synergy among complementary resources is maximized.

2.1.2.3 Scheduling

Scheduling is the designation of time segments for specific tasks or activities, the nature of which is defined during the allocation or coordination stages. Scheduling typically uses time as its base variable; tasks are expected to start at a specified time and to execute for a fixed time interval.

2.1.2.4 Mode control

In case of sensors offering multiple modes, the sensor management should make use of the most optimal mode for a given task, provided that there is no other overriding reason not to. When changing sensor modes, the data stream may be halted during the transition. The sensor management must address whether it is more important to maintain the operation in a possibly sub-optimal mode to capture a live data stream, or to switch to a more optimal mode.

Other potential issues in tactical surveillance, for which strategies within sensor management are required at all levels, would include: emission control (trade off the use of active sensors for gathering more complete information over self-security); failure recovery (alter

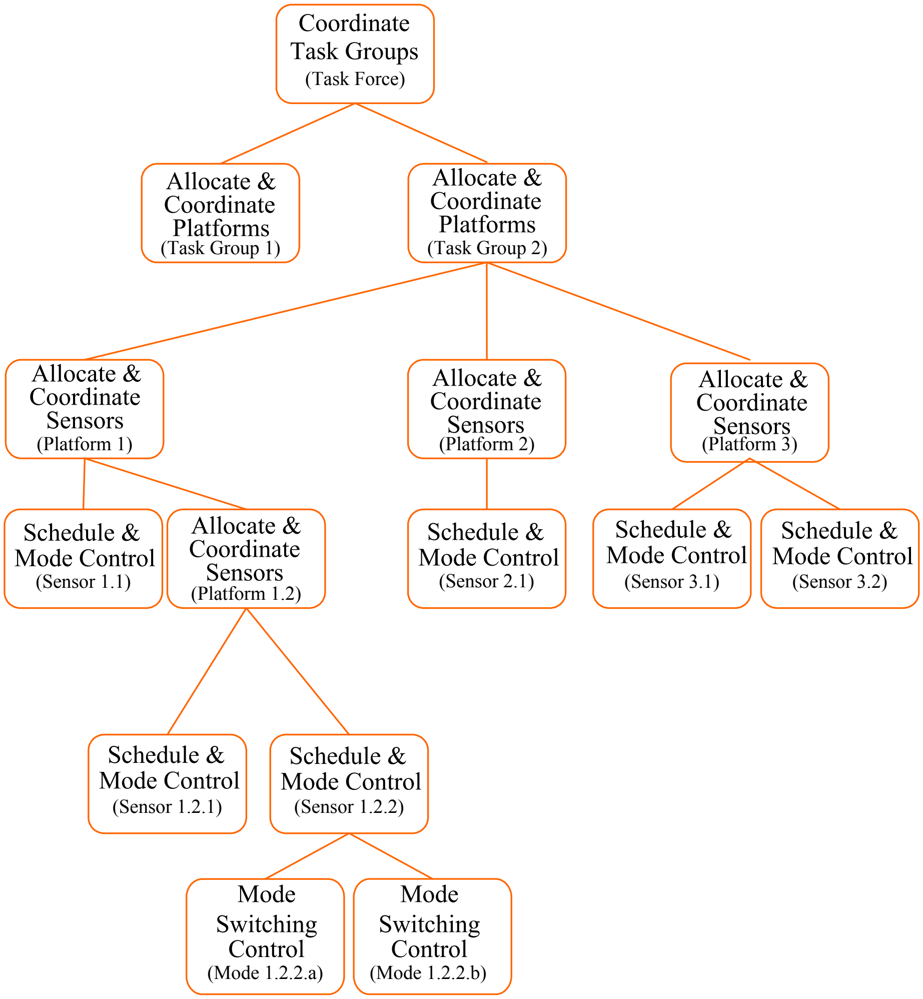

Coordinate Task Groups (Task Force)
Allocate & Coordinate Platforms (Task Group 1)
Allocate & Coordinate Platforms (Task Group 2)
Allocate & Coordinate Sensors (Platform 1)
Allocate & Coordinate Sensors (Platform 2)
Allocate & Coordinate Sensors (Platform 3)
Schedule & Mode Control (Sensor 1.1)
Allocate & Coordinate Sensors (Platform 1.2)
Schedule & Mode Control (Sensor 2.1)
Schedule & Mode Control (Sensor 3.1)
Schedule & Mode Control (Sensor 3.2)
Schedule & Mode Control (Sensor 1.2.1)
Schedule & Mode Control (Sensor 1.2.2)
Mode Switching Control (Mode 1.2.2.a)
Mode Switching Control (Mode 1.2.2.b)

Figure 2: Hierarchy of sensor management problems in military context

the sensing allocation and schedule in case of disabled or diminished sensing capability); and contingency handling (address when and how to make the necessary changes if the situation/objectives change).

The sensor management at each level makes decisions on how to most effectively use the sensing resources under its direction to achieve the tasks requested by the level above, while addressing a changeable working environment. Control challenges at each level differ from each other, but maintain a constant scheme.

2.2 Holonic control

In this section, we examine a spectrum of architectures and show that holonic control offers the best matching to the underlying command structure (Figure 1) while affording autonomy and self-organization at different resource levels.

In distributed control schemes, each node in the architecture has a controller that allows it to work collectively with its neighbours to achieve some overall goal. There are several organizational structures that allow nodes and their controllers to work together. Given the hierarchical and recursive nature of the sensor management problems, the ideal control architecture must present the following characteristics: 1) hierarchical structure to provide a clear chain of command; 2) adaptability to the current situation; 3) sufficient autonomy of each node to perform its function without being encumbered by actions taken at the top level; 4) sufficient robustness to maintain operations even if elements of the network are incapacitated; and 5) recursiveness: each node is composed of one or more nodes of a lower abstraction level.

The following presents a list of candidate control architectures to address sensor management problems.

2.2.1 Centralized

This architecture has the advantage of relatively simple control; however, if the situation for which it is configured changes, then a massive effort is required to reconfigure it. Centralized architectures are characterized by a single point of failure, a high computational burden at the central node, and a lack of general robustness and flexibility. Such an architecture is not suitable for military applications because it requires that a central node be kept intact at all times, which is a significant risk in the military context.

2.2.2 Hierarchical

This structure consists og a top-down decomposition of tasks and reflects a division of labour approach. This structure is efficient in that it forces an expected behaviour; but it is inflexible and branches can become uncontrollable if an intermediate element is incapacitated. The degree of autonomy of an element in a hierarchy is quite limited because of the master/slave relationships that exist between levels. The top-down approach is convenient for planning purposes and the dissemination of instructions and goals. However, if the situation changes significantly, then a new entire plan must be derived, which can be a significant computational and communications burden.

2.2.3 Heterarchical (or fully decentralized)

The heterarchical architecture is relatively robust because there is very little to break, but this lack of structure also makes it very difficult to control. The heterarchical architecture is also unsuitable for the problem at hand because of the risk of an undesirable chaotic behaviour. Design of heterarchical architectures is typically by trial and error because of the difficulties associated with predicting overall emergent behaviour from the simple rules of individual agents.

2.2.4 Federated

This architecture is a compromise between the hierarchical and the heterarchical structures. Like the heterarchical approach, the nodes have a high degree of autonomy but communicate through specialized middle nodes. This approach has improved robustness and flexibility over the other architectures but does not allow for the dynamic restructuring that is a basic feature of holonic architectures, as described below.

2.2.5 Holonic

This is a hybrid structure that combines the best aspects of different architectures and avoids many of their pitfalls. The holonic architecture (Figure 4) takes advantage of the distributed capabilities of classical Multi-Agent Systems (MAS) while incorporating the benefits of the hierarchical command structure that allows for strong goal orientation. This architecture is discussed in more details in the next section.

These architectures can be represented along a spectrum, as illustrated in Figure 3.

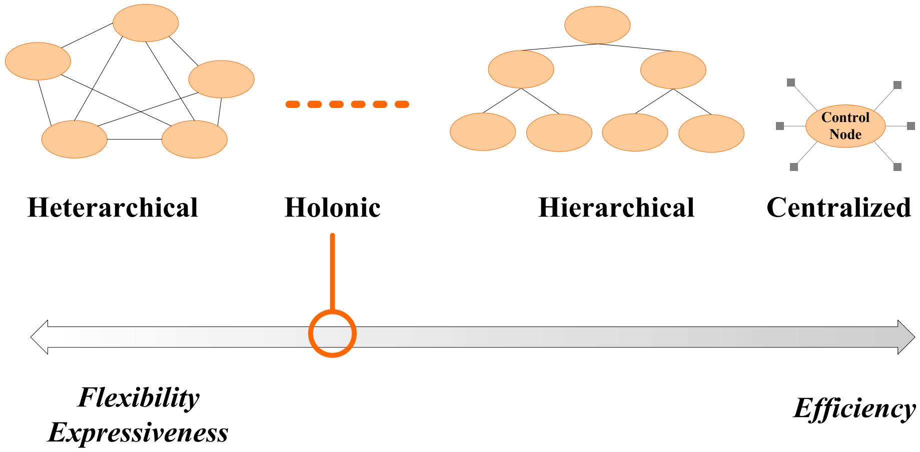


Figure 3: Spectrum of architectures

2.3 Holonic architecture

To explain complex biological and social systems, Arthur Koestler [8] made two key observations:

1. These systems evolve and grow to satisfy increasingly complex and changing needs by creating stable intermediate forms which are self-reliant and more capable than the initial systems; and
2. In living and organizational systems, it is generally difficult to distinguish between wholes and parts: almost every distinguishable element is simultaneously a whole (an essentially autonomous body) and a part (an integrated section of a larger, more capable body).

To refer to this concept, Koestler suggested a new term: *holon*, from the Greek *holos* meaning whole and the suffix *on* implying particle as in proton or neutron.

2.3.1 Characteristics of holons

A holonic system, intended to address the difficulty of coordination in decentralized systems, consists of autonomous, self-reliant units, called *holons* that co-operate to achieve the overall system objectives [9]. Some key properties of a holonic system developed in Koestler's model are [10]:

Autonomy - the capability of a holon to create and control the execution of its own plans and/or strategies (and to maintain its own functions);

Cooperation - the processes whereby a set of holons develop mutually acceptable plans and execute them;

Self-organization - the ability of holons to organize themselves in order to achieve an overall system goal;

Reconfigurability - the ability of the function of a holon to be simply altered in a timely and effective manner.

Another important holonic concept is the notion of *functional decomposition.* The complexity of dynamic systems can be dealt with by decomposing the systems into smaller parts. A consequence of this is the idea that holons can contain other holons (*i.e.*, they are recursive). Problems are solved by *holarchies* (hierarchies of holons), or groups of autonomous and co-operative basic holons and/or recursive holons that are themselves holarchies.

As illustrated in Figure 4, the recursive and hierarchical structure of holonic architecture and its ability to generate dynamic linkages to form an impromptu command structure to perform a task make it well suited to the above described military sensor management problems.

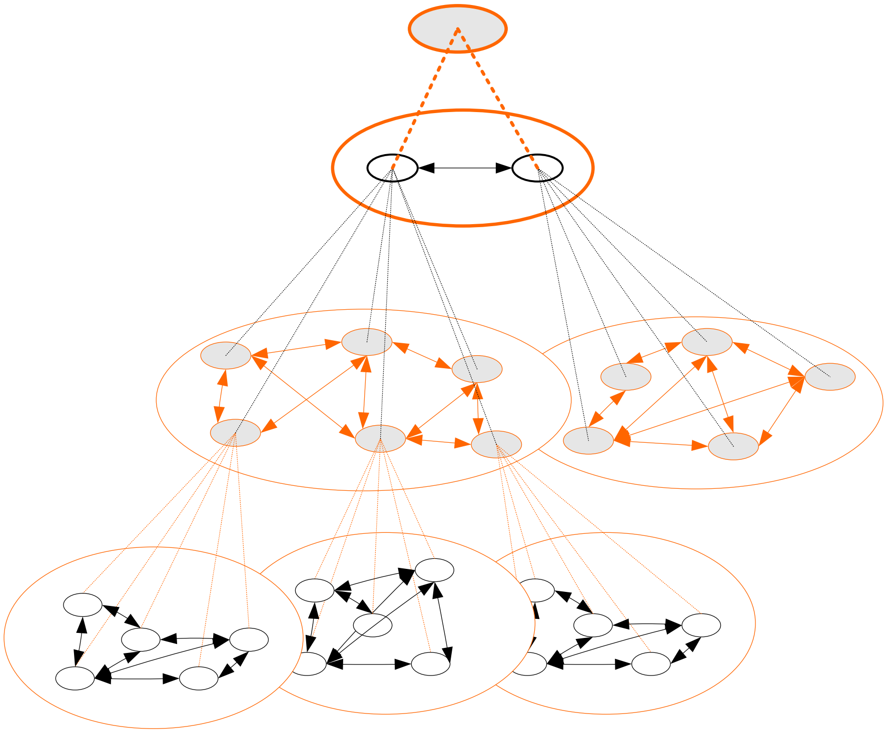


Figure 4: Holonic organization

3 Holonic-based sensor management

This chapter details a holonic control strategy for sensor management in military settings. As discussed in the previous chapter, the primary benefit of holonic control is its ability to form a localized structure, or holarchy, to address the needs as they arise. The holonic control approach allows for a hybrid hierarchy that will retain the benefits of stability of a true hierarchy while providing the flexibility and robustness of a distributed system.

The flexibility displayed by holonic structures is the result of the combined behaviour of the holarchy and not the actions of individual holons. The elements and the processes of the holarchy are presented in the following sections.

3.1 Elements and relationships

The idea of applying the holonic control methodology to the task of sensor management is illustrated in Figure 5. The following paragraphs describe, briefly, the main elements (holons) of the holonic sensor management system.

3.1.1 Command center

The command centre requests a specific piece of information from the entire surveillance network through a single point of contact, the Service Interface and Command Holon (SICH). The priority assigned to the information defines the regional significance and the need significance. For example, a request with a local significance for self-preservation requires real-time response that would take priority over a request with global significance for monitoring a large area. Both are viewed as high priority by their respective command-level; however, the self-preservation goal of the resources will take priority because if it does not, the resource will no longer be able to fulfill the global request.

3.1.2 Service Interface & Command Holon (SICH)

Acting as a facilitator, the SICH is considered as the most important holon. It interfaces with requesting holons to define the constraints of their requests and then spawns a task holon. The SICH then releases this task holon into the network. When the task holon returns the resulting information, this information is presented to the requesting holon and the task holon that gathered the information is then terminated. A fixed communications hierarchy (orange lines) is defined with respect to the SICHs. SICHs require constant two-way communications with each other within the hierarchy in order to keep track of resource availability and to transfer information up the chain of command. This communications hierarchy is defined according to functional significance, *i.e.*, group-level SICH connects to platform holons while each platform-level SICH connects to its own internal resources.

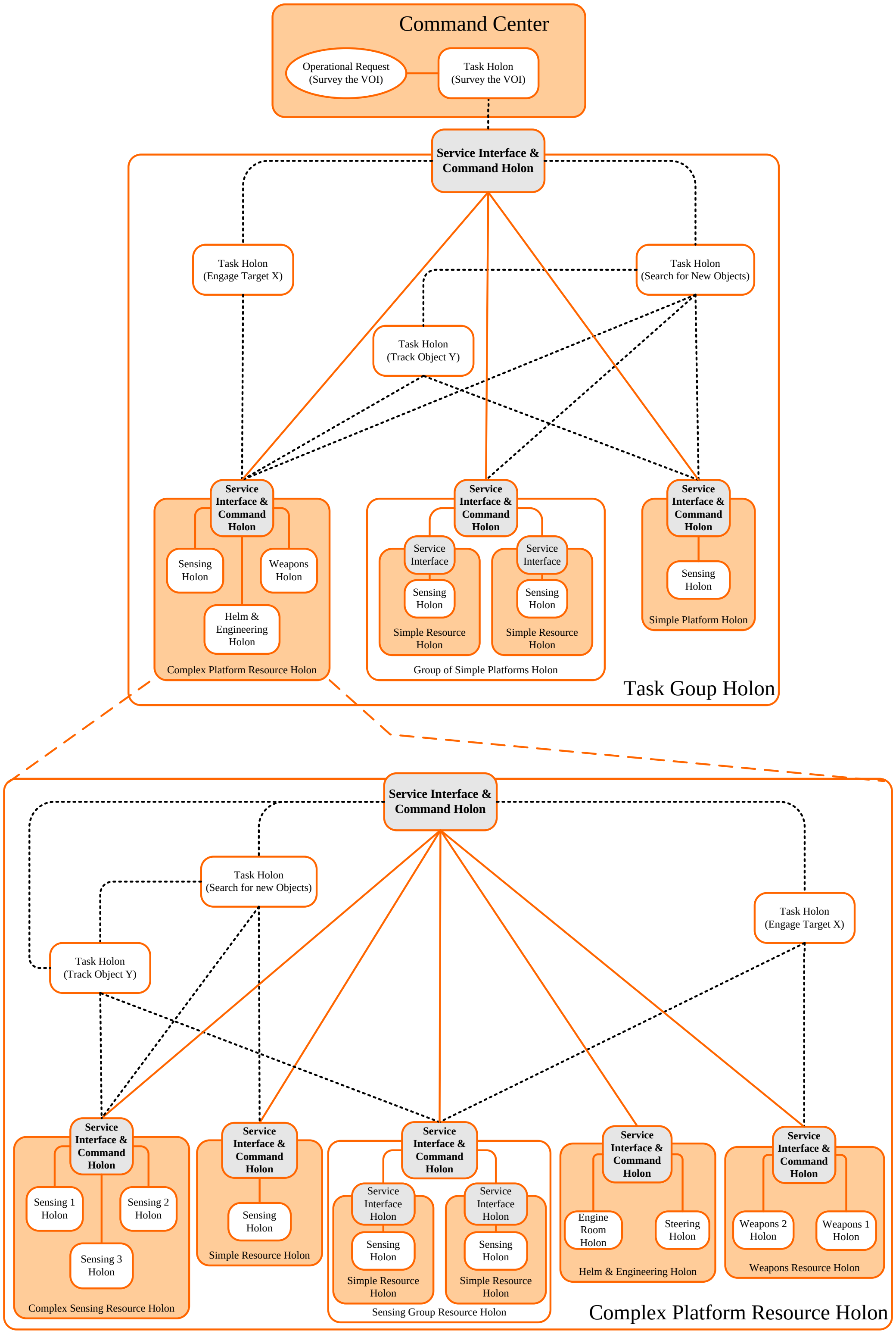

Command Center
Operational Request (Survey the VOI)
Task Holon (Survey the VOI)
Service Interface & Command Holon
Task Holon (Engage Target X)
Task Holon (Search for New Objects)
Task Holon (Track Object Y)
Service Interface & Command Holon
Sensing Holon
Weapons Holon
Helm & Engineering Holon
Complex Platform Resource Holon
Service Interface & Command Holon
Service Interface
Sensing Holon
Simple Resource Holon
Service Interface
Sensing Holon
Simple Resource Holon
Group of Simple Platforms Holon
Service Interface & Command Holon
Sensing Holon
Simple Platform Holon
Task Goup Holon
Service Interface & Command Holon
Task Holon (Search for new Objects)
Task Holon (Track Object Y)
Task Holon (Engage Target X)
Service Interface & Command Holon
Sensing 1 Holon
Sensing 2 Holon
Sensing 3 Holon
Complex Sensing Resource Holon
Service Interface & Command Holon
Sensing Holon
Simple Resource Holon
Service Interface & Command Holon
Service Interface Holon
Sensing Holon
Simple Resource Holon
Service Interface Holon
Sensing Holon
Simple Resource Holon
Sensing Group Resource Holon
Service Interface & Command Holon
Engine Room Holon
Steering Holon
Helm & Engineering Holon
Service Interface & Command Holon
Weapons 2 Holon
Weapons 1 Holon
Weapons Resource Holon
Complex Platform Resource Holon

Figure 5: Recursive structure of holonic sensor management

3.1.3 Task holons

A task holon is a holon that once created autonomously makes use of the resources that are either allocated to it or that it negotiates for itself. This holon differs from other holons in the network as it only exists for the duration of the request and is in effect a roaming client in a network of services. The main purpose of this class of holons is to utilize resource holons to fulfill a given mandate. Task holons establish communication links (dashed black-lines) with the resources directly below them and are also in communication with the commanding SICH above.

3.1.4 Resource holons

Resource holons are the systems and devices that are coordinated at each level in the hierarchy (Figure 1). The general form of resource holons architecture is illustrated in Figure 6. Complex resources (composite resource holons), such as platforms, would be subdivided into holons that represent functional systems on board that must function together but are well defined in their own right. Simple resources, such as a single sensor, are at their most basic level already and would not benefit from further subdivision. Composite resource holons have an internal SICH that acts as an interface to the external holarchy and coordinates the access of task holons to internal resources holons.

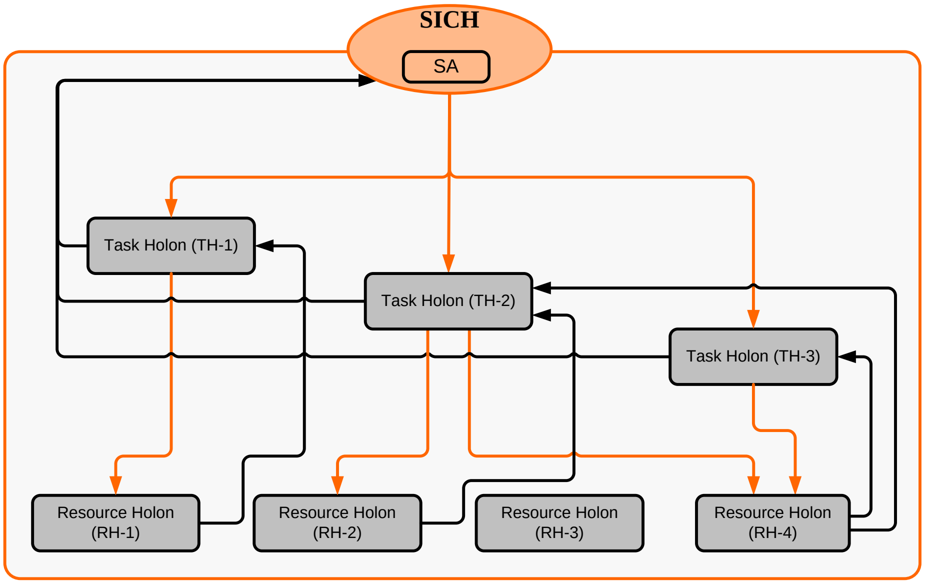


Figure 6: Generic structure of a resource holon

To illustrate how the structure operates, a sensor management holon at the group level is considered. When a task request is made to the group holon, its SICH negotiates with the requesting task holon (external to the group). If the task is accepted, the SICH spawns a (group-level) task holon. This new task holon negotiates with the SICHs of the resources holons (*i.e.*, the platforms internal to the group holon in this case) that it is aware of in order to get serviced. The task holon can be terminated by the group-level SICH if it has been supplanted by a newer one. If a Task holon cannot complete its objective because a resource holon has recently become unavailable, then the task holon will look for alternate solutions by negotiating with other resource holons (*i.e.*, platforms) for a replacement or by completing its task without the resource. If no solution is found, the task holon will report to the SICH above it, which will find an alternate solution either by creating an additional task holon to aid the first one or by terminating the task holon and creating a new one. In this way, problems are addressed locally and information about the problem is propagated up through the holarchy to the level ay which it is needed.

The same mechanism is used, within the platform holon, when the platform-level SICH accepts to service a task.

3.2 The hierarchical structure

The proposed holonic control architecture is decomposed into three main levels: sensor, platform, and group. The levels are related to each other in a recursive hierarchical manner typical of holonic structures. The sensors represent the lowest level of the hierarchy. Each platform coordinates the sensors that are located onboard it, but does not control the sensors onboard other platforms. Likewise, the group-level manager coordinates sensing activities between platforms but does not directly manage the sensors onboard these platforms.

The SICHs are connected to each another through fixed links (orange lines) that represents the resources hierarchy (see Figure 1), while task holons are responsible for making the dynamic hierarchical links (dotted black lines) and operating within them. A task holon makes requests to the SICH to negotiate access to available resource holons, which comprise the next level down in the holarchy. The result of a successful negotiation allows direct access to the allocated portion of the resource holons.

This holonic structure consists of a loosely defined architecture where the different levels address the problem at different scales. The system is basically a distributed one, but a hierarchy (orange lines) is imposed to demark the logical division of effort and resources. Each structural level operates at a different level of abstraction. For example, at the higher levels, the holons are concerned with "big picture" building problems. At the lower levels, they are concerned with more detailed problems.

3.3 Task planning, allocation and scheduling

Each level in the holarchy consists of a single holon, itself containing a number of holons (recursive architecture), as shown in Figure 5. The SICH acts as the negotiator (mediator) between holarchy levels, while the resource holons represent the services available within each level. Resource holons may consist of a subordinate holarchy level such as a sensor onboard a platform, or a platform as a member of a group. Task holons are generated by the SICH and are responsible for utilizing the resources to meet level-specific sensing objectives. Resource holons are responsible for implementing the requested tasks.

As an illustration, consider a platform-level holon. This holon represents an entire platform (such as a frigate), while the associated resource holons represent the resources onboard the platform such as sensors, communications, power, etc. The task holons act as autonomous agents onboard the platform. They are charged with a specific task such as "track this target" or "perform a search operation in this sector" and must negotiate with the resources to perform their task. Task holons are created by the SICH and typically exist only while they are needed. For example, a holon tracking a single target exists while the target is in the sensing domain of the platform and can be tracked. Once the target leaves this region, the associated tracking holon is no longer useful and is therefore terminated.

Figure 7 illustrates the control, coordination, and negotiation dynamics within the proposed holarchy. Note that the components shown in the figure are required in all the levels of the holarchy, but are not implemented within a single holon.

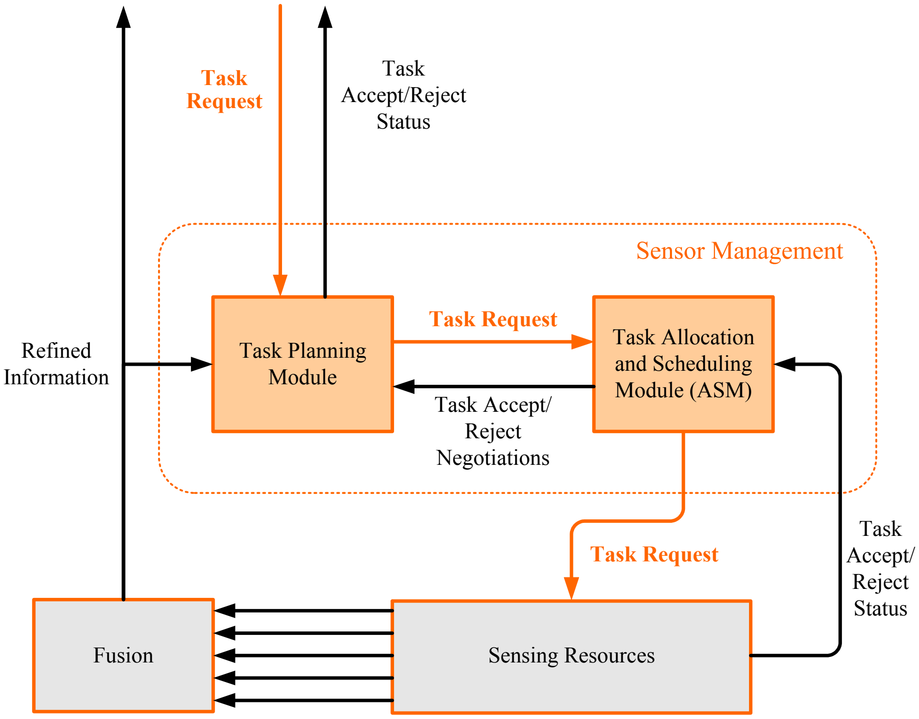

Task Request
Task Accept/Reject Status
Sensor Management
Refined Information
Task Planning Module
Task Request
Task Allocation and Scheduling Module (ASM)
Task Accept/ Reject Negotiations
Task Request
Task Accept/ Reject Status
Fusion
Sensing Resources

Figure 7: Control and coordination functions

In the proposed holarchy, task holons take on the role of task allocation and scheduling module, while the SICH acts as the task planner and handles negotiations with higher-level controllers and subordinate resources. Typically, the SICH will plan tasks that address sensing objectives and each task plan will lead to the creation of a task holon. Utilization of the resources (sensor allocation) is left up to the task holon negotiation process. The SICH plans are devised on the basis of situation analysis and predicted evolution of the situation. Clearly, the predictions will not always be accurate and a revised task plan will be generated. In the interim, the task holons attempt to implement existing task plans using available resources.

In the strategy under consideration, since there are three main levels of sensor management, each level contains a SICH as well as a number of task holons and resource holons. The task holons utilize the resource holons through a negotiation process. The resources may be limited in their capacity to serve the task holons and may not be able to service all of the jobs requested of them. In particular, two of these resources, the active-radar sensors and the wireless communications, are fundamentally limited to serve one task at a time (*e.g.* update track, transmit message). However, by using a time-sharing strategy, these resources can seemingly service many tasks simultaneously. For example, the time it takes to update a track (*i.e.*, get a new position measurement) is generally much shorter than the time period between track updates; therefore by scheduling updates intelligently, the sensor can support the tracking of multiple targets simultaneously.

To simulate this, the time-limited resource will accept a number of tasks that it will sequentially execute over a predefined time period. This time period will be chosen to be long in comparison with the typical task execution time (*e.g.* time to update a single track), but short in comparison with the task revisit time (*e.g.* time between track updates). In this way, the resource can seemingly service multiple tasks simultaneously while still being able to quickly address new tasks as they arise (*i.e.*, adapt to the changing environment).

The synchronous control (with predefined control period) concept is a departure from the holonic strategy of (asynchronous) event-driven control, but is adopted here mainly due to project constraints. It is assumed that the sensors can schedule tasks rapidly (*e.g.* less than one second), while the holonic control plans, allocates and schedules tasks over longer time periods. This simplification is a realistic one, which demonstrates that holonic control can be implemented using resources that are not designed with a holonic or event-driven control strategy in mind.

The synchronous control strategy is only imposed on the sensors and the wireless communications resources (at the lowest level), and does not concerns resource holons at the group or platform level. As a result, platform-level task holons can interface with the sensors (resource holons) only once during each control period. Likewise, tasks at the group level can access the wireless communications only once per control period. The control periods for the communications and the sensors are not linked.

3.4 Task negotiations

Task holons negotiate with the resource holons available to them to achieve their task goals. This negotiation, as will be seen, is most significant between platform-level task holons and the sensors. Task holons must also negotiate with the communications resource (transmitter) to send messages between the group and the platform levels.

The basic negotiation strategy is as follows:

1. At some time instant, the task holon evaluates its need for access to resource holons.
2. If needed, task holons request a service quality estimate from each resource holon.

3. Resource holons return Quality Of Service (QOS) that measures how well the service can be performed (if at all). This is a quality of service measure, not a measure of resource holon availability.
4. Task holons request service from the resource holon that provides the best QOS.
5. Resource holons return an accept/decline notification.

The negotiation process is complicated when there are multiple task holons and multiple resource holons. In this case, the task holons attempt to get service from the resource holon offering the best QOS, while the resource holons attempt to service the task holons in order of priority. Since the resource holons are limited in the number of tasks to be serviced during a control period, a number of iterations may be required to achieve the best matching of tasks with resources. This iterative process may be trivial (no iterations) if the resources holons are under-utilized and available to service all jobs requested of them. When resource holons loads are high, it may take several iterations to derive an optimal solution. The negotiation process occurs each time a resource control period begins. The negotiations will be assumed to occur instantaneously, although in practice, negotiations consume a finite amount of time and may in some situations affect sensor management performance.

Note that the result of the negotiation is the allocation of task holons to individual resource holons. This is a one-to-one pairing[1] and does not preclude the use of multiple resource holons by individual task holons since the pairing is renegotiated at the beginning of each resource holon control period. It is not inherent to the design that each task holon access only one resource holon at a time, but this strategy is adopted here for the sake of computational efficiency.

3.5 Communications

Communications between task holons and the SICH are assumed to take place over a very high bandwidth medium and are therefore not modeled here. Additionally, communication limitations between the task holons and the resource holons at the platform level are ignored in this work. However, the communication limitation between the platforms and the group level are addressed by introducing a communications holon that manages the (typically low-bandwidth) wireless connection between them.

3.6 Assumptions underlying the design of the holonic-based sensor management

Chapters 4 to 6 present the detailed structure and functions of the sensor management holarchy. The following are a few comments that may be useful in understanding the structure and functions:

1. The control strategy is designed having in mind the scenario that consists in protecting some high value asset (or unit). The main sensing objective is to maintain tracks on all

[1] The task holons only get service from one resource holon at a time.

targets in the area of interest while searching for new targets, giving special attention to those targets that are assessed to be threatening to the high value unit and to those areas where new targets are expected to appear.

2. Sensors described in Appendix A will be assumed here, although the holonic strategy may easily be adapted to include other types of sensors. Each sensor will be assigned a resource holon that manages the utilization of that particular sensing resource. These resource holons interface with the platform-level SICH.
3. The platform SICH coordinates the activities onboard each platform. In the current work, we will limit these responsibilities to include searching for and tracking of targets (sensor allocation), sensor configuration adjustments (mode control) and intra-platform cue/handoff (cooperation and control).
4. The role of the group-level sensor management holon is to coordinate the inter-platform sensing activities, such as cueing and handoff of tracks.

4 Sensor-level holons

The sensors represent the lowest level that will be considered in this work and therefore the holonic model used to describe them is simpler than the model used at the platform and group levels.

Each sensor holon consists of a local manager (*i.e.*, SICH) which generates sensor-level task holons. This general structure is common also to the platform and group levels. The main difference at the sensor level is that the sensor holon contains only one resource holon representing the sensor hardware. In this manner, a holonic control structure was constructed around an existing piece of hardware (or a model of it).

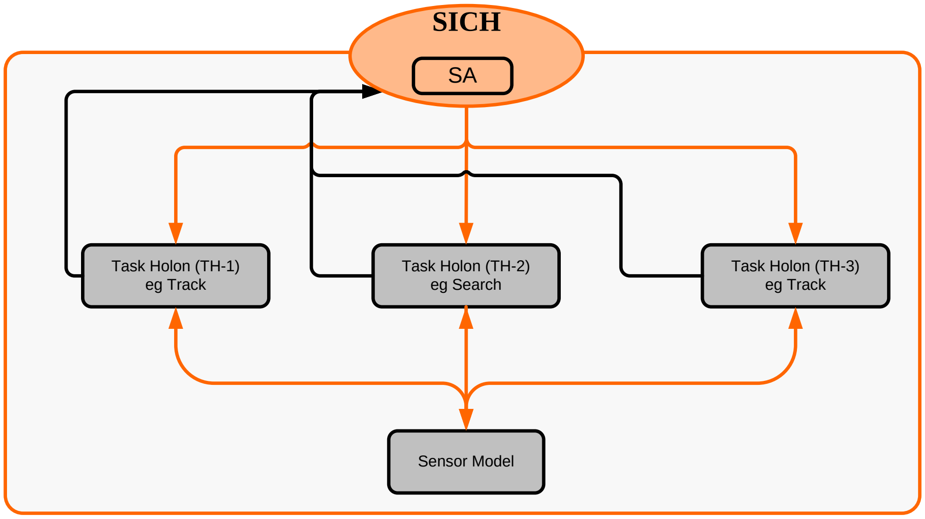


Figure 8: Sensor-level holon

Sensor-level task holons, created by the sensor-level SICH, contain control instructions for the sensor hardware pertaining to specific tasks (*e.g.* update existing tracks of existing objects or search for new ones). These detailed instructions are hardware specific and beyond the scope of the design presented here. Task holons are created to perform tasks that the SICH has negotiated for the next control period. After the control period has been completed, the sensor SICH reports the data generated by the task holons and the process then repeats itself for the next control period. Tasks that do not complete during a control period are reported as such (no service performed). Table 1 details the holons contained within the sensor level of the holarchy.

The sensor-level SICH incorporates elements of situation analysis (*i.e.*, data fusion algorithms). Typically, the situation analysis portion would be separated from the SICH, treated more as a resource. Due to the simplicity of the model at this level, it is sufficient to consider the data fusion algorithms existing within the SICH. Sensor-level situation analysis converts raw sensor data into measurements representing target detections.

Table 1: Sensor-level holons

Name	**Type**
SICH	Interface
Track Update Holon	Task
Search Holon	Task
Configuration Holon	Task
Sensor Hardware Holon	Resource

4.1 Sensor-level SICH

Sensor-level sensor management is concerned with scheduling the use of sensor hardware. Since only one task holon can be serviced at a time by the sensor hardware (resource holon), the role of sensor management at this level is to schedule these task holons as to best address the sensing objectives. At the sensor level, this scheduling problem is guided by a priority attribute attached to task holons that rates their relative importance. This priority ranking is generated by the platform-level SICH (Chapter 5).

The sensor-level SICH is responsible for negotiating with the platform-level task holons for the task acceptance of task requests. Once a request is accepted, the sensor-level SICH becomes responsible for creating, coordinating and terminating the sensor-level task holons. The data collected by the sensor-level resource holon (*i.e.*, the physical sensor) is delivered and interpreted by the sensor-level SICH and at the end of each control cycle, reported to the requesting platform-level task holon.

The sensor-level SICH creates, executes, and terminates sensor-level task holons at a regular time control period t_{sensor}. This period should be chosen to be small in comparison with the minimum update period of the different tracks. It is assumed that the time t_{task} for individual tasks to be completed, such as a single track update or search, will be short in comparison with the sensor control period t_{sensor}. As a result, a single sensor-level resource holon can address multiple task holons during each control period.

An optimal sensor utilization strategy should consider all the task holons that are requested of it and develop a plan to implement these task holons in a manner that best meets the sensing objectives. This problem has been closely examined by a number of researchers (see [11, 12, 13]) and solutions have been suggested for individual sensor optimization.

Task holons requiring service within each short time period will be scheduled on the basis of priority without regard to future task holon requirements. This myopic (*i.e.*, single control period) scheduling approach is not an optimal technique but is sufficient for the purpose of this work, that is demonstrating the applicability of holonic control approach to sensor management.

4.1.1 Platform to sensor task negotiation

Platform-level task holons will make requests for service from the set of sensor-level resource holons (*i.e.*, sensors) available to them. Each sensor-level SICH is responsible for returning an estimate of the Quality Of Service (QOS) that can be provided for each request for service. At the sensor level, QOS indicates whether or not the task holon can be serviced during the next control period, t_{sensor} and a measure of the performance with which the task holon can be serviced. This performance measure consists of a metric incorporating the range and bearing accuracy that can be anticipated from the measurement. The QOS reported by a sensor-level SICH will be zero if the target is not within the sensing domain of its resource holon, or if the resource holon is otherwise not available (*e.g.* not operational).

As an example, consider the case of updating several tracks with a single sensor. The QOS measure specifies the performance of the sensor against the target. In this case, t_{task} is the time to perform a single track update, while t_{sensor} is the total amount of time available for the sensor to perform track updates.

The sensor-level SICH determines execution times t_{task} for the individual tasks requested of it and accepts tasks in order of priority until the control period t_{sensor} is filled (see Figure 9). While the sensor-level SICH attempts to service the highest priority task holons, the platform-level task holons are attempting to secure service from the sensor-level resource holon that provides the highest QOS. Thus, the task negotiations become iterative in nature as multiple platform-level task holons negotiate with a number of sensor-level SICHs. In the design presented, the time for this iteration will be neglected and it will be assumed to take place instantaneously at the beginning of each control cycle.

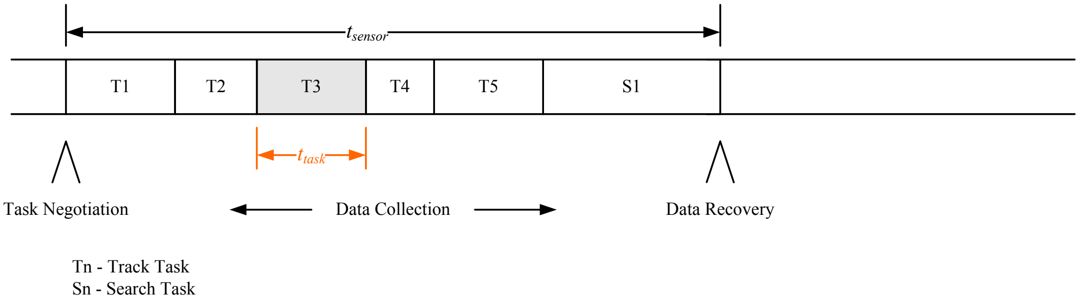


Figure 9: Scheduling within sensor control period

The task execution times t_{task} can be considered in two different ways:

1. The time t_{task} is the actual execution time: this approach is deterministic as all accepted task holons will be executed during the active control cycle t_{sensor}. This approach will simplify the simulation without significantly altering the demonstration of holonic control.
2. A more realistic variation would be to consider the computed execution times t_{task} as estimates of the true time that each task holon will take. Thus, one drawback of this scheduling approach is that the accepted task holons may be completed sooner than expected, leaving the sensor-level resource holon idle for small periods of time.

> In contrast, task holons may take longer than expected to complete, resulting in some accepted task holons not being fully serviced. As a result, task holons that are accepted but do not get serviced must be notified and attempt to obtain service during the next control period. This approach is more realistic as it will include some variability in task holon execution times and will force some task holons to be rescheduled. This is a complication that may not significantly increase the quality of the demonstration of holonic control, but could be addressed in follow-on activities.

Using the first approach, the sensor-level SICH creates sensor-level task holons after the negotiation process is complete to service the accepted platform-level task holons. All of the sensor-level task holons will be completed during the control period and the results stored in the sensor-level SICH. At the end of the sensor control period t_{sensor}, all of the task results are reported to the appropriate platform-level task holons.

4.1.2 Task execution time

A key to the presented sensor design is the computation of task execution time t_{task} for tasks requested of the sensor. This time estimate should take into consideration the attributes of the task in computing this measure. Search task holons and tracking task holons are treated differently. Track update holons are typically given first priority with the sensors and then any remaining time is dedicated to search task holons. While the interval between track updates is typically no smaller than one second, the time to acquire a single track update is (typically) measured in milliseconds. Thus a target track can be maintained with only a fraction of the total sensor capacity. The remainder of the capacity can be used to maintain other tracks or conduct target search operations. This is a scheduling problem that is solved by the holonic architecture through negotiations between sensor-level task holons and resource holons.

Track updates require a narrow sector scan to be conducted at a prescribed time on the basis of the track prediction. If the track quality is high, then the track update will require a very narrow search during a short time window. If the track quality is low, then a broader search may need to be conducted over a wider time window in order to detect the target in its new position. This issue is dealt with in detail in [14, 15]. Thus the time it takes for a track update to complete is on the basis of the track quality, *i.e.*, the (average) track-update time is inversely related to track quality. The sensor-level SICH can use this time estimate as a conservative measure of the task-completion time when negotiating with the platform-level holons.

As illustrated in Figure A.3, a scan sector is defined by the sector start angle ϕ_{start}, the sector stop angle ϕ_{stop} and scan rate ω. The time to complete a scan of the sector is

$$t_{scan} = \frac{\phi_{stop} - \phi_{start}}{\omega} \tag{1}$$

The time to complete a track update is calculated using the estimated scan width

$$\psi = \phi_{stop} - \phi_{start} \tag{2}$$

that is inversely proportional to the track quality

$$\psi_{update} = \frac{k}{r_{target}} \sqrt{P_{11}^2 + P_{33}^2} \tag{3}$$

where P_{11} and P_{33} are respective diagonal elements of error covariance matrix $\boldsymbol{P}$ (see Appendix B). Therefore, the time to update a track is computed as follows

$$t_{update} = \frac{\psi_{update}}{\omega} \tag{4}$$

Once all track updates are scheduled, any remaining time can be used to address the highest priority search task, *i.e.*,

$$t_{search} = t_{sensor} - \sum_{tracks} t_{update} \tag{5}$$

This search time corresponds to the search sector width

$$\psi_{search} = t_{search} \cdot \omega \tag{6}$$

This is an approximate method for determining track update times. More in-depth approaches are detailed in [16, 15, 11, 12].

Search tasks require a scan to be performed over a specific sector, an operation that takes a finite amount of time. Based on the mode of the sensor, the SICH can compute (estimate) the time it will take a scan to complete. Search tasks typically require repeated scans of specified regions and could theoretically consume 100% of the sensor time if not interrupted by other sensing tasks. In the holonic design outlined here, tasks are assigned a priority attribute that is used to manage the sensor utilization. Track updates may be given higher priority and the search tasks allotted the sensor time remaining after the track updates are secured.

4.1.3 Quality Of Service (QOS)

In order to facilitate platform-sensor negotiations the sensor-level SICH must be able to provide a QOS estimate to the platform-level task holons. This measure should reflect the ability of the sensor to service the task as well as a measure of how well the task can be performed given the current sensor configuration. This information is computed at the beginning of each control period in response to the tasks requested from the sensor.

For tracking tasks, the sensor mode combined with the relative position of the target can be used to determine the range and bearing accuracy of the measurement. These quantities can be provided to the requesting platform-level task holon so that the best sensor-level resource holon for the job can be selected. Note that, for tracking tasks, QOS does not depend on the sensor load, unless too many update task holons ask for service simultaneously. In this case, those task holons that cannot be scheduled during the control period are reported with QOS of zero.

For the purpose of the design reported here, the scalar QOS_{track} for tracking tasks is calculated according to the following equation

$$QOS_{track} = \left[(k_1 \eta_r)^2 + (k_2 r \sin \eta_b)^2 \right]^{-1/2} \tag{7}$$

The equation rates the effectiveness of the sensor with range-variance η_r and bearing variance η_b against a target located at range r. The constants k_1 and k_2 are chosen to balance the importance of range variance against bearing variance.

The quality of a search task depends on the configuration of the sensor and the amount of time that can be dedicated to the search t_{search}. For a given sensor, the maximal QOS_{search} is achieved when the sensor can dedicate an entire control period to the task. Search quality is also related to the configuration of the sensor, essentially relating to the sensor variances η_r, η_b, and the scan rate ω. Thus, for sensors whose configuration allows them to perform the search, the reported search quality is computed according to the following

$$QOS_{search} = \frac{k_3 \omega \cdot r_{\max}}{\left[(k_1 \eta_{r_{\max}})^2 + (k_2 r_{\max} \sin \eta_b)^2 \right]^{1/2}} \tag{8}$$

This metric reflects the fact that searches benefit from high scan rate and long detection range $r_{\max}$ moderated by sensor variance. The constant k_3 balances the importance of scan rate and range against sensor variance. Note that sensor variance is calculated for the maximum sensor range $r_{\max}$ since this is the desired range for target detection.

SICH data analysis, at the sensor level, consists in converting the raw hardware measurements into detections for platform-level analysis. This is an operational detail of the sensors themselves and will be overlooked here. Instead, the sensor model described previously will be used to compute target detections during simulation.

It should be noted that the process of computing target tracks and directing sector searches is the responsibility of the platform-level task holons and not part of the sensor-level sensor management. The sensor-level sensor management is only responsible for managing a particular sensor onboard a platform.

4.1.4 Average sensor load

Average sensor load is computed by the sensor-level SICH and can be communicated to the platform-level task holons during negotiations or directly to the platform-level SICH through other on-board communications pathways. For the purpose of the design presented here, the fraction of a control period t_{sensor} dedicated to high priority tasks will be defined as the instantaneous sensor load L_i. High priority tasks are defined as those with a priority attribute greater than the one of a normal search (see §5.6).

The idea is to base sensor load mainly on tracking tasks rather than search tasks. Thus, L_i will increase as the number of target tracks serviced by the sensor increases. In order to be a useful average measure, L_i is averaged over a sliding time window of the most recent N_{CI} control periods. Thus average sensor load L_S is computed according to

$$L_S = \frac{1}{N_{CI}} \sum_{i=1}^{N_{CI}} L_i \qquad (9)$$

4.2 Sensor-level task holons

Platform-level tasks that are accepted by the sensor-level SICH result in the creation of sensor-level task holons. While this representation is consistent with the holon architecture, the role of the task holon is, in this case, very limited due to the fact that only one resource holon is being utilized. The task holons are created with their execution order provided as an attribute, and therefore executed sequentially.

The main uncertainty at the sensor level is that tasks may not complete in the time estimated by the SICH. This uncertainty may or may not be taken into consideration in the implementation depending on the required level of detail; however, it is to be expected in practice. If tasks consume more time than estimated, the ones planned for execution near the end of the control period may not be performed. In this case, at the end of the control period when the task holon reports to the SICH, this failure to execute is returned instead of sensory data.

Tasks may also be executed in a shorter time than predicted, leaving sensor capacity unused. To compensate for this, the SICH may overbook tasks during the negotiation phase, though in most cases a search task will be available to utilize the excess sensor capacity.

Three types of task holons will be used to interface with the sensor hardware: tracking, searching, and configuration. Each holon is created with two attributes: execution order and instructions for the sensor-level resource holon. The sensor-level resource holon accepts the tasks according to their execution-order attribute and follows their instructions. Tracking holons specify a narrow region to search for a known target, while search holons specify a scan sector. The configuration holon is required for the configuration changes of sensor-level resource holon, typically active/inactive status and mode setting.

Configuration holons are meant to interrupt normal sensing (search and update) tasks and therefore consume the entire control period of the sensor. They are assumed to complete in all cases in the design presented in this report. Configuration adjustments can be expected to occur on an infrequent basis. Search holons consume a variable amount of time, depending on sensor-level resource holon load. If a target is found, a search holon will return a confirmation of the detection, as well as an estimation of the area it has searched. Track update holons result in either a new target measurement, which will serve for track update, or a failure message in case no measurement of the target is taken.

The following summarizes the roles of the sensor-level task holons.

4.2.1 Sensor-level search holons

Their role is to deliver control instructions to sensor-level resource holon and measurement data back to the sensor-level SICH. Their task results delivered to the SICH are: detection, no-detection, task failed to execute (sensor malfunction).

4.2.2 Sensor-level tracking holons

Their role is to deliver control instructions to sensor-level resource holon and measurement data back to the sensor-level SICH. Their task results delivered to the SICH are: track initiation, track update measurement, missed update, task failed to execute (sensor malfunction).

4.2.3 Sensor-level configuration holons

Their role is to deliver control instructions to sensor-level resource holon and status information back to the sensor-level SICH. Their task results delivered to the SICH are: coverage adjustment complete, or sensor inoperable (*i.e.*, sensor malfunction).

5 Platform-level holons

The platform as a holon represents a military asset (such as a frigate) and as such contains one or more sensor holons as resources (see Figure 10). The links show how tasks holons are being serviced by resource holons. A task holon can only be serviced by one resource holon, while a resource holon may service more than one task holon.

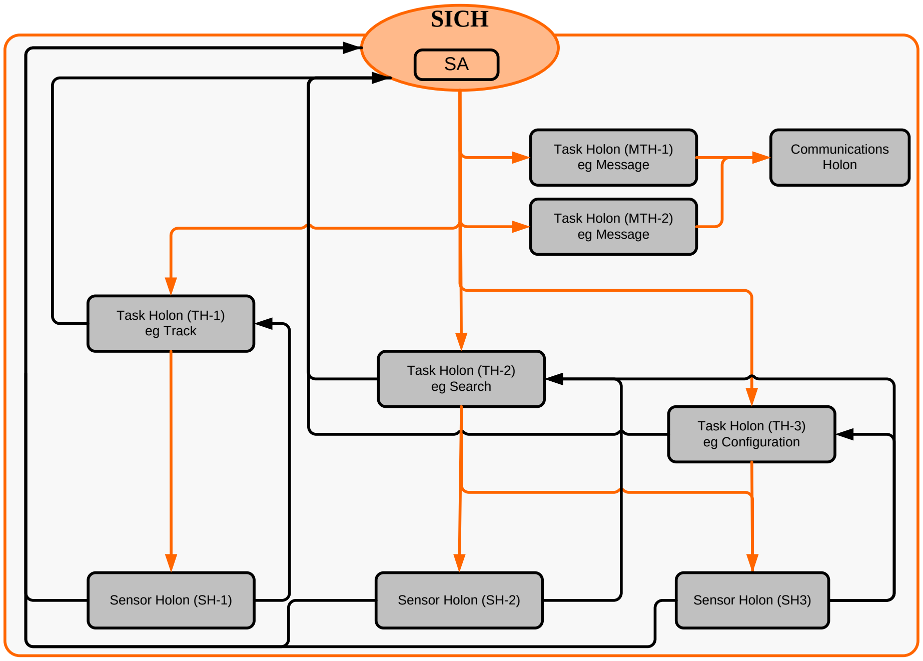


Figure 10: Simplified platform holon

Although the holon structure appears similar to the sensor level, the function of the platform-level task holons are much different. An additional platform-level resource holon included at this level is the "communications holon". This new holon manages a wireless communications link (*e.g.*, link 11) between the platform and the group-level holon.

5.1 Platform-level communications resource holon

The platform-level communications resource holon operates in a manner analogous to the sensor-level resource holons. The holon has an internal structure (Figure 11) that is the same as sensor holons except that the resource in this case is a transmitter/receiver rather than a sensor. The communications holon has a SICH that operates within control periods t_{comm} during which communication messages are sent. The basic process is that, at the beginning of each control period, the SICH negotiates with any task holons requesting access and accepts tasks in order of priority until a transmit queue is filled. It will be assumed that the 'receive' operations can proceed unhindered and that any received message is immediately relayed to the platform-level SICH.

Platform-level task holons requiring wireless communications will request communications

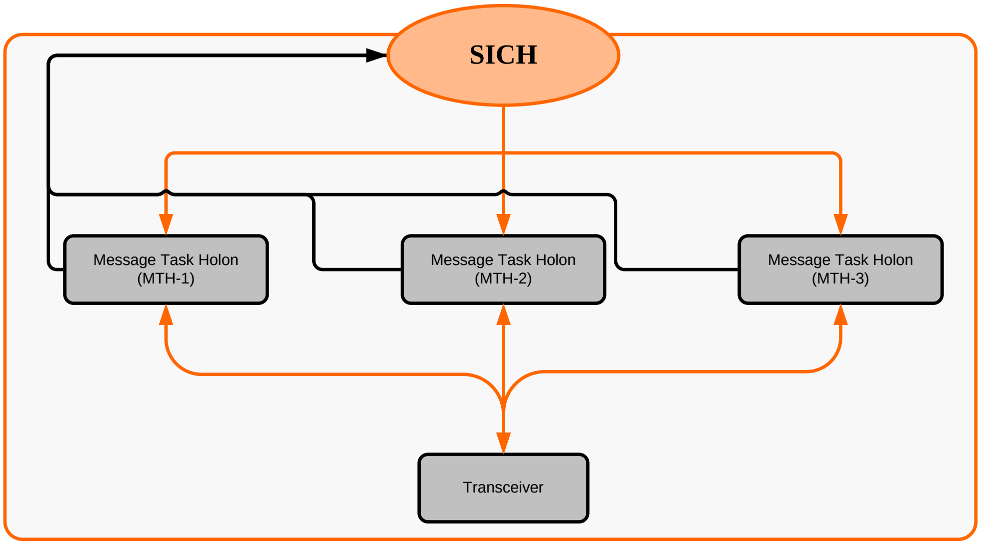

SICH
Message Task Holon (MTH-1)
Message Task Holon (MTH-2)
Message Task Holon (MTH-3)
Transceiver

Figure 11: Communications resource holon

holon service at the beginning of each control period t_{comm}. As part of the request, both the message priority $P_{message}$ and the message length $L_{message}$ will be provided to the communications holon SICH. Since a transmitter can only send one message at a time, there is a fixed amount of data that can be transmitted during t_{comm}, *i.e.*, the transmit queue. The SICH of the communications holon accepts messages on the basis of priority until the transmit queue is filled, and rejects the remainder. During the next control cycle, these remaining tasks may request service again, eventually getting their message out on the basis of priority level. Once accepted, the message will be assumed sent and received by the group holon SICH after the control period t_{comm} is lapsed.

The conversion of message length $L_{message}$ into transmission-time depends on the data transmit rate $R_{transmit}$. The transmission time $t_{transmit}$ is computed according to

$$t_{transmit} = \frac{L_{message}}{R_{transmit}} \tag{10}$$

The control period is chosen to be longer than the largest message transmission time so that, in general, several messages may be transmitted per control period. Thus, platform-level task holons will receive a simple message indicating whether or not their message can be transmitted during the following control period on the basis of the queue filling procedure.

5.2 Operations of platform holons

At the platform level, data analysis can be treated as a high level situation analysis and is depicted in Figure 10 as part of the platform-level SICH. The bandwidth limitations in communications will be ignored between the resource holons (sensors), the task holons and the SICH although in a practical implementation this issue could arise. Generally, however, the more significant communication bottleneck may occur in the platform-group link.

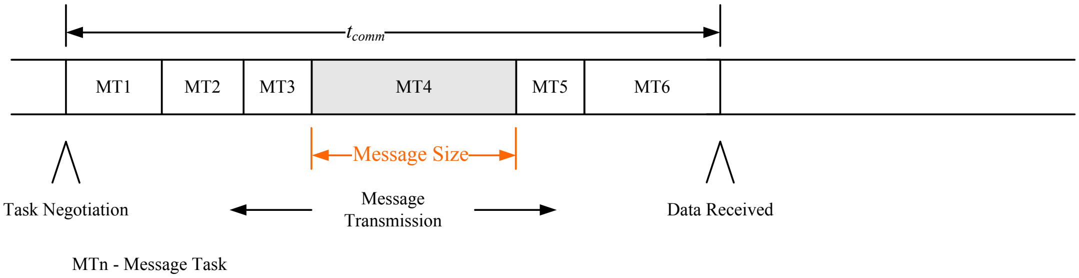


Figure 12: Scheduling of communications control period

The group level represents the point where all of the sensor data gathered by the platforms is assembled and inter-platform sensor management is coordinated. The group level may be located onboard a platform or at a remote location. The remote location will be assumed in the current work, so that the issue of communication bandwidth can be explored within the presented sensor management design. To model this, a communications resource holon is incorporated as a resource designed to facilitate platform-group communications. This holon (see Section 5.1) functions in a manner very similar to the sensor holons, using a control period to book outgoing messages. Platform-level SICH communications with the group holon, required for data transmission and task negotiations (*i.e.*, data and control), must utilize the communications resource holon. As an abstract model of communications, it will be assumed that there is no bandwidth restriction on receiving messages but outgoing transmissions are bandwidth limited.

The general operation of the platform-level holon is similar to the sensor-level holon, yet, much more complex. One of the functions of the platform-level SICH is to create platform-level task holons. There are four types of task holons that we will specify: (i) tracking; (ii) searching; (iii) configuration; and (iv) message. Note that tracking and search task holons provide most of the sensor management functionality onboard the platform.

The holons are independent agents operating according to internal programming and controlled by a number of parameters or attributes. The SICH initiates the task holons with a set of attributes and may also modify these attributes as events dictate. Table 2 details the various holons contained within the platform level of the holarchy.

Table 2: Platform-level holons

Name	Type
SICH	Interface
Search Holon	Task
Track Holon	Task
Message Holon	Task
Configuration Holon	Task
Communications Holon	Resource
Sensor Holon	Resource

5.2.1 Tracking holons

The tracking task holon role is to establish a target track (once detected) and subsequently acquire updates in order to maintain specified track quality. The tracking calculations are performed by the track holon and the results (position, velocity) are reported to the platform-level SICH as they become available.

5.2.2 Search holons

Search task holons are created to detect new targets and may be terminated after a specified period has lapsed or remain active until events dictate termination. Sensing objectives are used to control search task holons. For example, a platform participating in a military operation may want to continually look for targets approaching from a given sector. According to the design presented, a search task holon would be created to fulfill this objective and be active until the mission is complete. Alternately, a particular target may be expected in a predetermined time and location, in which case the corresponding search task holon would be created for the expected time window only.

5.2.3 Configuration holons

The general sensor management approach at the platform level is to define a configuration for the sensors and then allocate sensing duties on the basis of this configuration. Changes in sensor configuration should occur infrequently, on the basis of the situation analysis at either the platform or the group level. For example, the platform-level SICH may set one sensor in long-range mode for tracking aircraft and one in short-range mode for tracking water-based craft. In this case, tracking tasks would be distributed amongst the sensors on the basis of target characteristics, *i.e.*, aircraft would be tracked by one sensor while the other would track watercraft. As the situation changes, such as the elimination of all watercraft, the sensor configuration may change to facilitate the aircraft tracking tasks. Sensor modes are controlled by the configuration holon.

The configuration holon must have access to sensor for a period of time in order to implement the required parameter changes. For the purpose of the implementation presented, a sensor configuration change (mode control) will be simulated by temporarily disrupting service availability of the sensor. Configuration holons are created with configuration parameters for one sensor only. If multiple sensors need to be reconfigured, multiple configuration holons must be created. Configuration holons are generally the highest priority holons and get service immediately, but they may also tie up a resource for a period of time encompassing a number of sensor control periods. During this time, other task holons will be unable to get service from the affected sensor.

5.2.4 Message holons

The communications between the platform-level SICH and the group holon require the creation of a message task holon. The message holons are the only task holons that interface with the communications resource holon and each message task holon is assigned a priority level. Access to the communications resource holon depends on priority level and the relative size of the messages to be sent. A low-priority message may have to wait for many communications control periods t_{comm} before securing access to transmit its data. Thus, by means of negotiations with the communications resource holon, platform-level messages are transmitted to the group level.

All task holons, excluding message holons, compete for access to the sensing resource holons. The competition occurs each at the beginning of each sensor control period t_{sensor} and the resulting data is delivered at the end of the period (see Figure 9). The control period of each sensor may be different and unrelated to the other sensors onboard a platform. Once a sensor becomes available to schedule tasks (for the next control period), those task holons requiring sensor service begin negotiating with it.

This competition is mediated by the priority attribute attached to each task holon. These task holons asking for service from the same sensor within the same control period are accepted on a priority basis. When sensor loading is high, some tasks may not get service during their desired control period and may have to try and negotiate service during subsequent control periods. This process provides a heuristic for use of the sensor resource holons, providing the best service to the most important sensing task holons at the expense of less important task holons.

5.3 Platform-level situation analysis

Situation analysis acts as a resource for the SICH. Its main purpose is to analyze the incoming data and make assessments of the evolving situation. Situation analysis decomposes into functions that are described below and used in the sensor management design.

5.3.1 Maintenance of tracks database

The situation analysis component at the platform-level holon maintains a database of all known tracks, including target dynamics, confidence (performance) measures and non-kinematical information such as identity. This represents the local tactical picture of the situation. The database contains track information for all targets falling within the sensing domain of the platform. New data provided by a tracking task holon is integrated according to the track update model employed. A description of the proposed tracking algorithm is included in Section B.

5.3.2 Assessment of tracking performance deviation

Tracking performance is assessed for each track by the tracking task holons as part of their functions. These measures can be labeled Q_i, where i indicates the individual track. Here, Q_i will represent the quality of the track. These performance measures are reported to the platform-level SICH and are stored as part of the track database. The average deviation of these measures from their desired[2] values $Q_{desired,i}$ represents the average tracking performance deviation of the platform for known tracks Q_{track}, an online performance metric. This can be written for N_T tracks as

$$Q_{track} = \frac{1}{N_T}\sum_{i=1}^{N_T}(Q_i - Q_{desired,i}) \tag{11}$$

5.3.3 Threat evaluation

In the current work, track priority is closely associated with threat level. Threat can be assessed by the direction and speed with which targets approach protected assets. A target can pose a threat to a platform, a protected asset, or any other point in the area of interest (*e.g.*, group centre), as illustrated in Figure 13.

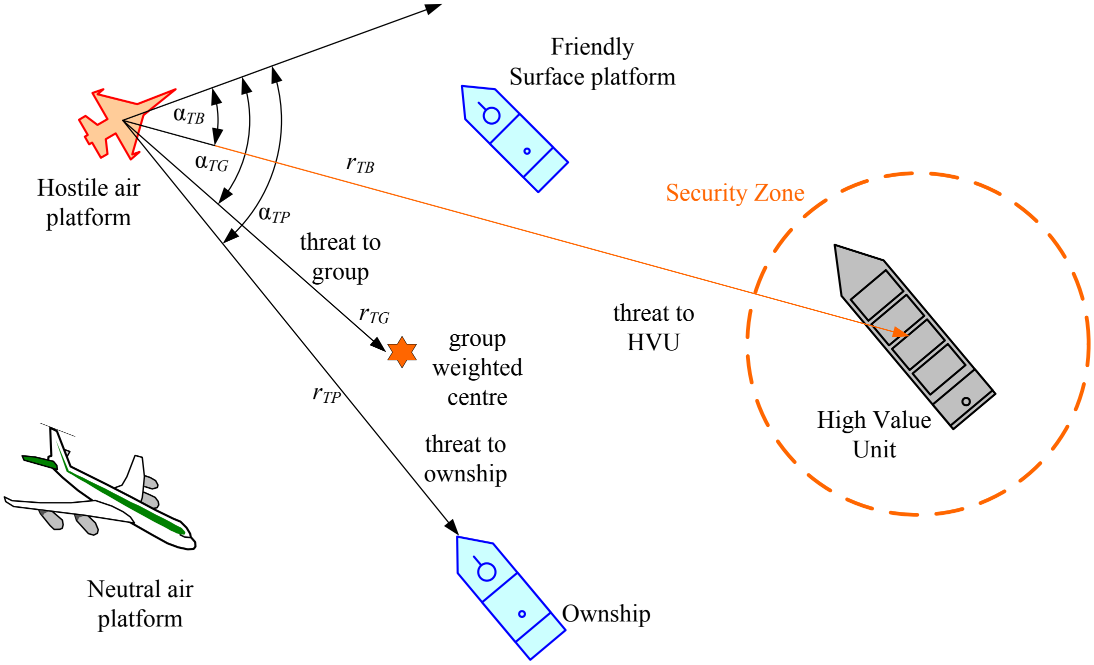


Figure 13: Threat evaluation

The output of the threat evaluation process is used, as depicted in Figure 14, to drive the sensing operations. Those targets that are deemed to be a greater threat to the protected assets will have to be tracked more closely than those that are benign.

Based on kinematics only, two typical measures of threat level for target tracks are the Closest Point of Approach (CPA) and the Time to Closest Point of Approach (TCPA). Consider a straight-line platform trajectory whose two-dimensional position $\boldsymbol{x}_p(t)$ can be

[2] The desired track quality is defined by the user as part of the initial system configuration. Nevertheless, in the design presented, mechanisms are included to allow for a dynamic setting of this parameter.

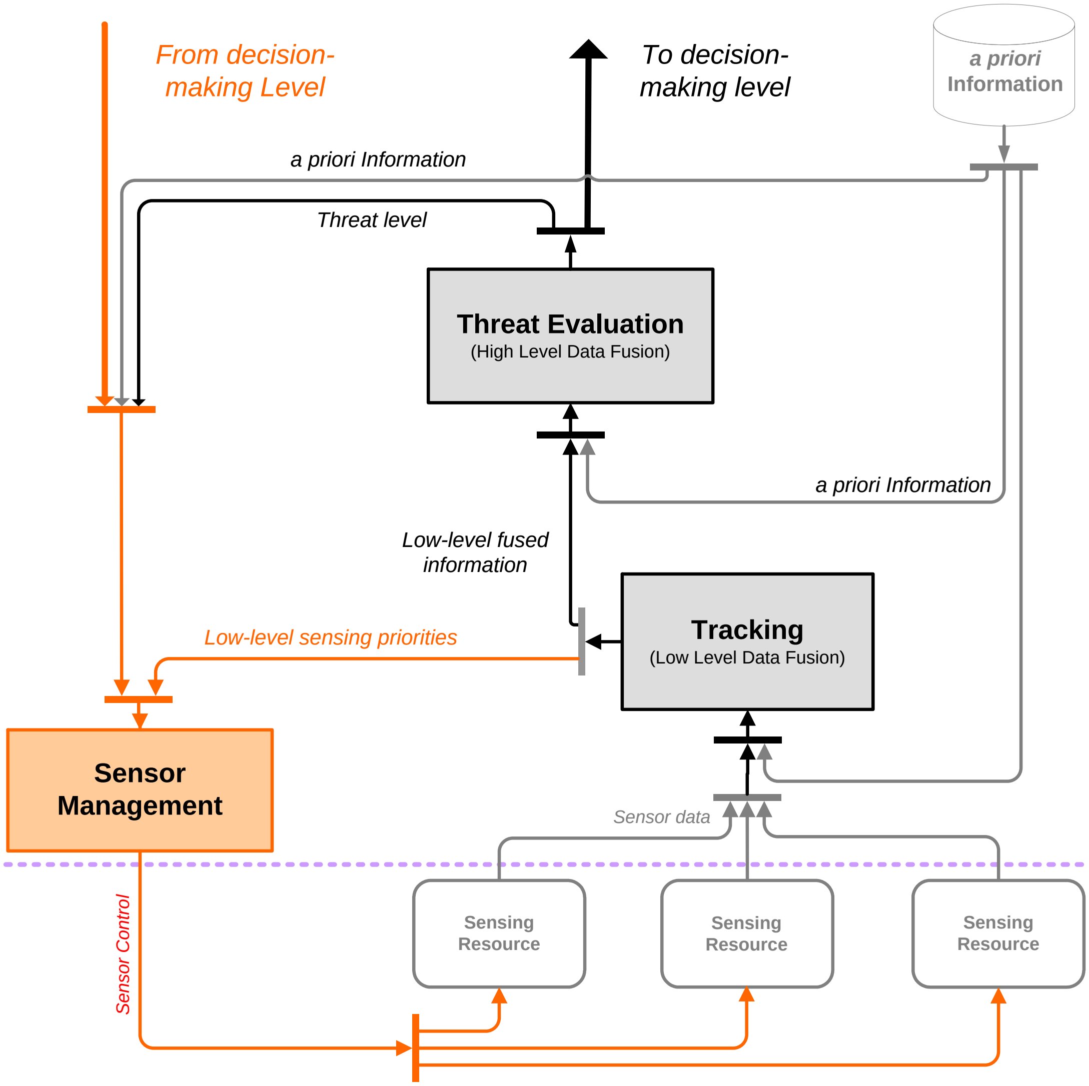


Figure 14: Exploitation of threat evaluation in sensor management

described using the measured velocity $\dot{\boldsymbol{x}}_p$ and the initial position $\boldsymbol{x}_p(0)$

$$\boldsymbol{x}_p(t) = \boldsymbol{x}_p(0) + t\dot{\boldsymbol{x}}_p \tag{12}$$

and also consider a point of interest (*e.g.*, target) moving with velocity $\dot{\boldsymbol{x}}_q$ described by $\boldsymbol{x}_q(t)$

$$\boldsymbol{x}_q(t) = \boldsymbol{x}_q(0) + t\dot{\boldsymbol{x}}_q \tag{13}$$

$\boldsymbol{x}_q(0)$ being the initial position (at $t = 0$).

Given these conditions, TCPA may be computed using the following (refer to Annex C)

$$TCPA = \frac{(\boldsymbol{x}_q(0) - \boldsymbol{x}_p(0))(\dot{\boldsymbol{x}}_p - \dot{\boldsymbol{x}}_q)}{|\dot{\boldsymbol{x}}_p - \dot{\boldsymbol{x}}_q|^2} \tag{14}$$

A negative TCPA value indicates an outbound target. The CPA is computed using the trajectory equations, thus

$$CPA = |\boldsymbol{x}_q(TCPA) - \boldsymbol{x}_p(TCPA)| \tag{15}$$

In order to compute a scalar measure of target threat Z, these two measures are combined according to

$$Z = k_1 \cdot TCPA + k_2 \cdot CPA \tag{16}$$

The constants k_1 and k_2 are chosen to scale Z and to balance the relative importance of TCPA versus CPA. When the threat-level of the target is computed against a platform, it will be denoted as Z_P, while threats against the protected asset are denoted Z_B, and other points, such as the group weighted center are indicated with Z_G.

The threat evaluation computation described here provides a method to rate threat based purely on the track kinematics. Other non-kinematical factors can also be used. Factors such as the identity of the target, determined either through radio contact or radar signature, or targets moving in known civilian air and waterways (such as an air corridor) may provide rationale for reducing the assessed threat level of the targets.

5.3.4 Prediction of likely location of new targets

Often it is possible to anticipate where new targets will appear in the area of interest. Generally, this involves some a priori knowledge of the situation but may also be computed from statistical measures. A predicted location of new targets helps refine the search tasks. This computation is limited to the sensing domain of the individual platforms.

The region around the platform will be divided into a number of sectors and maintain a record of the number of targets that have previously been discovered in each sector. This measure will be used to guide the creation of a new search holon or modify existing search holons attributes. In order to respond to new developments in the environment, these records should extend backwards in time to only a finite time horizon. The further back in time this horizon is chosen, the less reactive the sensor management system will be. If the horizon is too short however, the assessment is not likely to be very accurate.

For $N_{detect}(s)$ target detections in each sector s, the sectors are ranked in order of largest $N_{detect}(s)$. Those sectors with the highest ranking are considered the most likely for new target arrivals. Note that this calculation is independent of a similar one performed at the group level and may also be influenced by a priori knowledge of the environment, such as the presence of civilian airports, air bases, or established travel routes (air lanes, water lanes).

The group-level assessment of target arrivals may be used to refine the platform calculation, but this will not be implemented in current design. The group level target arrival calculation will still provide feedback to the platforms by initiating platform-level searches and coordinating cueing/handoff events between platforms. These initiated "special-searches" are given priority on the basis of the threat level of the cueing/handoff track.

5.3.5 Resource coverage assessment

This function considers sensor sector allocation, sensor operation modes, as well as the positioning of the platform to assess how well configured the sensors are for the current situation. Two (possibly competing) aspects are involved, resource coverage for optimal tracking and resource coverage to maximize the likelihood of detecting new targets (search). This assessment should also consider the coverage specification suggested by the group-level sensor management on the basis of a broader view of the situation.

For our purposes, we will consider every mode of every sensor and evaluate each mode combination for effectiveness against the current set of tracked targets. This measure ignores search priorities.

Sensor Quality Of Service (QOS) can be determined on the basis of knowledge of the sensor operating parameters. In this case, the bearing variance η_b and range variance η_r are used to compute the effectiveness of the sensor against the target. For target n at range r and sensor s, this quality of service $QOS(n, s, m)$ is computed according to Equation 7. Note that η_b and η_r, in Equation 7, are dependent on mode m, so this quality of service measure is mode dependent.

To assess the resource coverage, each sensor mode combination is ranked on the basis of the best match of track to sensor. This can be written as follows

$$Q_{\text{mode}}(m) = \sum_{n=1}^{N_T} \max_{s} |QOS(n, s, m)| \cdot P_n \tag{17}$$

Here, P_n is the current track priority level (base level) for target n. N_T is the number of targets currently tracked by the platform. The max operation is conducted over the set of platform sensors.

It is assumed that each track is assigned to the most effective sensor (max operation over all sensors). The sum provides an indication of how well the tracks could be serviced for each combination of sensor modes onboard the platform. For example, a platform with two sensors, each offering one of three modes of operation, can generate six mode combinations. Each mode combination is rated using the above equation and this is provided to the SICH as a vector for task planning.

The mode-specific QOS sum is weighted by track priority in order to penalize mode combinations that would ignore high-priority tracks. The measure also ignores sensor loads as it is designed for use in task planning rather than task allocation and scheduling. In other words, this calculation is used by the SICH in conjunction with a separate sensor load calculation in order to plan sensor configuration changes.

5.3.6 Prediction of cueing/handoff events

Tracks that leave the current platform sensing domain or tracks that cross from the domain of one sensor into another, on the same platform, require cueing/handoff coordination (Figures 15 and 16). Predicting these events in advance is necessary for the platform to maintain tracks and is also useful for sensor management at the group level (Chapter 6). This prediction is on the basis of the current sensor mode configuration for the platform and the current track information. At each track update, a new time and location prediction for cueing-handoff is computed. This amounts to a basic geometric/dynamic calculation that is not detailed here.

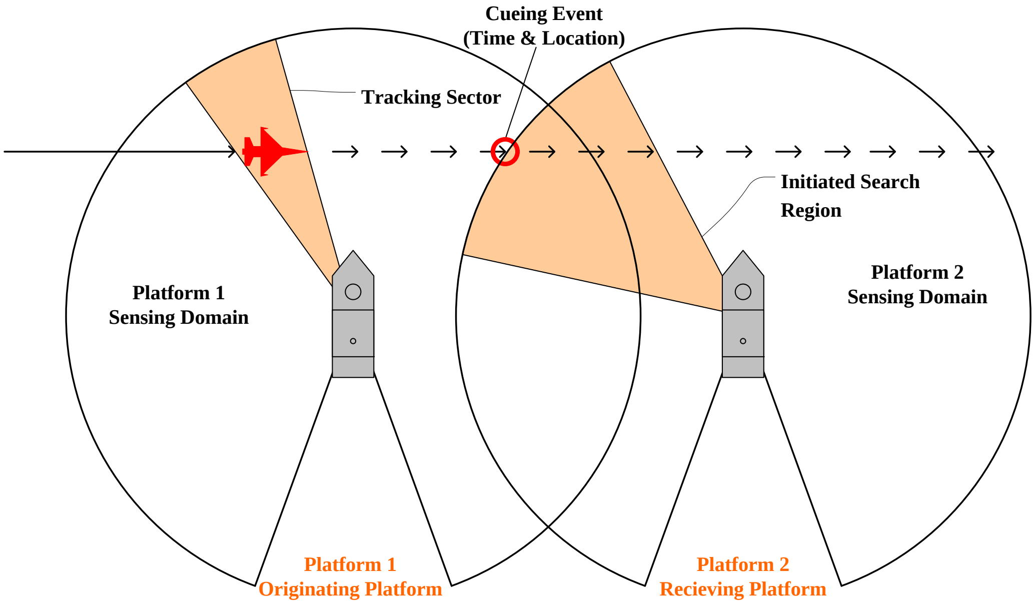


Figure 15: Inter-platform target cueing

The prediction includes time and location of target departure/arrival as well estimates of error in both measures. These measures are predicted from the current track information.

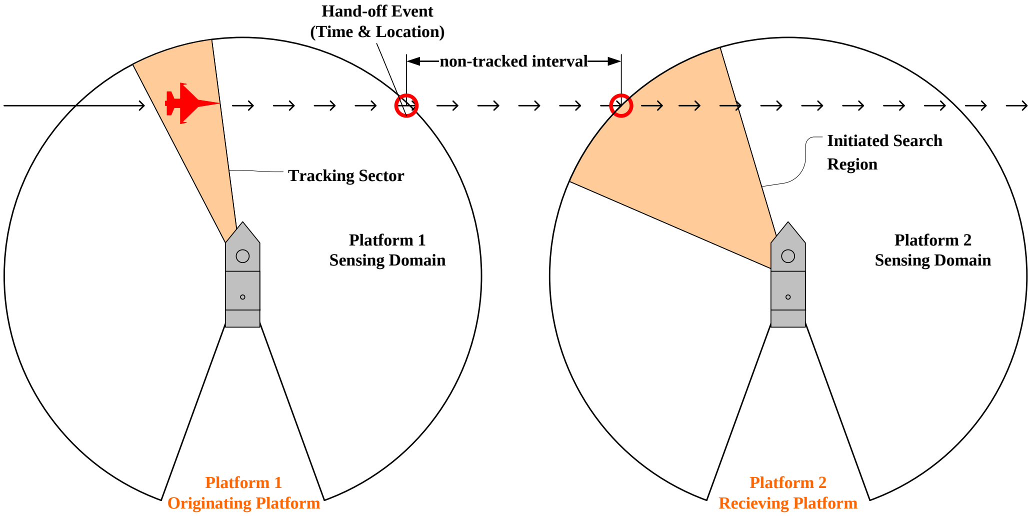


Figure 16: Inter-platform target handoff

The error estimate in time and location are based on the track error covariance.

5.3.7 Average platform load evaluation

The fraction of its time allocated to track update tasks determines the load of a platform. The load measure of the platform is derived from the load of the different sensors it carries.

Suppose, for example, that sensor s reports average load L_s. For a platform with n_s sensors, the average platform load L_p is computed according to

$$L_p = \sum_{s=1}^{N_s} w_s L_s \qquad (18)$$

Here the constants w_s weight the sensor loads relative to each another to account for the varying capacities amongst the sensors, with higher capacity sensors being weighted more. The weights w_s are platform-specific and are determined by the complement of sensing resources the platform is equipped with. One method of determining the weights is to base them on the scan rate ω of the sensors. The higher the scan rate, the more quickly track updates can be obtained, and thus the higher the capacity of the sensor will be to track targets. The following equation can be used

$$w_k = \frac{1}{\omega_k} \sum_{i=1}^{N_s} \omega_i \qquad (19)$$

where ω_i is the scan rate of sensor i. Since the individual sensor loads L_s vary in the range $[0, 1]$, the average platform load L_p will fall within this same range since

$$\sum_{s=1}^{N_s} w_s = 1 \qquad (20)$$

The average platform load represents the fraction of the total sensing capacity (*i.e.*, all sensors) onboard a platform that has been utilized for track update tasks. It should be noted that, when sensor scan rates are altered[3], the weights w_s must be recomputed in order to reflect the change in capacity.

The total sensing capacity C_p of a platform may be determined by summing the scan rate of all sensors

$$C_P = \sum_{s=1}^{N_s} \omega_s \tag{21}$$

and, therefore, the residual average sensing capacity RC_p of a platform can be computed according to

$$RC_p = C_P \cdot (1 - L_P) \tag{22}$$

The quantities L_p, C_p and RC_p will vary and must be intermittently reported to the group-level for task-planning purposes. The platform-level SICH is capable of gathering sensor load and configuration information through high bandwidth communications onboard the platform, but must generate a message holon to transmit this data to the group.

5.4 Platform-level SICH

At the platform level, the SICH is responsible for creating and terminating task holons in response to both group-level and platform-level sensing objectives. The general sensing approach is to configure the sensors (*i.e.*, mode, power, sector allocation) to best address the evolving situation and then manage the tracking and searching tasks from within this sensor configuration. The idea is that mode/power adjustments interrupt tracking and search operations and only need to occur infrequently in response to the evolving situation. Tracking and search occur much more frequently. Minimizing the occurrence of sensor reconfigurations also helps simplify the task of optimizing sensor utilization with respect to search and tracking [11].

Sensor management at the platform level must also balance the platform specific priorities with those of the group. The main mechanism to manage this is the priority attached to tasks. For example, tracks maintained by the platform are rated in terms of threat level to the platform. Track updates are assigned a priority that is directly related to the threat to the platform. If a track threatens a group asset, and its presence is unknown to the platform, the group-level holon may issue a high priority track assignment to the platform. This priority rating insures that the task will receive prompt attention despite the apparent lack of threat to the platform maintaining the track.

5.4.1 Task negotiations

The platform-level SICH negotiates with the group-level task holons for acceptance of sensing tasks. Group-level task holons are related to the broadest scope of the situation analysis

[3] Due to a configuration adjustment or sensor malfunction.

derived from data provided by multiple platforms. Due to communications limitation, the group-level sensor management is not capable of directing all sensing tasks and must delegate these to the platforms. Group-level task holons are generally limited to broad sensing objectives, such as adjusting search objectives and track responsibilities, as well as coordinating inter-platform sensing tasks such as cueing and handoff.

Group-platform task holon negotiations are complicated by the fact that wireless bandwidth limited communications must be used. This bottleneck introduces a time delay into all group-platform communications, as messages must be queued before being sent. To minimize communications, group-platform negotiations are simplified. Group-level task holons do not request a QOS estimate from the platforms, but instead simply delegate tasks to the platforms. The platforms do not respond directly to the group, instead they inform the group when an action such as initiating a track has been taken. This approach requires that the platforms regularly send status updates to the group.

This “task delegation” process is complicated by the fact that wireless communications are involved in transmissions to the platforms and in the return transmission from the platforms. The process is as follows:

1. Group-level message holons negotiate with the group-level communications resource holon to transmit a message, and are then terminated.
2. Once the message is received by the specified platform-level SICH, the platform-level SICH creates both a task holon to implement the delegated task and a message holon to inform the group-level holon that the task was initiated.
3. This message holon negotiates with the platform-level communications resource holon for transmission of the response message. When communication loads are high, the message holons may not be able to send the messages immediately, which creates a time delay.

5.4.2 Cue/Handoff group communications

For the most part, the platform-level SICH responds to data requests from the group-level task holons, but may sometimes require a message to be initiated at the platform level. Most notably, this occurs when an impending cueing/handoff event is detected that cannot be handled by the platform itself. For example, if a target is leaving the range of the platform sensors, the platform will lose the track. In this situation, a message should be sent to the group-level holon with all the information to allow a handoff of the tracking responsibility to another platform.

5.4.3 Task holon creation

Figure 17 illustrates a flowchart for the creation of task holons at the platform level. Holons are created in an event-driven manner on the basis of the analysis of the evolving situation. The flowchart can be considered as the logic of the state transitions and not a chronological description of task holon creation. The number and type of task holons defines the state of the platform-level sensor management.

The process is initiated with the creation of a number of search holons. The flowchart shows the process resulting from a single search holon. When a target is detected, a track holon and a message holon are created. The message holons inform the group-level holon when search and/or tracking tasks are initiated or terminated. Figure 17 also indicates that configuration changes, on the basis of local situation analysis, lead to the concurrent creation of configuration holons and message holons. Configuration changes at the platform level are based on tracking performance and can only occur after at least one track update task has been initiated.

Figure 18 illustrates the process of holon creation in response to group-directed cueing and handoff events. The process begins when a cueing/handoff event is recognized at the group level and a directive is issued to a platform. The platform assesses the received directive and either initiates a track (handoff) or a search to acquire the track (cueing). The platform-level SICH may also generate a configuration holon.

5.4.4 Determining priority and desired performance

The platform-level SICH is responsible for setting task holon attributes in response to the situation analysis. Among the various attributes, the most important are the base priority level P_0 and the task performance objective $Q_{desired}$.

5.4.4.1 Tracking holons

For tracking holons, these attributes are a function of the threat level that the track poses to the platform Z_p. In general terms, we can write

$$P_0 = k_1 \cdot Z_p + U \tag{23}$$

$$Q_{desired} = k_2 \cdot Z_p - k_3 \cdot L_p + U \tag{24}$$

Here, the constant k_1 relates the priority level directly to the threat level. Allowance is made for human (user) input through U. Likewise, the constants k_2 and k_3 relate the desired track quality directly to the threat Z_p and platform sensor load L_p. Note that the desired track quality is lowered in response to an increase in the platform sensor load.

5.4.4.2 Search holons

For search holons, a standard priority level $P_{standard}$ is defined from which the base priority level P_0 is modified, on the basis of the expected rate R_T of target arrivals in the search sector. The desired quality attribute is also modified from a standard $Q_{standard}$ on the basis of the expected target rate, but is moderated by the platform sensor load.

$$P_0 = P_{standard} + k_4 \cdot R_T + U \tag{25}$$

$$Q_{desired} = Q_{standard} + k_5 \cdot R_T - k \cdot L_p \tag{26}$$

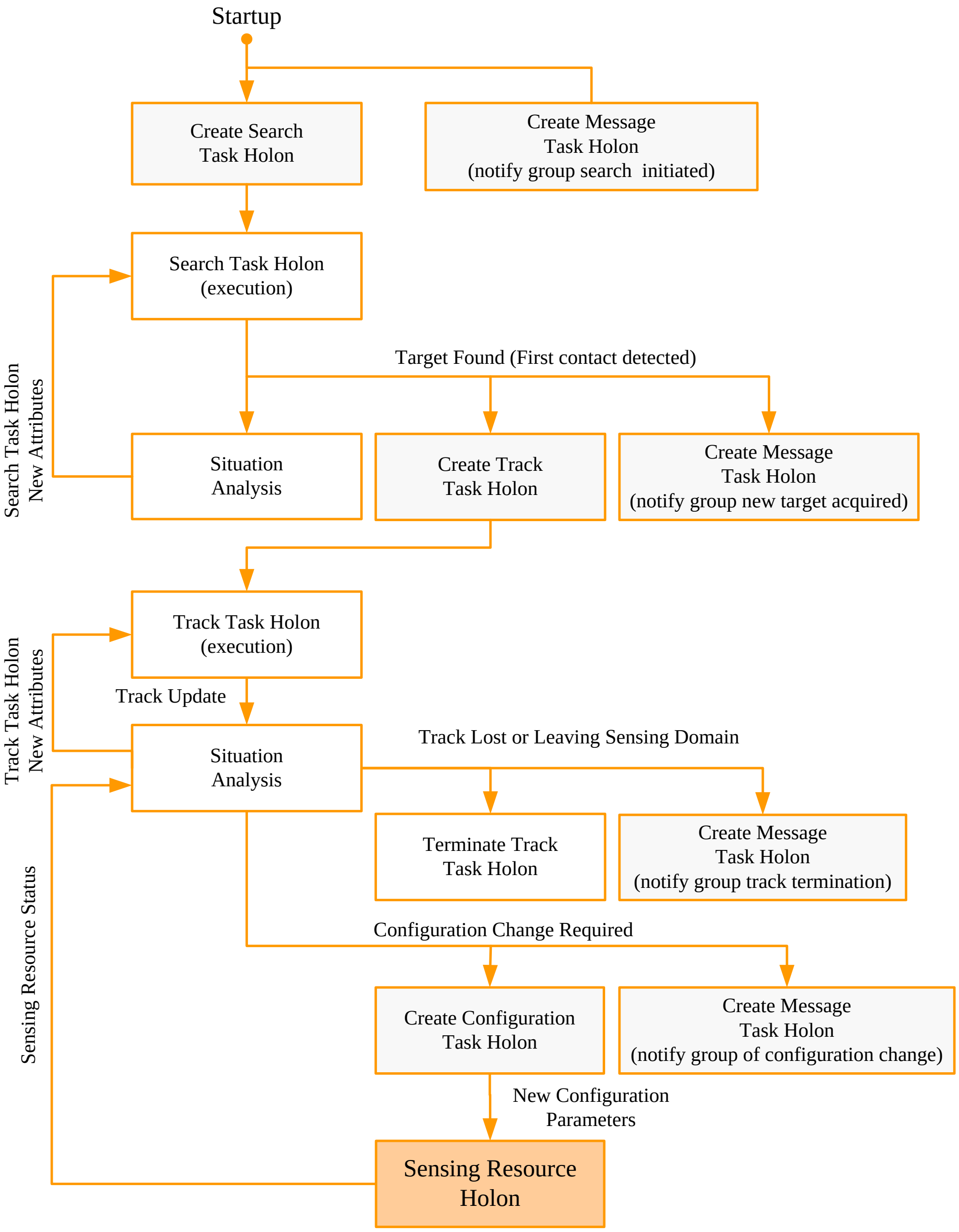

Startup
Create Search Task Holon
Create Message Task Holon (notify group search initiated)
Search Task Holon (execution)
Search Task Holon New Attributes
Target Found (First contact detected)
Situation Analysis
Create Track Task Holon
Create Message Task Holon (notify group new target acquired)
Track Task Holon (execution)
Track Task Holon New Attributes
Track Update
Situation Analysis
Track Lost or Leaving Sensing Domain
Terminate Track Task Holon
Create Message Task Holon (notify group track termination)
Sensing Resource Status
Configuration Change Required
Create Configuration Task Holon
Create Message Task Holon (notify group of configuration change)
New Configuration Parameters
Sensing Resource Holon

Figure 17: Platform-level holon management logic

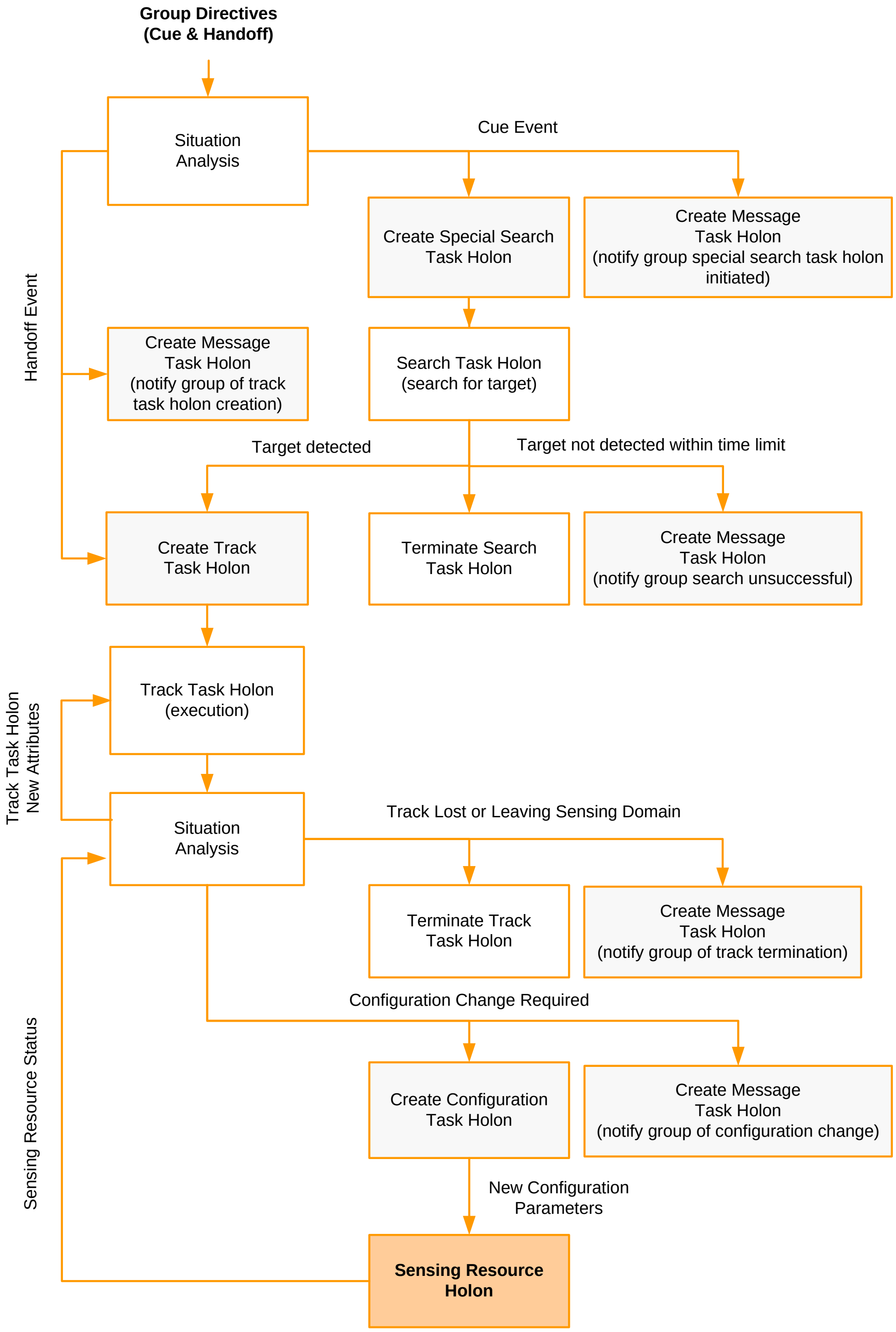

Group Directives
(Cue & Handoff)
Situation Analysis
Cue Event
Create Special Search Task Holon
Create Message Task Holon (notify group special search task holon initiated)
Handoff Event
Create Message Task Holon (notify group of track task holon creation)
Search Task Holon (search for target)
Target detected
Target not detected within time limit
Create Track Task Holon
Terminate Search Task Holon
Create Message Task Holon (notify group search unsuccessful)
Track Task Holon (execution)
Track Task Holon New Attributes
Situation Analysis
Track Lost or Leaving Sensing Domain
Terminate Track Task Holon
Create Message Task Holon (notify group of track termination)
Configuration Change Required
Sensing Resource Status
Create Configuration Task Holon
Create Message Task Holon (notify group of configuration change)
New Configuration Parameters
Sensing Resource Holon

Figure 18: Platform-level cueing & handoff holon management logic

Special search holons are created with higher priority to detect specific targets for cueing/handoff event coordination. They behave as tracking holons and their attributes are calculated in a similar manner:

$$P_0 = k_6 \cdot Z_p + U \tag{27}$$

$$Q_{desired} = k_7 \cdot Z_p - k_8 \cdot L_p + U \tag{28}$$

5.4.4.3 Configuration holons

Configuration holons are high priority tasks and can be assigned the same priority level each time they are created. This should be a high value to insure that the task is implemented promptly. The configuration quality specification should be associated with how many sensor control periods the configuration task takes to implement. This quality specification can also be made task-independent, *i.e.*, constant for all tasks. Thus, for configuration holons, one would have

$$P_0 = k_9 \tag{29}$$

$$Q_{desired} = k_{10} \tag{30}$$

The above equations are presented in a general form without specifying the relating constants k. The values for these constants must be related to the task holon performance evaluation Q_{task}. It is important to note that the equations above must be properly defined to provide robust (and stable) performance of the holonic control system.

5.4.4.4 Message holons

Message holons should take into account the priority of the message that they are trying to transmit. For example, if the message pertains to a track update task, the base priority of the associated track holon will be used. Message quality specification is set as a constant.

5.5 Platform-level search task holons

Platform-level sensor management tasks sensors to search regions for undetected targets. Generally, regions of interest are searched periodically. The platform-level SICH creates search task holons that negotiate access to the sensors to perform a search. Search objectives are typically ongoing and periodic, *i.e.*, specific areas need to be scanned for a short time, but repeatedly over a longer time horizon. For this reason, the search holon is not terminated after a single search but remains active, repeatedly negotiating access to the sensors. When a search holon is created, its attributes specify:

1. The sector to search,
2. The minimum searching quality to maintain (if possible), and
3. The priority of the task.

If the resources are under-utilized, the search task can proceed with no interval between searches. However, if the resources are very busy, some search tasks may exhibit too low a priority to acquire sensor access.

To overcome this problem, the search holon is allowed to adjust its own priority. This functionality is limited by making the platform-level SICH impose a variable priority and a priority rate parameter upon the search holon that controls the rate at which the search priority is increased. The priority level starts at the assigned level and increases on the basis of an internal assessment of the search performance. It must also have a specified limit in order to prevent very important tasks (*e.g.*, incoming targets) from being interrupted. Once the search performance objective is achieved, the task priority can be gradually lowered, provided that the performance is maintained. If the search performance falls below some minimum level, the task will be considered as having failed and this will be reported to the platform-level SICH immediately.

5.5.1 Assessment of search performance

The performance of a search task holon is related directly to the amount of sensor time the task holon is able to attain and the QOS that the sensor is able to provide. To be useful, this should be averaged over a period of time and not be based on just one sensor allocation.

Sensor QOS is based on the range and bearing variance of the sensor (mode specific), as well as on the scan rate. Equation 8 describes one method for computing QOS for a given sensor s. A high QOS measure reflects the desirable sensor properties of high scan rate and long detection range with little sensor variance.

The task performance Q_{search} is assessed on the basis of the amount of time allotted to the search task multiplied by the QOS of the sensor responding. This performance measure is averaged over a number N_{CI} of sensor control periods. Thus, for a holon that is serviced by N_s different sensors, Q_{search} is computed according to

$$Q_{search} = \frac{1}{N_{CI}} \sum_{i=1}^{N_{CI}} \sum_{s=1}^{N_s} t_i(s) \cdot QOS_{search}(s) \tag{31}$$

Here, $t_i(s)$ is the sensor time allotted to the search during period i by sensor s. During each control period, the holon receives service from only one sensor s_{win}, thus one can write

$$Q_{search} = \frac{1}{N_{CI}} \sum_{i=1}^{N_{CI}} t_i(s_{win}) \cdot QOS_{search}(s) \tag{32}$$

5.5.2 Adjustment of priority level

Search task holons are created with a desired task quality $Q_{desired}$ that the holon seeks to achieve. One means of gaining access to resources is through increasing the priority level of a task. The following attributes, given by the platform-level SICH to task holons, limit an increase in priority level in response to Q_{search} values falling below $Q_{desired}$.

- Priority base value: P_0
- Priority increase rate: m_p
- Priority maximum: $P_{\max}$

If the search task quality Q_{search} is greater than $Q_{desired}$, the task priority P_{search} will be set at the base value P_0. If Q_{search} falls below $Q_{desired}$, the increase rate m_p specifies the increase per time unit according to the following

$$P_{search} = P_0 + m_p \cdot (t - t_0) \tag{33}$$

where t_0 is the most recent time when Q_{search} became less than $Q_{desired}$. Finally, P_{search} is not allowed to increase beyond $P_{\max}$.

5.5.3 Search task failure

In the event that the task holon cannot acquire service for a long period of time, or that service attempts repeatedly fail, the task holon will report a task failure to the platform-level SICH. This is controlled by the holon attribute $Q_{\min}$. The condition for task failure is then trivial: when Q_{search} falls below $Q_{\min}$, the search task is considered to have failed.

5.6 Platform-level track task holons

At the platform level, the SICH generates one track holon for each target track. Each track holon is generated with a number of attributes that are required for the holon to perform its job. These include:

- Target identity and current tracking data
- Desired track performance
- Minimum tracking performance
- Priority of the task

The task holons compete for access to the resources. Tracking tasks require only intermittent access to the sensors to maintain a given track quality. The higher the specified quality, the more often the holon needs to secure an update. If the holon cannot secure a track update and the track quality falls below a specified minimum, then the task is considered to have failed and this is reported to the SICH. The specified minimum is determined by the SICH when the track holon is created and may be modified at any time by the SICH. For example, when sensor loads increase dramatically, it may be necessary to lower the specified minimum in order to maintain all of the tracks.

5.6.1 Assessment of tracking performance

The tracking performance is central to the operation of the tracking holon. For the purpose of the design presented, tracking performance will be defined as the reciprocal of a tracking error measure (*i.e.*, the information measure). In this case, it is derived directly from the tracking algorithm error covariance matrix $\boldsymbol{P}$, as described in Annex B.

For our purpose, we will consider positional terms of error covariance to be the tracking quality measure

$$Q_{track} = \left[\sqrt{P(1,1)^2 + P(3,3)^2}\right]^{-1} \tag{34}$$

where $P(1,1)$ and $P(3,3)$ are the diagonal elements of the error covariance matrix $\boldsymbol{P}$ (see Annex B).

When sensors are under-utilized (low sensor load), track updates can be performed as desired, maintaining the tracking quality. If the sensors are busy (high sensor load), then the task priorities are used to decide amongst the various task executions, resulting in possible delays in track updates [16]. To address this problem, tracking-holons are equipped with a variable task priority parameter. This allows the track holon to increase its priority level over time as tracking performance Q_{track} decreases. The priority level increase rate is specified when the holon is created, as are the starting and maximum priority levels. The priority maximum insures that the task does not interfere with more important ones.

5.6.2 Track update request

Tracks must be updated in order to maintain a specified track quality. This quality specification $Q_{desired}$ is determined by the platform-level SICH and is used to control when a track is updated. The process is simple. The Kalman filter tracking prediction (Annex B) provides a means to compute the state covariance matrix $\boldsymbol{P}$ and therefore predict how track quality will deteriorate in the future without additional track updates. The time at which the predicted track quality $Q_{predicted}$ falls below the desired quality $Q_{desired}$ indicates the time before which a track update must be acquired in order to maintain Q_{track} greater than $Q_{desired}$. It is the responsibility of track holons to negotiate sensor service prior to this point. For newly acquired tracks, this will lead to a quick succession of track updates until $Q_{desired}$ is met, after which the updates should occur at longer and more regular intervals.

Since the tracking process is based in Cartesian coordinates, a regularly updated track moving away from the observer will result in a track quality Q_{track} that decreases with time, according to the definition adopted here. As a result, the performance specification $Q_{desired}$ cannot be a constant value but, instead, must be scaled by target range r. Thus, the platform-level SICH must specify a base performance specification Q_{base} and a scaling factor k for target range

$$Q_{desired} = k \cdot r \cdot Q_{base} \tag{35}$$

5.6.3 Task priority adjustment

The track task priority is adjusted by the task holon in response to Q_{track} falling below a desired task quality measure $Q_{desired}$. This adjustment is handled in the same manner as with the search holon (§5.6), where the task priority is adjusted on the basis of the task performance assessment Q_{track}.

Tracking task base priorities are generally set higher than search task priorities, as illustrated in Figure 19. The figure depicts a priority scale ranging from $P_{\min}$ to $P_{\max}$ with larger numbers indicating higher priority. Tracking tasks are assigned a base priority anywhere within the scale, while search tasks are assigned a priority in the lower range below $P_{search,\max}$. This reflects the design methodology of sacrificing search performance to maintain target tracks when sensor loads are high. The priority scale helps provide a measure of relative importance in sensor management tasks. Target tracks considered to be important (*e.g.*, high threats) are given high priority and are not interrupted by search tasks. However, tracks that are not as critical (such as friendly targets) may be temporarily interrupted in order to maintain the search for new targets.

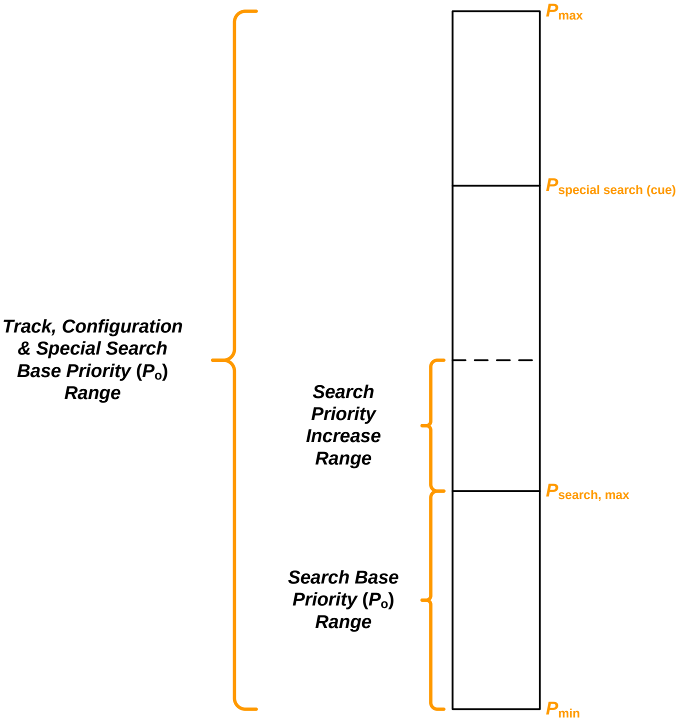

P_{max}
$P_{special\ search\ (cue)}$
Track, Configuration & Special Search Base Priority (P_o) Range
Search Priority Increase Range
$P_{search,\ max}$
Search Base Priority (P_o) Range
P_{min}

Figure 19: Task priorities

Since task holons can adjust their priority levels, a search task priority level P_{task} may grow larger than $P_{search,\max}$. In this way, low-priority tracks can be interrupted (temporarily) to perform high-priority search tasks. Additionally, search tasks (special searches) that are delegated by the group-level holon typically involve track cueing. In these cases, the search task priority is based on the threat level of the target involved and may fall anywhere within the priority scale.

5.6.4 Track task failure

In the event that the track holon cannot acquire service for a long period of time, or that service attempts repeatedly fail, the track holon will report a task failure to the SICH. The failure condition is set by the platform-level SICH as the holon attribute $Q_{\min}$. The condition for track task failure is similar to that of a search task failure, *i.e.*, when Q_{track} falls below $Q_{\min}$, the task is considered to have failed.

Determination of the value of $Q_{\min}$ is based upon the platform load, *i.e.*, the number of tracks the platform is currently maintaining. When excess sensing capacity is available on the platform, $Q_{\min}$ can be set higher than when little capacity remains. $Q_{\min}$ can be related to the number of misses, *i.e.*, unsuccessful track update attempts, that can be tolerated. For example, if the platform load is high, $Q_{\min}$ may be set so that more misses may be tolerated.

5.6.5 Track verification phase

The track holon is created by the SICH in response to a target detection by a search holon. The track update holon therefore must first verify the track before starting the tracking process as described above is implemented. This phase of the track holon operation is characterized by a rapid succession of target updates that can be used to initialize a track. If the track is verified in this manner then the normal tracking process above begins. If the track holon fails to verify the track, then the track holon is terminated.

During this phase, the track holons seek track updates at a predetermined frequency (*e.g.*, one second apart) in order to initialize the tracking filter. If the target can be detected several times, then the track can be established and track quality can be computed. If the target cannot be repeatedly detected, the verification will fail and the holon will be terminated. If it is detected, the verification phase continues for a specified period of time during which the track quality Q_{track} is brought above $Q_{\min}$. At this point, the verification stage is over and normal tracking can proceed

5.6.6 Handling cueing/handoff

To maintain track quality, tracks predicted to cross from one sensor coverage area to another onboard the same platform must be handled through handoff cooperation [1]. When a cueing/handoff event is predicted, the track holon responsible for that target will automatically negotiate access to the new sensor for track updates. If the target is passing into a region that is within the platform sensing domain, but not within the current coverage due to mode/power settings or sector allocations, the target cannot be tracked without a configuration adjustment. This situation will be predicted by the situation analysis and a configuration holon may be created to modify the sensing parameters. The configuration holon will only be created if the configuration adjustment is appropriate. This can be measured by weighing the utility of tracking the target versus the cost of the resultant disruption in other tracking and searching tasks. Configuration changes and/or dropped tracks are reported by the platform-level SICH back to the originating group-level SICH, through the associated group-level tracking holon [4].

A more complicated situation arises when the track passes through a blind spot in the sensor coverage where track updates are unavailable [1]. A blind spot may be due to a permanent

[4] See Chapter 6 for more details on these last two.

physical sensor blockage (such as the ship radar mast) or to a sensor failure. In these cases, a track handoff is required. Within the design presented here, this is handled as follows: the track quality will degrade as the target passes through the blind spot and when the track quality Q_{track} falls below the $Q_{\min}$ threshold, the track holon will be terminated. A special search holon will be created to re-establish the track as it emerges from the blind spot, according to estimates derived from the previous track. The time window of the search operation and the width of the search region are derived from the track uncertainty value.

Track handoff may be required even when there is no blind spot present; the determining factor is the uncertainty associated to the track. For example, when a configuration adjustment is needed to continue tracking a target, there will be a short period where the sensor will be unavailable for track updates. If the track is also newly acquired (*i.e.*, low Q_{track}), then the time delay may cause Q_{track} to fall below $Q_{\min}$ and a handoff, rather than a cueing operation, will occur.

Track handoff onboard a single platform is generally not necessary, as the transit time between sensor coverage areas is usually small enough to prevent the tracking holon from terminating. The platform-level SICH may facilitate this by reducing $Q_{\min}$ when cueing/handoff events are predicted. In this case, the track quality may degrade to the point at which the next track update requires a search over a broad area. Thus, cueing in this case becomes very much like handoff, only the track is not dropped in the interim.

5.7 Platform-level advanced functionalities

This section describes possible advanced functionalities of the above described platform-level task holons, which are not considered in the design presented. They are yet described here for the completeness of the report.

5.7.1 Handling irresolvable tracks

Considering the observed 2D space, tracks that align themselves along the missing dimension relative to a given sensing platform are irresolvable from the platform point of view. Recognizing and predicting this occurrence is part of the platform-level situation analysis. Irresolvable track situations are typically short-lived and only temporarily affect the tracking quality of the targets involved. There is little that the platform can do to overcome this situation. When target tracks are irresolvable for longer periods, such as with aircraft formations, the platform may request assistance from the group-level sensor management for a separate point of view.

Irresolvable tracks are discovered at the platform-level situation analysis. The platform-level SICH will inform the group-level tracking holons, associated with the tracks in question, of the irresolvable complaint. These holons will in turn inform the group-level SICH, which can issue additional tracking holons to gather tracking information from other platforms (*i.e.*, target(s) allocated to more than one platform). The original tracking holons at the group-level pass the updated track information to the originating platform. The *helper* tracking holons at the group-level exist until the targets are deemed resolvable from a single platform again.

5.7.2 Handling merging tracks

In the case of closely spaced targets, the tracking resolution can be increased by specifying a more appropriate sensor mode, or through increasing the frequency of track updates (*i.e.*, specify higher quality track). In the first case, the platform-level SICH could issue a configuration holon to improve the sensing performance of a particular platform. In the second case, the original tracking holon(s) could be terminated and (a) new one(s) issued in its place with a higher quality of track specification. This same strategy can be employed when the tracks are merging or crossing. a detailed method of handling merging tracks is described in [16].

5.7.3 Track while search

This is an advanced search technique where track updates are combined with searching. If a target happens to be in the sector where a search is occurring, a track update can be supplied at the same time as the sector is searched. This provides for more efficient sensor use. The complication here is that a search task will actually provide a track update. The fusion algorithms must be able to distinguish whether the measurement is a new target or part of an existing track. Either way, the search holon will detect the target and report it, but this provides no savings unless the track holon associated with the target is informed and does not try to negotiate a track update during the search period.

5.8 Platform-level configuration task holons

Configuration holons are issued when a change in sensor configurations is required. Generally, this is in response to a request from the group-level sensor management, although it may also be in response to the platform-level situation analysis. Configuration adjustments, such as a mode change, can interfere with tracking and searching tasks, and therefore configuration holons are only issued occasionally but with a high priority attribute. Each configuration holon is created to negotiate with a single sensor for configuration adjustments (mode control). Once the sensor reports that the configuration adjustment is complete, the task holon is terminated.

Sensor rate and power setting specify the range and resolution of the sensor (*i.e.*, sensor mode), which can be adjusted to compensate for fast moving targets tracked over long ranges, or slower moving targets tracked over shorter ranges. Configuration may also include sector allocations that specify the operating sector of individual sensors. If, for example, a platform is equipped with two identical sensors, it might make sense to dedicate each one to tracking within non-overlapping 180° sectors. More pointedly, given a platform with different sensor types, sector responsibilities can be tailored to the situation. As an example, consider the situation where all airborne targets approach from one direction while all water-based targets are approaching from another. A suitable sensor configuration might be to use short range sensors to track the water-based targets and a long-range sensor to track the aircraft.

Configuration adjustments temporarily interrupt other tasks from being performed by the reconfigured sensor. In addition, existing task holons such as search and track may have to utilize different sensors after a configuration adjustment. Within the design presented, these task holons would simply need to negotiate service from a different sensor in order to continue their tasks. In this way, configuration adjustments can be implemented without the need to explicitly re-plan all ongoing sensing tasks.

5.9 Platform-level navigation task holons

Platform navigation is an important aspect of sensor management especially when remotely guided vehicles, such as Unmanned Air Vehicles (UAVs), are involved. The design proposed here can incorporate such navigation requirements, but requires an additional set of situation analysis and control metrics to be derived. Due to time and budget limitations of the work reported, the effort was focused on sensor management onboard platforms, such as frigates, ignoring the design of sensor management navigation. The following short description of this functionality is provided for completeness.

Platform navigation is generally the responsibility of the platform, although, in the case of a UAV, navigation is remotely controlled. It is the role of the platform-level SICH to decide whether or not to accept the navigation commands specified by the group-level sensor management. Once accepted, a platform-level navigation holon that secures the resources and implements the position change is created. For platforms that are currently under navigation, the current navigation holon is replaced with the new navigation holon.

Navigation holons are issued with a desired new platform position/orientation, time-frame for completion, and priority level. Since platform navigation affects tracking and searching tasks, the holon may revise the initial navigation plan on the basis of its negotiations with other task holons (*e.g.*, delay before moving) and the platform-level SICH may accept this revision or terminate the holon and reissue a higher priority or revised navigation holon.

5.9.1 Holon creation

Navigation requests received from the group-level sensor management are balanced with navigation objectives derived at the platform-level. These two objectives are compared and a compromise between the two is used to generate navigation plans. Navigation plans are then assessed to see if the disruption to searching and tracking is more harmful than not implementing the navigation. The following calculations may be used:

- Generate navigation plans to address platform-level sensor management concerns
- Generate navigation plans to address group-level sensor management concerns
- Find compromise plans and attach priority to task

This page intentionally left blank.

6 Group-level holons

The group-level sensor management holon is the most abstract level of sensor management considered here. The group-level sensor management acts as a coordinator between multiple platforms, each implementing its own sensor management. Situation analysis at the group level is based on information provided by the platforms, creating a broader picture of the environment than is available to any of the platforms individually. Because of the limitations in the group-platform communications link (*i.e.*, wireless), sensor data is analyzed at the group level and inter-platform coordination is achieved by issuing tasks to the platforms. With target tracking, for example, data sent from one platform may indicate a track threatening a second platform. The group-level sensor management would recognize this situation and issue a search task to the second platform.

The group-level sensor management relies on the platforms to detect and maintain target tracks (*i.e.*, secure track updates). Track information is relayed to the group where a global track database is assembled. Since group-platform communications are bandwidth limited, the track database at the group level will not be updated each time a platform updates an individual track. The approach taken here is to conserve communications bandwidth by prioritizing which tracks are updated in the group track database. For instance, tracks with high priority (*i.e.*, actual threats) will be updated often, while low priority tracks will be updated much less frequently. In this manner, the most important information is also the current information at the group level.

Three main tasks that will be coordinated at the group level are: search, tracking, and configuration. The nature of these tasks differs from their counterparts at the platform level. In searching, for example, the group-level sensor management can use tracking information from one platform to predict the arrival of new targets at a second platform. With this knowledge, a search to acquire tracks can be initiated on the second platform before their arrival. Once a search is initiated, the group-level sensor management can only periodically monitor the platform as it conducts the search.

Subsequent to any track initiation at the platform level, a message will be sent to the group sensor management with the track information. In response, the group-level sensor management will begin a process of periodically monitoring and requesting new track data from the platform that initiated the track. The group requests these updates with a frequency that is based on an assessment of the track importance (*e.g.*, threat) relative to the other tracks it is aware of.

Platform sensor configuration is primarily the responsibility of the platforms themselves. However, the group-level sensor management may be in a position where the information gathered by one platform is used to suggest a configuration adjustment at another one. In these cases, the platform must balance its own configuration assessment with that of the group and reconfigure the sensors accordingly. One example of this is when targets are approaching a platform sensing domain but are yet beyond the range of the sensors in their current configuration. In this case, sensor/track information from a second platform may be used as the basis for requesting a configuration adjustment.

In addition to monitoring track and search tasks, the group-level sensor management must respond to the situation analysis provided by the platforms and use this information for its own assessment of the larger situation. This primarily comes in the form of messages sent by the platforms that indicate some critical situation arising. Most notable among these would be the loss of a track or the prediction of an impending cueing/handoff event.

The group-level sensor management holon is depicted in Figure 20. The main difference between this holon and the platform-level one is that at the group level, the task holons interface with the platform resource holons using the communications holon.

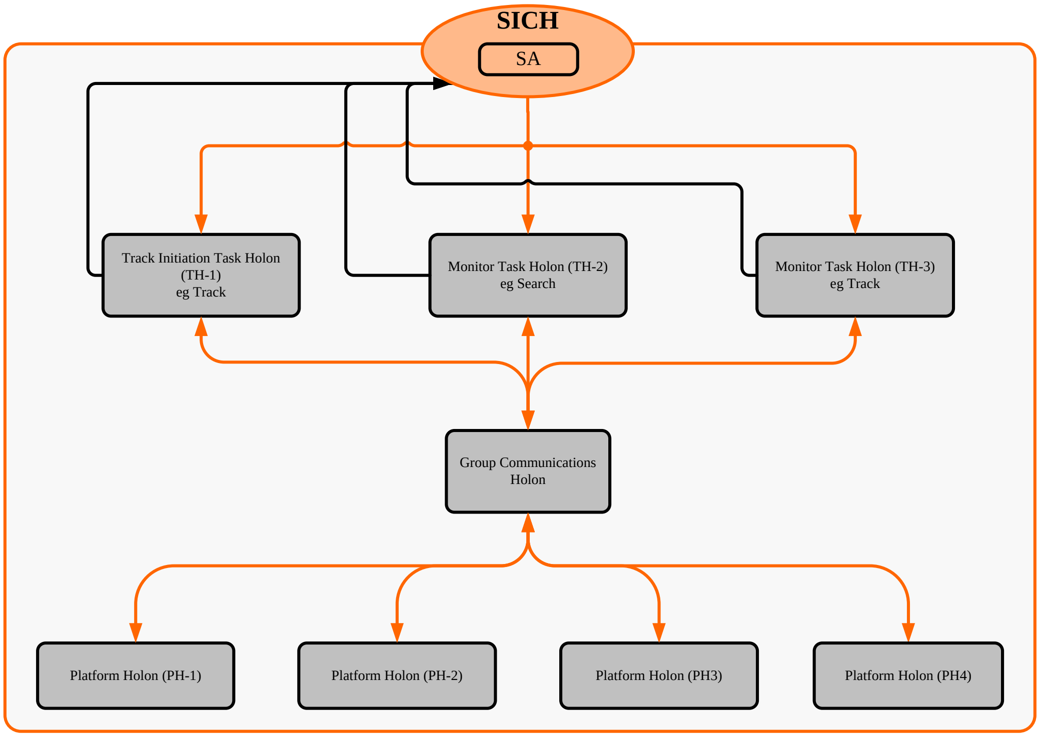

SICH
SA
Track Initiation Task Holon (TH-1) eg Track
Monitor Task Holon (TH-2) eg Search
Monitor Task Holon (TH-3) eg Track
Group Communications Holon
Platform Holon (PH-1)
Platform Holon (PH-2)
Platform Holon (PH3)
Platform Holon (PH4)

Figure 20: Simplified group holon

At the platform level, only message holons access the communications holon. At the group level, however, all of the task holons require access to the communications holon in order to send messages to the platforms. The group-level communications holon (§6.8) operates in the same manner as the communications holon at the platform level, transmitting task messages in order of priority.

For the most part, the group-level holarchy operates in the same manner as at the platform level. The group-level SICH generates task holons using the results of the situation analysis. These task holons are created with an attached priority that can be adjusted by either the group-level SICH or the task holon itself on the basis of an internal assessment of performance. Group-level task holons negotiate with the platforms to complete their task and the data is reported to the group-level SICH. In the design presented, it is assumed that there is a high-bandwidth connection between the SICH, the situation analysis function, the task holons and the communications holon. Table 3 details the holons contained within the group-level holarchy.

Table 3: Group-level holons

Name	Type
Service Interface Command Holon (SICH)	Interface
Track Holon	Task
Search Holon	Task
Monitor Holon	Task
Configuration Holon	Task
Communications Holon	Resource
Platform Holon	Resource

6.1 Group-level situation analysis

Group-level situation analysis is a capability that the group-level SICH can draw upon to make decisions regarding task creation or to modify task holon attributes (such as their priority level). The following functions will be employed in our design.

6.1.1 Track database for all known objects

The situation analysis component at the group level assembles and analyzes track information provided by the different platforms. This information is used to update the group-level understanding of the situation. The track database consists of target dynamics as well as confidence measures and non-kinematical information such as identity. The update rate at the group level is based on the track priority as deemed by the group-level SICH.

6.1.2 Database of platform status

Platform status information is transmitted by the platforms intermittently and stored in a database as part of the situation analysis. This information includes details such as platform location, orientation, heading, platform loading, sensor configuration and sensor operation status. This data is used in group-level SICH task planning.

6.1.3 Prediction of domain transitions

Each track in the database can be extrapolated to predict an approximate time and location where the track is expected to exit a platform sensing domain, and more importantly, when it is to enter the sensing domain of another platform. This is a straightforward kinematical calculation and is not presented here.

6.1.4 Threat evaluation

At the group level, track information reported by the platforms is rated in terms of threat level. Depending on the mission, this assessment may differ from the threat evaluation performed at the platform level, since the group-level sensor management is concerned with a broader picture. For example, a target heading away from one platform might be considered as low priority by the platform. However, if that target is heading in the direction of another platform, the group-level sensor management should recognize this fact and rate its threat level much higher. In this case, the group-level sensor management may ask the platform that a higher quality track be maintained on that object.

6.1.5 Prediction of likely location of new targets

It is often possible to anticipate where new targets will appear in the area of interest. Generally, this involves some a priori knowledge of the situation, but may also be computed from statistical measures. However, generally, targets may appear from any direction at any time. Most important, at the group level, is the prediction of targets entering the sensing domain of individual platforms. Endowed with a broader view of the situation than the platform, the group-level sensor management is in a position to make such determinations on the basis of available tracking information. Both the time and location of target arrival can be estimated and used to initiate a high priority search (special search) task to the receiving platform in order to re-acquire the track.

In a manner similar to the platform, the area of interest is divided into sectors. New target detections within each sector are recorded for a specified period and used to rate sectors in terms of likelihood of new targets. For N_T targets overall and for $N_{detect}(s)$ target detections in each sector s, the sectors are ranked in order of largest $N_{detect}(s)$.

6.1.6 Resource load estimation

Platform load L_p, total capacity C_p and residual capacity RC_p are calculated at each platform as part of the local situation analysis. This information is transmitted to the group intermittently and provides sufficient information for task planning at the group level. Group loading L_G, group capacity C_G and group residual capacity RC_G are useful metrics for task planning and coordination.

Group loading can be estimated from the reported platform loading L_p, as follows

$$L_G = \sum_{P=1}^{N_p} w_P L_P \tag{36}$$

Here, N_p platforms are assumed each with sensing capacity C_p and the weights are computed from the sensing capacities according to

$$w_P = \frac{C_P}{\sum\limits_{i=1}^{N_p} C_i} \tag{37}$$

The total sensing capacity of the group is computed by summing the individual platform capacities

$$C_G = \sum_{i=1}^{N_p} C_i \tag{38}$$

Thus the residual group capacity is computed according to

$$RC_G = C_G \cdot (1 - L_G) \tag{39}$$

6.1.7 Estimation of communications load/delay

In order to plan for cueing/handoff between platforms, the load of the communications resource is critically important. One means of estimating wait times is to average individual message wait times over a finite interval on the basis of priority level. Thus, for cueing/handoff tasks, the average wait time for high priority tasks can be used to estimate how long the communications delay will be.

For this calculation, the task priority scale is divided into low, medium and high priority ranges. Task holon communications wait times are determined for each category by averaging the wait times of recently completed tasks.

6.2 Group-level SICH

At the group level, SICH is responsible for creating task holons to address group-level sensing tasks. Once created, the task holons negotiate with the resources to achieve their designated task. The general strategy is that the platforms acquire and maintain target tracks and the group level sensor management acquires this information from the platforms and analyzes it. Based on the assessment of the broader view of the situation provided by the platform data, the group-level SICH may generate new sensing directives for the platforms. These directives lead to the creation of group-level task holons. We will consider three types of task holons:

1. **Search/Track** - Search/track holons are generated in response to two main types of situations.
   (a) When particular regions may, on the basis of the situation analysis, stand out as likely sources of new targets. In this situation, search/track holons are created to negotiate initiation (through the communications holon) of an appropriate search onboard one of the platforms.
   (b) When cue or handoff events between platforms needs to be facilitated. In this instance, the group-level SICH generates a search/track holon that hands-off (search) or cues (track) a target track onboard a particular platform.
2. **Monitoring** – Monitoring holons are created to manage the communications required to secure track information from the platforms for delivery to the group-level SICH. As each platform initiates a new search, acquires a new track, or changes its sensor

configurations, a message is sent to the group-level SICH notifying it. In response, the group-level SICH generates a monitor holon to obtain periodic updates on these tasks. This holon negotiates with the communications holon to send its update request to the platform. Monitor holons may also request an increase in a platform-level task priority, if the platform itself is not updating the task often enough. For example, a target track that is deemed high-priority at the group level may be at a lower priority at the platform level and must be increased in order to provide suitable track quality for the group-level sensing objectives.

3. **Configuration** – Configuration holons are generated to request a configuration change for a specific platform. The configuration specification will address a situation development, such as an impending cueing/handoff event. The configuration holon requests a configuration change from the platform, which in turn generates platform-level configuration holon(s) to ultimately change the sensor configuration. As an example, consider a target that falls within a platform sensing domain but is not tracked by that platform because the sensors are not configured to detect it. The group-level sensor management holon may be in a position to recognize this fact (due to detection by another platform, for example) and can issue a configuration holon to the platform. The group-level configuration holon specifies only where the platform should be looking to detect the target. The platform, on the basis of this information, computes the best sensor configuration that will detect the new target while maintaining the tracks it is currently tracking. Based on this computation, a platform-level configuration holon will be created to implement the configuration change(s).

Any activities undertaken by the platforms such as track initiation, target detection and configuration changes, are related to the group level. Thus, there is a constant exchange between the group and the platforms. The group specifies directives and the platforms notify the group about implementation of those directives. In this manner, the group is able to follow the actions of the platforms while choosing which information (*e.g.*, tracks) it would like to have updated and at what frequency.

6.2.1 Task holon creation

Figure 21 illustrates the process of creating task holons at the group level. Initially, the group level awaits messages from the platforms indicating the initiation of search or track tasks and generates monitoring holons in response. The monitoring holons exist until the platform terminates its search/track task. In cases where the group deems the platform task performance insufficient, a configuration holon may be created to request an adjustment to the search or track task.

When track domain transitions are notified by a platform, the group facilitates cueing or handoff of the track between platforms. If the track can be handed-off to another platform, the group-level SICH creates a search holon. If a platform needs to initiate a search to acquire the track, then a group-level search holon is created. Finally, if the group-level SICH determines that the platform requires a configuration adjustment in order to acquire the track, then a configuration holon is created to suggest a configuration change.

Each holon communicates a sensing directive to the platform. Once an action is taken at the platform, a platform-level message holon is created to inform the group and an appropriate monitoring holon is created.

6.2.2 Determining priority and desired performance

The base priority P_0 and desired performance $Q_{desired}$ of monitoring holons are determined by the task they are created to monitor. For holons monitoring tracking tasks, these attributes may be based on the threat level of the track relatively to the group, *i.e.*, Z_g. Thus, we can write

$$P_0 = k_{g_1} \cdot Z_g + U \tag{40}$$

$$Q_{desired} = Q_0 - k_{g_2} \cdot Z_g + k_{g_3} \cdot L_c + U \tag{41}$$

Here, the constant k_{g_1} relates the priority level directly to the group threat level. The performance measure for monitoring holons $Q_{monitor}$ is a time to acquire data updates, a quantity to be minimized. The above equation is defined by a base update time Q_0 that is modified on the basis of the target threat-level and communication load. Allowance is made for human (user) input through U. Similarly, the constants k_{g_2} and k_{g_3} relate the desired track quality directly to the threat (to the group) and group communications resource load, L_c.

For holons monitoring search tasks, a standard priority level $P_{standard}$ is used and the base priority level P_0 is based on the expected rate R_T of target arrivals in the search sector. The desired quality attribute is also modified from $Q_{standard}$, on the basis of the expected target rate and also platform communication load, L_c. These can be written as

$$P_0 = P_{standard} + k_{g_4} \cdot R_T + U \tag{42}$$

$$Q_{desired} = Q_{standard} + k_{g_5} \cdot R_T - k_{g_6} \cdot L_c \tag{43}$$

Configuration holons are high priority tasks and are created with a constant and very high priority level in order to be quickly implemented. The desired quality is associated with how many sensor control periods the configuration takes to be implemented. It can also be made standard. Thus, for configuration holons, these are given by

$$P_0 = k_{g_9} \tag{44}$$

$$Q_{desired} = k_{g_{10}} \tag{45}$$

Search and tracking holons involved in cueing and/or handoff coordination have their base priority and desired performance specification determined by the threat to the group Z_g

$$P_0 = k_{g_1} \cdot Z_g + U \tag{46}$$

$$Q_{desired} = k_{g_2} \cdot Z_g - k_{g_3} \cdot L_p + U \tag{47}$$

Note here that the platform sensor load L_p and not the group communications resource load L_c moderate the desired performance.

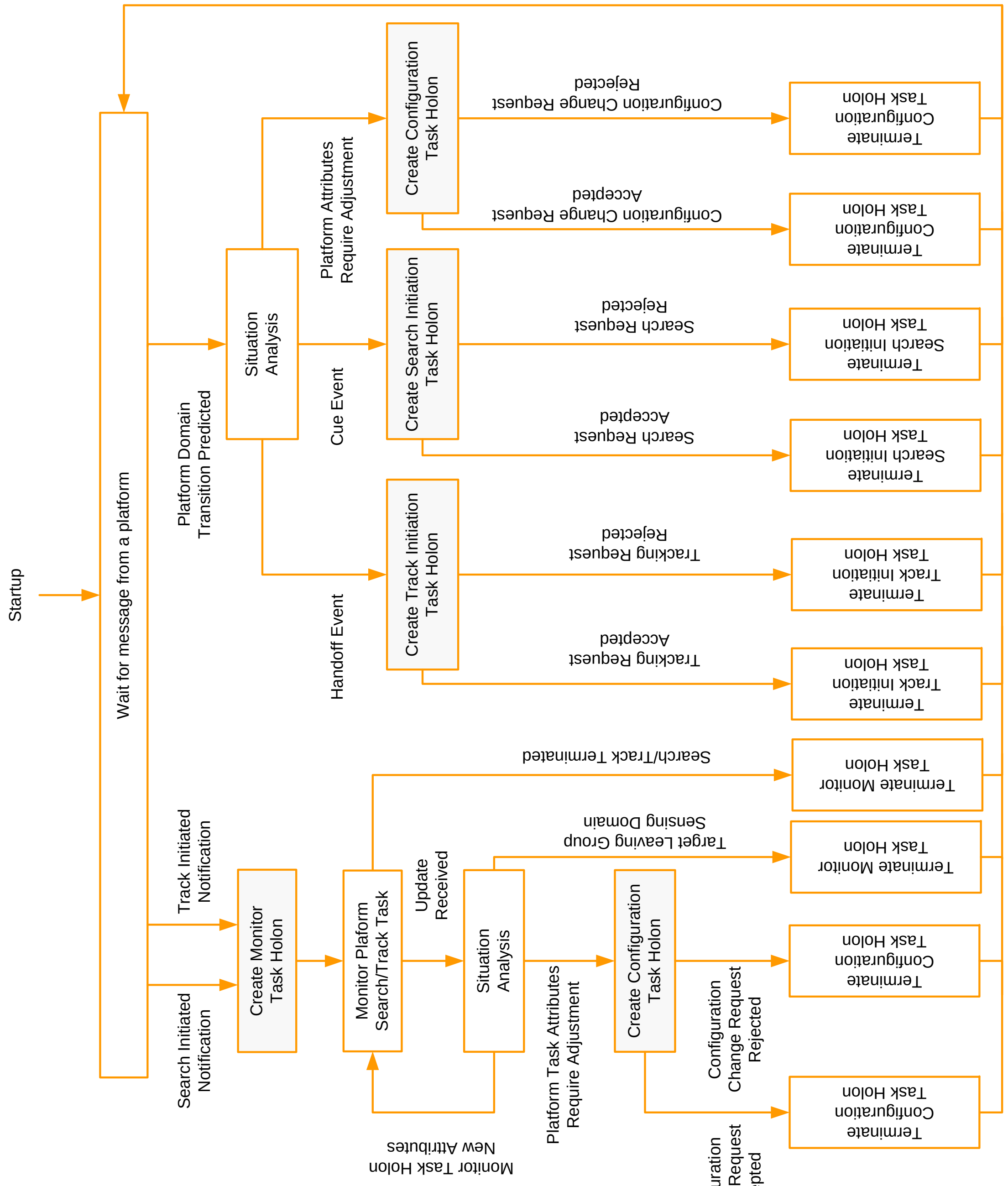

Startup
Wait for message from a platform
Platform Domain Transition Predicted
Situation Analysis
Platform Attributes Require Adjustment
Cue Event
Handoff Event
Create Configuration Task Holon
Create Search Initiation Task Holon
Create Track Initiation Task Holon
Configuration Change Request Rejected
Configuration Change Request Accepted
Search Request Rejected
Search Request Accepted
Tracking Request Rejected
Tracking Request Accepted
Terminate Configuration Task Holon
Terminate Configuration Task Holon
Terminate Search Initiation Task Holon
Terminate Search Initiation Task Holon
Terminate Track Initiation Task Holon
Terminate Track Initiation Task Holon
Track Initiated Notification
Search Initiated Notification
Create Monitor Task Holon
Monitor Platform Search/Track Task
Update Received
Situation Analysis
Monitor Task Holon New Attributes
Platform Task Attributes Require Adjustment
Create Configuration Task Holon
Configuration Change Request Rejected
Configuration Change Request Accepted
Search/Track Terminated
Target Leaving Group Sensing Domain
Terminate Monitor Task Holon
Terminate Monitor Task Holon
Terminate Configuration Task Holon
Terminate Configuration Task Holon

Figure 21: Group-level holon management logic

6.2.3 Task allocation

Unlike the platform level, task holons at the group level are limited in their communication with the resources. As a result, the task negotiation does not resolve task allocation problems but rather manages the allocation of tasks to platforms. This means that task allocation must be determined before group-level task holons are created. An aid to this decision making process is the rating of platform suitability for tasks on the basis of known platform status information.

Both search and tracking tasks specify a geographic region to be sensed and require that at least one sensor be in a suitable position to perform the task. From the group perspective, it is not important which sensor will be used to perform the task but which platform to allocate the task to. A simple means for deciding amongst platforms is to determine which platform is closest to the geographic region specified by the task. This approach does not consider sensor blind zones or mode configurations that may be important when platform sensing-domains overlap. Although ignored in this project, these considerations could be used, in future projects, to improve the task allocation process. For our purpose, it will be sufficient to base task allocation on the proximity of the platform to the task geographic region as well as platform loading.

The task allocation process works as follows: for tracking tasks, the relative threat level $Z_P(p)$ to each platform p is determined. This is based on the Time to Closest Point of Approach (TCPA) and the Closest Point of Approach (CPA) as described earlier. The platform with the highest threat level would be considered as the most likely candidate for task allocation. Once the threats are computed, the remaining capacity of each platform $RC_P(p)$ is taken into consideration and a task-fitness measure $F_P(p)$ for each platform is computed according to

$$F_P(p) = \kappa_1 \cdot Z_P(p) + \kappa_2 \cdot RC_P(p) \tag{48}$$

where, the constants κ_1 and κ_2 are chosen to balance the importance of the threat over the remaining sensing capacity. Once $F_P(p)$ is computed for all platforms, the task is allocated to the platform with the largest $F_P(p)$.

The search task allocation process is similar to the track task. Search tasks allocated from the group level are associated with cueing/handoff events and are therefore associated with a target track. The threat level of this track is used in the fitness calculation above and the remainder of the process is identical.

6.3 Group-level search/track task holons

The search and track task holons utilize the communications holon to request that search or track tasks be implemented at a platform level. The holon contains a priority attribute and details of the search/track, including a desired performance parameter, which it sends to the platform. The platform makes a decision as to how to fulfill the request (or reject it) and sends an update message back to the group. This message includes the task attributes.

In other words, the creation of a task holon at the platform level triggers the creation of an associated message holon that informs the group-level SICH of the particulars (attributes) of the platform response.

Both search and track tasks may be requested. Once the initial directive is sent to the platform, the group-level search or track holon is terminated. However, once an action is taken by the platform, a group-level monitoring holon is created to secure task updates. The desired performance measure $Q_{desired}$ is transmitted to the platform by the search or track holon, as well as details regarding task completion. For tracking tasks, this would include the track kinematical data, while search tasks would specify which region to search. Although it is up to the platform to assign a task priority to search and track tasks, the group-level search or track holon can also provide a (group-level) threat evaluation that the platform may use in determining platform-level task priority.

6.4 Handling cueing/handoff

If a target is predicted to cross the coverage area of more than one platform, the group-level SICH will recognize this as an upcoming cueing/handoff event. Only cueing/handoff events amongst platforms are dealt with at the group level. Cueing/handoff amongst individual sensors is an issue for the platform-level sensor management.

If a target is passing from the coverage area of one platform to another, it may be necessary to cue the receiving platform in advance of the target arrival (see Figures 15 and 16). The general methodology is to handoff a track to a second platform, if the time between sensing domains is small enough to maintain sufficient track quality. Otherwise, a high-priority search (*i.e.*, cueing) is required from the receiving platform in order to reacquire the target track.

In the case of cueing operations (Figure 15), the receiving platform is cued and a search is initiated at the platform level on the basis of the likely location of the incoming target. This is handled by the creation of a group-level search holon issuing a search directive (search holon) to the receiving platform. The receiving platform-level SICH creates, in response, a search holon and a communications holon to inform the group-level search holon that the search has been initiated (or has failed to initiate). Once the target is detected by the receiving platform, a platform-level track holon is created and the platform informs the group that the track has been initiated. If the cued platform requires an adjustment to its current sensor modes, the platform-level SICH will issue an appropriate platform configuration holon. If the platform is not capable of tracking the target, it will inform the group-level SICH.

Track handoff between platforms (Figure 16) is initiated when the group-level situation analysis detects a potential handoff event. In this case, the group-level SICH creates a group-level tracking holon, which communicates the tracking data, threat level and desired performance measure to the receiving platform-level SICH. In response, the receiving platform-level SICH creates a tracking holon to continue the track and a communications holon to inform the group-level tracking holon that the track has been initiated (or has failed to initiate).

Cueing/handoff tasks coordinated by the group-level sensor management must take into account the communications delay that is inherent in bandwidth-limited communications. This is a function of communications load and it will result in a longer delay as the number of tracked targets grows. If the communications delay is long enough, potential track handoff tasks may be infeasible and track cueing may be employed instead.

6.5 Group-level advanced functionalities

Handling of merging/irresolvable tracks and sensor emissions are advanced functions that might be added to the group-level SICH. These functions are briefly presented hereafter.

6.5.1 Handling merging/irresolvable tracks

The vantage point of individual platforms makes the task of resolving individual targets in the environment difficult or impossible when targets are closely spaced, fall along approximately the same bearing (relative to an individual platform), or have tracks that are merging or crossing. Generally, the problem of associating tracks with targets is a fusion process and is not addressed in this work; however, upon recognition of an impending association difficulty, the sensor management system can take steps to provide additional data to the fusion process.

In the case of closely spaced targets, the tracking resolution can be increased by specifying a more appropriate sensor mode, or by increasing the frequency of track updates, *i.e.*, request higher quality track (see [16] for details). In the first case, the group-level SICH could issue a coverage[5] holon to improve the sensing performance of a particular platform. In the second case, the original monitoring holons could negotiate a higher quality-of-track specification. This same strategy can be employed when the tracks are merging or crossing. In the case where target tracks are not resolvable from a single platform, sensor readings from a remote location such as a different platform provide the only means of solving the association problem. In this case, the group-level SICH may issue a tracking holon to provide short-term complementary track updates from a different perspective.

In most cases, tracks will only be irresolvable for a short period of time and track predictions will enable the track differentiation just before and just after the irresolvable period. This reduces the track quality during the irresolvable period. If another platform has a sensor already configured for tracking in the same area, then track updates from that platform may allow tracking quality to be maintained throughout this period. If this is not the case, platform sensors may need to be reconfigured (*i.e.*, configuration holon issued) or repositioned (*i.e.*, navigation holon issued). This requires that the event be predicted early enough so that the sensor reconfiguration can be completed. In many situations, it will be necessary to accept the temporary degradation in track quality.

[5] Not considered in the current design.

6.5.2 Electromagnetic emissions

Group-level tracking holons can also be issued with an emission attribute that specifies a limit on emissions. The holon must then negotiate with the platforms for access to the appropriate type of sensor to complete the task. If no platform can be found, the tracking holon reports to the group-level SICH, which in turn can issue a configuration holon to make the necessary adjustments in platform sensor configuration.

6.6 Group-level monitoring task holons

Monitoring task holons are created in response to a message received by the group-level sensor management holarchy from a platform holarchy regarding the initiation of a track, a search or a configuration change. At the group level, a monitor holon is created for each platform-level task. The group-level SICH, on the basis of the relative importance of each platform-level task, assesses the priority of the monitoring holon and provides it with the following operation attributes:

– Desired monitoring task performance goal (update frequency of group-level data)
– Specifications of the platform-level task attributes
– Task priority limits (for self-adjusting priority)

The first attribute above is used by the monitoring holon to modify its priority level, when it is unable to secure task updates in the specified period. The period specification relates to information update at the group level on a platform-level task: the shorter the update period, the more up to date the information.

Specifications of the platform-level task attributes allow the group-level monitoring holon to determine when platform-level task attributes need to be modified in order to meet the group-level reporting needs. For example, if the group-level sensor management looks for very current knowledge about a particular track, the platform-level sensor management must update that track frequently. If this is not the case, the group-level monitoring holon may send a message to increase the update rate.

In addition to task data, the attributes of the associated platform-level task holon are also returned from the platform at each update. Since these may change from time to time, it is imperative that the monitoring holon be aware of these attributes. Moreover, when platform-level task holons are terminated or fail, this information must also be provided to the monitoring holon through the platform-level SICH.

The monitoring holon assesses its own performance relative to the performance specification provided by the group-level SICH. In this case, the performance specification $Q_{desired}$ is simply the desired interval during which task updates are acquired from the communications holon. Correspondingly, the task holon performance measure Q_{task} is simply the time interval following the reception of the last task update (*i.e.*, data delivery). This delivery time includes the time for the platform to respond, not just the time it takes to secure a communications holon transmission.

6.6.1 Monitoring quality

The main responsibility of the monitoring holon is to keep the group-level SICH informed about the tasks being executed at the platforms level. Thus, the main indicator of how well this holon is performing is to measure the time since the last task update. This measure will be taken as the on-line quality assessment, $Q_{monitor}$. Monitoring holons are created with a $Q_{desired}$ attribute, representing a desired time between updates. The monitoring holon will attempt to keep $Q_{monitor}$ less than $Q_{desired}$ and is considered as having failed when $Q_{monitor}$ grows larger than a specified time limit $Q_{\max}$.

6.6.2 Priority level adjustment

Priority adjustment is achieved in the manner described previously. Monitoring task holons are created with a desired task quality $Q_{desired}$ that the holon seeks to achieve. One means of gaining access to resources is through increasing its priority level. The group-level SICH creates monitoring task holons with attributes that limit the increase in priority level in response to $Q_{monitor}$ values growing smaller than $Q_{desired}$. These attributes are the following:

- Priority base value: P_0
- Priority increase rate: m_p
- Priority maximum: $P_{\max}$

If the monitoring task quality $Q_{monitor}$ is larger than $Q_{desired}$, then task priority $P_{monitor}$ will be set at the base value P_0. If $Q_{monitor}$ grows smaller than $Q_{desired}$, the increase rate m_p specifies the increase per time unit according to the following

$$P_{monitor} = P_0 + m_p \cdot (t - t_0) \tag{49}$$

where t is the time index and t_0 is the most recent time when $Q_{monitor}$ became less than $Q_{desired}$ (*i.e.*, when $Q_{desired}$ is achieved, reset $P_{monitor}$). Finally, $P_{monitor}$ is not allowed to grow beyond $P_{\max}$.

6.6.3 Task failure

In the event that the monitoring holon cannot acquire service for a long period of time, or that service attempts repeatedly fail, the monitoring task holon will report a task failure to the group-level SICH. This is controlled by the group-level SICH issued holon attribute $Q_{\min}$. The condition for monitoring failure is then trivial: when $Q_{monitor}$ becomes smaller than $Q_{\min}$, the task is considered to have failed.

6.7 Group-level configuration task holons

The configuration holon at the group level is very similar to the search or track holons in that it is created to send a message and is then terminated. Configuration changes derived by the group-level SICH, through its situation analysis algorithms, are used to create a group-level configuration holon. This holon is generated with a priority attribute that is generally high, so that it will be implemented at the platform quickly. The group-level

SICH is notified by the platform when an actual configuration change is implemented. This provides sufficient feedback to control the system.

6.8 Group-level communications resource holon

The communications resource holon at the group level functions in the same manner as at the platform level. The only difference here is that the communications holon is utilized by the group-level task holons directly or by the group-level SICH through the issuance of message holons.

7 An overview of holarchy operations

This chapter summarizes the operation of the proposed holonic control system for sensor management in military settings. It should be noted that a number of aspects of sensor management have been simplified or not fully addressed in this design. This chapter discusses the role of sensor management in acquiring and maintaining target tracks across multiple platforms but avoids the more low-level situation analysis of target associations, track separation and sensor emission control. These issues are not addressed in this work but it is important to note that they can be addressed within the sensor management design proposed here and may be part of some future follow-on work.

In the sensor management design presented here, the platforms perform most of the sensing activities, including:

1. Detecting targets
2. Tracking targets
3. Modifying sensor configurations on the basis of the analysis of the local situation

The group level sensor management role is to:

1. Selectively acquire data from the platforms
2. Modify platform sensing operations on the basis of the group-level situation analysis

The sensors themselves are responsible for managing the various tasks specified by the platform.

7.1 Initialization

At the beginning of the operation, each platform will first be required to configure its sensors for searching. This proceeds with the platform-level SICHs, creating platform-level configuration holons that control their sensor resources. Once the sensors have implemented the configuration adjustments, and reported to the platform-level SICH, the platform-level SICH generates a message holon for the purpose of informing the group-level sensor management of the current platform sensor configuration.

In parallel with the sensor initiation, the platform-level SICH creates a number of platform-level search holons in order to detect targets in the platform sensing domains. Aboard each platform, the platform-level search holons will have to compete with the platform-level configuration holon for access to the sensors.

While the configuration adjustment is proceeding, the platform-level search holons are negotiating with the sensors for service. Since the configuration holon is of higher priority than search holons, the sensor configuration will be changing as the search tasks are conducted. Once the configuration adjustment is complete, only platform-level search holons will remain active until a target is detected.

7.2 Target tracking

When a platform-level search holon reports a target detection, the platform-level SICH issues a platform-level tracking holon that maintains the target track. Once the track is established, the platform-level SICH creates a message holon in order to notify the group-level SICH. The group-level SICH, in response, creates a group-level monitoring holon that intermittently updates the group-level track database. Thus, as new tracks are formed at the platform level, the group level is notified and monitors the tracking data. There are two levels of situation analysis that are implemented, group-level and platform-level.

1. The platform-level situation analysis is used to balance resource usage amongst the tracking tasks. For example, targets that are deemed to be threatening to the platform are to be tracked more closely than non-threatening targets.
2. At the group level, the communications limitations changes the nature of the sensor management. The group cannot control target tracking onboard the platforms directly, so it plays more a supervisory role. Situation analysis provides a means for balancing the limited communications resource with the delay in the group-level tracking data. For example, the group-level interest in tracking data would be to keep up to date on targets that are deemed to be threatening to the protected asset, while relaxing the data update rate on targets that are non-threatening. These non-threatening targets may be tracked closely by the platform, but the group-level sensor management does not need to use the valuable communications resource to keep the group-level track database updated.

In some instances, the group-level sensor management may look for a better tracking than the one the platform is currently providing. In this case, the group-level SICH issues a communications holon containing a new desired track performance specification. The platform-level SICH will attempt to modify the attributes of the appropriate platform-level tracking holon in order to meet this specification, or possibly change sensor modes in response. This course of action is generally only needed when the group-level and the platform-level situation analysis processes differ in their assessment of the target priority.

7.3 Cueing/Handoff events

There are two main types of cueing/handoff events that can occur:

1. Those due to targets transiting between sensor domains onboard a single platform (Figure 22).
2. Those due to targets transiting from one platform sensing domain to another (Figure 23).

In both cases, cueing and handoff can occur, depending on how long the target will be out of sensing range.

Within a single platform, the platform-level tracking holon responsible for a transiting target usually can handle the handoff by simply acquiring service on a different sensor as the situation warrants. Between platforms, cueing/handoff requires coordination from the group-level sensor management. In this case, it is up to the group-level situation analysis

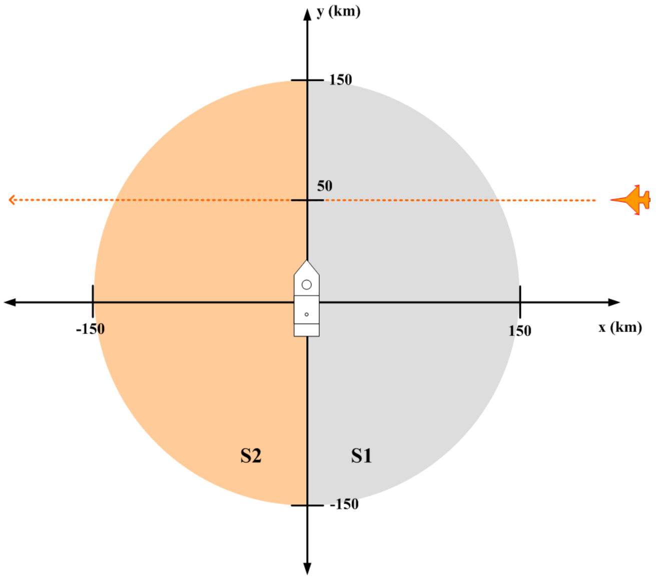


Figure 22: Single-platform cueing/hand-off

to recognize the impending event and cue for a search or handoff the track to the receiving platform. If the track is lost or degraded, due to the long delay before the target reaches the receiving platform, the group-level SICH will issue a group-level search holon that will delegate a search task to the receiving platform. The platform-level SICH will in turn create a platform-level search holon that will attempt to reacquire the target track.

7.4 Group holon located onboard a platform

The group-level holon, as described here, is an abstraction of a military command centre, and as such, may operate from any location. In fact, it is common for the command centre to be based onboard one of the platforms under its command. In this case, the platform that supports the group holon is of more inherent value as it performs not only as a platform resource but also acts as host to the group command. As a result, the platform supporting the group would, in many situations, be considered a high value unit (HVU), as it should have a greater level of protection compared to other platforms that comprise the group. The design presented here may be used in these situations although some modification may be necessary.

The main difference between the system described here and the one in which the group holon resides onboard a platform is the group-platform communication issues. Unlike the restrictions addressed so far, group communications with the hosting platform need not be (significantly) bandwidth restricted. This may alter some of the sensor management strategy, since the group will be better able to communicate with the hosting platform. Two differing strategies for addressing this modification can be employed:

1. Keep the same design
2. Combine group and platform sensor management

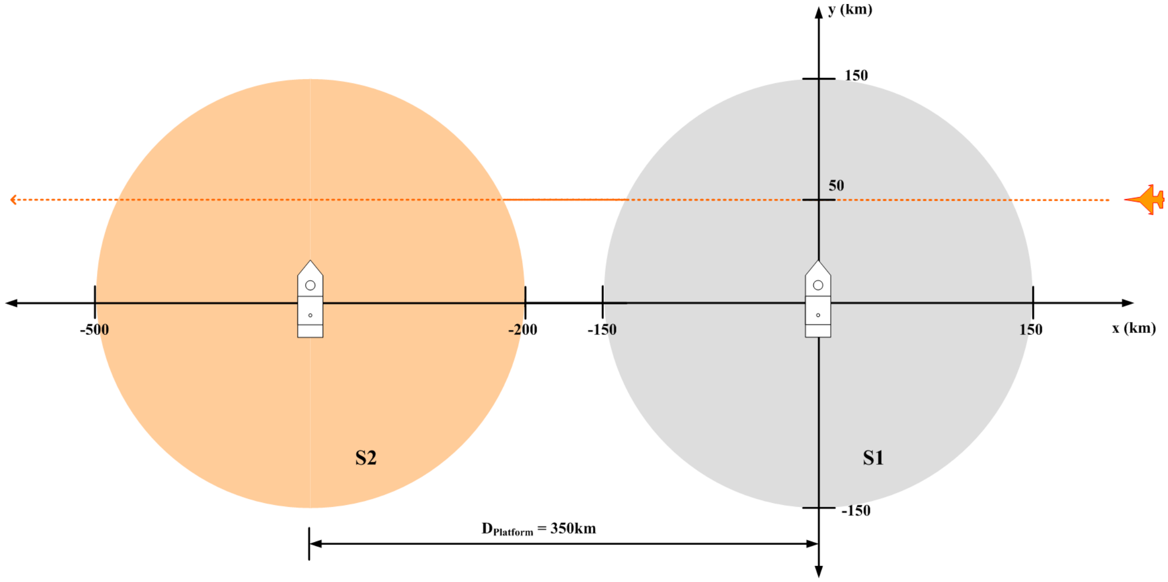


Figure 23: Inter-platform cueing/hand-off

The first strategy ignores the physical location of the group and requires no change from the design presented in this document. The advantage of this approach is that the command centre, *i.e.*, the group-level sensor management holarchy, is portable. With this strategy, the group-level sensor management holarchy always assumes bandwidth limited communication with the platforms, so it does not matter where it is located as long as some form of communication is available. For example, if the the group-level sensor management holarchy is located onboard a platform that sustains heavy damage, it may be transferred to a more secure platform and continue operating. The main problem with this approach is that the overall design is not optimized for the collocation of the group sensor management and a platform sensor management, *i.e.*, the hosting platform is treated just as another resource.

The second strategy combines the group and platform sensor management onboard the platform that is hosting the group-level sensor management holarchy. This approach requires that the group-level SICH issue both group-level task holons and platform-level task holons. The platform-level task holons would interact with the platform resources while the group-level task holons would coordinate sensing activities across all of the remaining platforms. The benefit of this approach is that the group-level sensor management holarchy incorporates the high bandwidth connection to the host platform in its task planning. This may alter the overall sensor management strategy. For example, the platform hosting the group-level sensor management holarchy may be used as the first choice for (group-directed) sensing tasks, since it is the easiest one to communicate with.

8 Conclusion

This report presented a conceptual design for sensor management in military settings using a holonic control approach. Three levels of sensor management were considered: sensor, platform, and group. Sensor-level sensor management involves the control and task scheduling for individual sensors. Platform-level sensor management involves the allocation of tasks to the sensors, control of sensor configurations and coordination of sensing activities between sensors. Platform-level sensor management is confined to individual platforms. Group-level sensor management coordinates sensing tasks between platforms and controls platform sensing configurations.

A sensor management strategy, which is based on perception of threat to protected assets, was described. Threat evaluation of individual tracks is used as the basis upon which the sensor management system allocates sensor resources. Threatening targets are assigned more sensor time and are therefore tracked more closely than non-threatening targets. This approach saves resource usage and allows the system to focus resources on the most important targets.

The design presented here did not specifically address all of the issues related to sensor management in military settings. In particular, issues such as emission control, platform navigation and various aspects of situation analysis were either overlooked or simplified in order to devise a workable sensor management design. The design presented focused on problems related to target search and tracking, as well as techniques for handling track cueing and handoff between sensors onboard a single platform and between platforms.

A number of sensor management aspects were not investigated in the design presented here due to time and budget limitations. Some of these aspects, which may be pursued in any related future work, are described below:

Sensor diversity – The current design utilizes only ESA-type sensors but is not limited to the use of these sensors only. Conventional Scanning Radar (CSR) and Electronic Support Measures (ESM) type sensors (see Annex A) can be incorporated with some additional design work.

Secondary platforms – The design may be extended to secondary platforms such as UAVs and helicopters dispatched from the main platforms described in this document. UAVs and other aircraft that may be dispatched by a platform result in another level of sensor management. These secondary platforms typically report to the platform from which they were dispatched and, as such, represent another level of sensor management. To this end, the dispatching platform acts as a coordinator for the secondary platform(s), much in the way the group level acts as a coordinator for the platforms. Communication issues are important in this situation as well.

Optimization – The design presented here provides a number of heuristics for sensor management using a holonic architecture. Specifically, the equations presented in this document for setting task priorities, task holon desired performance and holon creation, all rely on the proper selection of scaling constants. These constants must be selected to achieve a reasonable performance of the sensor management system, but there is no guarantee of optimal performance. Furthermore, should the design become complex, the selection of these constants will become a laborious if not an impossible task. A proper design methodology must be developed to extend this work beyond the scale that is presented here. These methodologies may involve the application of control techniques to the holonic architecture developed here, thus taking the next logical step towards a more in-depth sensor management design.

Although the motivating application of the presented holonic design is sensor management, its properties and generic implementation makes it potentially exploitable in several others domains requiring planning with resources, such as force-level combat power management.

References

[1] Benaskeur, A. and Irandoust, H. (2007), Sensor management for tactical surveillance operations, Technical Report DRDC Valcartier TM 2006-767. UNCLASSIFIED.

[2] Benaskeur, A. and Irandoust, H. (2008), Holonic Approach for Control and Coordination of Distributed Sensors, Technical Report DRDC Valcartier TR 2008-015. UNCLASSIFIED.

[3] Adrian, R. A. (1993), Sensor Management, *Proceedings of the 1993 AIAA/IEEE Digital Avionics System Conference, Ft. Worth, TX*, pp. 32–37.

[4] Manyika, J. and Durrant-Whyte, H. (1994), Data Fusion and Sensor Management : A Decentralized Information-Theoretic Approach, New York, NY: Ellis Horwood.

[5] Blackman, S. and Popoli, R. (1999), Design and Analysis of Modern Tracking Systems, Norwod, MA, USA: Artech House.

[6] Ng, G. W. and Ng, K. H. (2000), Sensor management - what, why and how, *Information Fusion*, 1, 67–75.

[7] Steinberg, A., Bowman, C., and White, F (2001), in Handbook of Multisensor Data Fusion, The Electrical Engineering and Applied Signal Processing Series, Ch. Revisions to the JDL data fusion model, Quebec City: CRC Press.

[8] Koestler, A. (1967), The Ghost in the Machine.

[9] Bussmann, S. (1998), An agent-oriented architecture for holonic manufacturing control, In *First Open Workshop on Intelligent Manufacturing Systems Europe*, pp. 1–12.

[10] McFarlane, D.C. and Bussmann, S. (2000), Developments in holonic production planning and control, *Production Planning and Control*, 11(6), 522–536.

[11] Wintenby, J. (2003), Resource Allocation in Airborne Surveillance Radar, Ph.D. thesis, Department of Signals and Systems, School of Electrical Engineering, Chalmers University of Technology, Götenborg, Sweden. Technical Report No. 463.

[12] Wintenby, J. and Krsihamurthy, V. (2004), Hierarchical Resource Management in Adaptive Airborne Surveillance Radars, In *Proceedings of the Seventh International Conference on Information Fusion*, pp. 791–798, Stockholm, Sweden.

[13] Lopez, J., Herrero, J., Rodriguez, F., and Corredera, J. (2003), Cooperative Management of a Net of Intelligent Surveillance Agent Sensors, *International Journal of Intelligent Systems*, 18, 279–307.

[14] Rhéaume, F. and Benaskeur, A. (2006), Impact of the Tactical Picture Quality on the Fire Control Radar Search-Lock-on Time, Technical Report DRDC Valcartier TR 2003-284. UNCLASSIFIED.

[15] Rhéaume, F. & A. Benaskeur (2007), Fire Control-Based Adaptation in Data Fusion Applications, In *Proceedings of Fusion 2007*, Quebec City, Canada.

[16] Benaskeur, A. and Rhéaume, F. (2007), Adaptive data fusion and sensor management for military applications, *Aerospace Science and Technology*, 11(4), 327–338.

[17] Sunday, D (2001), Distance between Lines and Segments with their Closest Point of Approach. http://softsurfer.com/Archive/algorithm_0106/algorithm_0106.htm.

This page intentionally left blank.

Annex A: The sensors

This annex presents the sensors involved in the scenario and that could be controlled with the holonic sensor management application. These include Electronic Support Measures (ESM), Conventional Scanning Radar (CSR) and Electronically Scanned Array (ESA). As a simplification, sensors will only detect the presence of enemy or neutral objects. All friendly platforms will be transparent to the sensors and the corresponding position data will not be reported back to a search task.

A.1 Electronic support measures

Electronic Support Measures (ESM) is a passive device for detecting, intercepting, identifying and locating sources of radiated electromagnetic energy [1]. Figure A.1 illustrates its parameters. The ESM output is a list of the measured bearing β for each detected contact relative to the platform position in the scanned coverage sector.

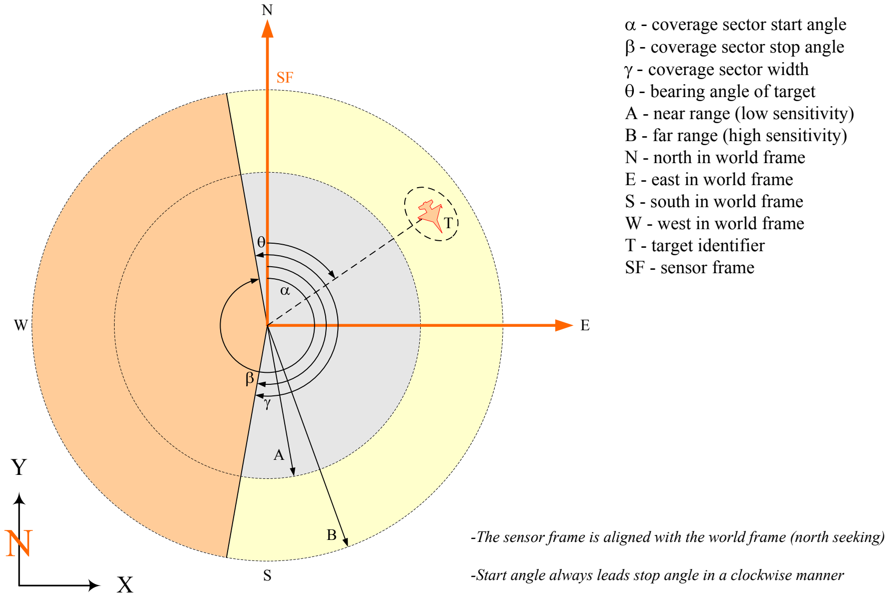
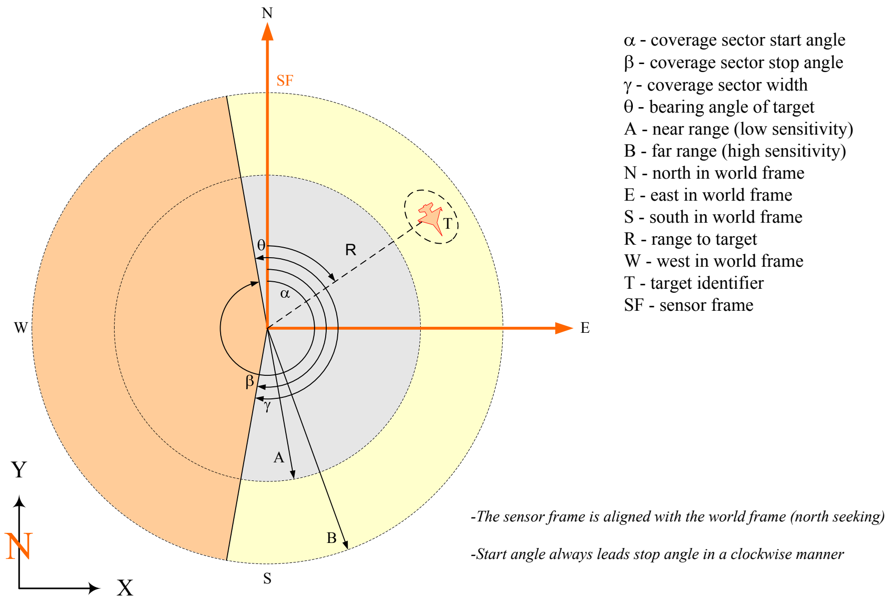

N
SF
W
E
S
θ
α
β
γ
A
B
T
Y
N
X
α - coverage sector start angle
β - coverage sector stop angle
γ - coverage sector width
θ - bearing angle of target
A - near range (low sensitivity)
B - far range (high sensitivity)
N - north in world frame
E - east in world frame
S - south in world frame
W - west in world frame
T - target identifier
SF - sensor frame
-The sensor frame is aligned with the world frame (north seeking)
-Start angle always leads stop angle in a clockwise manner

Figure A.1: ESM parameters

A.2 Conventional scanning radar

Conventional Scanning Radars (CSRs) are mechanically rotating active sensor usually employed for medium to long range surveillance. CSR outputs a list of the range and bearing of each contact relative to the platform position in the scanned coverage sector. Data are provided at the end of a sweep. For this work, we assume two different scan rates for the CSR: 60°$/s$ (Low) and 120°$/s$ (High). Moreover, CSR resolution is assumed to be 1.5° in bearing. Figure A.2 presents some parameters of the CSR.



N
SF
E
S
W
R
T
θ
α
β
γ
A
B
Y
N
X
α - coverage sector start angle
β - coverage sector stop angle
γ - coverage sector width
θ - bearing angle of target
A - near range (low sensitivity)
B - far range (high sensitivity)
N - north in world frame
E - east in world frame
S - south in world frame
R - range to target
W - west in world frame
T - target identifier
SF - sensor frame
-The sensor frame is aligned with the world frame (north seeking)
-Start angle always leads stop angle in a clockwise manner

Figure A.2: CSR parameters

Table A.1 gives examples of maximum range and resolution for different power modes and for a small, medium and large platform.

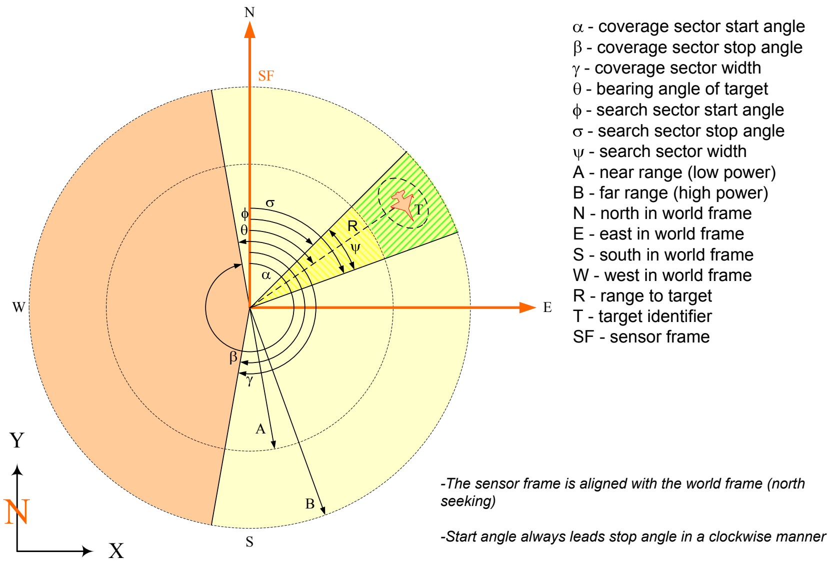
Table A.1: CSR Range and (Resolution) Matrix

Platform	**Power**	**Scan Rate**	
		Low	**High**
Small	**Low** **High**	120 km (40 m) 180 km (90 m)	60 km(40 m) 120 km (90 m)
Medium	**Low** **High**	130 km (50 m) 210 km (95 m)	60 km(50 m) 140 km (95 m)
Large	**Low** **High**	150 km (60 m) 270 km (100 m)	100 km(60 m) 180 km (100 m)

A.3 Electronically scanned array

Electronically Scanned Array (ESA) radars are active sensors that can almost instantly direct their beam toward a specific area. Figure A.3 presents their parameters. ESAs output a list of the range and bearing of each detected contact relative to the platform position in the scanned search sector(s). In this work, ESA are assumed to have two different scan rates, low (100 °/s) and high (180 °/s), and two different power modes, low and high. The bearing resolution is 2°.

ESAs are controlled through combinations of their scan sector, scan rate and power. By


N
SF
E
S
W
σ
ϕ
θ
α
β
γ
R
ψ
T
A
B
Y
N
X
α - coverage sector start angle
β - coverage sector stop angle
γ - coverage sector width
θ - bearing angle of target
ϕ - search sector start angle
σ - search sector stop angle
ψ - search sector width
A - near range (low power)
B - far range (high power)
N - north in world frame
E - east in world frame
S - south in world frame
W - west in world frame
R - range to target
T - target identifier
SF - sensor frame

-The sensor frame is aligned with the world frame (north seeking)

-Start angle always leads stop angle in a clockwise manner

Figure A.3: ESA parameters

manipulating the values of these, the sensor can change the region that is scanned and the quality of the readings returned by the scan. The scan sector is controlled by its start and stop angles. The start angle always leads the stop angle and both are between 0° and 360°.

A given scan sector may not have access to the whole 360°. Things like its position on a platform may make parts of the scan sector unusable. These are called the sensor's blind. The blind is defined similarly to the scan sector with start and stop angles ranging from 0° to 360° and the start angle always leading the stop angle.

The rate at which we sweep through a given scan sector determines how frequently the region is sampled. The slower the rate, the more samples will be taken and thus the region will be sampled more accurately. Small targets in the region will have a higher probability of being detected, but it will take longer to cover a given sector. The faster the rate, the less samples will be taken and thus the region will be sampled less accurately. This increases the likelihood that smaller targets will not be detected.

A.4 Probability of detection

Each target has a Radar Cross-Section (RCS), also known as x-section, and a radar emission signature. The radar x-section determines how well a particular target is visible to active sensing devices. The radar emission signature determines the amount of radar power that the target emits and, consequently, how easy it can be detected by a passive sensor. The radar emission signature has not been implemented for targets in this simulation (it has only been implemented for sensors on the friendly platforms). The target detection has been implemented such that the smaller a target radar cross section, and the farther away it is, then the less likely it will be detected. In addition to the targets size and range from sensors, the scan rate of a sensor also affects the probability that it will detect a particular

Table A.2: Probability of detection matrix

Range	**X-Section or Emission Signature Magnitude**		
	Small	**Medium**	**Large**
Near	1.00 (0.95)	1.00 (0.98)	1.00 (1.00)
Mid	0.40 (0.20)	0.90 (0.75)	1.00 (0.95)
Far	0.02 (0.01)	0.30 (0.15)	1.00 (0.90)

target. A higher radar scanning rate results in a lower probability of detecting targets but provides higher data update rates that are beneficial when tracking fast agile targets (*i.e.*, a fighter).

Table A.2 illustrates the dependence of probability of detection (POD) on target signature and distance from sensor. It shows the POD for both a slow scan rate and a fast scan rate. The POD at a fast scan rate is shown in brackets. The normalized range is the distance to the target relative to the maximum sensor range of current mode setting. Near means that the normalized range to the target, r_{PT}, is less than or equal to 40%($r_{PT} \leq 40\%$). Mid means that $40\% < r_{PT} \leq 75\%$ and far means that $75\% < r_{PT}$.

Annex B: Target Tracking Using Kalman Filter

A Kalman filter that is suitable for target tracking from a stationary observer is described below. The main objective of the filter is to estimate the position of a target using a range-bearing sensor (equivalent to radar). The velocity of the target is assumed to be unknown to the observer.

B.1 Models

The target motion model is assumed to be a constant velocity model. The Cartesian coordinate system is used to represent the workspace. The state vector of the target is defined as, $\boldsymbol{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^T$.

B.1.1 System Model

The evolution model (system model) of the target position and velocity can be described by the following equations.

$$x_{k+1} = x_k + \dot{x}_k h \tag{B.1}$$

$$\dot{x}_{k+1} = \dot{x}_k \tag{B.2}$$

$$y_{k+1} = y_k + \dot{y}_k h \tag{B.3}$$

$$\dot{y}_{k+1} = \dot{y}_k \tag{B.4}$$

Where, $h = t_{k+1} - t_k$. Thus, the state transition equation can be written as,

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{w}_k \tag{B.5}$$

Where,

$$\boldsymbol{A}_k = \begin{bmatrix} 1 & h & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.6}$$

and $\boldsymbol{w}_k$ is the process noise associated with the motion model and is assumed to have a zero mean Gaussian distribution.

B.1.2 Measurement model

A range-bearing sensor (measured in polar coordinates) is assumed to be used in measuring the position of the target. Thus, a measurement of the system can be represented as $\boldsymbol{z} = [r, \theta]^T$.

A nonlinear measurement model is used and the model can be represented by

$$r = \sqrt{x^2 + y^2} \tag{B.7}$$

$$\theta = \arctan(x, y) \tag{B.8}$$

In a general form, the above relationship can be represented by

$$\boldsymbol{z}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{v}_k \tag{B.9}$$

Where $\boldsymbol{H}_k$ is the observation matrix and $\boldsymbol{v}_k$ is the measurement noise and it is assumed to have a zero mean Gaussian distribution.

B.2 Extended Kalman filter

To initialize the Kalman filter the initial state vector and state covariance vector ($\boldsymbol{x}_0$ and $\boldsymbol{P}_0$) must be selected. The target position states in the state vector can be initialized with the initial measurement of the target. The target velocity vectors can be initialized to the most probable velocity of a particular type of a target (e.g. airplane) or using two first positions to estimate velocity and its variance. The covariance matrix can be initialized with the position variance of the initial measurement and the variance of the initial velocity estimate. The filtering process uses the regular prediction and update process.

B.2.1 Prediction stage

During this stage given the time step h for the next measurement, the next state vector, state covariance matrix and measurement vector is predicted by the following equations.

$$\hat{\boldsymbol{x}}_k = \boldsymbol{A}_{k-1} \boldsymbol{x}_{k-1} \tag{B.10}$$

$$\boldsymbol{P}_k = \boldsymbol{A}_k \boldsymbol{P}_{k-1} \boldsymbol{A}_k^T + \boldsymbol{Q}_{k-1} \tag{B.11}$$

$$\hat{\boldsymbol{z}}_k = \boldsymbol{H}_{k-1} \hat{\boldsymbol{x}}_k \tag{B.12}$$

Where, $\boldsymbol{Q}$ is the process noise covariance matrix. The position uncertainty is directly proportional to the velocity and to the time duration for the next measurement. The uncertainty in velocity is directly proportional to the time duration for the next measurement. Thus,

$$\boldsymbol{Q}_k = \begin{bmatrix} k_1 \dot{x} h & 0 & 0 & 0 \\ 0 & k_2 h & 0 & 0 \\ 0 & 0 & k_1 \dot{y} h & 0 \\ 0 & 0 & 0 & k_2 h \end{bmatrix} \tag{B.13}$$

Where, k_1 and k_2 are suitably chosen parameters.

B.2.2 Update stage

During the update stage a new measurement, $\boldsymbol{z}_k$, is taken and then the state vector and the state covariance matrix is updated using the following equations.

$$\boldsymbol{v}_k = \boldsymbol{z}_k - \hat{\boldsymbol{z}}_k \tag{B.14}$$

$$\boldsymbol{S}_k = (\nabla_x \boldsymbol{H}_{k-1}) \hat{\boldsymbol{P}}_k (\nabla_x \boldsymbol{H}_{k-1})^T + \boldsymbol{R}_k \tag{B.15}$$

$$\boldsymbol{W}_k = \hat{\boldsymbol{P}}_k (\nabla_x \boldsymbol{H}_{k-1})^T \boldsymbol{S}_k^{-1} \tag{B.16}$$

$$\boldsymbol{x}_k = \hat{\boldsymbol{x}}_k + \boldsymbol{W}_k \boldsymbol{v}_k \tag{B.17}$$

$$\boldsymbol{P}_k = \hat{\boldsymbol{P}}_k - \boldsymbol{W}_k \boldsymbol{S}_k \boldsymbol{W}_k^T \tag{B.18}$$

Where,

$$\boldsymbol{R}_k = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \tag{B.19}$$

is the measurement covariance matrix, while σ_r, σ_θ, are the standard deviations of range and bearing, respectively.

$$\nabla_x \boldsymbol{H}_k = \frac{\partial \boldsymbol{H}_k}{\partial x} \tag{B.20}$$

can be expanded to,

$$\nabla_x \boldsymbol{H}_k = \begin{bmatrix} \dfrac{x}{\sqrt{x^2+y^2}} & 0 & \dfrac{y}{\sqrt{x^2+y^2}} & 0 \\ \dfrac{-\,y}{x^2+y^2} & 0 & \dfrac{x}{x^2+y^2} & 0 \end{bmatrix} \tag{B.21}$$

This page intentionally left blank.

Annex C: Closest Point of Approach (CPA)

The following material is adopted from [17]. In the context of target tracking the dynamically changing points discussed below represent platforms (p) and targets (q).

The *Closest Point of Approach* refers to the positions at which two dynamically moving objects reach their closest possible distance. This is an important calculation for collision avoidance. In many cases of interest, the objects, referred to as *tracks*, are points moving in two fixed directions at fixed speeds. That means that the two points are moving along two lines in space. However, their closest distance is not the same as the closest distance between the lines since the distance between the points must be computed at the same moment in time. So, even in $2D$ with two lines that intersect, points moving along these lines may remain far apart. But if one of the tracks is stationary, then the CPA of another moving track is at the base of the perpendicular from the first track to the second's line of motion.

Consider two dynamically changing points whose positions at time t are $\boldsymbol{x}_p(t)$ and $\boldsymbol{x}_q(t)$. Let their positions at time $t = 0$ be $\boldsymbol{x}_p(0)$ and $\boldsymbol{x}_q(0)$ and let their velocity vectors per unit of time be $\dot{\boldsymbol{x}}_p$ and $\dot{\boldsymbol{x}}_q$. Then, the equations of motion for these two points are

$$\boldsymbol{x}_p(t) = \boldsymbol{x}_p(0) + t\dot{\boldsymbol{x}}_p \tag{C.1}$$

$$\boldsymbol{x}_q(t) = \boldsymbol{x}_q(0) + t\dot{\boldsymbol{x}}_q \tag{C.2}$$

which are the familiar parametric equations for the lines. However, the two equations are coupled by having a common parameter t. So, at time t the distance between them is

$$d(t) = |\boldsymbol{x}_p(t) - \boldsymbol{x}_q(t)| = |\boldsymbol{w}(t)| \tag{C.3}$$

where,

$$\boldsymbol{w}(t) = \boldsymbol{w}_0 + t(\dot{\boldsymbol{x}}_p - \dot{\boldsymbol{x}}_q) \tag{C.4}$$

with,

$$\boldsymbol{w}_0 = \boldsymbol{x}_p(0) - \boldsymbol{x}_q(0) \tag{C.5}$$

Now, $d(t)$ is a minimum when $D(t) = d(t)^2$ is a minimum, and we can compute:

$$D(t) = \boldsymbol{w}(t) \cdot \boldsymbol{w}(t) = (\dot{\boldsymbol{x}}_p - \dot{\boldsymbol{x}}_q) \cdot (\dot{\boldsymbol{x}}_p - \dot{\boldsymbol{x}}_q)t^2 + 2\boldsymbol{w}_0 \cdot (\dot{\boldsymbol{x}}_p - \dot{\boldsymbol{x}}_q)t + \boldsymbol{w}_0 \cdot \boldsymbol{w}_0 \tag{C.6}$$

which has a minimum when

$$0 = \frac{d}{dt}D(t) = 2t[(\dot{\boldsymbol{x}}_p - \dot{\boldsymbol{x}}_q) \cdot (\dot{\boldsymbol{x}}_p - \dot{\boldsymbol{x}}_q)] + 2\boldsymbol{w}_0 \cdot (\dot{\boldsymbol{x}}_p - \dot{\boldsymbol{x}}_q) \tag{C.7}$$

This in turn can be solved to get the time of CPA to be:

$$t_c = \frac{-\boldsymbol{w}_0 \cdot (\dot{\boldsymbol{x}}_p - \dot{\boldsymbol{x}}_q)}{|\dot{\boldsymbol{x}}_p - \dot{\boldsymbol{x}}_q|^2}$$

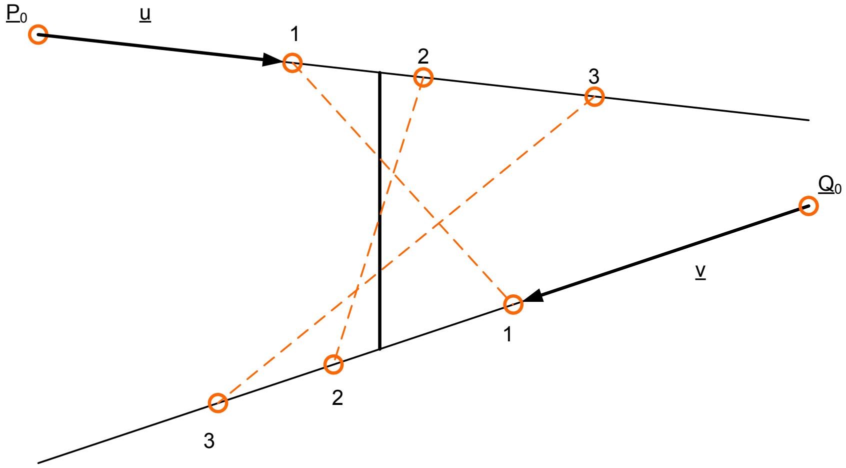


Figure C.1: Closest point of approach

Whenever, $|\dot{\boldsymbol{x}}_p - \dot{\boldsymbol{x}}_q|$ is nonzero. If $|\dot{\boldsymbol{x}}_p - \dot{\boldsymbol{x}}_q| = 0$ then the two points are traveling in the same direction at the same speed, and will always keep the same distance apart, so one can use, tc = 0. In both cases we have that:

$$d_{CPA}(\boldsymbol{x}_p(t), \boldsymbol{x}_q(t)) = |\boldsymbol{x}_p(t_c) - \boldsymbol{x}_q(t_c)|$$

Note that when, $t_c < 0$, then the CPA has already occurred in the past, and the two points are going further apart as they move on in time.

List of symbols

α	Coverage sector start angle
β	Coverage sector stop angle
Δ	Time allocated to message
γ	Coverage sector width
ϕ_{start}	Search sector start angle
ϕ_{stop}	Search sector stop angle
Ψ	Search sector width
η_n	Bearing variance
η_r	Range variance
ω	Scan rate
ρ	Heading of platform
θ	Bearing angle of target
C	Sensing capacity
CL	Communications load
CP	Cue performance
d	Distance measure
F_P	Task-fitness measure
h	Kalman filter time increment
$\boldsymbol{H}_k$	Observation matrix
k	Constant
L	Load
m	Sensor mode
m_p	Priority increase rate
n	Target index
N_s	Number of sensors
N_p	Number of platforms
N_{CH}	Number of communication task holons
N_{CI}	Number of control intervals
N_{SH}	Number of search holons
N_T	Number of targets (or tracks)
N_O	Number of observable targets
p	Platform indice
P	Priority level
$\boldsymbol{P}$	Error covariance matrix
q	Target indice
$\boldsymbol{Q}$	Process noise covariance matrix
r	Range
RC	Sensing capacity
R_T	expected rate of target arrivals
s	Search sector
t	Time index
T	Time period

U	User input
$\boldsymbol{v}_k$	Measurement noise
w	Constant
$\boldsymbol{w}_k$	Process noise
W	Time window length
x	Position on the X axis
$\boldsymbol{x}$	Position vector
$\dot{\boldsymbol{x}}$	Velocity vector
X	X axis of the 2D plane
$\hat{\boldsymbol{x}}_k$	Estimated state vector
y	Position on the Y axis
Y	Y axis of the 2D plane
$\boldsymbol{z}_k$	Measurement vector
Z	Threat level

List of acronyms

CPA	Closest Point of Approach
CSR	Conventional Scanning Radar
DRDC	Defence Research & Development Canada
ESA	Electronically Scanned Array
ESM	Electronic Support Measures
HVU	High Value Unit
POD	Probability Of Detection
QOS	Quality Of Service
RCS	Radar Cross-Section
RH	Resource Holons
SA	Situation Analysis
SICH	Service Interface Command Holon
SM	Sensor Management
TCPA	Time to Closest Point of Approach
TH	Task Holons
UAV	Unmanned Aerial Vehicle

This page intentionally left blank.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Defence R&D Canada – Valcartier
2459 Pie-XI Blvd. North Quebec city, Quebec, Canada
G3J 1X5

2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

UNCLASSIFIED

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

Design of a holonic control architecture for distributed sensor management

4. AUTHORS (Last name, followed by initials – ranks, titles, etc. not to be used.)

Benaskeur, A.; Irandoust, H.; McGuire, P.

5. DATE OF PUBLICATION (Month and year of publication of document.)

September 2009

6a. NO. OF PAGES (Total containing information. Include Annexes, Appendices, etc.)

108

6b. NO. OF REFS (Total cited in document.)

17

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Report

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

Defence R&D Canada – Valcartier
2459 Pie-XI Blvd. North Quebec city, Quebec, Canada G3J 1X5

9a. PROJECT NO. (The applicable research and development project number under which the document was written. Please specify whether project or grant.)

11bs

9b. GRANT OR CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DRDC Valcartier TR 2009-056

10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

(X) Unlimited distribution
(   ) Defence departments and defence contractors; further distribution only as approved
(   ) Defence departments and Canadian defence contractors; further distribution only as approved
(   ) Government departments and agencies; further distribution only as approved
(   ) Defence departments; further distribution only as approved
(   ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11)) is possible, a wider announcement audience may be selected.)

Unlimited

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

Military operations are typically conducted in demanding, dynamic, semi-structured and large-scale environments. The nature of these operating environments makes it difficult to detect, identify and track all the targets within the Volume Of Interest (VOI). To deal with this problem, sensing resources may have to be distributed across a large area, collecting a wealth of data. Yet, to effectively use that data, the sensing resources need to be properly managed.

This report presents the design of holonic sensor management architecture. It follows two previous documents, which detailed the issues involved in military sensor management and the properties of holonic control, respectively. The holonic control proposed here is a novel approach to sensor management, in that its architecture supports dynamic linkages, thus allowing the achievement of changing objectives.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Defence R&D Canada

Canada's Leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE DÉFENSE

www.drdc-rddc.gc.ca