HMM and Auction-based Formulations of ISR Coordination Mechanisms for the Expeditionary Strike Group Missions

Topic 6:
Modeling and Simulation

Woosun An [STUDENT]

Email: woa05001@engr.uconn.edu

Chulwoo Park [STUDENT]

Email: chp06004@engr.uconn.edu

*****Krishna Pattipati

University of Connecticut

Dept. of Electrical and Computer Engineering

371 Fairfield Way, U-2157

Storrs, CT 06269-2157, United States

FAX: 860-486-5585

Phone: 860-486-2890

Email: krishna@engr.uconn.edu

David L. Kleinman

Naval Postgraduate School, Dept. of Information Sciences

Monterey, CA 93943, United States

Email: dlkleinm@nps.edu

William G. Kemple

Naval Postgraduate School, Dept. of Information Sciences

Monterey, CA 93943, United States

kemple@nps.edu

* To whom correspondence should be addressed: krishna@engr.uconn.edu

ABSTRACT

In this paper, multi-stage auction-based intelligence, surveillance, and reconnaissance (ISR) coordination mechanisms are investigated in the context of dynamic and uncertain mission environments, such as those faced by expeditionary strike groups. Each attribute of the mission task is modeled using a hidden Markov model (HMM) with controllable emission matrices, corresponding to each ISR asset. For each HMM-sensor pair, we evaluate a matrix of information gains (uncertainty reduction measures); the elements of this matrix depend on the coordination structure and the concomitant delays accrued. We consider three coordination structures (distributed ISR coordination, ISR officer serving as a coordinator, ISR officer serving as a commander) here. We evaluate these structures on a hypothetical mission scenario that requires the monitoring of ISR activities in multiple geographic regions. The three structures are evaluated by comparing the task state estimation error cost, as well as travel, waiting and assignment delays. The results of analysis were used as a guide in the design of a mission scenario and asset composition for team-in-the-loop experimentation at NPS. Our solution has the potential to be a mixed initiative decision support tool to an ISR coordinator/commander, where the human provides possible sensor-task pairings and the tool evaluates the efficacy of assignment in terms of task accuracy and delays.

Keywords: Sensor scheduling, Sensor assignment, Hidden Markov model (HMM), Auction algorithm, Information gain heuristic, Coordination delays

1. Introduction

1.1. Motivation

Complex surveillance applications, such as multi-target tracking and use of unmanned aerial vehicles (UAVs) for monitoring activities in remote or hostile environments, require one to trade off sensor performance (e.g., detection, identification, and tracking accuracies) and the sensor usage cost (e.g., power and bandwidth consumption, distance traveled, risk of exposure, deployment requirements). The objective of dynamic sensor scheduling is to judiciously allocate sensing resources to exploit the individual sensors' capabilities, while minimizing their usage cost. As an example, consider a target identification scenario where an incoming aircraft needs to be identified as an enemy or a friendly target using active or passive sensors available at a surveillance station. This scenario requires sensor scheduling because active sensors (e.g., radar) tend to reveal clues about the location of the surveillance station to a potential enemy aircraft, whereas the more stealthy passive sensors tend to be inaccurate. Thus, in this case, the sensor scheduling algorithm needs to trade-off accuracy versus risk of exposure. As another example, unmanned aerial vehicles (UAVs) are preferred assets for monitoring nearly all the intelligence, surveillance, and reconnaissance (ISR) activities; however, they cannot be deployed in large numbers due to their limited availability. Thus, astute allocation of scarce resources is a major issue in ISR coordination.

In this paper, we develop analytic models of an expeditionary strike group (ESG) with different ISR coordination structures tasked with executing a surveillance mission. An ESG provides a flexible Navy-Marine force, capable of tailoring itself to a wide variety of missions. An important ESG mission involves dealing with asymmetric threats, such as terrorist groups who carry out attacks while trying to avoid direct confrontation. This stealthy behavior makes it very difficult to predict when and where they will strike. Moreover, the increased geographical range and unpredictable nature of this behavior require effective allocation and appropriate scheduling of sensors to achieve mission objectives. Effectively performing the ISR activities is a key step to gain situational awareness, which, in turn, enables the allocation of resources for the interdiction of asymmetric threats.

We model the asymmetric threats using hidden Markov models (HMMs), because these activities are concealed and their true state can only be inferred through the uncertain observations obtained using various ISR sensors. A pattern of these observations and its dynamic evolution over time provides the information base for inferring a potential realization of an asymmetric threat. Thus, each state of a HMM is characterized by a set of attributes, and a sensor package consisting of a subset of sensors is needed to accurately estimate these attributes and, consequently, to infer the task state.

This is the type of problem considered by Hutchins *et al*. [12], where they have examined how an ESG with alternative structures and processes affect the decision performance and information flow in information-rich planning and execution environments according to the authority level of an ISR officer. In this environment, operationalized in the Distributed Dynamic Decision-making (DDD-III) simulator, multiple tasks, each having multiple attributes (e.g., illegal weapon running, crowd behavior, and terrorist activities), need to be monitored. For each task, multiple ISR assets, each having different capabilities to measure a subset of the task‘s attributes, need to be allocated to gain situational awareness. The experiment has been designed to emphasize the (limited) ISR resources of the ESG, and participants are required to assign ISR sensors over time to learn the attributes of tasks for subsequent task processing. The fact that subsets of sensors are needed to measure a task’s attributes leads to a non-traditional, dynamic and many-to-one assignment (i.e., multiple sensors for a given task) problem. No known solutions have been devised for this problem.

In our previous work [24], assuming that any sensor can measure all the attributes of a task (i.e., one-to-one assignment problem), a hidden Markov model (HMM)-based dynamic sensor scheduling problem was formulated, and solved using information gain and rollout concepts to overcome the computational intractability of the dynamic programming recursion. The problem involves dynamically sequencing a set of sensors to monitor multiples tasks, which are modeled as multiple HMMs with multiple emission matrices corresponding to each of the sensors. The dynamic sequencing problem is to minimize the sum of sensor usage costs and the task state estimation error costs. The rollout information gain algorithm proposed in [24] employs the information gain heuristic as the base algorithm to solve the dynamic sensor sequencing problem. The information gain heuristic selects the best sensor assignment at each time epoch that maximizes the sum of information gains per unit sensor usage cost, subject to the assignment constraints that at most one sensor can be assigned to a HMM and that at most one HMM can be assigned to a sensor. The rollout strategy involves combining the information gain heuristic with the Jonker-Volgenant-Castañon (JVC) assignment algorithm and a modified Murty's algorithm to compute the κ-best assignments at each decision epoch of rollout. The capabilities of the rollout information gain algorithm were illustrated using a hypothetical scenario based on [12] to monitor intelligence, surveillance, and reconnaissance (ISR) activities in multiple fishing villages and refugee camps for the presence of weapons and terrorists or refugees.

In this paper, motivated by the many-to-one assignment inherent in the ISR coordination problem [12], we extend the algorithms in [24] by proposing a three-phase approach for solving it. During phase I, at each decision epoch, we compute M-best sensor packages for measuring the attributes of each task modeled as a HMM, where M is a user-specified parameter. For N_t tasks and M sensor packages each, there will be at most MN_t sensor packages. However, these sensor packages may have overlapping sensors (i.e., same sensor in two different packages) when viewed across tasks. In order to ensure that no sensor is allocated to more than one task at a decision epoch, we generate, during phase II of our solution approach, L disjoint sensor package sets over all tasks, where L again is a user parameter. We accomplish this by solving a set packing problem, extended to generate L-best solutions. The result is L one-to-one assignment problems of allocating disjoint sensor packages to tasks. Each of the L problems is similar to the one considered in [24]. However, one now needs to compute the information gain for a sensor package, rather than a single sensor. The solution of L one-to-one assignment problems constitutes the Phase III of our solution approach. Decomposing the original problem into three sequential subproblems overcomes the computational intractability of the dynamic many-to-one assignment problem.

We evaluate the many-to-one assignment algorithm in terms of state estimation errors and delays using a realistic ISR scenario on the three coordination structures defined in [12]: self-synchronization, ISR officer as a coordinator, and ISR officer as a commander. In a self-synchronizing structure, decision makers (DMs), e.g., Sea Component Commander (SCC) and Marine Expeditionary Unit (MEU) Commander, take individual responsibility for assigning and coordinating ISR assets to the mission tasks. Communication channels among DMs are activated for exchanging information of unused assets and unassigned tasks so that DMs with excess assets can assist DMs with scarce resources and yet are responsible for processing tasks. This coordination is accomplished via a multi-step auction mechanism. In the ISR commander structure, DMs provide lists of tasks, assets and information gains for each task-asset pair (i.e., efficacies of assigning an ISR asset to a task) to the ISR commander. The ISR commander solves a centralized task-asset assignment problem and transmits his assignments to the subordinate DMs. In the ISR coordinator structure, DMs solve their own individual task-asset assignment problems for their own tasks first. The ISR coordinator compares rewards of self-synchronization with that of a centralized structure. If the reward of ISR asset assignment using a self-synchronizing structure is less than a specified fraction of that accrued from a centralized assignment, the ISR coordinator recommends centralized task-asset assignments to the DMs. The value of the fraction enables us to model a number of coordination behaviors ranging from a fully-engaged coordinator to a hands-off coordinator.

1.2. Previous Work

A simplified version of the problem considered here is also related to the dynamic sensor scheduling problem; this problem has been widely studied in the area of target tracking [22], [6]. Mathematically, the problem is to solve a sequential stochastic allocation problem that seeks to minimize the expected scheduling cost under a given set of constraints over time [6]. For linear Gaussian state space systems, one can obtain an analytic solution for the posterior distribution of the system state given the sensor measurements and a sensor sequence via a Kalman filter [16]. Shakeri *et al*. [19] formulated the sensor scheduling problem subject to a fixed total budget and the cost of individual sensor varying inversely with its measurement variance. They obtained an optimal measurement schedule that minimizes the trace of a weighted sum of the estimation error covariance matrices of a discrete-time vector stochastic process, when the auto-correlation matrix of the process is given. The study showed that the problem can be transformed into a nonlinear programming problem with linear equality and inequality constraints. In the special case of a linear finite-dimensional stochastic system, they showed that the problem can be formulated as a nonlinear optimal control problem, where the gradient and Hessian of the objective function with respect to the sensor accuracy parameters can be derived via a two-point boundary value problem. The resulting optimization problem was solved via a projected Newton Method [19].

In [21], Singh *et al*. provided a summary of previous research on sensor scheduling for tracking targets, whose dynamics are modeled by linear Gauss-Markov processes. They formulated the sensor scheduling problem as one of minimizing the variance of the estimation error of hidden states of a continuous-time HMM with respect to a given action sequence [21]. The authors proposed a stochastic gradient algorithm to determine the optimal schedule for the HMM. Another effort, related to our work, using a discrete HMM framework was considered by Krishnamurthy in [14]. Here, the author proposed a stochastic dynamic programming (DP) framework to solve the sensor scheduling problem, which is intractable for all but simple HMMs with a few states (e.g., at most 15 states).

Sub-optimal approaches, based on information-theoretic criteria, have been developed for overcoming the computational intractability of determining the optimal sensor schedule. For a linear Gauss-Markov system, Logethitis *et al*. [15] formulated the sensor scheduling problem as one of determining a sequence of active sensors to maximize the mutual information between the states of the unobserved dynamic process and the observation process generated by the sensors. In the context of sensor networks, Zhao *et al*. [23] and Chu *et al*. [7] formulated the target tracking problem as a sequential Bayesian estimation problem, where the participants for sensor collaboration are determined by minimizing an objective function comprised of information utility e.g., measured in terms of entropy, Mahalanobis distance and the sensor usage cost.

1.3. Scope and Organization of the paper

In section 2, the many-to-one sensor scheduling problem is formulated and solved using the three-phase approach. In section 3, the scheduling model is adapted to the three coordination structures by including waiting, travel and coordination delays pertinent to each coordination structure. In section 4, we discuss how auction mechanisms are modeled for the three ISR coordination structures, viz., self-synchronization, ISR

Coordinator and ISR Commander. In section 5, we apply our model to an ISR mission scenario and present the analysis results. Finally, section 6 concludes with a summary.

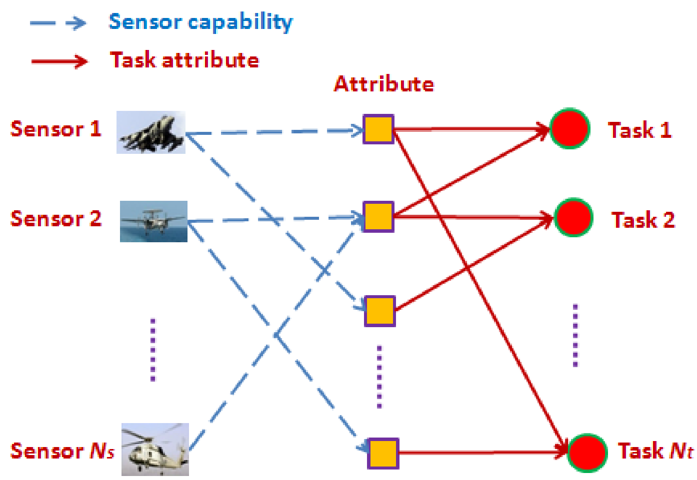

Figure.1. Many-to-one assignment problem

2. Dynamic Sensor Scheduling Problem and its Solution

Consider a scenario with N_t discrete HMMs (representing tasks, r=1, 2, .., N_t) that are evolving independently, but are coupled via the sensor allocation policy. This model is also known as a factorial hidden Markov model (FHMM) in the machine learning literature [11][24]. Each HMM state is characterized by a set of task attributes, and each sensor can measure certain of these attributes; this implies that multiple sensors may be needed to cover the attributes of a task state as shown in Fig.1. Suppose there are N_s sensors, and $\mu(k) \subseteq \{1,2,..,N_s\}$ are the set of available sensors at decision epoch $k \in \{1,2,..,K\}$. Let $|\mu(k)| = N_s(k)$ denote cardinality of available sensors at decision epoch k. We assume that a sensor package $s_{lr}(k)$, l=1,2,..,M out of available sensors, $\mu(k)$, that covers the attributes of the hidden state of a HMM, r is assigned at each time k. Evidently, this is a many-to-one assignment problem. The objective is to minimize the task state estimation error and sensor usage cost over all tasks for a specified planning horizon of K time epochs.

We decompose the solution approach for the many-to-one assignment problem at each decision epoch into three sequential phases: M-best sensor package generation for each task, L-best disjoint set generation via modified set packing that generates multiple solutions, and solving the resulting L one-to-one assignment problems using the algorithms in [24]. For simplicity of presentation, we omit the time index, k in sections 2.1 and 2.2. The notation employed in this section is listed in Table 1.

Let

N_a = Number of distinct attributes over all tasks

N_t = Number of tasks

N_s = Number of sensors

$B =$ Binary N_a by N_t attribute-HMM matrix defined by,

$$B_{ir} = \begin{cases} 1: & \text{if } i \in \text{attribute set of HMM } r \\ 0: & \text{otherwise} \end{cases}$$

$A =$ N_a by N_s Binary sensor capability matirx defined by,

$$A_{iq} = \begin{cases} 1: & \text{if sensor } q \text{ is capable of measuring the } i^{th} \text{ attribute} \\ 0: & \text{otherwise} \end{cases}$$

$$x_{qr}^{(l)} = \begin{cases} 1, & \text{if sensor } q \text{ is allocated to HMM } r \text{ as part of package } s_{lr}, l = 1, \ldots, M \\ 0, & \text{otherwise} \end{cases}$$

S_r = Set of M-best sensor packages capable of measuring HMM r

$= \{s_{1r}, \ldots, s_{Mr}\}$ (output of Phase I)

$$S = \bigcup_{r=1}^{N_t} S_r$$

W = Distinct (unique) set of sensor packages obtained from S

$= \{w_m : w_m \in S; w_m \neq w_{m'}, m \neq m'\}$

$$y_{qm} = \begin{cases} 1, & \text{if sensor } q \text{ is assigned to distinct sensor package } w_m \\ 0, & \text{otherwise} \end{cases}$$

where $\begin{cases} w_m = s_{lr}, \text{ if } y_{qm} = x_{qr}^{(l)}, \forall\ q = 1, \ldots, N_s \\ w_m \neq s_{lr}, \text{ otherwise} \end{cases}$

$D^{(i)}$ = The i^{th}-best disjoint set of sensor packages that pack the set of sensors, $i = 1, 2, .., L$

$= \{w_n^{(i)} : \sum_{n=1}^{|D^{(i)}|} z_{qn}^{(i)} \leq 1,\ q = 1, \ldots, N_s\}$ (output of Phase II)

$$z_{qn}^{(i)} = \begin{cases} 1, \text{ if sensor } q \text{ is assigned to } w_n^{(i)} \\ 0, \text{ otherwise} \end{cases}$$

$$\text{where, } \begin{cases} w_n^{(i)} = w_m, \text{ if } z_{qn}^{(i)} = y_{qm}, \forall\ q = 1,\ldots,N_s \\ w_n^{(i)} \neq w_m, \text{ otherwise} \end{cases}$$

$$u_m^{(i)} = \begin{cases} 1, \text{ if } w_m \text{ is assigned to } D^{(i)}, m = 1,\ldots,|W| \\ 0, \text{ otherwise} \end{cases}$$

Table.1. Summary of notation

2.1 Phase I: Multiple sensor packages for each task

Consider a task *r*. Our objective is to generate *M*-best sensor packages of minimal cardinality that are capable of measuring all the attributes of task *r*, while minimizing the sum of travel costs of sensors in each sensor package from their current locations to the location of task *r*.

The problem of finding the best sensor package corresponds to the following binary programming problem (BPP), one for each task *r*:

$$\begin{aligned} f &= \min_{x_{qr}} \sum_{q=1}^{N_s} \left(d_{qr} + C\right) x_{qr}^{(1)} \\ \text{s.t. } & \sum_{q=1}^{N_s} A_{iq} x_{qr}^{(1)} \geq B_{ir}, \quad i = 1,2,..,N_a \\ & x_{qr}^{(1)} \in \{0,1\}, \qquad q = 1,2,..,N_s \end{aligned} \tag{1}$$

Here the superscript (1) denotes the best solution to the BPP. In addition, d_{qr} is the cost of moving sensor q from its current location to the location of task *r*. This cost is computed via:

$$d_{qr} = \frac{\|(a_r, b_r) - (a_q, b_q)\|_2}{v_q} \tag{2}$$

where (a_r, b_r) and (a_q, b_q) denote the Cartesian coordinates of the location of task *r* and the location of sensor *q*, respectively, and v_q denotes the velocity of sensor *q* (or the mobile platform on which it is resident .) In (1), the term $C\sum_{q=1}^{M} x_{qr}$ is a penalty factor used to generate *M*-best sensor packages of minimal cardinality. The coefficient C is selected such that $C > \max_q(d_{qr})$.

We employed a branch-and-bound algorithm for generating *M*-best solutions to the BPP in (1) by partitioning the space of feasible solutions such that the top *j*-best solutions are precluded when finding the $(j+1)^{th}$ best solution. A simple and efficient method for partitioning the feasible space is to express each sensor package as a binary vector $s_{lr} \to [x_{1r}^{(l)}, ..., x_{qr}^{(l)}, ..., x_{N_s r}^{(l)}], l = 1, 2, .., M$.

(1) Initialize $S_r = \varnothing$, constraints of (1) $\to P$, set of optimal costs $H = \varnothing$.

(2) Solve (1) to obtain optimal sensor package s_{1r}. If there is no feasible solution, $H \to H \cup \{\infty\}$ stop. Otherwise, $S_r \to \{s_{1r}\} \cup S_r$, $H \to H \cup \{f\}$, set *j*= *2* and go to step 3.

(3) Suppose we have the $(j-1)^{th}$ best solution $s_{(j-1)r} \to [x_{1r}^{(j-1)}, ..., x_{qr}^{(j-1)}, ..., x_{N_s r}^{(j-1)}]$

(4) Form two subproblems by adding the following constraint to the constraints set of BPP of node $s_{(j-1)r}$:

$$\sum_{q=1}^{|N_s|} 2^{q-1} x_{qr}^{(j)} > \sum_{q=1}^{|N_s|} 2^{q-1} x_{qr}^{(j-1)} \cup P \to P_i$$

$$\sum_{q=1}^{|N_s|} 2^{q-1} x_{qr}^{(j)} < \sum_{q=1}^{|N_s|} 2^{q-1} x_{qr}^{(j-1)} \cup P \to P_g$$

(5) Solve the two BPPs with P_i and P_g as constraints, respectively. If there is no feasible solution for both, stop. If a candidate solution $s'_{jr} \to [x'^{(j)}_{1r}, ..., x'^{(j)}_{qr}, ..., x'^{(j)}_{N_s r}]$ is redundant (i.e., a superset of a previous best

solution), remove the solution, but add the corresponding constraints to the constrain set. Set $x_{qr}^{(j-1)} = x_{qr}'^{(j)}, q = 1,..., N_s$ and go to step 4. Otherwise, save the costs $H \to H \cup \{f_i, f_g\}$ and the corresponding sensor packages. Obtain the j^{th} best sensor package from H. Set j=j+1 and go to step 3.

2.2 Phase II: Multiple disjoint set generation from sensor packages

For N_t tasks and M sensor packages each, there will be as many as MN_t sensor packages. However, these sensor packages may have sensor overlaps when viewed across tasks. In order to ensure that no sensor is allocated to more than one task at a decision epoch, we generate, during phase II of our solution approach, L disjoint sensor package sets over all tasks, where L again is a user parameter. To do this, we define a set of distinct sensor packages W by grouping all unique sensor packages without regard to HMMs from which such packages were obtained. Then, the problem of obtaining a disjoint sensor set with maximum number of sensors can be formulated as a set packing problem (here superscript (1) denotes the best solution for the set packing problem):

$$\begin{aligned} &\max_{u_m} \sum_{m=1}^{|W|} u_m^{(1)} \\ &\text{s.t.} \sum_{m=1}^{|W|} y_{qm} u_m^{(1)} \le 1, \qquad q = 1,\ldots N_s \\ &\qquad u_m^{(1)} \in \{0,1\}, \qquad m = 1,\ldots,|W| \end{aligned} \tag{3}$$

The L-best solutions $D = \{D^{(1)},..., D^{(L)}\}$ to the set packing problem in (3) are obtained by combining the set packing algorithm, coupled with the branch-and-bound algorithm outlined earlier.

2.3 Phase III: Sensor package assignment

In this phase, we solve L assignment problems, one for each $\{D^{(i)}(k)\}_{i=1}^{L}$. The one-to-one assignment problem of allocating a disjoint sensor package set, say $D^{(i)}(k)$, at decision epoch k to N_t tasks at time k is solved, subject to assignment constraints that each task is assigned no more than one sensor package and each sensor package is assigned to no more than a single task. The objective here is to maximize the sum of information gains (IG) at each time epoch k:

$$\psi^{*k} = \arg\max_{1\le i\le L} \arg\max_{\psi^{(i)}(k)} \sum_{n=1}^{|D^{(i)}(k)|} \sum_{r=1}^{N_t(k)} \frac{I_{rn}(\underline{\pi}_r(k\,|\,k\text{-}1),\, w_n^{(i)}(k)\in S_r(k))}{\sum_{q=1}^{N_s(k)} d_{qr}(k) z_{qn}^{(i)}(k)} \psi_{rn}^{(i)}(k)$$

$$\text{subject to} \quad \sum_{n=1}^{|D^{(i)}(k)|} \psi_{rn}^{(i)}(k) \le 1 \quad r=1,\ldots,N_t(k)$$
$$\sum_{r=1}^{N_t(k)} \psi_{rn}^{(i)}(k) \le 1 \quad n=1,2,..,N_s(k) \tag{4}$$

Here $\underline{\pi}_r(k\,|\,k\text{-}1)$ is information state at decision epoch k based on all the observations up to and including time (k-1) [24]. The one-to-one assignment problem in (4) is solved using the JVC algorithm [13].

3. Including Delays into the HMM-based Sensor Scheduling Model

When a DM (e.g., MEU, SCC, ISR officer) assigns a sensor package $s_{lr}(k)\in S_r(k)$ to a task r at time epoch k, there will be a delay of $\Delta_C(s_{lr}(k))$ due to delays caused by assignment, travel, and waiting for a busy sensor to become available. The net effect is that allocation decisions at time k results in a task attribute being measured at time epoch $[k+\Delta_C(s_{lr}(k))]$. The delay time $\Delta_C(s_{lr}(k))$ can be written as sum of the three constituent delays:

$$\Delta_C(s_{lr}(k)) = \Delta_C(k) + \Delta(s_{lr}(k)) + \Delta_r(k) \tag{5}$$

where $\Delta_C(k)$ is the assignment delay, $\Delta(s_{lr}(k))$ is the travel delay and $\Delta_r(k)$ is the waiting delay.

3.1 Assignment delay

The assignment delay $\Delta_C(k)$ depends on the sensor allocation process, which in turn depends on the ISR coordination structure (Distributed (self-synchronization), ISR officer as a coordinator, ISR officer as a commander). We model this delay as being comprised of two components: synchronization and coordination. In the synchronization phase, DMs assign their available assets to their tasks first at time epoch *k*. This can be done in parallel. Consequently, it is reasonable to choose the maximum assignment delay among DMs as the synchronization delay. During the coordination phase, DMs exchange information on their unassigned tasks and unused assets among them. When there are available assets and unassigned tasks, a coordination delay of $\Delta_\beta(k)$ is accrued. Thus,

$$\Delta_C(k) = \max_\alpha(\Delta_{DM\alpha}(k)) + \Delta_\beta(k) \tag{6}$$

where $\Delta_{DM\alpha}(k)$ and $\Delta_\beta(k)$ can be functions of available assets and unassigned tasks.

3.2 Travel and waiting delays

The travel delay d_{qr} is the time it takes for an ISR asset $q \in s_{lr}(k)$ to reach the task's location (a_r, b_r) so that a measurement on the attribute of task *r* can be made. This delay is given by Eq. (2). Thus,

$$\Delta(s_{lr}(k)) = \max_{q \in s_{lr}(k)} d_{qr} \tag{7}$$

When one or more of the ISR assets is not available for assigning it to task *r* at time epoch *k*, the task needs to wait until the asset package becomes available. This delay is the waiting delay $\Delta_r(k)$, which can be computed by keeping track of the sensor availability status.

3.3 Including Delays in HMM State Propagation

Suppose we assign the ISR asset package $s_{lr}(k)$ to HMM representing task *r*. The measurement from this asset becomes available at time $[k + \Delta_C(s_{lr}(k))]$. The information state, $\underline{\pi}_r(k \mid k)$, is propagated up to time epoch $[k + \Delta_C(s_{lr}(k))]$ via

$$\underline{\pi}_r(l \mid k) = A_r^T(l-1)\underline{\pi}_r(l-1 \mid k); l = k+1, k+2, ..., k + \Delta_C(s_{lr}(k)) \tag{8}$$

where $A_r(l-1)$ denotes the transition probability matrix of HMM *r* at time epoch (*l*-1).

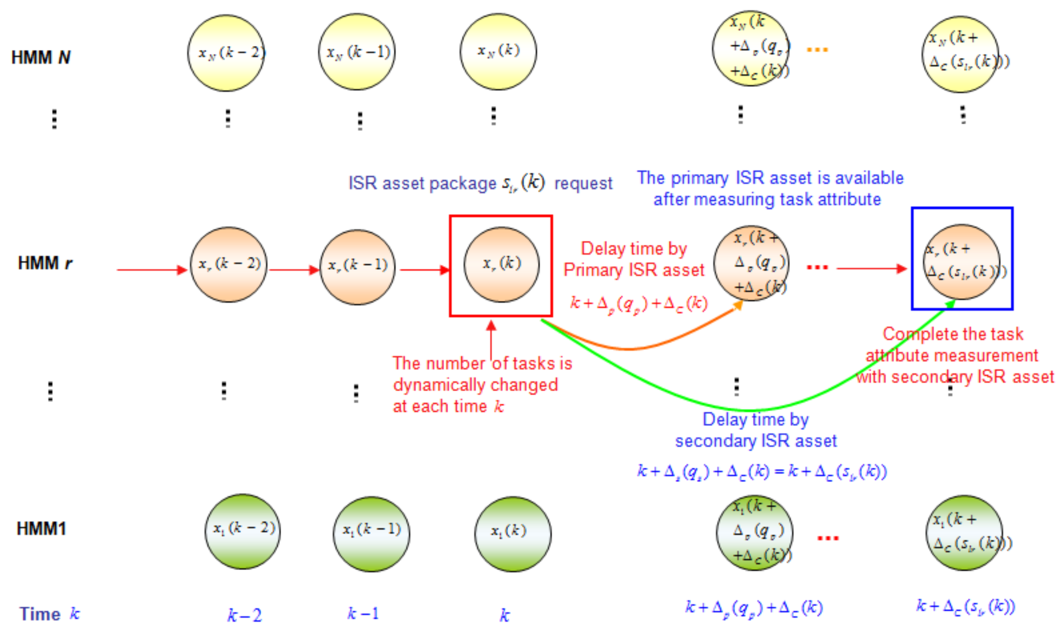

Figure.2. Delay model combined with HMM sensor scheduling model

At time epoch $k+\Delta_C(s_{lr}(k))$, the information state $\underline{\pi}_r(k+\Delta_C(s_{lr}(k))\,|\,k+\Delta_C(s_{lr}(k)))$ is updated by the measurement made by the assigned asset package $s_{lr}(k)$:

$$\pi_{ri}(k+\Delta_C(s_{lr}(k)\,|\,k+\Delta_C(s_{lr}(k)) = \frac{b_{rhil}(k+\Delta_C(s_{lr}(k))\pi_{ri}(k+\Delta_C(s_{lr}(k))\,|\,k)}{\sum_{i=1}^{n_r} b_{rhil}(k+\Delta_C(s_{lr}(k))\pi_{ri}(k+\Delta_C(s_{lr}(k))\,|\,k)} \tag{9}$$

Here, b_{rhil} represents emission probability of observing h^{th} element of emission matrix B_{lr}, when observing the i^{th} state of HMM r with l^{th} sensor package $s_{lr}(k)\in S_r$ [24]. This process is shown in Fig.2, where we have assumed, for illustrative purposes, the sensor package to be comprised of a primary asset q_p and secondary asset q_s, i.e., $s_{lr}(k)=\{q_p,q_s\}$ and $x_r(k)$ represents the hidden state of HMM r at time epoch k.

4. Models for ISR Coordination Structures

4.1 Distributed ISR coordination (Self-synchronization)

A self-synchronizing structure has the attributes of distributed intelligence, diversity, self-organization, and lateral accountability [25]. In this structure, DMs communicate as peers; there are no fixed supervisor/subordinate relationships. Coordination among DMs is realized by using a market mechanism, such as the ‘contract net protocol’ or the ‘request for bid’ protocols, etc. Some inherent capabilities of this structure include self-configuration, flexibility, fault-tolerance, reduced complexity, and emergent behaviors [26].

The distributed ISR coordination mechanism is modeled as a multi-stage auction as follows:

1. Each DM solves his own ISR problem for his own mission tasks first via the auction algorithm.
2. At the end of this phase, each DM broadcasts the availability of unused assets and any unassigned tasks (requiring assets) for which he is responsible.
3. Each DM solves the resulting assignment problem.
4. If all the tasks are assigned or there are no available assets, stop. Else, go back to step 2.

Evidently, the number of auction stages, i.e., information exchanges and assignment problems to be solved, is dynamic. It depends on the number of tasks to be assigned, available assets, and the assumed delay models.

4.2 ISR coordination by a Commander

Traditional C2 hierarchy keeps authority and information at the center. The control flow in this structure is typically top-down and the feedback information is bottom-up. The DMs at the upper level make asset allocations and coordinate lower level units, while DMs at the lowest level execute the tasks. One of the many merits of a hierarchical C2 structure is that it provides unity of command, which refers to the principle that a subordinate should have one and only one superior to whom he or she is directly responsible. Because military power is the product of multiple capabilities, a centralized C2, as an embodiment of the principle of unity of command, is essential to effectively fuse these capabilities.

The ISR coordination by a commander is modeled as a single-stage auction. This is because the ISR commander has the authority to control all the ISR assets (including his own) assigned to the mission. The coordination process in this structure proceeds as follows:

1. Each DM computes information gain for all tasks when his assets are assigned to them.
2. Each DM provides the list of tasks, assets and information gain data to the commander.
3. Commander solves the centralized assignment problem via the auction or the JVC algorithm.

4.3 ISR coordination by a Coordinator

The ISR coordinator does not own assets, but facilitates information flow and effective planning. We model the coordinator structure as an intermediate between a self-synchronizing and commander structures. Specifically, when the sum of information gains obtained by the distributed ISR coordination structure is not substantially less than that of the centralized (commander-derived) solution, the ISR coordinator does not intervene; otherwise, the coordinator suggests the implementation of a centralized solution to the DMs involved in mission planning and execution. Formally, if

$$\frac{\sum_{n=1}^{|D^{(Dis)}|}\sum_{r=1}^{N_t(k)} I_{rn}(k)\psi_{rn}^{*Dis}(k)}{\sum_{n=1}^{|D^{(Com)}|}\sum_{r=1}^{N_t(k)} I_{rn}(k)\psi_{rn}^{*Com}(k)} > \gamma \tag{10}$$

where $\psi_{rn}^{*Com}(k)$ is the asset assignment under an ISR commander structure and $\psi_{rn}^{*Dis}(k)$ is the asset assignment of a self-synchronizing structure at time epoch k. When the threshold $\gamma = 0$, the coordinator does not intervene. On the other hand, when $\gamma \approx 1$, the coordinator acts like an ISR commander. Thus, the threshold parameter γ enables us to model a number of coordination behaviors ranging from a fully-engaged coordinator to a hands-off coordinator.

5. Computational Results

5.1 A Hypothetical Mission Scenario

This scenario, motivated by ESG missions, involves simultaneous monitoring of multiple geographically dispersed threat activities. Here, an ISR officer needs to dynamically allocate sensors to monitor asymmetric threat activities in a notional area (e.g. fishing villages, refugee camps) that involves primarily two fictitious countries, Asiland and Bartola [12]. Asiland is an unstable state, where maritime smugglers and anti-western terrorist groups have supported the insurgent factions hostile to the government of Bartola. Local terrorists and sea rovers use Asiland's as a base. The scenario considers that nearly a month ago, the northern shore of Asiland was struck by a tsunami that destroyed several fishing villages and caused enormous casualties. Large numbers of Asiland citizens sought refuge in south for help and assistance. However, this exodus quickly drained the resources of Asiland. Consequently, many Asiland refugees began to move to fishing villages and refugee camps in Bartola. Within a few days, insurgents and terrorist factions in and around Asiland began to exploit the situation, infiltrating their operations into Bartola by disguising as refugees and smuggling weapons onboard fishing boats and merchant ships. Bartola's military was overwhelmed with controlling massive influx of refugee boats, as well as tracking the terrorist/insurgent's activities using these boats and ships for illegal transfers. The government of Bartola sought help from the United States to provide Humanitarian Assistance/Disaster Relief (HA/DR) to Bartola and the organizations operating relief activities within it. The ESG sensor assets are deployed and begin to monitor strategically significant areas (e.g. major sea and air lanes as well as several major ports, villages, refugee camps, roads, and cities/sites) as shown in Fig. 3.

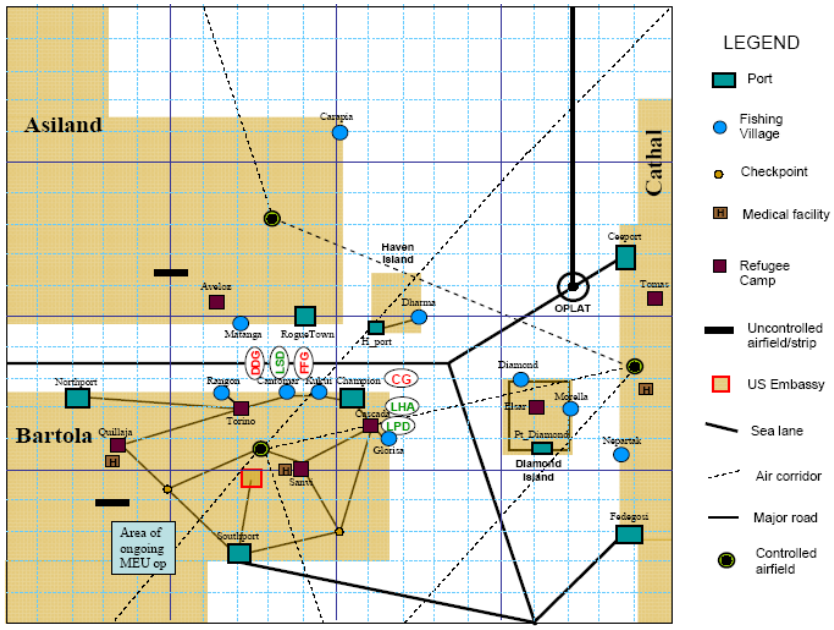

Figure.3. Notional area for scenario development

5.2 Delay and Cost Models

The assignment delay for the distributed ISR structure is computed as follows:

$$\Delta_C(k) = \sigma(\max_{\alpha}(N_{s(DM\alpha)}(k) \times N_{t(DM\alpha)}(k))^{\rho}) + \beta(U_s(k) \times U_t(k))^{\rho} \qquad (11)$$

where σ is a *structural delay* factor, $N_{sDM\alpha}(k)$ is the number of available assets (sensors) and $N_{tDM\alpha}(k)$ is the number of tasks that DM_{α} is responsible for at time epoch *k*. Here, we refer to β as a delay factor for coordination between DMs. Here, $U_s(k)$ is the number of available (unassigned) assets and $U_t(k)$ is the number of unassigned tasks after ISR asset assignment of synchronization phase. ρ is used as a coordination delay model parameter. Similarly, the assignment delay for the ISR commander structure is set as follows:

$$\Delta_C(k) = \sigma(N_s(k) \times N_t(k))^{\rho} \qquad (12)$$

where N_s is the total number of available assets and N_t is the total number of tasks at time epoch k. Cost for sensor assignment and information state update is not considered in this experiment. The estimation error per completed task is computed as follows:

$$J(\mathbf{\Pi}(k\,|\,k)) = \frac{1}{\text{Completed tasks}} \sum_{r=1}^{N} \sum_{k=1}^{K} [1 - \underline{\pi}_r(k/k)^T \underline{\pi}_r(k/k)] \tag{13}$$

5.3 Pre-experiment Analysis for Selecting Assets for Experimentation

The dynamic sensor scheduling model was used as a guide in the design of a mission scenario and asset composition for A2C2 team-in-the-loop experiment 11 at NPS. The purpose of Experiment 11 was to investigate whether coordination structures make a difference in resource scarce environments. The purpose of this pre-experiment analysis

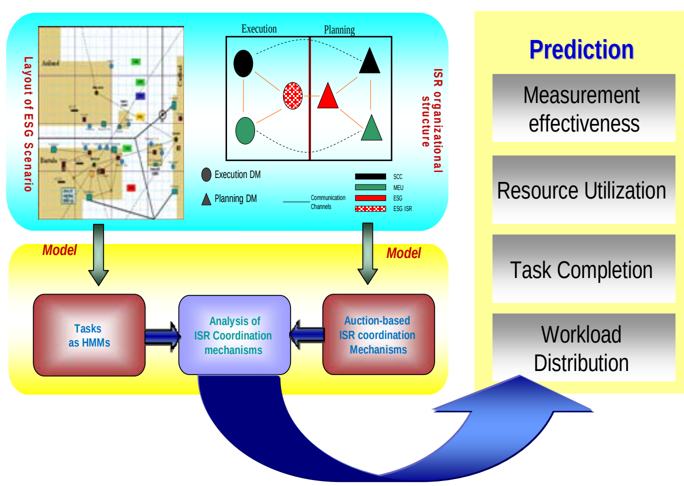

Figure.4. ESG Pre-experiment Analysis using Model-based Approach

was to evaluate the performance of the three coordination structure on the ISR mission scenario in [12] with the asset packages shown in Table 4. Here, the baseline asset package S_0 corresponds to the same as that used in [12], while scenarios S_1-S_6 correspond to progressively reduced asset packages. The assets have different speeds, and measurement ranges, as shown Table 2. We assume that each asset can measure a single attribute. Asset measurement capabilities for task classes are as shown in Table 3, where blue color represents attributes of task classes that assets can monitor. The ISR assets need to measure 84 task attributes, which are modeled as 84 HMMs. The structural delay

	MEU				SCC			
	RECC	XH30	UAV	MSPF	RHIB	XH30	HH60	UAV
Unit Number	4	2	2	2	2	2	2	2
Velocity (Kt)	90	90	75	90	50	90	90	75
Measurement Range	3.5	3.5~10	12~15	4.5	4.5	3.5~10	3.5~6	12~15

Table.2. Velocities and measurement ranges of assets

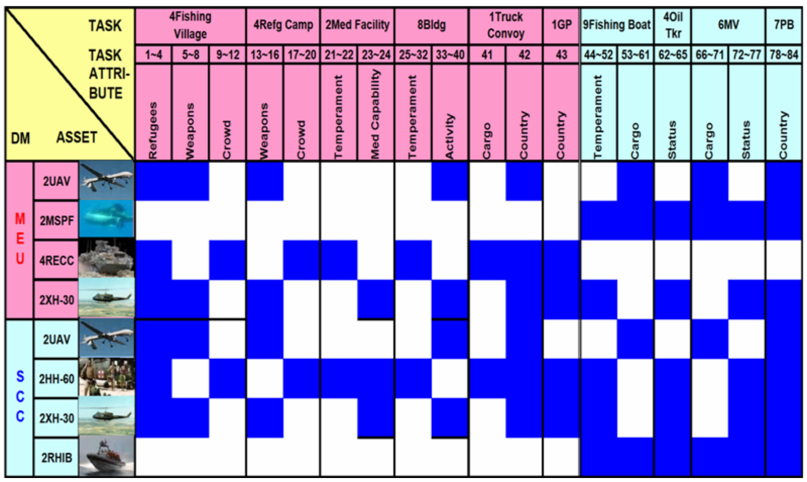

Table.3. Asset-task capability table

factor σ is set as 0.01, and the coordination delay parameter β is set as 0.01, and the threshold γ is set as 0.5, and the coordination delay model parameter ρ is set as 1/2. We specify the transition probability matrix for each HMM based on task arrival patterns specified in the DDD mission scenario definition file [12], [24]. The emission matrices are set by considering asset measurement capabilities for various task classes. The mission in notional area consists of ground tasks (e.g. fishing village, refugee camp, medical facility, building, truck convoy, and ground patrol) and maritime tasks (e.g. fishing boat, oil tanker, merchant ship, and patrol boat). In this scenario, MEU has the responsibility for monitoring the ground tasks, while SCC has the responsibility for monitoring the maritime tasks. The total DDD simulation time for this mission scenario was 90 minutes, which corresponded to 18 hours of actual mission operation. We used a sampling interval of 1 minute so that the number of time epochs $K = 90$. The tasks need to be monitored periodically for persistent surveillance every 15 minutes. We model this by reintroducing a completed ISR task as a new arrival after 10 time epochs so that this task is revisited for persistent surveillance. We ran the simulations for the seven asset availability scenarios shown in Table 4. The model was used to compute a number of performance measures, including the number of tasks completed, cumulative task state estimation error, delay per completed task, and the asset utilization rates under the three coordination structures for the seven scenarios.

Scenario (Assets)	Assets
S0 (18)	4UAV, 2MSPF, 4RECC, 2RHIB, 4XH30, 2SH60
S1 (17)	3UAV(-1)[*], 2MSPF, 4RECC, 2RHIB, 4XH30, 2SH60
S2 (16)	3UAV(-1)[*], 2MSPF, 3RECC(-1), 2RHIB, 4XH30, 2SH60
S3 (15)	3UAV(-1)[*], 2MSPF, 2RECC(-2), 2RHIB, 4XH30, 2SH60
S4 (14)	3UAV(-1)[*], 2MSPF, 2RECC(-2), 2RHIB, 3XH30(-1)[**], 2SH60
S5 (13)	3UAV(-1)[*], 2MSPF, 2RECC(-2), 1RHIB(-1), 3XH30(-1) [**], 2SH60
S6 (11)	2UAV(-2)[*,**], 2MSPF(-1), 2RECC(-2), 1RHIB(-1), 3XH30(-1) [**], 2SH60

Table.4. Asset Availability Scenarios

Figs. 5-8 display these metrics. The results suggest that the ISR commander structure has better performance than the other two ISR structures. In addition, as the number of assets decreases, the state estimation error per task and the delay per task in all ISR structures tend to increase. Another point to note is that team performance under asset package scenario S_6 degrades significantly compared to the other scenarios, as shown in the Fig. 5-8. We can infer the reason from Fig. 7 where the waiting delay significantly increases. This implies that the number of ISR assets may not be adequate to handle the mission scenario, because the workload of each ISR asset tends to increase as shown in Fig. 8. This also shows that UAV and MSPF are bottleneck resources. Experiment 11 was run under asset scenario S_5 at NPS. After action reviews from human teams revealed that UAVs and MSPFs are indeed the bottleneck resources.

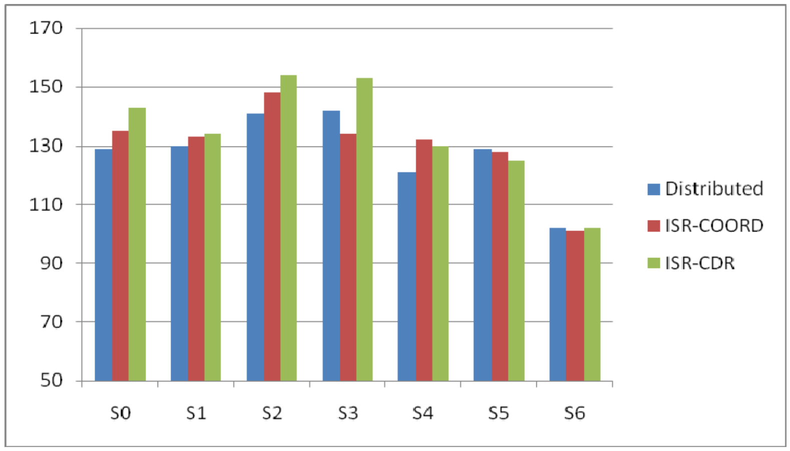

Figure.5. Number of Completed tasks for the three ISR coordination structures

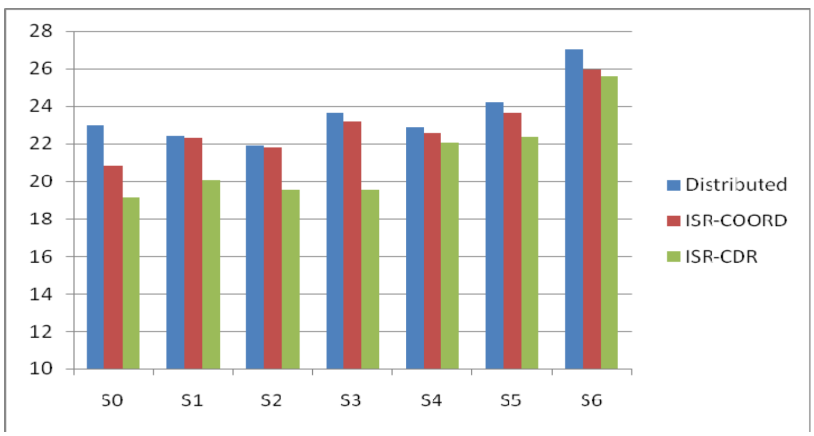

Figure.6. Estimation error per completed tasks of three ISR organizational structures

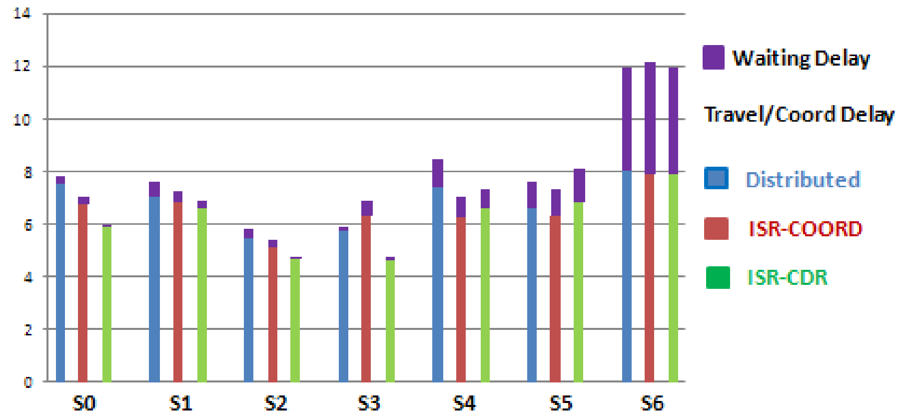

Figure.7. Delay per completed task of three ISR organizational structures

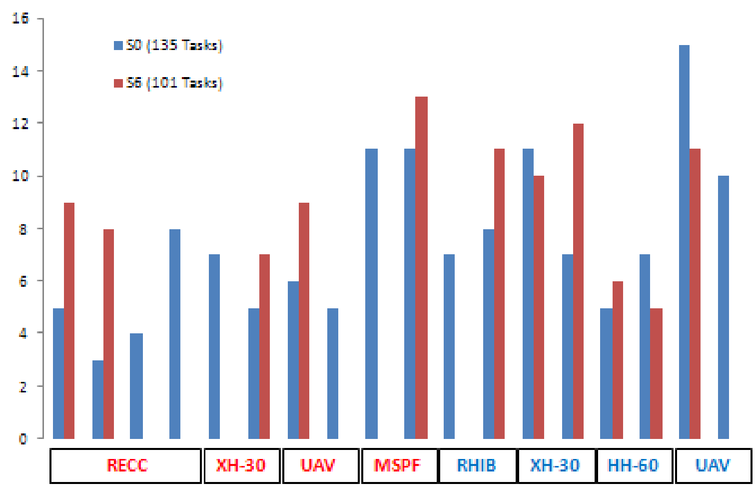

Figure.8. Asset usage rate for scenario S0 and S6

6. Conclusions

This paper formulated the sensor scheduling problem using factorial HMM formalisms for various ISR coordination structures. The scheduling model was applied to a realistic mission scenario to analyze different assert availability scenarios. The performance of various ISR coordination mechanisms was evaluated by comparing the state estimation error cost, as well as travel, waiting and assignment delays. The analytic model developed in this paper provides a quantitative framework to systematically analyze a number of organizational issues. For example, our modeling framework can be used to study:

- Does the structure facilitate adaptation in the face of novel situations?
- How fast does the structure respond to major disturbances (e.g., asset breakdowns, DM failures, changes in task workload)?
- How does information flow in the structure?
- How does absence of global information in the self-synchronizing structure impact its performance?
- How do coordination protocols impact a structure's decision processes?
- Under what conditions does a structure exhibit chaotic behaviors?

These and other issues are under continuing investigation.

7. Reference

[1] L. E. Baum, T. Petrie, G. Soules and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. of Math. Stat*, vol. 41, pp. 164-171, 1970.

[2] D. P. Bertsekas, "An auction algorithm for shortest paths," *SIAM Journal on Optimization*, vol. 1, pp. 425-447, 1991.

[3] D. P. Bertsekas, "Auction algorithms for network flow problems: A tutorial introduction," *Computational Optimization and Application*, vol. 1, pp. 7-66, 1992.

[4] D. P. Bertsekas, "Network optimization: continuous and discrete models," *Athena Scientific*, Belmont MA, 1998.

[5] M. Brand, N. Oliver and A. Pentland, "Coupled hidden Markov models for complex action recognition," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, pp. 994-999, 1997.

[6] S. Chhetri, "Sensor scheduling and efficient algorithm Implementation for Target Tracking," *Ph. D. Dissertation*, Arizona State University, 2006.

[7] M. Chu, H. Haussecker and F. Zhao, "Scalable information-driven sensor querying and routing for Ad Hoc heterogeneous sensor networks," *International Journal of High Performance Computing Applications*, vol. 16, pp. 293-313, 2002.

[8] I. Cox and M. Miller, "On finding ranked assignments with application to multi-target tracking and motion correspondence," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, pp. 486-489, 1995.

[9] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 296-271, 1959.

[10] S. Fine, Y. Singer and N. Tishby, "The hierarchical hidden Markov model: Analysis and Applications," *Machine Learning*, vol. 32, pp. 41-62, 1997.

[11] Z. Ghahramani and M. I. Jordan, "Factorial hidden Markov models," *Machine Learning*, vol. 29, pp. 245-273, 1997.

[12] S. G. Hutchins, S. A. Weil, D. L. Kleinman, S. P. Hocevar, W. G. Kemple, K. Pfeiffer, D. Kennedy, H. Oonk, G. Averett and E. E. Entin, "An investigation of ISR coordination and information presentation strategies to support expeditionary strike groups," 12th *ICCRTS*, Newport, RI, 2007.

[13] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol.38, pp. 325-340, 1987.

[14] V. Krishnamurthy, "Algorithms for optimal scheduling and management of hidden Markov model sensors," *IEEE Transactions on Signal Processing*, vol. 50, pp. 1382-1397, 2002.

[15] A. Logothetis and A. Isaksson, "On sensor scheduling via information theoretic criteria," *IEEE American Control Conference*, San Diego, CA, vol. 4, pp. 2402-2406, 1999.

[16] L. Meier, J. Perschon and R. M. Dressier, "Optimal control of measurement subsystems," *IEEE Transactions on Automatic Control*, vol. 12, pp. 528-536, 1967.

[17] K. G. Murty, "Algorithm for ranking all the assignments in order of increasing cost," *Operations Research*, vol. 16, pp. 682-687, 1968.

[18] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257-286, 1989.

[19] M. Shakeri, K. R. Pattipati and D. L. Kleinman, "Optimal measurement scheduling for state estimation," *IEEE Transactions on Aerospace and Electronic System*, vol. 31, pp. 716-729, 1995.

[20] S. Singh, W. Donat, H. Tu, K. Pattipati and P. Willett, "Anomaly detection via feature-aided tracking and hidden Markov models," *IEEE Aerospace Conference*, Big Sky, Montana, pp. 1-18, 2007.

[21] S. S. Singh, N. Kantas, B. Vo, A. Doucet and R. J. Evans, "Simulation based optimal sensor scheduling with application to observer trajectory planning," *Technical Report*, Cambridge University, 2005.

[22] He. Ying and K. P. Chong, "Sensor scheduling for target tracking in sensor networks," *IEEE Conference on Decision and Control*, vol. 1, pp. 743-748, 2004.

[23] F. Zhao, J. Shin and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine*, vol. 19, pp. 61-72, 2002.

[24] W. An, S. Singh, K. R. Pattipati S. S. Gokhale and D. L. Kleinman, "Dynamic scheduling of multiple hidden Markov model-based sensors," *Journal of advances in Information fusion*, vol. 2, No. 2, June 2008.

[25] D.C. McFarlane, and S. Bussmann. Developments in Holonic Production Planning and Control. *Int. Journal of Production Planning and Control*, vol. 11, No. 6, pp. 522-536, 2000.

[26] D. Stark, "Heterarchy: Distributing Intelligence and Organizing Diversity", *The Biology of Business: Decoding the Natural Laws of Enterprise*, pp. 153-179 San Francisco: Jossey-Bass. 1999