Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Robust Tasking of Airborne Sensing Nodes for Network Availability ⋆

N. Eva Wu and Yan Guo, and Matthew C. Ruschmann

Department of Electrical and Computer Engineering, Binghamton University Binghamton, NY 13902-6000 USA (Tel: 607-777-4375; e-mail: evawu,Yan.Guo,Matthew.Ruschmann @binghamton.edu).

Abstract: This paper considers tasking a finite number of cooperative agents to randomly emerging targets for their removal. Faults occur when some agents engaged in a mission are expired. Agents are subject to threat at a level determined by the number of targets present. On the other hand, the rate at which a target is removed depends on the number of cooperative agents assigned to it. Faults effectively change the network architecture and therefore degrade the network performance. Designs of control policies that determine the number of agents assigned are based on the network life when expired agents cannot be replenished, and on the network availability when expired agents are replenished at a certain rate. Tasking process is described by a discrete event system in the form of a queuing network, where agents are servers and targets are customers. Optimal policies are determined by solving a Markov decision problem. To facilitate the reader's understanding of the motivation, and of the problem, the agents are specialized to networked pairs of airborne sensors that are tasked to locate non-cooperating microwave transmitters as targets.

1. INTRODUCTION

This paper formulates and solves a new problem of tasking a finite number of cooperative agents to randomly emerging targets for their removal. It focuses on enhancing the network performance in the face of expiration of its agents. The resulting network is said to be fault-tolerant. When the agents considered are networked airborne sensors, the mobility and a multiplicity of the sensing nodes make fault-tolerance possible. Fault-tolerant tasking in this paper is achieved by implementing operation policies optimized for network availability.

Control of networked multiple agents has been an intensively discussed topic recently in the controls literature Antsaklis et al. [2007]. Oh et al. [2007], for example, describes a pursuit evasion game, where mobile agents are to chase and capture multiple moving targets in a minimum amount of time, and a network of stationary sensors serves to help enhance the target observability in the game.

With unmanned aerial vehicles (UAVs) replacing stationary networks and manned vehicles, significant improvements in network performance can be expected. Networking in a hostile environment, however, poses new challenges. Data exchange inherent to a networked operation and prolonged mission time due to poor execution expose the otherwise passive location sensors, thus increase the likelihood of the vehicles being destroyed.

Examine a situation where the motion of two unmanned aerial vehicles (UAV) and a hostile radar lie within a plane, as illustrated in Fig. 1. Let us assume that the two

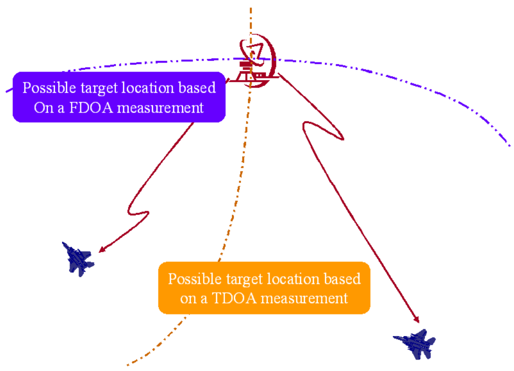


Fig. 1. Transmitter location using a TDOA measurement and an FDOA measurement by two airborne sensors.

vehicles are equipped to acquire both the time difference of arrival and the frequency difference of arrival of the radar signal Ho et al. [1997]. The sensors are mounted on the vehicles, and thus are subject to the same speed and curvature constraints as that of the vehicles. The sensors are passive nodes when acquiring data from the transmitter, but become active when exchanging data between them in order to provide a location estimate.

It can be seen from Figure 1 that at least two sensor carrying vehicles, which make both a TDOA measurement and an FDOA measurement, are needed in a 2-dimensional setting to locate the target. A noiseless measurement of time difference of arrival by a pair of sensors on the two vehicles is given by

$$s_T = \frac{1}{c}[\sqrt{(x_2-x_e)^2+(y_2-y_e)^2} - \sqrt{(x_1-x_e)^2+(y_1-y_e)^2}], \tag{1}$$

⋆ This work was supported in part by the US Air Force Office of Scientific Research under Grants FA9550-06-0456 and in part by the US Air Force Research Laboratory under Contract FA8750-07-1-0172.

and a noiseless measurement of frequency difference of arrival by the same pair of sensors is given by

$$s_F = \frac{f_e}{c}\Big[\frac{(x_2 - x_e)u_2 + (y_2 - y_e)v_2}{\sqrt{(x_2 - x_e)^2 + (y_2 - y_e)^2}} - \frac{(x_1 - x_e)u_1 + (y_1 - y_e)v_1}{\sqrt{(x_1 - x_e)^2 + (y_1 - y_e)^2}}\Big], \tag{2}$$

where (x_e, y_e) is the transmitter location to be estimated, (x_1, y_1) and (x_2, y_2) are the positions of the two vehicles, respectively, (u_1, v_1) and (u_2, v_2) are the velocities of the vehicles, f_e is the carrier frequency of the transmitted signal, and c is the speed of light.

Since the measurements are always noisy, multiple measurements are needed for an accurate location estimation of the emitter. Such measurements can be distributed temporally along the trajectories of motion of a pair of sensors, or spatially over multiple pairs of sensors, or both. Measurements made by multiple pairs of sensors, which form a network, offer greater degree of fault-tolerance, and greater potential for improved speed and accuracy in target location. The reader is referred to Ho et al. [1997] and Torrieri [1984] for more detailed discussion on methods for location estimation and accuracy analysis.

A tasking problem that is specific to this application refers to that of allocating a finite number of sensor pairs to randomly emerging microwave transmitters to maximize the network availability. A tasking policy that is too greedy tends to exhaust resources before the arrival of unanticipated radars, whereas a tasking policy that is too conservative tends to lengthen the exposure of the sensor carrying vehicles. Tasking is treated as a server allocation problem of a queuing network. Optimal policies are sought as the solutions of Markov decision problems.

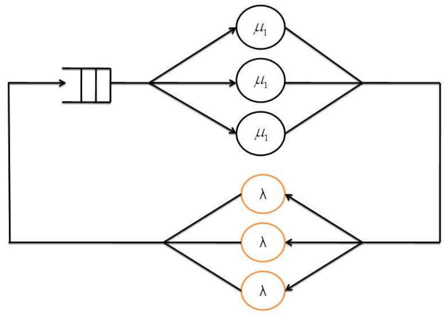
The paper is organized as follows. Section 2 modeling the tasking process for a small scale sensor network. Section 3 designs supervisory control policies for optimal tasking for the cases where lost sensors can and cannot be replenished by solving appropriate Markov decision problems. Section 4 evaluates the network performance in terms of expected network life and steady-state availability. Section 5 concludes the paper.

2. MODELING OF TASKING PROCESS

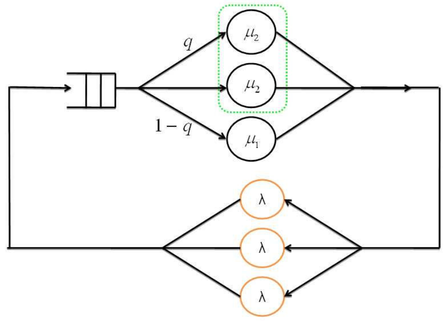
An optimized tasking is one that maximizes the expected life of the network where the lost airborne sensors cannot be replenished, or one that maximizes the expected steady-state availability of the network where the lost sensors can be replenished. In this study fault-tolerance refers to the network's tolerance to vehicle loss.

Figure 2 is a queuing network model of a six-sensor, finite target population tasking process, where each server represents a pair of sensors capable of independently locating a target to a certain accuracy in the absence other pairs, and a customer is a randomly emerging target.

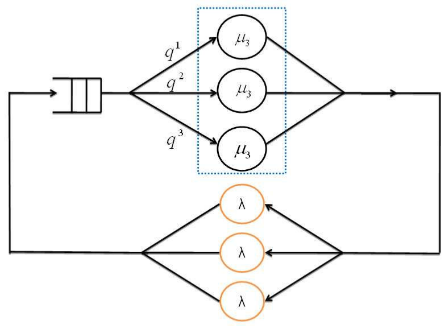
Each customer resides in the queue or a server is regarded as a detected target which is being or to be served by one or more servers or sensor-pairs. Service is complete as soon as the target location is determined to a required accuracy. A



(a) One sensor-pair/target allocation



(b) Two sensor-pair/target allocation



(c) Three sensor-pair/target allocation

Fig. 2. A queuing network model of a three sensor-pair, finite target population airborne sensor network.

target is then considered removed. A sensor-pair allocated to a target is tied to the target until its service is complete, or the life of the sensor-pair is terminated, whichever comes first. The three delay elements of average delay $1/\lambda$ imply that target population is limited by three at any given time. A new target is generated or replenished at a delay element with rate λ upon the service completion of a target at one or multiple servers.

An supervisory control policy determines whether to allocate one, or two, or three pairs of sensors to each reported target, with a corresponding mean service time of $1/\mu_1, (\geq)1/\mu_2, (\geq)1/\mu_3$, respectively, where μ_i denotes the service rate of committing i pairs of sensors to a target. Given the sensing mechanism, the mean service time by a single pair of sensors is in the range of seconds to tens of seconds, dominated by the time required to adjust sensor positions and velocities for continued data collection, exchange, and processing needed for target location to a required accuracy. Each sensor-pair has a mean lifetime

$1/\nu_0 \geq 1/\nu_1 \geq 1/\nu_2 \geq 1/\nu_3$, depending on the threat level quantified by the number of targets present as indexed by the subscript. $1/\nu_0$ is the server life representing the expected natural endurance of a vehicle, which is "often an hour or so at best" Samad et al. [2007]. It also reflects sensor lives affected by undetected targets. The network is said to have expired when there is no longer a single surviving sensor-pair.

Tasking process model is built in this study with the premise that event life distributions have been established for the process of target arrival ($\exp(\lambda) \equiv 1 - e^{-\lambda t}$), the process of target location ($\exp(\mu_i)$), the process of loss of a sensor-pair ($\exp(\nu_i)$), and the process of sensor replenishment ($\exp(\omega)$) when new sensor carrying vehicles are supplied for an expired network. Since all event lives are assumed to be exponentially distributed, the database unit can be conveniently modeled as a Markov chain specified by a state space $\mathcal{X}$, an initial state probability mass function (pmf) $\pi_x(0)$, and a set of state transition rates λ, μ_i, ν_i, and ω.

A state name is coded with a 4-digit number indicative of the number of targets present and the network configuration. A valid state representation is given by QS, where queue length $Q \in \{0, 1, 2, 3\}$, and server state $S = (i, j, k)$, with $i \in \{0, 1, 2, 3, 4\}$, and $j \in \{0, 1, 2, 3, 4, 5\}$, and $k \in \{0, 1, 3, 4, 5\}$. A server state "0", represented by the value of i, or j, or k, indicates an idle sensor-pair, a "1" indicates a target's being located by one server (or one sensor-pair), a "2" and a "3" indicate that a target's being located by two and three cooperating servers, respectively, a "4" indicates a lost server, and a "5" indicates that the lost server has been tied to another server in serving a target. The expired network requires 4 distinct states to memorize the possible queue length distributions. Note that this state specification has assumed homogeneous sensor-pairs and homogeneous targets, and has made use of the symmetry which results in 37 states. A set of alternative state names are assigned from $\mathcal{X} = \{1, 2, ..., 37\}$ with 0000 mapped to $x = 1$ and the network expiration states mapped to $x = 34, 35, 36$, and 37.

Events that trigger the transitions and the corresponding transition rates are given as follows. An emerging target enters with rate $(3 - Q) \times \lambda$. A target is located by one sensor-pair with rate μ_1, and $i (> 1)$ cooperative sensor-pairs with rate μ_i. In the latter case, the i servers are configured as a single hyper-exponential server with i parallel stages Cassandras et al. [1999]. An arriving target enters any one of the servers with probability $1/i$, which has a service time distribution $\exp(\mu_i)$. When service is completed, the target is removed, while no new target can enter service when the hyper-exponential server is busy. The service time distribution of a hyper-exponential server is

$$F_i(t) = \sum_{j=1}^{i} \frac{1}{i}(1 - e^{-\mu_i t}) = 1 - e^{-\mu_i t}, \tag{3}$$

which assumes homogeneity of the servers. Loss of a sensor-pair occurs at rate $m\nu_0$ when the network is idle with m remaining sensor-pairs, $m\nu_1$ when one target emerges, and $m\nu_i$ when $i (> 1)$ targets emerge. Replenishment process begins at the network expiration with rate ω. If one of the sensor-pairs is lost while locating a target with other sensor-pairs, the surviving sensor-pairs continue to locate the target at the same rate. This is a simple way to memorize the service already being provided without resorting to a more complex model.

Let $X \in \mathcal{X}$ denote the random state variable at time t. The set of state transition functions is given by

$$p_{i,j}(t) \equiv P[X(t) = j | X(0) = i], \quad i, j = 1, 2, ..., 37. \tag{4}$$

The continuous-time Markov chain can be solved from the forward Chapman-Kolmogorov equation Cassandras et al. [1999],Kao [1997]

$$\dot{P}(t) = P(t)Q(u(x)), \quad P(0) = I, \quad P(t) = [p_{i,j}(t)] \tag{5}$$

and $Q(u(x))$ is called an infinitesimal generator or a rate transition matrix whose $(i, j)^{th}$ entry is given by the rate associated with the transition from current state i to next state j. Table 1 summarizes information contained in transition rate matrix $Q(u(x))$. Control variable $u(x)$ will be discussed shortly. State probability mass function at time t

$$\pi(t) = [\pi_1(t) \ \pi_2(t) \ \cdots \ \pi_{37}(t)], \quad t \geq 0 \tag{6}$$

can be solved from

$$\dot{\pi}(t) = \pi(t)Q(u(x)), \quad \text{given } \pi(t = 0). \tag{7}$$

A Markov chain for the tasking process of Figure 2 has been established so far. Since transition rate matrix Q is dependent on control actions, the state transition functions $p_{i,j}(t)$ are being controlled, and so are the state probabilities. Rate transition matrix Q is given in the form of a table in Table 1.

$\mathcal{X}$	Coded $\mathcal{X}$				Arrivals						Completions						Losses & Replenishements					
	Q	S1	S2	S3	$\mathcal{X}'$	Rate	$\mathcal{X}'$	Rate	$\mathcal{X}'$	Rate	$\mathcal{X}'$	Rate	$\mathcal{X}'$	Rate	$\mathcal{X}'$	Rate	$\mathcal{X}'$	Rate	$\mathcal{X}'$	Rate	$\mathcal{X}'$	Rate
1	0	0	0	0	4	$3*\lambda*u1$	7	$3*\lambda*u2$	9	$3*\lambda*u3$	x	x	x	x	x	x	2	$3*v_0$	x	x	x	x
2	0	4	0	0	5	$3*\lambda*u1$	8	$3*\lambda*u2$	x	x	x	x	x	x	x	x	3	$2*v_0$	x	x	x	x
3	0	4	4	0	6	$3*\lambda$	x	x	x	x	x	x	x	x	x	x	34	v_0	x	x	x	x
4	1	1	0	0	10	$2*\lambda*u1$	13	$2*\lambda*u2$	x	x	0	μ_1	x	x	x	x	5	$3*v_1$	x	x	x	x
5	1	1	4	0	11	$2*\lambda$	x	x	x	x	2	μ_1	x	x	x	x	6	$2*v_1$	x	x	x	x
6	1	1	4	4	12	$2*\lambda$	x	x	x	x	3	μ_1	x	x	x	x	35	v_1	x	x	x	x
7	1	2	2	0	13	$2*\lambda$	x	x	x	x	0	μ_2	x	x	x	x	22	$2*v_1*u1$	8	v_1*u2	x	x
8	1	2	2	4	14	$2*\lambda$	x	x	x	x	2	μ_2	x	x	x	x	23	$2*v_1$	x	x	x	x
9	1	3	3	3	15	$2*\lambda$	x	x	x	x	0	μ_3	x	x	x	x	26	$3*v_1$	x	x	x	x
10	2	1	1	0	16	λ	x	x	x	x	4	μ_1	x	x	x	x	11	$3*v_2$	x	x	x	x
11	2	1	1	4	17	λ	x	x	x	x	5	μ_1	x	x	x	x	12	$2*v_2$	x	x	x	x
12	2	1	4	4	18	λ	x	x	x	x	6	μ_1	x	x	x	x	36	v_2	x	x	x	x
13	2	2	2	1	19	λ	x	x	x	x	7	μ_1*u1	4	μ_2*u2	x	x	24	$2*v_2*u1$	14	v_2*u2	x	x
14	2	2	2	4	20	λ	x	x	x	x	5	μ_2*u1	8	μ_2*u2	x	x	30	$2*v_2$	x	x	x	x
15	2	3	3	3	21	λ	x	x	x	x	4	μ_3*u1	7	μ_3*u2	9	μ_3*u3	28	$3*v_2$	x	x	x	x
16	3	1	1	1	x	x	x	x	x	x	10	μ_1	x	x	x	x	17	$3*v_3$	x	x	x	x
17	3	1	1	4	x	x	x	x	x	x	11	μ_1	x	x	x	x	18	$2*v_3$	x	x	x	x
18	3	1	4	4	x	x	x	x	x	x	12	μ_1	x	x	x	x	37	v_3	x	x	x	x
19	3	2	2	1	x	x	x	x	x	x	10	μ_2*u1	13	μ_2*u2	x	x	25	$2*v_3*u1$	20	v_3*u2	x	x
20	3	2	2	4	x	x	x	x	x	x	11	μ_2*u1	14	μ_2*u2	x	x	33	$2*v_3$	x	x	x	x
21	3	3	3	3	x	x	x	x	x	x	10	μ_3*u1	13	μ_3*u2	15	μ_3*u3	31	$3*v_3$	x	x	x	x
22	1	2	5	0	24	$2*\lambda$	x	x	x	x	2	μ_2	x	x	x	x	6	v_1*u1	23	v_1*u2	x	x
23	1	2	5	4	30	$2*\lambda$	x	x	x	x	3	μ_2	x	x	x	x	35	v_1	x	x	x	x
24	2	2	5	1	25	λ	x	x	x	x	22	μ_1*u1	5	μ_2*u2	x	x	12	v_2*u1	30	v_2*u2	x	x
25	3	2	5	1	x	x	x	x	x	x	24	μ_1*u1	11	μ_2*u2	x	x	18	v_3*u1	33	v_3*u2	x	x
26	1	3	3	5	28	$2*\lambda$	x	x	x	x	2	μ_3	x	x	x	x	27	$2*v_1$	x	x	x	x
27	1	3	5	5	29	$2*\lambda$	x	x	x	x	3	μ_3	x	x	x	x	35	v_1	x	x	x	x
28	2	3	3	5	31	λ	x	x	x	x	5	μ_3*u1	8	μ_3*u2	x	x	29	$2*v_2$	x	x	x	x
29	2	3	5	5	32	λ	x	x	x	x	6	μ_3	x	x	x	x	36	v_2	x	x	x	x
30	2	2	5	4	33	λ	x	x	x	x	6	μ_2	x	x	x	x	36	v_2	x	x	x	x
31	3	3	3	5	x	x	x	x	x	x	11	μ_3*u1	14	μ_3*u2	x	x	32	$2*v_3$	x	x	x	x
32	3	3	5	5	x	x	x	x	x	x	12	μ_3	x	x	x	x	37	v_3	x	x	x	x
33	3	2	5	4	x	x	x	x	x	x	12	μ_2	x	x	x	x	37	v_3	x	x	x	x
34	0	4	4	4	x	x	x	x	x	x	x	x	x	x	x	x	1	w	x	x	x	x
35	1	4	4	4	x	x	x	x	x	x	x	x	x	x	x	x	4	w*u1	7	w*u2	9	w*u3
36	2	4	4	4	x	x	x	x	x	x	x	x	x	x	x	x	10	w*u1	13	w*u2	15	w*u3
37	3	4	4	4	x	x	x	x	x	x	x	x	x	x	x	x	16	w*u1	19	w*u2	21	w*u3

u1=I_1, u2=I_2, u3=I_3, $\mathcal{X}$: current state, $\mathcal{X}'$: next state

Fig. 3. Transitions and transition rates of the tasking process

3. DESIGN OF SUPERVISORY CONTROL POLICY

Several possible supervisory control policies associated with tasking are examined. An aggressive policy allocates as many available sensor-pairs to as many targets present; A greedy policy allocates all available sensor-pairs to one target at a time; A conservative policy always allocates only one sensor-pair to every target present to reserve assets in anticipation of new targets. In addition, four optimal policies have been attempted to minimize the cost of sensor loss, threat level, unattended targets, and time needed to replenish upon network expiration, respectively.

The optimal policies are obtained by solving Markov decision problems of appropriate penalty functions. A discrete-time Markov chain model suitable for this purpose can be derived under each cost criterion by the application of a uniformization procedure Kao [1997]

$$\pi(t_{k+1}) = \pi(t_k)[I + \frac{1}{\rho}Q(u(x_k))], \tag{8}$$

where the uniform rate ρ is greater than any total outgoing transition rates at any states of the original continuous-time Markov chain (7).

Each Markov decision problem considered in this paper assumes that a cost, denoted by $C(i, u)$, is incurred at every state transition, where i is the state entered and u is a control action selected from a set of admissible actions Cassandras et al. [1999], Bertsekas [1995]. A solution amounts to determining a stationary policy $\pi = \{u(x_k),\ k = 0, 1, \cdots\}$ that minimizes the following expected total discounted cost

$$V_\pi(x_0) = E_\pi \sum_{k=0}^{\infty} \alpha^k C(X_k, u_k) \tag{9}$$

where $0 < \alpha < 1$ is a discount factor. $C(X_k, u_k)$ in each of the four Markov decision problems takes the form of total number of lost sensor-pairs, ω^{-1} at the state of network expiration, ν_i at the state where it is the server loss, and Q at the state where it is the queue length, respectively.

Let $X_k \in \{1, 2, \cdots, 37\}$ denote the random state variable at $t_k = k/\rho$ in the discrete time Markov chain. Control action

$$u(x_k) = \begin{cases} 1, & \text{allocate one sensor-pair to a target} \\ 2, & \text{allocate two sensor-pairs to a target} \\ 3, & \text{allocate three sensor-pairs to a target} \end{cases} \tag{10}$$

Note that the indicator functions in Table 1 are defined follows

$$I_i = \begin{cases} 1, & u = i \\ 0, & \text{otherwise} \end{cases}, i = 1, 2, 3. \tag{11}$$

It is known Cassandras et al. [1999], Bertsekas [1995] that under the condition $0 \leq C(j, u) < \infty$ for all j and all u that belongs to some finite admissible sets U_j, the minimum cost $V^*(i)$ satisfies the following optimality equation:

$$V(i) = \min_{u \in U_i} \left\{ C(i, u) + \alpha \sum_{j=1}^{37} p_{i,j} V(j) \right\},\ u \in U_i, \tag{12}$$

$i = 1, \cdots, 37$, where $p_{i,j}$ is the $(i,j)^{th}$ entry of $I + \frac{1}{\rho}Q(u(x_k))$.

The solution to (12) can be obtained via linear programming Boyd et al. [2004], Bertsekas [1995]. In this case, the set of optimality equations is turned into a set of affine constraints on the set of optimization variables $\{V(i)\}$, and the problem can be formally stated as follows.

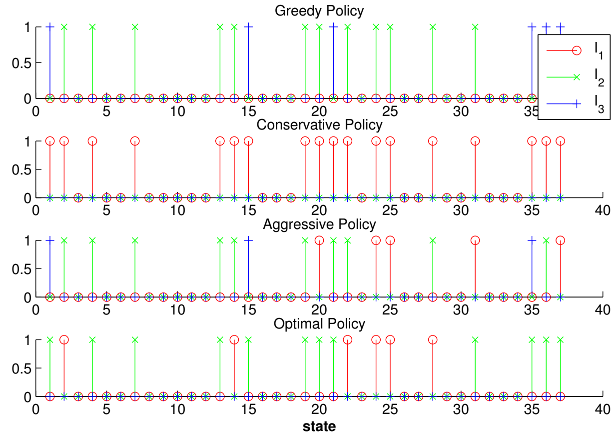


Fig. 4. Control policy indicators. red: one sever/target; green: two servers/target; blue: three servers/target

$$\text{Maximize } V(1) + V(2) + \cdots + V(36) + V(37) \tag{13}$$

$$\text{Subject to } V(i) \geq 0,\ \ i \in \mathcal{X} = \{1, \cdots, 37\} \tag{14}$$

$$V(i) \leq [C(i, u) + \alpha \sum_j p_{i,j} V(j)]\ |_u, \tag{15}$$

$\forall u \in U_i,\ \ i \in \mathcal{X}$.

In the tasking process considered, U_j is nonempty only at state j = 1, 2, 4, 7, 13, 14, 15, 19, 20, 21, 22, 24, 25, 28, 31, 35, 36, 37. Therefore, (15) leads to 99 affine inequality constraints. This problem is readily solvable by `linprog` in MATLAB's Optimization Toolbox MathWorks [2006]. The active constraints are checked with a MATLAB script to determine the optimal control policy.

Figure 4 shows an example of 4 stationary control policies depicted in terms of indicator functions, as defined in (11), of the state $x \in \mathcal{X}$. It can be seen that the optimal policy takes into consideration of anticipated targets more than the aggressive policy, but is much more aggressive in terms of use of resources than the conservative policy. Among the four optimal policies solved, only the policy derived under the least sensor loss is plotted (bottom), which will be shown shortly to outperform other three optimal policies in terms of both MTTNE and availability. The minimum queue length policy coincides with the greedy policy, as expected. The other two optimal policies make less aggressive use of resources than the optimal policy shown. The least sensor loss policy will be called the optimal policy from this point on.

The control policies are robust with respect to the range of parameter variations that have been examined: $\nu_1 \in [0.001, 0.01]$ $1/sec.$, and $\lambda \in [0.001, 0.1]$ $1/sec.$ The optimal policy is calculated at $\alpha = 0.0792$. All optimal policies drift slightly toward more conservative actions (using fewer resources) as the discount factor α increases, which is consistent with the outcome of the longer term policy making. Because of the finite target population setup, the effect of increasing the target arrival rate is not fully reflective of the target traffic intensity. Simulations with *MATLAB SimEvets* MathWorks [2006] are being performed without limiting the target population.

4. PERFORMANCE ANALYSIS

The seven policies developed in Section 3 are compared against one another with respect to two common measures of fault-tolerance: mean time to network expiration (MMTNE) and availability. These have been used in Wu et al. [2005] in a similar fashion as performance measures of a database unit.

When no replenishment is provided, the network life eventually expires when all sensor-pairs are lost. This occurs when the network enters one of its absorbing states at 34, 35, 36, or 37. Decompose the state probability vector

$$\pi(t) = [\underbrace{\pi_\tau(t)}_{1\times 33} \ \underbrace{\pi_\alpha(t)}_{1\times 4}] \tag{16}$$

where vector $\pi_\tau(t)$ contains transient state probabilities, and $\pi_\alpha(t)$ contains absorbing state probabilities. Decomposing the rate transition matrix Q accordingly yields

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ 0 & 0 \end{bmatrix} \tag{17}$$

From (17), it can be determined that mean time to network expiration is given by

$$\mathrm{MTTNE} = -\pi_\tau(0)Q_{11}^{-1}1_\tau, \ \ 1_\tau = [\underbrace{1 \ \cdots \ 1}_{1\times 33}]^T \tag{18}$$

Suppose as soon as the network expires, a replenishment process starts. Suppose with a rate ω the airborne sensors are replenished, and at the completion of the replenishment, the tasking process immediately resumes. In this case, the Markov chain (7) becomes irreducible, and a unique steady-state distribution exists Kao [1997]. The steady-state availability, which can be roughly thought of as the fraction of time the network has at least one surviving pair of sensors, is computed by

$$A_{net} = 1 - \pi_F(\infty), \tag{19}$$

where $\pi_F(\infty) = \pi_{34}(\infty) + \pi_{35}(\infty) + \pi_{36}(\infty) + \pi_{37}(\infty)$, the sum of state probabilities associated with network expiration, which can be determined by solving

$$\pi(\infty)Q = 0, \text{ and } \sum_{x=1}^{37} \pi_x(\infty) = 1. \tag{20}$$

A slightly different notion of availability is also examined, where the network is considered unavailable as long as unattended targets are present. In this case, the network availability is given by

$$\begin{aligned} A_{tgt} = & \\ & \sum_{i=1}^{11} \pi_i(\infty) + \pi_{13}(\infty) + \pi_{16}(\infty) + \pi_2(\infty) \\ & + \pi_{23}(\infty) + \pi_{24}(\infty) + \pi_{26}(\infty) + \pi_{27}(\infty). \end{aligned} \tag{21}$$

Mean time to network expiration is plotted in Figure 5 against target arrival rate with two sets of sensor loss rates as parameters at $\alpha = 0.0792$. It shows that compromise that optimal policy makes between being too greedy and too conservative enhances the network life consistently for all parameters values considered.

In Figure 6, availability is also plotted against target arrival rate with two sets of sensor loss rates as parameters

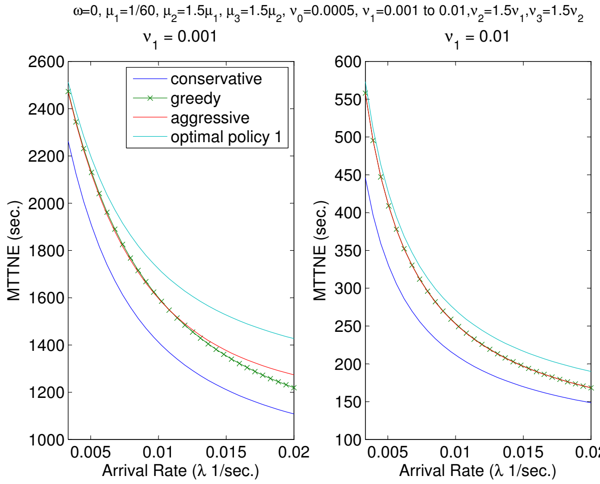


Fig. 5. Mean time to network expiration as a function of target arrival rate with two sets of sensor loss rates as parameters.

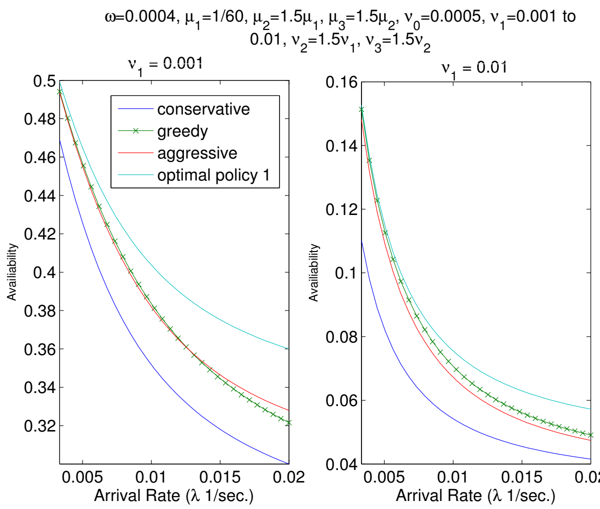


Fig. 6. Network availability with at least one surviving server.

at discount factor $\alpha = 0.0792$. The observations from the MTTNE apply in terms of the gain the optimal policy offers. It is noted that the availability is low. This is because of the low replenishment rate used in the computation, which corresponds to an expected time of more than 40 minutes to reestablish the lost network.

It is expected that the network availability defined as the probability that all targets is lower than the availability defined as the probability that there is at least one surviving server. On the other hand, the dependence of both notions of availability on the sensor loss rate and on the target arrival rate stays the same. Figure 7 shows the plot of the two availabilities against target arrival rate with two sets of sensor loss rates ν_i as parameters under the optimal policy.

Extensive simulations using MathWorks [2006] is being conducted to generate a more complete picture of the network performance in response to control policies for a larger size of networks. The results will be reported elsewhere.

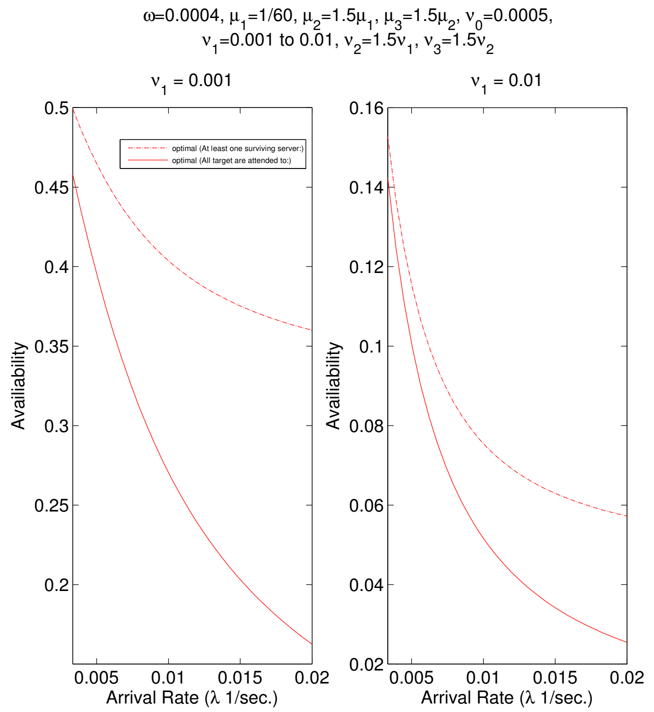


Fig. 7. Comparison between availability with at least one surviving server (dash) and availability with all targets being attended (solid).

5. CONCLUSIONS

This paper sought to determine supervisory control policies that best configure an airborne location sensor network to provide a high degree of guarantee of prompt completion of coordinated data acquisition and processing missions in the face of loss of vehicles. Use of redundancy and dynamic allocation of participating sensors was the key enablers.

The paper presented a queuing network approach to optimal sensor-pair assignment to locate detected targets for a small scale airborne sensor network, where use of redundancy is balanced with avoiding more vehicle exposure.

A number of related issues are being investigated. In addition to loss of sensors, degradation of network performance can also be the result of broken communication links. Such incidents are modeled as intermittent faults of the servers in queuing networks. The effects of such faults will be studied using discrete event simulations, which will also examine possible emergent phenomenon of larger networks, and non-homogeneous sensors and targets.

A highly relevant task is to solve a guidance and then a control problem of the vehicles. Guidance problem Huang et al. [2008] refers to that of establishing a criterion and deriving a set of desired vehicle trajectories under the criterion that the airborne sensors are expected to follow to expedite the target location estimation to a required accuracy. It is known that the quality of acquired data by the airborne sensors depends highly on both the network architecture which is determined by the number of sensor-pairs, and the *states* of all participating sensors relative to the target and to one another. A guidance principle based on the entropy Cover et al. [1991] of the noise distribution of the sensed signal has been established, based on which one seeks to adjust the states of the sensors to the most suitable positions and velocities for further data acquisition and processing.

Once the guidance principle is determined, feasible vehicle trajectories can be generated. Path following control can be performed with time coordination Kaminer et al. [2006] to achieve synchronous data acquisition by the sensors. Both the guidance and the control problems are being investigated, and will be reported separately in the near future.

REFERENCES

P. Antsaklis, and J. Baillieul, editors, Special issue on Technology of Networked Control Systems, *Proceedings of the IEEE*, vol.95, Issue 1, 2007.

D.P. Bertsekas, *Dynamic Programming and Optimal Control*, Volume 1 & Volume 2, Athena Scientific, 1995.

S.P. Boyd, and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

C.G. Cassandras, and S. Lafortune, *Introduction to Discrete Event Systems*, Kluwer, 1999.

T.M. Cover, and J.A. Thomas, *Elements of Information Theory*, John Wiley & Sons, 1991.

Ho, K.C., and Chan, Y.T., Geolocation of a known altitude object from TDOA and FDOA measurements, *IEEE Transactions on Aerospace and Electronic Systems,* vol.33, pp.770-782, 1997.

K. Huang, N.E. Wu, and M.L. Folwer, Optimal Guidance of Unmanned Aerial Vehicles for Emitter Location, *2008 Proc. IFAC World Congress*, 2008.

I. Kaminer, O. Yakimenko, A. Pascoal, and R. Ghabcheloo, Path Generation, Path Following and Coordinated Control for Time-Critical Missions of Multiple UAVs , *Proceedings of American Control Conference*, 2006.

E.P.C. Kao, *An Introduction to Stochastic Processes*, Duxbury Press, 1997.

MathWorks, *SimEvents User's Guide, For Use with Simulink*, The MathWorks, Inc., 2006.

MathWorks, *Optimization Toolbox User's Guide, For Use with MATLAB, Version 3*, The MathWorks, Inc., 2006.

S. Oh, L. Scheanto, P. Chen, and S. Sastry, Tracking and coordination of multiple agents using sensor networks: system design, algorithms, and experiments, *Proceedings of the IEEE*, vol.95, pp.234-254, 2007.

T. Samad, J.S. Bay, and D. Godbole, Network-centric systems for military operations in urban terrain: the role of UAVs, *Proceedings of the IEEE*, vol.95, pp.92-107, 2007.

D.J. Torrieri, Statistical theory of passive location systems, *IEEE Transactions on Aerospace and Electronic Systems,* vol.20, 1984.

N.E. Wu, J.M. Metzler, and M.H. Linderman, Supervisory Control of a Database Unit, *Proc. IEEE Conference on Decision and Control*, 2005.