

# Communication and Sensing Trade-Offs in Decentralized Mobile Sensor Networks: A Cross-Layer Design Approach

Yasamin Mostofi, Timothy H. Chung, Richard M. Murray and Joel W. Burdick  
California Institute of Technology  
Pasadena, CA 91125

**Abstract**—In this paper we characterize the impact of imperfect communication on the performance of a decentralized mobile sensor network. We first examine and demonstrate the trade-offs between communication and sensing objectives, by determining the optimal sensor configurations when introducing imperfect communication. We further illustrate the performance degradation caused by non-ideal communication links in a decentralized mobile sensor network. To address this, we propose a decentralized motion-planning algorithm that considers communication effects. The algorithm is a cross-layer design based on the proper interface of physical and application layers. Simulation results will show the performance improvement attained by utilizing this algorithm.

## I. INTRODUCTION

There has recently been considerable interest in sensor networks [1], [2]. Such networks have a wide range of applications such as environmental monitoring, surveillance and security, smart homes and factories, target tracking and military systems. A team of mobile agents equipped with sensing, wireless communication and local processing capabilities can further take advantage of the mobility to achieve sensor configurations that result in better networked sensing. To address and overcome technological challenges of such networks, different and non-conventional designs and strategies should be used. Such designs lie at the intersection of multiple disciplines like control, communication and computation, necessitating cross-disciplinary approaches.

Decentralized control of sensor motions is a key issue in such networks and has gotten considerable interest [3], [4], [5]. Most of the current research in this area, however, assumes ideal communication links, considering only sensing objectives. Communication plays a key role in the overall performance of such networks as each sensor relies on improving its estimate by processing the information received from others.

Considering the impact of communication channels on wireless estimation/control is an emerging area of research. Authors in [6], [7], [8] have looked at the impact of communication channels on Kalman filtering over a wireless link and the conditions required for stability. Authors in [9], [10], [11], [12], [13], [14] have looked at the impact of some aspects of a communication link like noise, quantization, fading, medium access and packet loss on wireless control of a mobile sensor. Authors in [15] have derived the minimum required rate for maintaining stability of control over a communication channel.

Considering the effect of communication on motion planning in a decentralized mobile network and addressing the introduced communication and sensing trade-offs, however, have not been studied before. It is the goal of this paper to investigate the relationship between sensing and communication in mobile sensor networks and demonstrate how the mobility can be utilized to achieve better overall performance. We show that without a proper interface of physical and application layers, the performance of the network can degrade considerably when considering non-ideal communication links. To improve the performance, we propose a decentralized

motion-planning algorithm that considers both sensing and communication objectives. The algorithm modifies the local information-processing and motion-planning functions of each sensor to account for communication channels. Finally, our simulation results show the performance improvement gained by using this algorithm.

## II. SYSTEM MODEL

Consider  $N$  mobile sensors that are cooperatively estimating the state of a target with the following dynamics:

$$x[k+1] = x[k] + w[k] \quad (1)$$

where  $x[k] \in \mathbb{R}^n$  is an  $n \times 1$  vector representing the state of the target at time  $k$  and  $w[k]$  is the process noise.  $w[k]$  is assumed zero mean, Gaussian and white with  $Q$  representing its covariance matrix. Let  $y_j[k]$  represent the observation of the  $j^{\text{th}}$  sensor at time  $k$ :

$$y_j[k] = x[k] + v_j[k] \quad (2)$$

where the observation noise,  $v_j[k]$ , is zero mean Gaussian noise with  $R_j[k]$  representing its covariance matrix:  $R_j[k] = \overline{v_j[k]v_j^t[k]}$  with “superscript  $t$ ” representing the transpose of a vector/matrix.

### A. Sensor Fusion and Decentralized Motion Planning

Each node transmits its local measurement and measurement error covariance to other nodes. Let  $\hat{y}_{i,j}[k]$  and  $\hat{R}_{i,j}[k]$  represent the measurement of the  $j^{\text{th}}$  sensor and its corresponding error covariance matrix received at the  $i^{\text{th}}$  sensor respectively. We will have the following for  $1 \leq i, j \leq N$ ,

$$\begin{aligned} \hat{y}_{i,j}[k] &= y_j[k] + c_{i,j}[k] & c_{i,i}[k] &= 0_{n \times 1} \\ \hat{R}_{i,j}[k] &= R_j[k] + L_{i,j}[k] & L_{i,i}[k] &= 0_{n \times n} \end{aligned} \quad (3)$$

where  $c_{i,j}[k] \in \mathbb{R}^n$  and  $L_{i,j}[k] \in \mathbb{R}^{n \times n}$  contain communication noises occurred in the transmission of each element of  $y_j[k]$  and  $R_j[k]$  respectively and  $0_{n \times 1}$  and  $0_{n \times n}$  represent the zero vector and matrix respectively.  $U_{i,j}[k]$  represents covariance matrix of  $c_{i,j}[k]$ :

$$U_{i,j}[k] = \overline{c_{i,j}[k]c_{i,j}^t[k]} \quad (4)$$

Each sensor would then fuse its own measurement with the received ones to reduce its measurement uncertainty. We assume that each sensor uses a Best Linear Unbiased Estimator (BLUE) [16] to process local and received information. It then makes a local decision about where to move next to minimize its local fused estimation error. Let  $p_j[k] \in \mathbb{R}^2$  represent the position of the  $j^{\text{th}}$  sensor at time  $k$ . The  $j^{\text{th}}$  sensor decides its next move as follows:

$$p_j[k+1] = \xi(p_j[k], \hat{y}_{j,1}[k], \hat{R}_{j,1}[k], \dots, \hat{y}_{j,N}[k], \hat{R}_{j,N}[k]) \quad (5)$$

where  $\xi(\cdot)$  represents the motion-planning function used locally at each node.

In order to provide a measure for evaluating the overall performance, in the next section we will first find optimal sensor configurations for networked sensing. To highlight communication/sensing trade-offs,

we show how the optimal locations change when considering imperfect communication. This analysis provides the basis of comparisons and conclusions made later in this paper.

### III. PROVIDING A BENCHMARK: OPTIMAL SENSING LOCATIONS

In this section we examine sensing and communication trade-offs by finding the optimal locations of the sensors in the presence of imperfect communication. This investigation will serve two purposes:

- 1) To give insight on how communication impacts sensing,
- 2) To provide a benchmark for evaluating the performance of the decentralized network in the subsequent sections.

We are interested in finding optimal sensor locations:  $p_j[k]$ . The observation and communication noise covariances are functions of the locations of the sensors and target. Let  $g(\cdot)$  and  $h(\cdot)$  represent these functions respectively. Then,

$$\begin{aligned} R_j[k] &= g(p_j[k], p_T[k]) & 1 \leq j \leq N \\ U_{i,j}[k] &= h(p_i[k], p_j[k]) & 1 \leq i, j \leq N \end{aligned} \quad (6)$$

where  $p_T[k]$  represents target location at the  $k^{\text{th}}$  time instant. The optimal locations considering only communication costs may differ from the optimal locations considering only sensing costs. This results in a trade-off between communication and sensing. We will consider the nature of these trade-offs in this section. The time index  $k$  will be implied except when explicitly necessary.

#### A. Case of Perfect Communication

First we will look at the optimal locations under perfect communication to focus on sensing costs. Let  $\Psi_j$  represent the error covariance matrix of the  $j^{\text{th}}$  sensor after processing the information received from others. We will have the following using a BLUE estimator:

$$\Psi_j = \left( \sum_{i=1}^N R_i^{-1} \right)^{-1} \quad (7)$$

Note that in the absence of communication noise, each sensor has the same fused error covariance,  $\Psi_j$ . We take the determinant of  $\Psi_j$  to be the cost to minimize. Then the optimal locations are the solution to the following optimization problem:

$$\text{Maximize } \det(\Psi_j^{-1}) \rightarrow \text{Maximize } \det\left(\sum_{i=1}^N R_i^{-1}\right) \quad (8)$$

where sensor locations,  $p_1, p_2, \dots, p_N$ , are the optimization variables and  $R_i$  for  $1 \leq i \leq N$  are functions of sensor locations as defined in Eq. 6.

#### B. Case of Imperfect Communication

In this case, we will have the following error covariance matrix after fusion at the  $j^{\text{th}}$  sensor:

$$\Psi_j = \left[ \sum_{i=1}^N (R_i + U_{j,i})^{-1} \right]^{-1} \quad (9)$$

where  $R_i$  and  $U_{j,i}$  are functions of the optimization variables:  $p_1, p_2, \dots, p_N$ , as indicated by Eq. 6. In this scenario, each sensor will have a different local cost function. Therefore, there are different ways of formulating the optimization problem. One possible way is to optimize an average measure,

$$\begin{aligned} \text{Maximize } \sum_j \det(\Psi_j^{-1}) \rightarrow \\ \text{Maximize } \sum_j \det\left(\sum_{i=1}^N (R_i + U_{j,i})^{-1}\right) \end{aligned} \quad (10)$$

To see communication and sensing trade-offs from Eq. 8 and 10, we have to be more specific about the task of the network and define the function  $g$  in Eq. 6. Therefore in the remainder of the paper, we consider a target moving in the plane, and its state is defined to be its position, i.e.  $x = p_T \in \mathbb{R}^2$ . The network will then be estimating the position of the target jointly. The conclusions drawn from this example are, nevertheless, applicable to other sensor network examples as well.

#### C. Cooperative Sensing for Target Location Estimation

1) *Observation Parameters:* To model observation noise of each sensor, we choose a typically used sonar model [18], [19], which results in the following measurement noise covariance,  $R_j$ :

$$R_j = T(\theta_j) D_j(r_j) T^t(\theta_j) \quad (11)$$

where  $T(\theta_j)$  is the rotation matrix:

$$T(\theta_j) = \begin{bmatrix} \cos(\theta_j) & \sin(\theta_j) \\ -\sin(\theta_j) & \cos(\theta_j) \end{bmatrix} \quad (12)$$

and

$$D_j(r_j) = \begin{bmatrix} f_j(r_j) & 0 \\ 0 & \gamma f_j(r_j) \end{bmatrix} \quad (13)$$

where  $r_j$  is the distance of the  $j^{\text{th}}$  sensor to the target and  $\theta_j$  is the corresponding angle in the global reference frame, as illustrated in Fig. 1. The function  $f_j$ , the model for the range noise variance of the  $j^{\text{th}}$  sensor, depends on  $r_j$  and  $\gamma$  is a scaling constant. Eq. 11 describes function  $g$  of Eq. 6 since  $r_j$  and  $\theta_j$  are functions of the locations of the target,  $p_T$ , and the  $j^{\text{th}}$  sensor,  $p_j$ . A common model for  $f$  is quadratic, with the minimum achieved at a particular distance from the target, namely the ‘‘sweet spot’’ of the sensor [3].

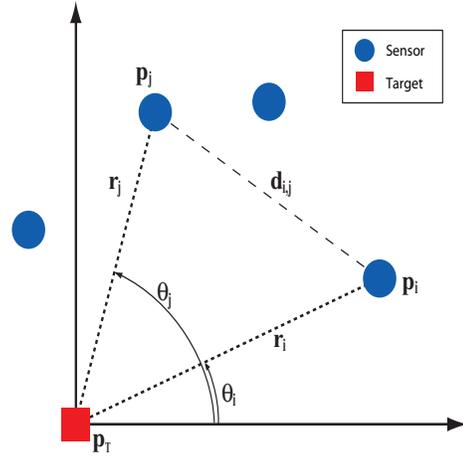


Fig. 1. Illustration of System Variables

2) *Communication Parameters:* We consider an AWGN channel and a distance-dependent path loss model to describe the communication link [20]. Communication noises of the received observation vector are taken to be zero mean and i.i.d, which results in

$$U_{i,j} = \sigma_{comm,i,j}^2 I_2 \quad (14)$$

with  $I_2$  representing a  $2 \times 2$  unit matrix. We assume symmetric uplink and downlink, which implies  $U_{i,j} = U_{j,i}$ .  $\sigma_{comm,i,j}^2$ , the communication noise variance of each element of the observation vector from the  $j^{\text{th}}$  to the  $i^{\text{th}}$  sensor, is a function of the transmission environment and receiver/transmitter

design parameters. Authors in [9] showed that for a uniform quantizer and BPSK modulation, using a distance-dependent path loss model, communication noise variance will be as follows:

$$\sigma_{comm,i,j}^2 = \frac{q^2}{12} + q^2 \times \frac{4^{N_b} - 1}{3} \times Q(\sqrt{SNR_{rec,i,j}}) \quad (15)$$

where  $q$  is the quantization step size and  $N_b$  is the number of bits per transmission of each element of the observation vector. We assume that all the sensors use the same  $q$  and  $N_b$ .  $Q(\mu) = \frac{1}{\sqrt{2\pi}} \int_{\mu}^{\infty} e^{-\frac{t^2}{2}} dt$  for an arbitrary  $\mu$ .  $SNR_{rec,i,j}$  is the average received Signal to Noise Ratio and will have the following relationship with  $d_{i,j}$ , the distance separating the  $i^{th}$  and  $j^{th}$  sensors:

$$SNR_{rec,i,j} = \frac{\alpha}{d_{i,j}^{n_p}} \quad (16)$$

where  $d_{i,j} = \sqrt{r_i^2 + r_j^2 - 2r_i r_j \cos(\theta_i - \theta_j)}$  and  $n_p > 0$  is the path loss exponent which depends on the environment. Furthermore,  $\alpha \geq 0$  is a function of the transmitted signal power, receiver noise, frequency of operation and the communication environment [20]. We assume the same  $\alpha$  and  $n_p$  for all the communication links.

3) *Perfect Communication*: It can be shown that the maximization problem of Eq. 8 is equivalent to the following using Eq. 11 [3]:

$$\begin{aligned} & \text{Maximize} \quad \frac{1}{\gamma} \left[ \sum_{i=1}^N \frac{1}{f_i(r_i)} \right]^2 + \\ & \frac{(1-\gamma)^2}{\gamma^2} \sum_{2 \leq i < j \leq N} \frac{1}{f_i(r_i) f_j(r_j)} \sin^2(\theta_i - \theta_j) + \\ & \frac{(1-\gamma)^2}{\gamma^2} \frac{1}{f_1(r_1)} \sum_{i=2}^N \frac{1}{f_i(r_i)} \sin^2(\theta_i) \end{aligned} \quad (17)$$

where  $r_1, r_2, \dots, r_N$  and  $\theta_2, \dots, \theta_N$  are the optimization variables and  $\theta_1$  is taken zero. For instance, consider two sensors with the same  $f$  functions. It can be easily shown that the optimal locations will be as follows:  $r_{1,opt} = r_{2,opt} = r_{ss}$  and  $\theta_{1,opt} - \theta_{2,opt} = \frac{\pi}{2}$ . The sweet spot radius,  $r_{ss}$ , represents the distance from sensor to target that achieves the minimum of function  $f(r)$ .

4) *Imperfect Communication*: In this case, Eq. 10 will be as follows using Eq. 11 and 14:

$$\sum_j \det(\sum_{i=1}^N T(\theta_i)(D_i(r_i) + U_{j,i})^{-1} T^t(\theta_i)) \quad (18)$$

To see the impact of communication more clearly, consider the case of two sensors. After much algebraic manipulation, Eq. 18 is given by (assuming  $f_1 = f_2 = f$ ):

$$\begin{aligned} & \text{Maximize} \\ & \frac{\gamma(f(r_1)+f(r_2))^2 + \sigma_{comm}^2(\gamma+1)(f(r_1)+f(r_2))}{\gamma f^2(r_1)(\gamma f^2(r_2) + \sigma_{comm}^2 + \sigma_{comm}^2(\gamma+1)f(r_2))} + \\ & \frac{(\gamma-1)^2 \sin^2(\Delta) f(r_1) f(r_2) + \sigma_{comm}^4}{\gamma f^2(r_1)(\gamma f^2(r_2) + \sigma_{comm}^2 + \sigma_{comm}^2(\gamma+1)f(r_2))} \end{aligned} \quad (19)$$

where  $\Delta = \theta_1 - \theta_2$  and  $\sigma_{comm}^2 = \sigma_{comm,1,2}^2$  is a function of optimization variables through Eq. 15 and 16. The optimal solution of the case of perfect communication, i.e.  $\frac{\pi}{2}$  angle difference and sweet spot radius, may not be the optimal solution for this case any more. Depending on the quality of the channel, sensors may have to compromise sensing quality for better communication, sacrificing either the  $\frac{\pi}{2}$  angle and/or the sweet spot radius. This is what we refer to as ‘‘sensing/communication trade-offs’’. To see this, Table II shows the optimal solution to Eq. 19, found using a brute-force search, for the parameters of Table I and for three different channels,  $\alpha = 570$ ,  $\alpha = 5700$  and  $\alpha = 57000$ . Values of  $\alpha$  are chosen based on realistic parameters for transmitted signal power, receiver noise and frequency of operation. As  $\alpha$  gets smaller, the quality of the channel degrades ( $SNR_{rec}$  is proportional to  $\alpha$ ). We can see

from Table II that as  $\alpha$  gets smaller, the optimal solution deviates more considerably from the solution of the perfect communication case. For instance at  $\alpha = 570$ ,  $\Delta_{opt}$  is  $18^\circ$  instead of  $90^\circ$  of the perfect communication case. The results highlight sensing and communication trade-offs in sensor networks.

Observation Parameters	
$f$	$0.0008(r - 15.625)^2 + 0.1528$
$\gamma$	5
$Q$	$.01I_2$
Communication Parameters	
$q$	0.0018
$N_b$	15
$n_p$	2

TABLE I  
SYSTEM PARAMETERS

	Perfect Comm	Distance-Dependent Path Loss		
		$\alpha = 570$	$\alpha = 5700$	$\alpha = 57000$
$r_{1,opt}$	15.625	15.2	14.1	15.625
$r_{2,opt}$	15.625	15.2	14.1	15.625
$\Delta_{opt}$	$90^\circ$	$18^\circ$	$70.56^\circ$	$90^\circ$

TABLE II  
OPTIMAL LOCATIONS FOR TWO SENSORS

#### IV. DECENTRALIZED MOBILE SENSOR NETWORK AND IMPERFECT COMMUNICATION

The previous section provided insight on the impact of communication on the optimal locations of the sensors. In this section we consider a mobile network that seeks to achieve the optimal configuration through local decentralized motion planning. We investigate the effect of imperfect communication on such networks. We further propose an algorithm that extends the work developed in [3] to improve the performance by taking communication effects into account in the motion-planning process of each sensor.

##### A. Decentralized Mobile Sensor Network

Before introducing the impact of imperfect communication, we will first discuss related work on decentralized motion planning neglecting communication impacts.

The objective in cooperative estimation is to determine the sensor motions which minimize fused estimation error. Furthermore, we seek a decentralized solution such that each sensor identifies its optimal location for the next time step. Given that the gradient provides the locally optimal direction of movement, authors in [3] use a gradient-based descent algorithm which defines the optimal control action as the one which will position each sensor to minimize a local measure function in the following time step. They further reduce the gradient descent algorithm to a discrete gradient search algorithm by restricting the possible control actions for each sensor to a finite, discrete set of motions. Their algorithm demonstrated performance improvement over other existing decentralized motion-planning algorithms with low computational complexity. Here we briefly describe the algorithm (for more details, see [3]).

Each sensor takes a local measurement as described by Eq. 2 and uses a local Kalman filter [21] to improve its local estimate. Let  $x_{est,j}[k]$  and  $Z_j[k]$  represent the local estimate of the  $j^{th}$

sensor and its corresponding error covariance matrix after Kalman filtering at time  $k$ . Each sensor then transmits its local information and receives the estimates of others. For this section we assume perfect communication. Therefore, the  $j^{th}$  sensor will have the exact copies of  $x_{est}[k]$  and  $Z[k]$  of other sensors. It then fuses all the information to improve its performance by using a BLUE estimator. Let  $x_{fused,j}[k]$  and  $Z_{fused,j}[k]$  represent the estimate of the target and the corresponding error covariance matrix after fusion at the  $j^{th}$  sensor. We will have,

$$Z_{fused,j}[k] = \left( \sum_{i=1}^N Z_i^{-1}[k] \right)^{-1} \quad (20)$$

To plan its next move, the  $j^{th}$  sensor takes the following steps:

- 1) It uses  $x_{fused,j}[k]$  and  $Q$  to predict the next state of the target (if an estimate of  $Q$  is not available, it assumes that the state of the target has not changed):

$$x_{predicted,j}[k+1] = x_{fused,j}[k] + s \quad (21)$$

where  $x_{predicted,j}[k+1]$  is the prediction of the  $j^{th}$  sensor of the state of the target and  $s$  is a sample of zero mean white Gaussian noise generated using covariance matrix  $Q$ .

- 2) Using the received local error covariances of other sensors, it then predicts the estimation error covariances of other nodes by propagating the corresponding Kalman filters one step ahead:

$$Z_{predicted,j,i}[k+1] = EP(Z_i[k]) \quad j \neq i \quad (22)$$

where  $Z_{predicted,j,i}[k+1]$  is the  $j^{th}$  node's prediction of the local error covariance of the  $i^{th}$  sensor.  $EP(\cdot)$  stands for a function that produces a prediction of the next error covariance using Kalman filtering.

- 3) For every possible motion vector,  $m$ , the  $j^{th}$  node then predicts its own error covariance by progressing its Kalman filter. Let  $Z_{predicted,j}[k+1, m]$  represent the prediction of the  $j^{th}$  sensor of its own error covariance as a function of  $m$ . Using these predictions, the  $j^{th}$  node produces the following cost to minimize:

$$MotionCost_j[k, m] = \det(Z_{predicted,j}^{-1}[k+1, m] + \sum_{i=1, i \neq j}^N Z_{predicted,j,i}^{-1}[k+1])^{-1} \quad (23)$$

$MotionCost_j[k, m]$  is the cost used by the  $j^{th}$  sensor in planning its motion at time  $k$ . Finally it chooses the motion vector that minimizes the cost:

$$m^* = \operatorname{argmin} MotionCost_j[k, m] \quad (24)$$

In this manner, the task of motion planning is given to each sensor, in place of a central computation node. To see the performance of the decentralized algorithm for the observation parameters of Table I under perfect communication, Fig. 2 shows sensor trajectories for 50 time steps. It shows the convergence of sensors to their optimal locations (defined by the solution of Eq. 17) when  $N = 3$ .

### B. Impact of Imperfect Communication

We observed in Section III that to optimize the performance in the presence of imperfect communication, the network may need to trade sensing quality for better communication performance. This means that the local motion-planning algorithm should take communication link qualities into account. Information on the quality of the link is available in the physical layer. Since motion planning is performed in the higher application layer, this requires proper interface between the two layers. Existing motion-planning algorithms do not

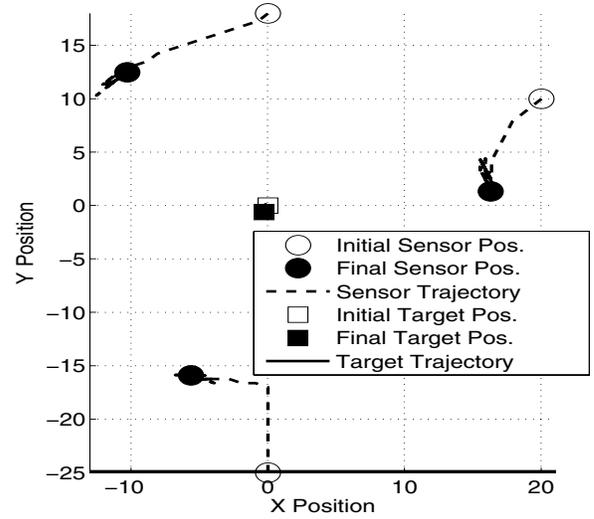


Fig. 2. Performance of the already-existing decentralized algorithm, case of perfect communication,  $N = 3$

take communication issues into account. Using such algorithms in the presence of non-ideal communication links can result in poor performance as each sensor will now receive noisy versions of  $x_{est}[k]$  and  $Z[k]$ . To illustrate this, we add non-ideal communication links to the simulation setup of Fig. 2. Since communication impacts are not taken into account in the motion-planning algorithm, we consider scenarios that have a distance-dependent packet drop mechanism in order to limit the amount of received estimation noise. This means that for  $d_{i,j} > d_{critical}$ , the  $i^{th}$  sensor will discard the data received from the  $j^{th}$  sensor. Scenario#1 refers to the case where  $\alpha = 5700$  when received packets are kept. Scenario#2 refers to an ideal case where received packets are noise-free if they are kept in the receiver and scenario#3 refers to the case where  $\alpha = 570$ . Communication parameters are as summarized in Table I and  $d_{critical} = 20m$ . Fig. 3 shows sensor trajectories for  $N = 2$ , scenario#1 and for 50 time steps. As can be seen, the algorithm does not converge and the sensors are acting independently. Fig. 4 shows the determinant of the error covariance of one of the sensors (after fusion) for scenario#1 and 2. For comparison, the determinant of the error covariance for  $N = 1$  and  $N = 2$  with perfect communication are also plotted. At the beginning the sensors can communicate and, due to the low level of communication noise for these two scenarios, can benefit from cooperative sensing for a short period of time. However, since the local information processing and motion planning algorithms of each sensor do not take communication effects into account, the sensors can not be guided toward finding the optimum locations. Instead, they move in the opposite directions of the optimum trajectories, which results in the sensors acting independently (see Fig. 3). The same situation happens after a few iterations as well. Therefore, the sensors can not benefit from networked sensing. Fig. 4 also shows the performance for scenario#2, the ideal case in which the received packets are noise-free if not dropped in the receiver. We can see that the network shows a similar behavior. It can not find the optimum locations and can not benefit from cooperative sensing. Fig. 5 shows the determinant of the error covariance of one of the sensors (after fusion) for scenario#3. In this scenario,  $\alpha = 570$ , which represents a weaker channel. It can be seen that since the communication noise is not accounted for in sensor fusion and motion planning algorithms,

the estimation error can get considerably high when received packets are not dropped. The trajectories for scenario#3 are similar to that of scenario#1 of Fig. 3. It should be noted that in the absence of a packet drop mechanism, the estimation error would have been even higher due to the lack of a proper interface of the physical and the application layers. To see this, Fig. 5 also shows a sample of the estimation error for scenario#4. Scenario#4 refers to the case where all the packets are kept in the receiver and  $\alpha = 5700$ . Although this is a stronger channel, the estimation error can get considerably high since the sensors move away from each other.

In general, even for the cases that the sensors start out closer, they can easily end up performing individual estimation. This behavior of the network is also independent of the value of  $d_{critical}$ . This is due to the fact that the motion-planning algorithm is not taking communication effects into account. This motivates designing decentralized motion-planning algorithms that are more robust to communication imperfection. The next section will show how to modify the aforementioned decentralized algorithm to include communication impacts, creating the possibility of sensing/communication trade-offs when planning the next move.

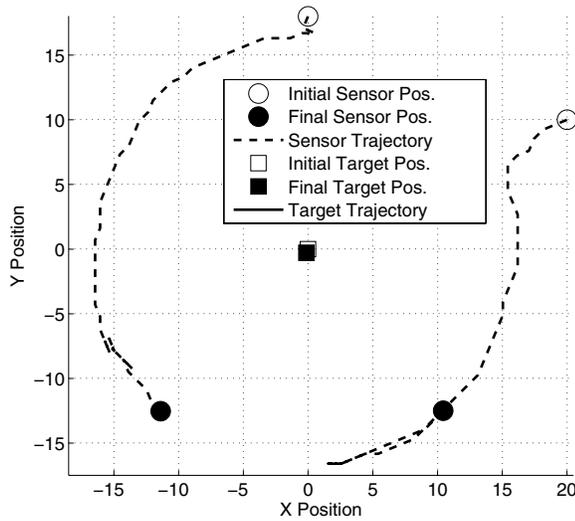


Fig. 3. Performance of the already-existing decentralized algorithm, case of imperfect communication,  $N = 2$ , scenario#1

### C. A Decentralized Motion-Planning Algorithm Considering Both Communication and Sensing: A Cross Layer Design Approach

To improve the performance in the presence of imperfect communication, we modify the algorithm to allow for an interface of application and physical layers. Physical layer will pass information on the quality of the link to the application layer. More specifically, it will pass knowledge of the communication noise variances,  $\sigma_{comm,i,j}^2[k]$ . Application layer of each sensor then uses this information for both fusion and motion planning. At the time of local fusion, Eq. 20 becomes:

$$Z_{fused,j}[k] = \left( \sum_{i=1}^N (\hat{Z}_{j,i}[k] + U_{j,i}[k])^{-1} \right)^{-1} \quad (25)$$

where  $\hat{Z}_{j,i}[k]$  is the noisy version of  $Z_i[k]$  received by the  $j^{th}$  sensor. Eq. 25 prevents noisy samples from degrading fusion performance. In practice physical layer can estimate  $\sigma_{comm,i,j}^2$  by measuring the

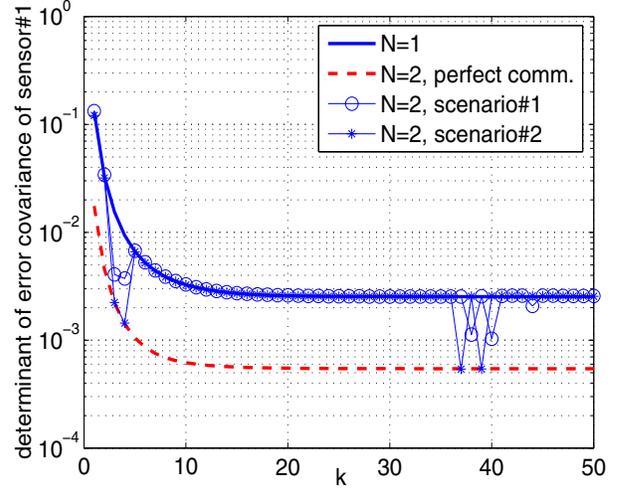


Fig. 4. Performance of the already-existing decentralized algorithm, case of imperfect communication,  $N=2$ , scenario#1 and scenario#2

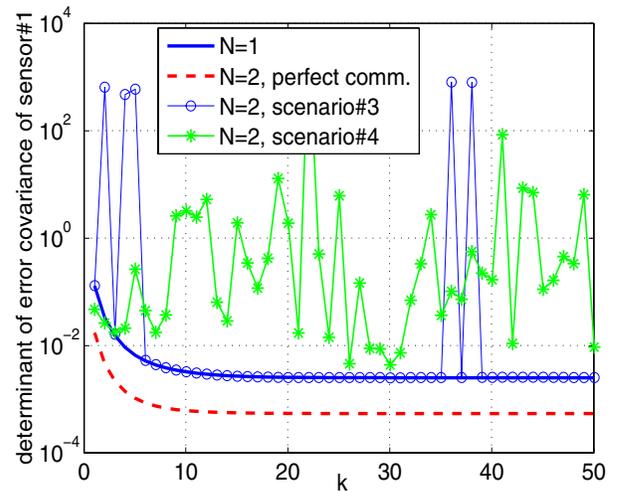


Fig. 5. Performance of the already-existing decentralized algorithm, case of imperfect communication,  $N=2$ , scenario#3 and scenario#4

received Signal to Noise Ratio. Then when planning the next move, Eq. 23 should be modified as follows:

$$MotionCost_j[k, m] = \det \left[ Z_{predicted,j}^{-1}[k+1, m] + \sum_{i \neq j} (Z_{predicted,j,i}[k+1] + U_{predicted,j,i}[k+1, m])^{-1} \right]^{-1} \quad (26)$$

where  $Z_{predicted,j,i}[k+1] = EP(\hat{Z}_{j,i}[k])$  and  $U_{predicted,j,i}[k+1, m]$  is the  $j^{th}$  sensor's prediction of the communication error covariance of the  $i^{th}$  sensor's transmission given motion vector  $m$ . Typically each sensor also transmits its position as well. In that case, the  $j^{th}$  sensor receives noisy estimates of positions of other nodes from which it can estimate its distances to other sensors for each motion vector. It can then use the model described by Eq. 15 to get  $U_{predicted,j,i}[k+1, m]$ . If the estimates of positions of other nodes are not available, the received observation estimates have implicit information on the positions of other nodes and can be used for prediction. Fig. 6-10 show the performance of

the proposed algorithm for the parameters of Table I. Fig. 6 shows sensor trajectories for 50 time steps,  $N = 2$  and  $\alpha = 570$ . We can see convergence of the sensors to their optimal locations. After 50 time steps, we have  $\Delta[50] = 13.5^\circ$ ,  $r_1[50] = 16$  and  $r_2[50] = 16.1$ . Comparing these values with the corresponding optimal ones in Table II, shows convergence of the decentralized algorithm to the optimal locations. We can see that by accounting for communication links in the application layer, we improve the performance considerably. Fig. 7-10 show the determinant of the error covariance for one of the sensors (after fusion) as a function of time, for two different channels and for  $N = 2, 3, 4$  and  $5$  respectively. We can see that in all the figures, for  $\alpha = 5700$ , the error stays very close to that of the ideal communication from the beginning. For  $\alpha = 570$ , the sensors start out acting individually but can find the optimum configuration quickly resulting in the error reaching very close to that of the ideal communication case after a few time steps. The convergence gets faster as the quality of the link improves. Convergence time is also a function of the initial positions of the sensors and may be different for different sensors of the network. The error is always bounded by that of a single sensor independent of the quality of the link. The results emphasize the importance of cross-layer feedback in decentralized motion-planning.

To see the performance of the proposed algorithm when the target is moving faster, we next simulate the proposed algorithm for the following target motion:  $x_{fast}[k + 1] = Ax_{fast}[k] + w[k]$ . Fig. 11, 12 and 13 show the performance of the proposed decentralized algorithm for  $A = .7I_2$ ,  $Q = .1I_2$  with the rest of the parameters as summarized in Table I. Fig. 11 shows how sensors track the target for  $N = 2$ ,  $\alpha = 570$  and 50 time steps. Fig. 12 and 13 show the determinant of the error covariance of one of the sensors (after fusion) as a function of time, for two different channels and for  $N = 2$  and  $4$  respectively. Fig. 11 shows that using the proposed algorithm, the sensors can track the target considerably well. Fig. 12 and 13 further demonstrate that the error reaches very close to that of the ideal communication case. We can see that the network benefits from cooperative sensing for target tracking.

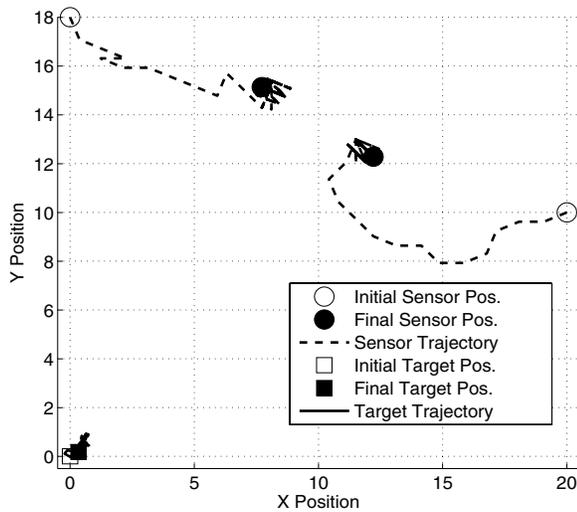


Fig. 6. Performance of the proposed decentralized algorithm, case of imperfect communication,  $N = 2$ ,  $\alpha = 570$

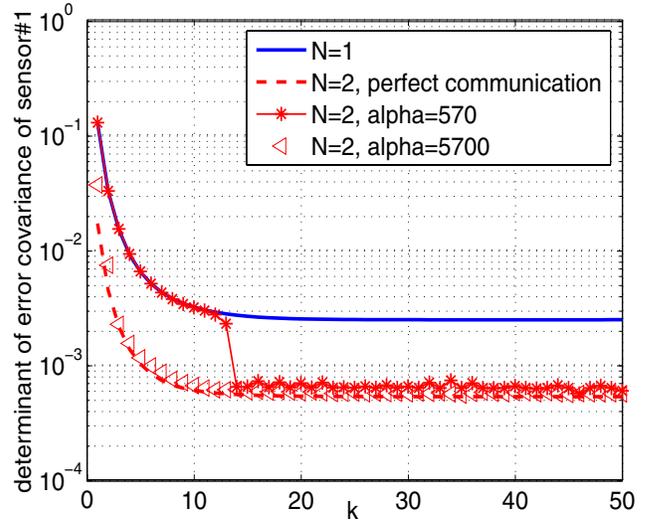


Fig. 7. Performance of the proposed decentralized algorithm, case of imperfect communication,  $N=2$

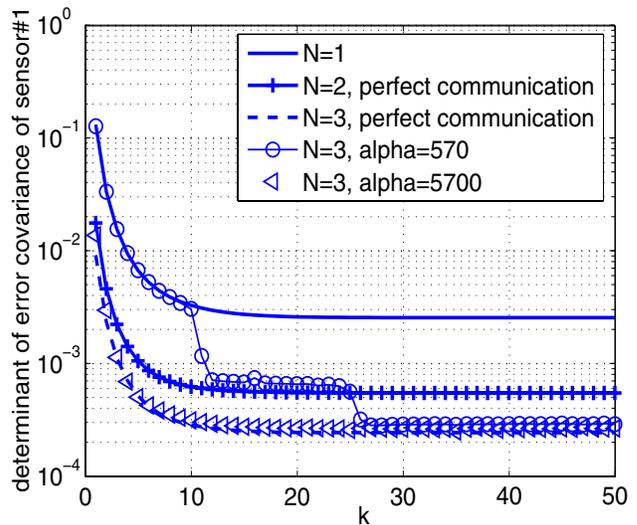


Fig. 8. Performance of the proposed decentralized algorithm, case of imperfect communication,  $N=3$

## V. SUMMARY

In this paper we considered the impact of imperfect communication on the performance of a decentralized mobile sensor network. We showed communication and sensing trade-offs in such networks by determining the optimal sensor locations in the presence of non-ideal communication links. To improve the performance, we proposed a decentralized motion-planning algorithm that takes both communication and sensing objectives into account. The algorithm was a cross-layer design and highlighted the importance of sharing the information of physical layer with the application layer. Finally simulation results showed the performance improvement gained by using this algorithm.

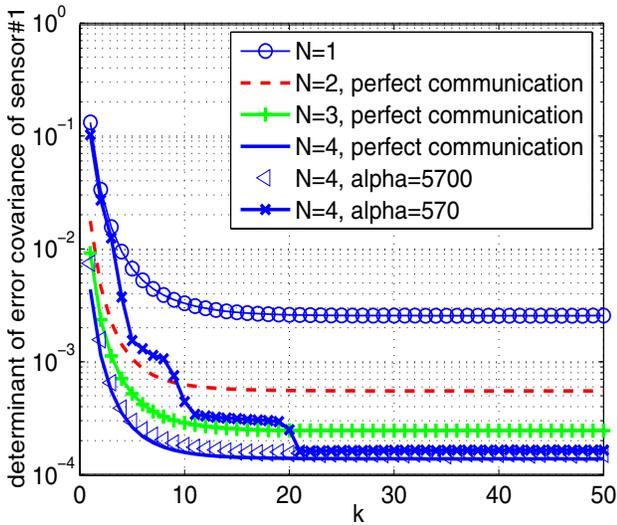


Fig. 9. Performance of the proposed decentralized algorithm, case of imperfect communication,  $N=4$

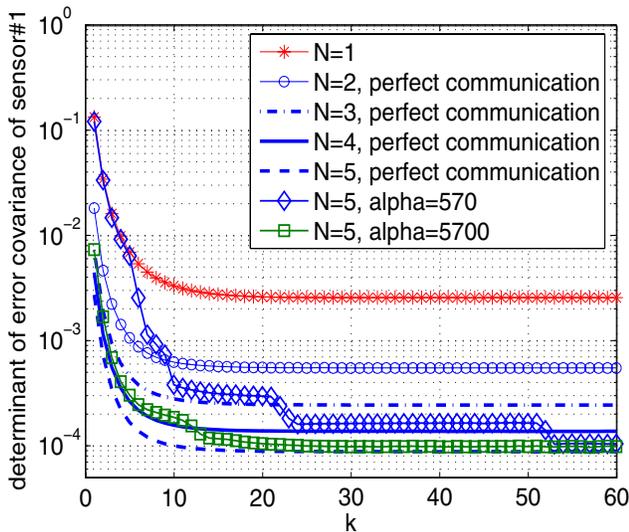


Fig. 10. Performance of the proposed decentralized algorithm, case of imperfect communication,  $N=5$

## VI. ACKNOWLEDGEMENTS

Authors would like to thank Vijay Gupta for useful discussions on the topic.

## REFERENCES

- [1] C. Chong and S. Kumar, "Sensor networks: evolution, opportunities and challenges," *Proceedings of the IEEE*, Volume: 91, Issue: 8, Aug. 2003, Pages:1247-1256
- [2] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert and S. Sastry, "Distributed control applications within sensor networks," *Proceedings of the IEEE*, Volume: 91, Issue: 8, Aug. 2003, Pages:1235-1246
- [3] T. H. Chung, V. Gupta, J. W. Burdick and R. M. Murray, "On a Decentralized Active Sensing Strategy using Mobile Sensor Platforms in a Network," 43rd IEEE Conf. on Decision and Control, 2004

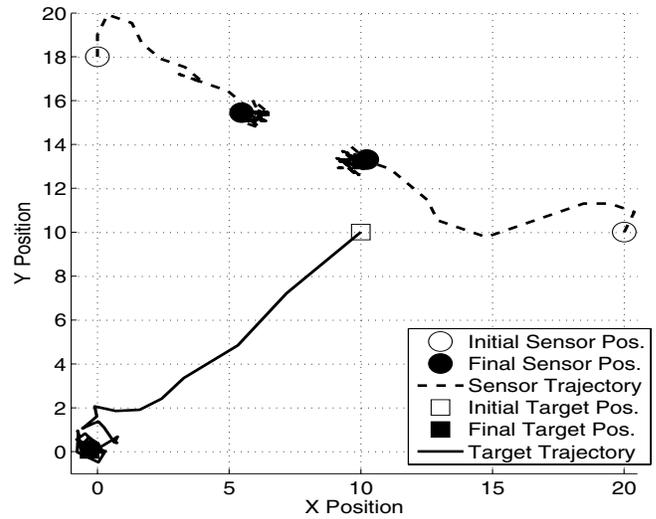


Fig. 11. Performance of the proposed decentralized algorithm when tracking, case of imperfect communication,  $N=2$

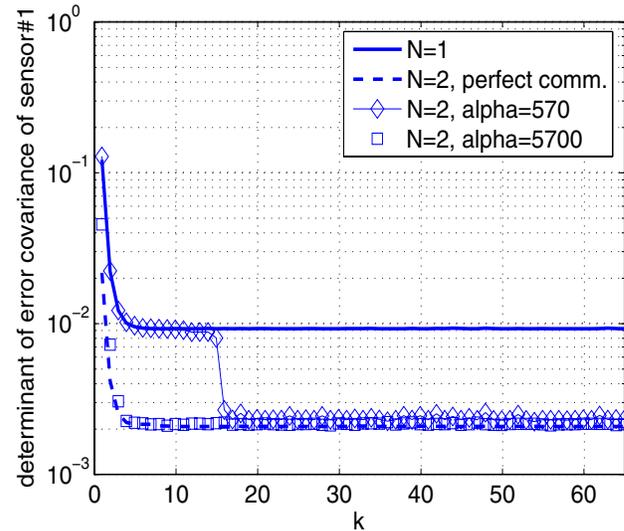


Fig. 12. Performance of the proposed decentralized algorithm when tracking, case of imperfect communication,  $N=2$

- [4] D. Yeung and G. A. Bekey, "A Decentralized Approach to the Motion Planning Problem for Multiple Mobile Robots," *Proc. of Intl. Conf. on Robotics and Automation*, vol. 4, pp. 1779-1784, 1987
- [5] Y. Guo and L. E. Parker, "A Distributed and Optimal Motion Planning Approach for Multiple Mobile Robots," *Proc. of Intl. Conf. on Robotics and Automation*, vol. 3, pp. 2612-2619, 2002
- [6] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, S., "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, Vol. 49, Issue: 9, Sept. 2004
- [7] X. Liu and A. J. Goldsmith, "Kalman Filtering with Partial Observation Losses," 43rd IEEE Conference on Decision and Control, 2004
- [8] Y. Mostofi and R. Murray, "On Dropping Noisy Packets in Kalman Filtering Over a Wireless Fading Channel," To appear in the 24th American Control Conference (ACC)
- [9] Y. Mostofi and R. Murray, "Effect of Time-Varying Fading Channels on the Control Performance of a Mobile Sensor Node," *Proceedings of 1st IEEE International Conference on Sensor and Adhoc Communications and Networks*, 2004
- [10] L. Xiao, M. Johansson, H. Hindi, S. Boyd and A. Goldsmith, "Joint

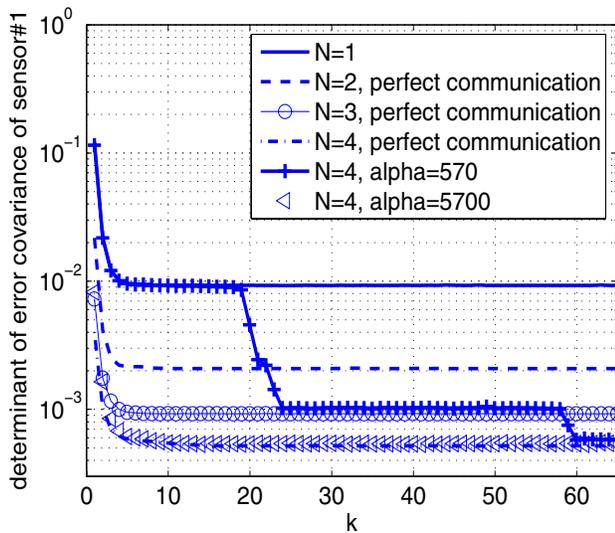


Fig. 13. Performance of the proposed decentralized algorithm when tracking, case of imperfect communication,  $N=4$

optimization of communication rates and linear systems," IEEE trans. on automatic control, vol. 48, no. 1, Jan. 2003

- [11] X. Liu and A. Goldsmith, "Wireless communication tradeoffs in distributed control," 42nd IEEE Conference on Decision and Control, volume: 1, 2003, pages:688 - 694
- [12] X. Liu and A. J. Goldsmith, "Wireless Medium Access Control in Distributed Control Systems," Allerton Conference on Communication, Control and Computing, 2003
- [13] S. Tatikonda, A. Sahai and S. Mitter, "Control of LQG systems under communication constraints," IEEE 37th Conference on Decision and Control, 1998, vol. 1, Pages:1165-1170
- [14] D. P. Spanos and R. M. Murray, "Robust Connectivity of Networked Vehicles," 43rd IEEE Conf. on Decision and Control, 2004
- [15] N. C. Martins and M. A. Dahleh, "Fundamental limitations of feedback control with communication constraints," <http://www.cds.caltech.edu/doyle-fest/presentations/pdfs/dahleh.pdf>
- [16] Steven M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1993
- [17] J. R. Spletzer and C. J. Taylor, "Dynamic Sensor Planning and Control for Optimally Tracking Targets," Intl. J. Robotics Research, vol. 22-1, pp. 7-20, 2003.
- [18] K.V. Ramachandra, *Kalman Filtering Techniques for Radar Tracking*. New York, NY: Marcel Dekker, Inc., 2000
- [19] K. Umeda and J. Ota and H. Kimura, "Fusion of Multiple Ultrasonic Sensor Data and Imagery Data for Measuring Moving Obstacle'S Motion," Intl. Conf. on Multisensor Fusion and Integration for Intelligent Systems, December 1996, pp. 742-748.
- [20] William Jakes, *Microwave Mobile Communications*. IEEE Press, 1974
- [21] T. Kailath, A. H. Sayed, B. Hassibi, *Linear Estimation*. Prentice Hall information and system sciences series