Experimental Validation of a Localization System Based on a Heterogeneous Sensor Network

José Araújo, Henrik Sandberg, and Karl Henrik Johansson

***Abstract*— The experimental implementation and validation of a localization system based on a heterogeneous sensor network is described. The sensor network consists of ultrasound ranging sensors and web cameras. They are used to localize a mobile robot under sensor communication constraints. Applying a recently proposed sensor fusion algorithm that explicitly takes communication delay and cost into account, it is shown that one can accurately trade off the estimation performance by using low-quality ultrasound sensors with low processing time and low communication cost versus the use of the high-quality cameras with longer processing time and higher communication cost. It is shown that a periodic schedule of the sensors is suitable in many cases. The experimental setup is discussed in detail and experimental results are presented.**

I. Introduction

Wireless sensor networks provide opportunities for the development of new applications in monitoring and automation [1], [2]. One such possible application area is location systems for tracking mobile objects in indoor and outdoor environments, e.g., [3], [10]. Tracking and localization are well-studied problems, see for example [4]. Nevertheless, implementation of such algorithms in wireless networks pose new challenges: Resource limitations such as battery power and communication bandwidth need to be taken into account in the design of estimation and control algorithms, as well as communication protocols. When not all sensors can be used simultaneously, they need to be actively selected. Sensor scheduling problems have been studied extensively in the literature, e.g., [5]. One class of scheduling problems is when there is a cost associated with the use of the sensors [6]. Another problem recently studied is when the schedule depends on the sensor quality, the processing time and the communication cost [7].

The problem considered in this paper is to estimate the position of a mobile robot using two types of sensors: ultrasound ranging sensors and web cameras. The system architecture is illustrated in Fig. 1. The sensors share the same communication medium, so the fusion center needs to both schedule which sensor to use and decide on how to incorporate the sensor data into a position estimate. The ultrasound sensors provide low-quality measurements, small processing delay, and a light communication cost. The web camera provide high-quality measurements, large processing delay, and a high communication cost. The solution described in the paper shows how optimal sensor schedules and fusion algorithms developed in [7] can be implemented and extended. The periodic and aperiodic schedules are found by means of search over a finite set. The main contribution of the paper is an experimental implementation and validation of our solution using a heterogeneous sensor network and a mobile robot.

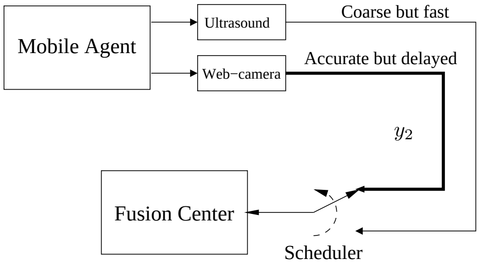


Fig. 1. The system architecture for the localization of a mobile agent using a heterogeneous sensor network.

The outline of this paper is as follows. Section II presents robot and sensor models used in the fusion center. A sensor fusion algorithm together with a scheduler for the sensor communication is described in Section III. The experimental setup is given in Section IV. In Section V the experimental results are presented and some filter design choices are discussed. The paper is concluded in Section VI.

II. Robot and Sensor Models

In this section, the kinematic mobile robot model and the sensor models are presented. It is also shown how the models are sampled and how stochastic noise enters in the models.

A. Kinematic Models

The mobile robot used for the experimental validation can be modeled by a simple non-linear kinematic model,

$$\begin{aligned} \dot{X} &= V\cos\theta, & \dot{Y} &= V\sin\theta \\ J\ddot{\theta} &= T, & m\dot{V} &= F, \end{aligned} \tag{1}$$

where V is the velocity, θ the heading, J the moment of inertia, T the applied moment, F the applied motor force, and m the vehicle mass. The model is depicted in Fig. 2.

To simplify the description further, we consider one-dimensional movement along the X-direction ($\theta = 0$), which reduces (1) to

$$\dot{X} = V, \quad m\dot{V} = F. \tag{2}$$

This work was supported by the Swedish Research Council, the Swedish Strategic Research Foundation, the Swedish Governmental Agency for Innovation Systems, and the European Commission through the FeedNetBack project.

The authors are with the ACCESS Linnaeus Centre, School of Electrical Engineering, Royal Institute of Technology, 100 44 Stockholm, Sweden. Email: araujo@kth.se,{hsan, kallej}@ee.kth.se

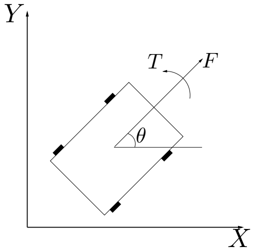


Fig. 2. Mobile agent kinematic model.

The driving force F is supposed to be unknown. We model the motion of the robot as a random walk and an integrated random walk.

Remark 1: If the model (1) is linearized, it is straightforward to extend the following theory to movement in two dimensions by doubling the state space dimension. Another option is to apply Extended Kalman Filtering theory directly to the nonlinear model (1), but that lies outside the scope of this paper since we here mainly want to experimentally verify the ideas in [7].

The sampled data models are as follows.

1) Random Walk Model: In the first model, we assume the robot motion is a random walk. This means that the velocity V is white noise. If the model (2) is sampled with a uniform sampling interval under this assumption, we obtain

$$X_{k+1} = X_k + w_k,$$

where w_k is a random process noise variable such that $\mathbf{E}w_k = 0$ and $\mathbf{E}w_k w_{k'} = W\delta_{k-k'}$.

2) Integrated Random Walk Model: In the second model, we assume instead that the force F acting on the robot is subject to white noise. If the model (2) is sampled with a uniform sampling interval h under this assumption, we obtain

$$X_{k+1} = X_k + hV_k, \quad V_{k+1} = V_k + w_k,$$

where w_k is a random process noise variable such that $\mathbf{E}w_k = 0$ and $\mathbf{E}w_k w_{k'} = W\delta_{k-k'}$.

As is well known, the random walk and the integrated random walk models can be written as a state-space model, commonly used for estimation and tracking purposes [4] as,

$$x_{k+1} = Ax_k + Bw_k, \qquad (3)$$

with state a vector $x_k \in \mathbb{R}^n$. For the first model, we have $x_k = X_k$, $A = B = 1$, and for the second model we have $x_k = (X_k, V_k)^T$, $A = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix}$, and $B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

B. Sensor Models

The heterogeneous sensor network consists of ultrasound sensors and web cameras.

In Fig. 3, the ultrasound-based localization system is illustrated. The ultrasound receivers are placed in the ceiling and an ultrasound transmitter is carried by the mobile robot. Each receiver can calculate its distance to the robot. Fusing the distances given by three receivers, it is possible to determine the robot position through multilateration [9].

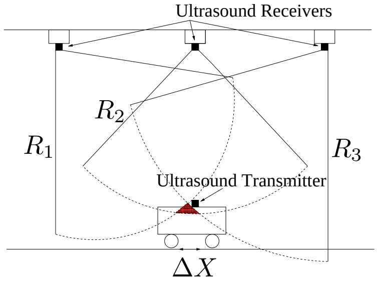


Fig. 3. Ultrasound-based localization system model with measurement error ΔX. Each pair of ultrasound receiver and transmitter calculates their absolute distance R_i and ultrasound transmitter position is achieved applying the multilateration localization method.

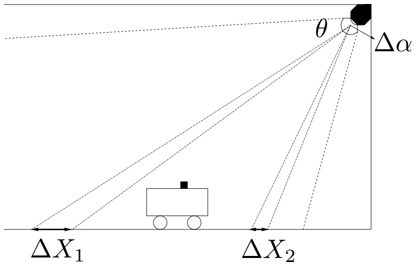


Fig. 4. Vision-based localization system model with measurement error ΔX_1,ΔX_2. The measurement error will increase when the mobile agent is more distant from the vision sensor.

In Fig. 4, the vision-based localization system is illustrated. The web camera span-angle is θ and its pixel resolution is denoted by $\Delta\alpha$. Through image processing it is possible to compute the agent centroid position by performing a transformation from pixel coordinates to world coordinates. A given area will map into the same pixel. The size of the area depends on the distance to the camera. The position error will increase when the mobile agent is more distant from the web camera.

To get mathematically tractable sensor models, the position measurements will be modeled as

$$X_{meas,k} = X_k + \Delta X_k,$$

where the measurement error is zero on average, $\mathbf{E}\Delta X_k = 0$. The variances of the measurement error are assumed to be in the form $\mathbf{E}\Delta X_k \Delta X_{k'} =: \Sigma\delta_{k-k'}$ for the ultrasound measurement, and $\mathbf{E}\Delta X_k \Delta X_{k'} =: \sigma\delta_{k-k'}$ for the camera measurement. The variances Σ and σ depend on the position X of the robot. If we consider movement in a small area, however, the dependance on position can be neglected. If the dependence of Σ and σ on X is known, that can be explicitly taken into account in the covariance-based scheduler described in Section III-B.2, but not in the periodic scheduler in Section III-B.1.

III. Sensor Fusion and Scheduling

In this section, the sensor fusion and the scheduling algorithms are presented. These algorithms were originally proposed in [7]. We recall them here and include a slight extension to the scheduler that can be used to take varying variance of measurement into account.

A. Sensor Fusion

Define two sets of discrete time instants T_{hq} and T_{lq}, such that when $k \in T_{\text{hq}}$ the high quality sensor is used by the sensor fusion center, and when $k \in T_{\text{lq}}$ the low quality sensor is used. The model (3) together with the sensors is then

$$\begin{aligned} x_{k+1} &= Ax_k + Bw_k, & k &\geq 0, & (4)\\ y_{1,k} &= C_1 x_k + v_{1,k}, & k &\in T_{\text{lq}}, & (5)\\ y_{2,k} &= C_2 x_{k-d} + v_{2,k}, & k &\in T_{\text{hq}}, & (6)\end{aligned}$$

where $y_{1,k}$ is the low-quality measurement and $y_{2,k}$ the high-quality measurement. The variables $v_{1,k}$ and $v_{2,k}$ are white measurement noise, $\mathbf{E} v_{1,k} v_{1,k'} = \Sigma \delta_{k-k'}$, and $\mathbf{E} v_{2,k} v_{2,k'} = \sigma \delta_{k-k'}$. It is assumed that the high-quality sensor measurement $y_{2,k}$ is more accurate than $y_{1,k}$, i.e., $\sigma < \Sigma$, but delayed by d samples because of a higher processing and communication time. It is assumed that the delay of the low quality sensor can be neglected since its processing and communication time is lower than one time step. Note that $y_{1,k}$ is not defined when $k \in T_{\text{hq}}$ and $y_{2,k}$ is not defined when $k \in T_{\text{lq}}$.

We next derive a time-varying Kalman filter for (4)–(6). We rewrite the system to accommodate for the time delay d by introducing a new state vector

$$\bar{x}_k = \begin{pmatrix} x_k & x_{k-1} & \dots & x_{k-d} \end{pmatrix}^T, \tag{7}$$

so that (4)–(6) becomes

$$\begin{aligned} \bar{x}_{k+1} &= \bar{A}\bar{x}_k + \bar{B}w_k, & (8)\\ \bar{y}_k &= \bar{C}_k \bar{x}_k + \bar{v}_k. & (9)\end{aligned}$$

where,

$$\bar{A} = \begin{pmatrix} A & 0 & \dots & 0 & 0 \\ I_n & 0 & & 0 & 0 \\ 0 & I_n & & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & I_n & 0 \end{pmatrix}, \bar{B} = \begin{pmatrix} B \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{10}$$

$$\bar{C}_k = \begin{cases} \begin{bmatrix} C_1 & 0 & \dots & 0 & 0 \end{bmatrix}, & k \in T_{lq} \\ \begin{bmatrix} 0 & 0 & \dots & 0 & C_2 \end{bmatrix}, & k \in T_{hq}. \end{cases} \tag{11}$$

Introduce the following time-varying recursive Riccati equation

$$\begin{aligned} \bar{P}^*_{k+1|k} &= \bar{A}\Big[\bar{P}^*_{k|k-1} - \bar{P}^*_{k|k-1}\bar{C}_k^T \\ &\times \Big[\bar{C}_k \bar{P}^*_{k|k-1}\bar{C}_k^T + \bar{V}_k\Big]^{-1} \times \bar{C}_k \bar{P}^*_{k|k-1}\Big]\bar{A}^T + \bar{B}W\bar{B}^T, \end{aligned} \tag{12}$$

where $\bar{P}^*_{k|k-1}$ is the minimum possible covariance of the estimation error of the state $\bar{x}_k$ given the measurements up until time $k-1$. The time-varying Kalman filter that gives the optimal estimate is given by

$$\begin{aligned} \hat{\bar{x}}_{k+1|k} &= \left(\bar{A} - \bar{K}_k \bar{C}_k\right)\hat{\bar{x}}_{k|k-1} + \bar{K}_k \bar{y}_k & (13)\\ \bar{K}_k &= \bar{A}\bar{P}^*_{k|k-1}\bar{C}_k^T\left(\bar{C}_k \bar{P}^*_{k|k-1}\bar{C}_k^T + \bar{V}_k\right)^{-1}, & (14)\end{aligned}$$

where $\hat{\bar{x}}_{k+1|k}$ is the new state estimate and $\bar{K}_k$ is the Kalman gain. See [7] for further details on this filter.

B. Sensor Scheduling

How to schedule the sensors, i.e., how to choose the sets T_{lq} and T_{hq}, is described next.

The overall estimation quality is defined as the average trace of the covariance matrix of the estimation error:

$$p_{average}(k) := \frac{1}{k+1}\sum_{i=0}^{k} \text{trace} P^*_{i|i}, \tag{15}$$

where $P^*_{k|k}$ is the block of $\bar{P}^*_{k|k}$ that corresponds to the current state x_k in (7), see [7]. It is desirable to minimize $p_{average}(k)$ through a proper choice of T_{hq}. To also take communication cost into account, the following performance criterion is considered:

$$V_T(k, M) := \frac{M\lambda}{k+1} + p_{average}(k), \tag{16}$$

where M is the number of times the high-quality sensor is used over the time interval $[0, k]$ and λ is a positive parameter that is proportional to the communication cost.

The average communication cost per time sample is $\dfrac{M\lambda}{k+1}$. When only the high-quality sensor is used, the performance is $V_T(k, k) = \lambda + p_{average}(k)$, and when only the low-quality sensor is used $V_T(k, 0) = p_{average}(k)$. Thus λ is a measure of how much better (measured in resulting average error covariance) than the low-quality sensor the high-quality sensor must be for us to prefer to use it ($V_T(k, k) > V_T(k, 0)$).

It is legitimate to ask when the high-quality vision sensor is useful at all, since data from that sensor are delayed and have a communication cost. It is shown in [7] that there exists non-trivial sensor schedules that incorporates the vision sensor while minimizing V_T. The optimization problem grows exponentially with the time horizon k. To find suboptimal solutions, we restrict the presentation to periodic schedules and to a local search for so called covariance-based schedules.

1) Periodic Scheduler: For the periodic scheduler, we assume that the high-quality sensor is used with period $N \geq 1$:

$$\begin{aligned} T_{\text{hq}}(N) &= \{N-1, 2N-1, 3N-1, \dots\} \\ &= \{k \geq 0 \mid (k+1) \mod N = 0\}, \\ T_{\text{lq}}(N) &= \{0, 1, 2, \dots, N-2, N, \dots\} \\ &= \{k \geq 0 \mid (k+1) \mod N \neq 0\}. \end{aligned}$$

One can show that $\bar{P}^*_{k|k-1}$ (and $\bar{P}^*_{k|k}$) converges to an N-periodic matrix function when $k \to \infty$, under weak assumptions on the system [7]. Hence, $p_{average}(k)$ tends to a constant $p_{average}(N)$. For sufficiently large k, the optimal sensor cycle period is given by

$$N^*_k = \arg\min_N \left(\frac{\lambda}{N} + p^*_{average}(k)\right). \tag{17}$$

Note that (17) is a simple minimization problem over a finite set. The steady-state optimal period N^* is given by

$$N^* = \arg\min_N \left(\frac{\lambda}{N} + p^*_{average}(N)\right).$$

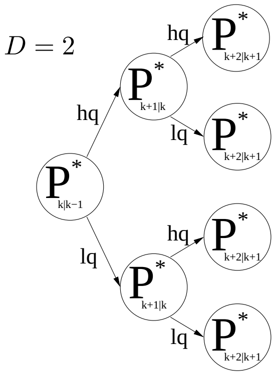


Fig. 5. Tree search for the covariance-based scheduler when $D = 2$.

2) Covariance-Based Scheduler: For the covariance-based scheduler, we let the sensor choice be given by the sensor that decreases the estimation error covariance the most in one step. Such an online scheduler is interesting to compare with the offline periodic scheduler discussed in previous section.

At a fixed time instant k, the covariance-based scheduler computes the error covariances $P^*_{k+i|k+i-1}$, $i = 0, \dots, D$, for all possible sensor combinations. The search depth D is a tuning parameter. To find the appropriate sensor schedule, the a tree search is done to find the lowest sum of $P^*_{k+i|k+i-1}$, $i = 0, \dots, D$ (adding the communication cost λ each time a high-quality measurement is used). An example of such a tree search is shown in Fig. 5 for depth $D = 2$. The schedule T_{hq} for the time interval $k, k+1, \dots, k+D$ is the path that generates the minimum. The algorithm is run when $k = 0, D, 2D, \dots$. Another alternative is to optimize the schedule at every k in a receding horizon approach, but that requires more computations. Based on the estimates of current and future position, it is also possible to let the noise covariance matrices Σ, σ, W, depend on position when computing $P^*_{k+i|k+i-1}$, as mentioned in Section II-B.

IV. Experimental Setup

The experimental setup is illustrated in Fig. 6. The system is composed of ultrasound sensors, web cameras, wireless network, fusion center, and a mobile agent to track. The web camera presents the high-quality sensor and is connected to a laptop enabling image processing and data exchange. A single camera was used in the tests described here. The ultrasound sensors use wireless sensor nodes for signal processing and data exchange. Both types of sensors communicate with a base-station placed in the same area which is used to perform the data fusion, filtering and estimation. The base-station's processing unit is connected to a wireless node in order to communicate with the ultrasound nodes over IEEE 802.15.4 and an IEEE 802.11g wireless card to communicate with the vision system's processing unit. The ultrasound sensor network consists of 16 wireless nodes with ultrasound sensors placed in the ceiling of our corridor. The web camera is placed on the wall in the area where the tests are performed. A radio-controlled car is used as the mobile agent to be located. The rest of this section describes in some more detail the ultrasound-based sensor system, the vision-based sensor system, and the sensor fusion center.

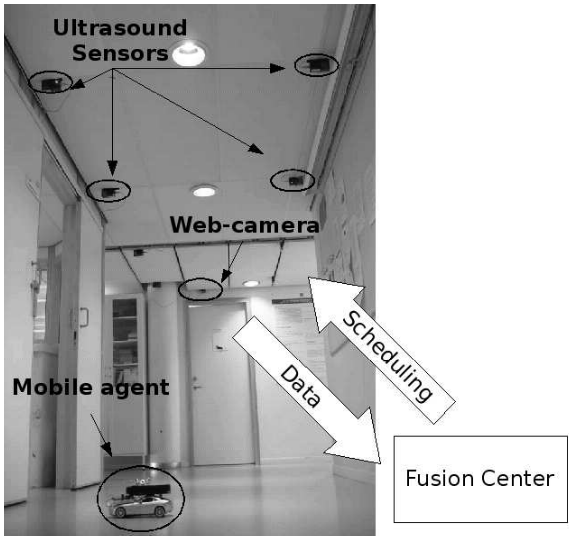


Fig. 6. Overview of the experimental setup that consists of ultrasound sensors, web cameras, wireless network, fusion center, and a mobile agent to track.

A. Ultrasound-Based Sensor System

The ultrasound transmitter and receiver circuitry are connected to wireless sensor nodes of the type Tmote Sky [8]. In order to perform the multilateration position computation, a method based on the time-delay-of-arrival (TDOA) technique was implemented. See [9] and [10] for further details on this method.

The ultrasound system works as follows. A transmitter circuitry placed on the mobile agent interact with the ultrasound receivers connected to the wireless nodes in the ceiling. With a $250\,\text{ms}$ period, the ultrasound transmitter node is simulatenously sending a wireless (IEEE 802.15.4) message and an ultrasound signal. The ultrasound receiver computes the time of flight of the ultrasound signal, using time calibration for the wireless message and the speed of sound. This gives the distances between each receiver node and the transmitter node. The position calculation is performed in the fusion center where all the distances achieved by the receiver nodes are collected and used in a multilateration computation. Filtering and outlier rejection are used to improve information.

B. Vision-Based Sensor System

A vision-based system that is composed of a Logitech Quickcam Fusion web camera [11] connected to a processing unit was developed. The processing unit performs the image acquisition, storage, and processing according to the scheduler in the sensor fusion center. The communication with the fusion center is over a IEEE 802.11g network. The output of the vision-based system is the 2D position of the mobile agent in the experimental area. It is computed through a linear transformation between pixel coordinates and world coordinates.

The image acquisition and storage is performed using the OPENCV library functions [12]. The image processing was done in MATLAB. The time spent for these three tasks is about 1 s, which is more than the communicating delay. An artificial delay was introduce to investigate the proposed filtering schemes. In the experiments presented in Section V, the artificial delay was set to 2 s, so the total delay it took from requesting a vision-based position measurement to having it available in the sensor fusion center was 3 s.

C. Sensor Fusion Center

The sensor fusion center is responsible for reading measurements from both sensors and using them according to the specified filter. It also has the task of triggering the vision-based system measurements. The ultrasound measurements are received periodically. The sensor fusion center implements the sensor fusion and scheduling algorithms of Section III.

V. Experimental Validation

This section presents some experimental results performed on the testbed. First, we discuss how some model parameters used in the filter were obtained, and then we compare the periodic and the covariance-based scheduler for some choices of tuning parameters.

A. Model Parameter Estimation

The ultrasound sensors give on average an error of 8 cm, with a variance of 12 cm^2. The vision-based system gives more precise measurements with average errors less than 2 cm, and with variance lower than 1 cm^2. These values were achieved by testing the system using fixed positions of the vehicle. Based on these measurements, we use for the model (4)–(6) $\Sigma = 12$ for the ultrasound-based system and $\sigma = 1$ for the vision-based system.

The mobile agent moves in the tests in the X-direction: Y is fixed and X is varying. The test trajectory is approximately 4.5 m long, and a single test takes about 12s. Since the measurements given by the camera are very accurate and taken at precise times, one can estimate the velocity by $V = (X_k - X_{k-1})/\Delta t$. The mobile agent starts up (first 2 s) with a velocity approximately equal to 11 cm/s, followed by an approximately constant velocity of 50 cm/s. For the final 3 s it has a constant velocity of 11 cm/s. Using these velocities, the process noises of the model 3 in Section II-A were set to $U = 0.5$ and $U = 0.05$, respectively.

B. Sensor Fusion Tests

Tests were made using the random walk and the integrated random walk models in (3). The periodic schedules and the covariance-based schedule were evaluated. In each test, 29 samples were taken with the sampling period $h = 1$ s. The performance of each filter was evaluated through the cost

$$V_E(k,M) = \frac{1}{k+1}(M\lambda + \sum_{i=0}^{k} |x_{real,i} - x_{est/raw,i}|^2), \tag{18}$$

where $x_{real,i}$ is the actual robot position at time i, and $x_{est/raw,i}$ is the measured or estimated position at time i, depending on what filter was used. The empirical cost V_E should be compared with the cost V_T in (16).

TABLE I

Optimal periodic high-quality sensor switching N^* for model 1 and 2 considering different communication cost λ.

λ	**Model 1**$_{\vert U=1.8}$	**Model 2**$_{\vert U=0.05}$
0	N^*= 9	N^*= 6, 8, 12
2000	N^*= 9, 14	N^*= ∞

We express V_T as roughly the average of V_E over several different identical experiments:

$$V_T(k,M) = \lim_{i\to\infty} \frac{1}{i} \sum_{j=1}^{i} V_E^j(k,M),$$

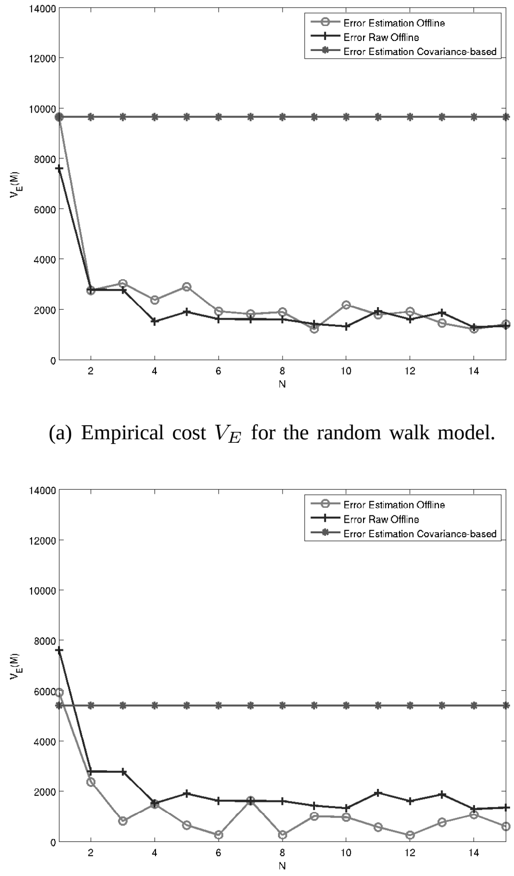
where $V_E^j(k,M)$ denotes the j-th repeated experiment and we assume the noise sequences in different experiments are independent.

Let us first consider the periodic scheduler in an experiment in which the communication cost was set to $\lambda = 0$. Fig. 7 shows the empirical cost V_E as a function of the the period N. The upper plots are derived for the random walk model and the lower plots for the integrated random walk model. The cost for the estimated positions with the periodic scheduler is shown together with the cost for the raw position measurements. The integrated random walk model (lower plots) provides better position estimates than the random walk model, which is expected since the motion of the mobile agent is closer to the integrated random walk model. It is also clear that in this case, the periodic scheduler presented in the paper gives much better estimates than the raw measurements. Fig. 7(a) indicates that the resulting performance for the proposed filter is for most values of N worse than using the direct measurements.

Also the performance of the covariance-based scheduler is shown in Fig. 7. The optimal schedule obtained was to never use the high-quality sensor in the case of the random walk model and to use $M = 12$ for the integrated random walk model. The parameter search depth parameter was tuned to $D = 4$. It can be seen Fig. 7 that the covariance-based scheduler does not give as good results as the periodic scheduler.

The optimal period N^* for communication cost $\lambda = 0$ and 2000 are given in Table I. It is natural that when the communication cost λ is large, the schedule period N^* is large (since the vision sensor is then too expensive to use).

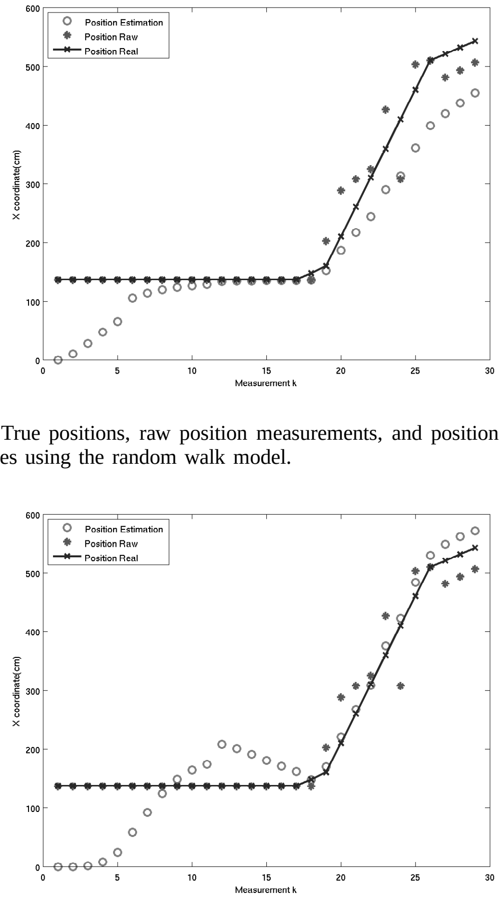
Fig. 8 shows a position trajectory. Both filtered and raw position measurements are compared to the real positions. The period for the high-quality sensor is $N = 6$. We note again the results using the integrated random walk model for the sensor fusion filter together with the periodic scheduler are good (lower plot). The initial transient in the estimation error is due to the initial error in the estimator.



(a) Empirical cost V_E for the random walk model.

(b) Empirical cost V_E for the integrated random walk model.

Fig. 7. Empirical cost V_E as a function of the the period N computed for the estimated positions with the periodic and the covariance-based schedulers. The cost for the raw position measurements are also shown. As expected, the integrated random walk model provides better position estimates than the random walk model.



(a) True positions, raw position measurements, and position estimates using the random walk model.

(b) True positions, raw position measurements, and position estimates using the integrated random walk model.

Fig. 8. Mobile agent trajectory and estimates with optimal period scheduler with $N = 6$.

VI. Conclusions

An implementation and evaluation of a localization system based on a heterogeneous sensor network was presented. The system is suitable for testing various networked control and estimation algorithms. A sensor fusion scheme with an intelligent scheduler that takes communication cost and delay into account was demonstrated with promising results. The sensor network consisted of ultrasound proximity sensors and web cameras. They were used to localize a mobile robot under sensor communication constraints.

Our ongoing work is on integrating other applications over the same wireless network and show how this type of wireless sensor infrastructure can provide a multitude of services with guaranteed quality. In doing so, it will be important to handle a variety of imperfections in the sensor scheduling and sensor fusion, such as packet losses and dynamic allocation of network resources.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, ”Wireless sensor networks: a survey” Computer Networks, vol. 38, no. 4, 2002.

[2] D. Culler, D. Estrin, and M. Srivastava, ”Overview of wireless sensor networks”, IEEE Computer, Aug 2004, special Issue in Sensor Networks.

[3] K.-E. Årzén, A. Bicchi, G. Dini, S. Hailes, K. H. Johansson, J. Lygeros, and A. Tzes. A component-based approach to the design of networked control systems. *European Journal of Control*, 13(2–3), 2007.

[4] Y. Bar-Shalom and X. R. Li, Estimation and Tracking: Principles, Techniques, and Software, Artech House, 1993.

[5] M. Athans, ”On the determination of optimal costly measurement strategies for linear stochastic systems”, Automatica, vol. 8, pp. 397-412, 1972.

[6] W. Wu and A. Arapostathis, ”Optimal control of markovian systems with observations cost: Models and LQG controls”, in Proceedings of American Control Conference, Portland, 2005, pp. 294-299.

[7] H. Sandberg, M. Rabi, M. Skoglund and K. H. Johansson. ”Estimation over heterogeneous sensor networks”. IEEE Conference of Decision and Control, 2008.

[8] Tmote Sky Datasheet - Moteiv corporation, 2007, http://www.sentilla.com/moteiv-endoflife.html

[9] R. Bucher and D. Misra. ”A Synthesizable VHDL Model of the Exact Solution for Three-dimensional Hyperbolic Positioning System”, in VLSI Design, Volume 15, 2002, Pp. 507-520.

[10] N. B. Priyantha, ”The Cricket Indoor Location System”, PhD Thesis - MIT, June 2003

[11] Logitech QuickCam Fusion, Logitech, http://www.logitech.com, 2008.

[12] OPENCV Library - http://opencvlibrary.sourceforge.net/, 2008

[13] B.D.O. Anderson and J.B.Moore, *Optimal Filtering*. Dover publications, 2005.