



LUND UNIVERSITY

Experimental Evaluation of a Distributed Kalman Filter Algorithm

Alriksson, Peter; Rantzer, Anders

2007

[Link to publication](#)

Citation for published version (APA):

Alriksson, P., & Rantzer, A. (2007). *Experimental Evaluation of a Distributed Kalman Filter Algorithm*. 3562-3567. Paper presented at 46th IEEE Conference on Decision and Control, 2007, New Orleans, LA, United States.

Total number of authors:

2

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Experimental Evaluation of a Distributed Kalman Filter Algorithm

Peter Alriksson and Anders Rantzer

Abstract—This paper evaluates the performance of a distributed Kalman filter applied to an ultrasound based positioning application with seven sensor nodes. By distributed we mean that all nodes in the network desires an estimate of the full state of the observed system and there is no centralized computation center after deployment. Communication only takes place between neighbors and only once each sampling interval. The problem is solved by communicating estimates between neighbors and then forming a weighted average as the new estimate. The weights are optimized to yield a small estimation error covariance in stationarity. The minimization can be done off line thus allowing only estimates to be communicated. In the experimental setup the distributed solution performs almost as good as a centralized solution. The proposed algorithm also proved very robust against packet loss.

I. INTRODUCTION

As battery and processing power of nodes in sensor networks increases the possibility of more intelligent estimation schemes become more and more important. The use of sensor networks was first driven by military applications, but with cheaper technology many other areas could make use of sensor networks, see for example [1] and [2]. Advantages with wireless sensor networks typically include increased flexibility and more robustness, as more than one unit is performing the same task.

However these advantages come with a cost: When communicating over wireless channels packet loss becomes a major problem and decentralized algorithms tend to be more complex than centralized solutions.

To illustrate the pros and cons a target tracking application will be considered. In simple target tracking applications, the task is to estimate the position of an external object. In some situations measurements are taken from spatially separated locations and an estimate is needed at each location. Ideally all measurements should be used at all locations, however this may require high bandwidth communication channels between all nodes.

To reduce the required bandwidth a distributed solution where only the position estimate is communicated among neighbors will be considered.

The problem where estimates are communicated has been given great attention in the literature. In for example [3] a decentralized Kalman filter was proposed. However, this algorithm requires every node to be able to communicate with every other node, which might not be possible. An alternative approach is to only allow nodes to communicate with their

neighbors. As opposed to the case where measurements are communicated no routing is required when estimates are used as information carriers.

Without direct communication between all nodes a new problem is introduced, namely how to combine estimates from just neighboring nodes. To optimally combine two estimates one has to know the mutual information between them. Computing this quantity for a general communication graph is a difficult task, that requires global knowledge of the topology. In the case of a loop-free graph the problem was solved in [4] by introduction of a channel filter. This approach was used in a coordinated search strategy application, see [5]. The problem has also been studied intensively in the dynamic consensus literature, see for example [6] and the references therein. In [7] a similar problem was studied, but for scalar systems.

This paper is organized as follows. Section II and III gives a brief overview of the theory used in the rest of the paper. The main part of the paper constitutes of sections IV, V and VI where the experimental setup together with the results are presented.

A. Target Tracking

In real target tracking applications sophisticated radar systems are used to take measurements of the position of a target moving in three dimensions. Here a simplified setup is used where the target to be tracked is a mobile robot moving in two dimensions.

Localization of mobile robots can be performed with a number of techniques. In laboratory experiments it is common to use vision, e.g., a ceiling-mounted camera combined with an image-processing system. Another possibility is dead-reckoning using a high-precision inertial measurement unit on board the robot. A problem with dead reckoning-based approaches, however, is that they do not use feedback and thus unmeasurable disturbances will cause position errors that cannot be compensated for. In an outdoor environment GPS would be another possibility.

The localization approach chosen here is based on ultrasound. The basic idea is to transmit a wireless radio packet simultaneously with an ultrasound pulse from each sender node. The receiver nodes measure the difference in time of arrival between the radio packet and the ultrasound pulse and can in this way calculate their distance to the sender node. By combining, or fusing, several distance measurements an estimate of the position can be obtained.

Two main approaches exist, [8]. In an *active mobile* system the infrastructure, in this case the sensor network, has receivers at known locations, which estimate distances

This work was partially funded by the RUNES Integrated Project contract IST-2004-004536.

P. Alriksson and A. Rantzer is with the Department of Automatic Control, Faculty of Engineering, Lund University, 221 00 Lund, Sweden {peter, rantzer}@control.lth.se

to a mobile device based on active transmissions from the device. These distances are then reported to a central node for processing. Examples of this approach are the Active Badge [9], and the Ubisense [10] systems. In a *passive mobile* system, instead, the infrastructure has active beacons at known positions that periodically transmits signals to a passive mobile device. The mobile device then use these signals to compute its current location. The most famous example of this is the Cricket system [11].

An advantage of the active approach is that it is more likely to perform accurate tracking than the passive approach. The passive approach, on the other hand, scales better with the number of mobile devices. Since the main objective here is for the sensor network to handle the localization, an approach similar to the active one was chosen. However here no central computation center is used.

As the robot has a practically unlimited power supply compared to the nodes in the sensor network, it is reasonable to assume that the robot can reach all nodes in the network with high probability. Thus all nodes can measure the distance to the robot at each sampling time and the robot can transmit its expected movement to the sensor network. The nodes however operate under severe power restrictions, thus only neighbor to neighbor communication is possible.

II. MATHEMATICAL FORMULATION

The motion of the robot is described by a discrete-time linear model of the following form, derived in [12].

$$x(k+1) = Ax(k) + Bu(k) + v(k) \quad (1)$$

Here $x(k) \in \mathbf{R}^n$ denotes the state of the system, $u(k) \in \mathbf{R}^m$ a known input and $v(k) \in \mathbf{R}^n$ a stochastic disturbance. The disturbance is assumed to be a white zero mean Gaussian process with covariance defined below. Note that in this simplified setup, it is assumed that the external input $u(k)$ is known to all nodes. As mentioned in Section I-A this assumption is satisfied in the experimental setup.

The process is observed by N agents each with some processing and communication capability. The agents are labeled $i = 1, 2, \dots, N$ and form the set V . The communication topology is modeled as a graph $G = (V, E)$, where the edge (i, j) is in E if and only if node i and node j can exchange messages. The nodes to which a node communicates are called neighbors and are contained in the set N_i . Note that node i is also included in the set N_i .

Each node observes the process (1) by a measurement $y_i(k) \in \mathbf{R}^{p_i}$ of the following form

$$y_i(k) = C_i x(k) + e_i(k) \quad (2)$$

where $e_i(k) \in \mathbf{R}^{p_i}$ is a white zero mean Gaussian process. The measurement- and process disturbances are correlated according to

$$E \begin{bmatrix} v(k) \\ e_1(k) \\ \vdots \\ e_N(k) \end{bmatrix} \begin{bmatrix} v(l) \\ e_1(l) \\ \vdots \\ e_N(l) \end{bmatrix}^T = \begin{bmatrix} R_v & 0 & \dots & 0 \\ 0 & R_{e11} & \dots & R_{e1N} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & R_{eN1} & \dots & R_{eNN} \end{bmatrix} \delta_{kl} \quad (3)$$

where $\delta_{kl} = 1$ only if $k = l$. Note that this is a heterogeneous setup where each agent is allowed to take measurements of arbitrary size and precision. Further the disturbances acting on the measurements are allowed to be correlated.

Each node is only allowed to communicate with its neighbors and only once between each measurement. Further the only assumption made on the graph structure is that it is connected, other assumptions such as requiring it to be loop free are not necessary. No node is superior to any other and thus after deployment no central processing is allowed.

The goal is to make sure that every node in the network has a good estimate $\hat{x}_i(k)$ of the state $x(k)$.

III. DISTRIBUTED KALMAN FILTER

When constructing an algorithm based on estimates instead of measurements care must be taken on how to combine estimates in a good way. The problem is that estimates in different nodes are not independent, as they contain the same process noise, and possibly also the same measurement information. To optimally combine two estimates the mutual information must be subtracted.

For a graph with loops, two nodes can not compute the mutual information by just using local information. Information can for example travel from node A to node C and then to node B. When node A and B are to compute their mutual information they may not be aware of the coupling through C.

To solve the problem for a general communication topology neighboring estimates are weighted so that the error covariance of the merged estimate is minimized. This approach will not give the optimal solution in general, but is applicable to graphs with loops. Weighted averaging can be seen as a generalization of the two-sensor track-fusion algorithm presented in [13]. There is great freedom when choosing both how to weigh local measurements and neighboring estimates. In [14] a procedure aimed at minimizing the covariance of the estimation error is presented, but other objectives such as minimizing the amount of communication for a given accuracy could also be considered.

A. On-Line Computations

The algorithm consists of the two traditional estimation steps, measurement update and prediction together with an additional step where the nodes communicate and merge estimates. We will refer to an estimate after measurement update as local and after the communication step as regional.

1) Measurement update

The local estimate $\hat{x}_i^{local}(k|k)$ is formed by the predicted regional estimate $\hat{x}_i^{reg}(k|k-1)$ and the local measurement $y_i(k)$

$$\hat{x}_i^{local}(k|k) = \hat{x}_i^{reg}(k|k-1) + K_i [y_i(k) - C_i \hat{x}_i^{reg}(k|k-1)] \quad (4)$$

where K_i is computed off-line using for example the procedure presented in [14]. The predicted estimate at time zero is defined as $\hat{x}_i^{reg}(0|-1) = \hat{x}_0$ where $\hat{x}_0$ is the initial estimate of $x(0)$.

2) Merging

First the agents exchange their estimates over the communication channel. This communication is assumed to be error and delay free. The merged estimate $\hat{x}_i^{reg}(k|k)$ in node i is defined as a linear combination of the estimates in the neighboring nodes N_i .

$$\hat{x}_i^{reg}(k|k) = \sum_{j \in N_i} W_{ij} \hat{x}_j^{local}(k|k) \quad (5)$$

The weighting matrices W_{ij} could for example be chosen using the procedure described in [14].

3) Prediction

Because the measurement- and process noises are independent the prediction step only includes

$$\hat{x}_i^{reg}(k+1|k) = A\hat{x}_i^{reg}(k|k) + Bu(k) \quad (6)$$

IV. EXPERIMENTAL SETUP

To generate measurements, corresponding to (2), an ultrasound based system together with trilateration will be used.

The stationary sensor nodes are each equipped with an ultrasound receiver and the mobile robot is equipped with an ultrasound transmitter. The stationary sensor nodes are implemented as Tmote Sky sensor network nodes together with a small ultrasound receiver circuit interfaced to the node via an AD converter, see Fig. 1.

Both the ultrasound transmitters and receivers are designed to be isometric, i.e., to transmit and receive in the full 360° degree plane.

The robot used in the experiments is a dual-drive robot developed in Lund. It is equipped with three Atmel AVR Mega16 processors and one TMote Sky node, see Fig. 1. Two AVR processors are used to control the wheel speeds using PI-controllers. The remaining AVR is used to generate ultrasound. For a detailed description of the hardware see [15]. Throughout all experiments the robot is controlled using a wireless joystick.

A. Ultrasound Based Localization

The implemented localization method works according to the following principles. At the beginning of each measurement cycle, the robot transmits a broadcast radio message to alert the nodes of the incoming ultrasound pulse. After a fixed time the robot then emits an ultrasound pulse. When the radio message reaches the node, it starts to sample the ultrasound microphone. Then the stationary nodes detect the beginning of the pulse using a moving median filter of length three.

The sample index where the pulse was detected is proportional to the distance between the stationary nodes and the robot when the pulse was emitted. If the speed of sound, the sampling interval in the nodes and the fixed delay between ultrasound- and radio transmission are known the actual distance can be computed. The position can then be computed using trilateration.

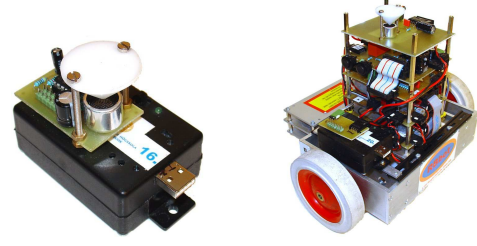


Fig. 1. Stationary sensor network node with ultrasound receiver circuit and robot with ultrasound sender. The nodes are packaged in a plastic box to reduce wear.

B. Trilateration

Trilateration is a method to find the position of an object based on distance measurements to three objects with known positions. In three dimensions the problem has two solutions, however the correct one can usually be determined from physical considerations. The basic problem is to find a solution $[p_x \ p_y \ p_z]^T$ to the following three nonlinear equations

$$\begin{aligned} (p_x - p_{x1})^2 + (p_y - p_{y1})^2 + (p_z - p_{z1})^2 &= d_1^2 \\ (p_x - p_{x2})^2 + (p_y - p_{y2})^2 + (p_z - p_{z2})^2 &= d_2^2 \\ (p_x - p_{x3})^2 + (p_y - p_{y3})^2 + (p_z - p_{z3})^2 &= d_3^2. \end{aligned}$$

where p_{xi} , p_{yi} and p_{zi} are known positions of the nodes and d_i is the distance from node i to the object to be positioned. The problem can be transformed to a system of two linear equations and one quadratic equation by e.g. subtracting the second and third equation from the first, see [16] for a detailed analysis.

An alternative more geometric approach was taken in [17] where the problem is solved using Cayley-Menger determinants. This approach has the benefit of a geometric interpretation of the solution in terms of volumes, areas and distances. Also the error analysis with respect to e.g. distance errors is simplified.

As the robot is assumed to only move in the xy -plane, the problem can be reduced to a set of two linear equations. The two linear equations will always have a solution unless all three known points are positioned on a line. The two linear equations define two lines, see Fig. 2, which can be represented as

$$a_0 y = a_1 + a_2 x \quad (7)$$

$$b_0 y = b_1 + b_2 x \quad (8)$$

where

$$\begin{aligned} a_0 &= 2(p_{y2} - p_{y1}) \\ a_1 &= d_1^2 - d_2^2 + p_{y2}^2 - p_{y1}^2 + p_{x2}^2 - p_{x1}^2 - 2p_z(p_{z2} - p_{z1}) \\ a_2 &= 2(p_{x1} - p_{x2}) \\ b_0 &= 2(p_{y3} - p_{y1}) \\ b_1 &= d_1^2 - d_3^2 + p_{y3}^2 - p_{y1}^2 + p_{x3}^2 - p_{x1}^2 - 2p_z(p_{z3} - p_{z1}) \\ b_2 &= 2(p_{x1} - p_{x3}). \end{aligned}$$

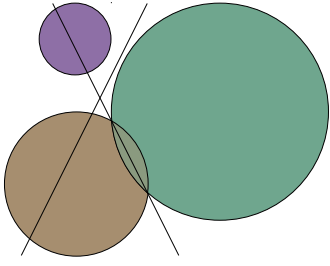


Fig. 2. Two lines defining the solution to the trilateration problem. Each line corresponds to one pair of circles. For clarity only two of the three possible lines are shown. This setup shows both overlapping and non-overlapping circles. Sensor nodes are located at the center of each circle.

Note that the z-coordinate p_z of the robot is assumed to be known, as it is only moving in the xy -plane. The intersection point of these two lines constitute the trilaterated position $[p_x^{tri} \ p_y^{tri}]^T$. Even though the three circles do not intersect in one point, the algorithm still provides a reasonable result. For a detailed discussion on how errors both in distance measurements and node positions influence the trilateration result, see [16] and [17].

C. Camera Based Localization

To evaluate the distributed ultrasound based localization system an independent localization system is needed. In the experimental setup a camera based system was used. The robot was equipped with two markers to aid the vision system. The camera system consists of one fixed mounted camera with a resolution of 640×480 pixels. Each marker is located in the image using an algorithm based on a Harris corner detector, see [18]. If the robot is assumed to move in a plane, an image coordinate can be transformed to a point p^{cam} in the plane using a linear transformation. The heading can then be computed using the two markers. In the experimental setup used, the vision based localization system had an accuracy of approximately 1cm.

D. Choice of Noise Covariance Matrices

The choice of measurement- and process covariance matrices (R_v and R_e) are crucial to the performance of the algorithm. The process noise covariance matrix determines the confidence in the model. Here R_v will be used as a tuning parameter to trade off between noise rejection and trust in the model.

The measurement noise covariance matrix R_e in the experimental setup is a symmetric 14×14 matrix, thus it has 105 free parameters. The somewhat standard choice of a diagonal matrix does not apply here as the trilaterated position measurements are highly correlated. Instead, using a test trajectory, R_e was estimated as a normalized version of

$$(\hat{R}_e)_{ij} = \frac{1}{T} \sum_{k=0}^{T-1} (p_i^{tri}(k) - p^{cam}(k))(p_j^{tri}(k) - p^{cam}(k))^T \quad (9)$$

where $p_i^{tri}(k)$ is the trilaterated position in node i at time k and $p^{cam}(k)$ is the position generated from the vision

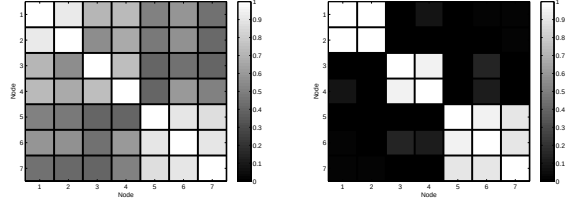


Fig. 3. Elements of the RMS-correlation matrix ρ associated with the x-position for measurements (left) and simulations (right). The correlation pattern introduced by trilateration is clearly visible.

system. The diagonal elements of $\hat{R}_e$ are thus the squared RMS-value of the trilateration error. The advantage of using squared RMS-values instead of for example the variance is that systematic errors are reflected in the RMS-value. As trilateration is a nonlinear operation, the error is dependent of the position. Thus the result of (9) is dependent on the specific trajectory the robot has followed. Ideally one would want to allow R_e to vary with time, however this would make the weights W time varying thus making it more complicated to compute them off-line.

To illustrate the correlation pattern let us define the RMS-correlation matrix as

$$\rho_{ij} = \frac{(\hat{R}_e)_{ij}}{\sqrt{(\hat{R}_e)_{ii}(\hat{R}_e)_{jj}}} \quad (10)$$

In Fig. 3 elements associated with the x -position for a typical trajectory are shown for both measurements (left) and simulations (right). The correlation pattern caused by trilateration is clearly visible both in the experimental and simulated data.

V. COMMUNICATION PROTOCOL

As discussed in Section I-A it is assumed that the robot can reach all nodes in the network with radio packets. Thus the robot constitutes a global clock which simplifies the communication protocol.

As a single node can only measure the distance to the robot, nodes need to form groups of three to be able to perform trilateration. One node in each group collects distance measurements from the other two and then computes the position of the robot using trilateration. To reduce utilized bandwidth, distance measurements and position estimates are transmitted in the same package.

The protocol implemented is illustrated by the schedule in Fig. 4.

- 1) At the beginning of each period the robot asks the wheel controllers for the current wheel velocities.
- 2) The robot sends a broadcast message to all nodes with its expected movement, that is previous heading estimate and wheel velocities. This corresponds to the term $Bu(k)$ in (1).
- 3) After sending the packet the robot updates its heading estimate based on wheel speeds and position estimates received from the network at step 7.

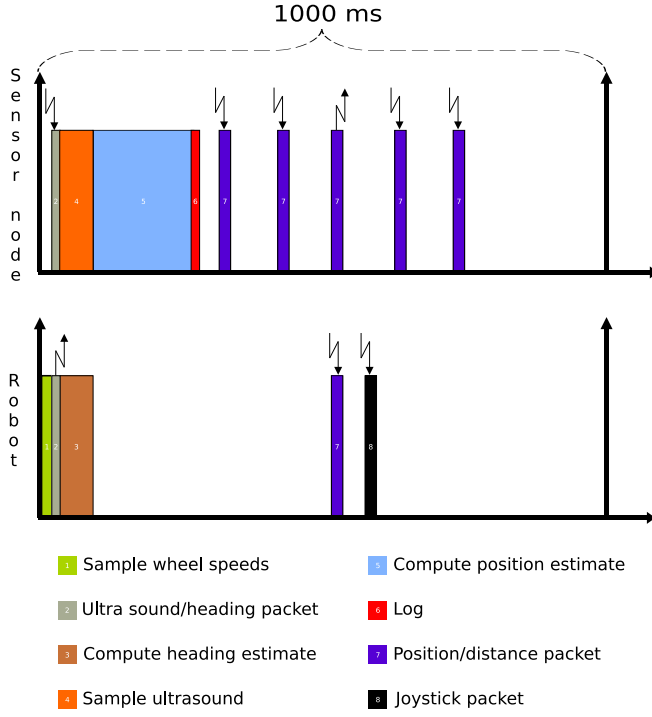


Fig. 4. Communication schedule used in experiments.

- 4) When the packet transmitted at step 2 reaches a node, it starts to sample the incoming ultrasound pulse. The sampling is interrupted when the edge of the pulse is reached.
- 5) The nodes compute their new estimate based on information received during the previous time interval.
- 6) Each node logs data using a wired network to reduce interference.
- 7) After a specified time based on its identity number the nodes broadcast their new position estimate together with the distance measurement taken at step 4. These messages are then received by neighboring nodes, including the robot if it is in range.
- 8) Finally the robot receives commands from a joystick used to control it.

In the implementation, data is transmitted after the prediction step instead of between the update- and prediction step as described in Section III. However it is straightforward to modify the algorithm for this scenario.

VI. EXPERIMENTAL RESULTS

To evaluate performance, the root mean square (RMS) of the difference to the position estimated by the vision system p^{cam} will be used.

When evaluating the impact of different parameters and design choices it is crucial that experiments are repeatable. However, as the radio environment where the experiments were performed is highly non-stationary, repeatability was a big problem. This issue was resolved by studying estimates generated in Matlab using logged trilateration-, wheel speed- and packet arrival data as input. The average RMS difference

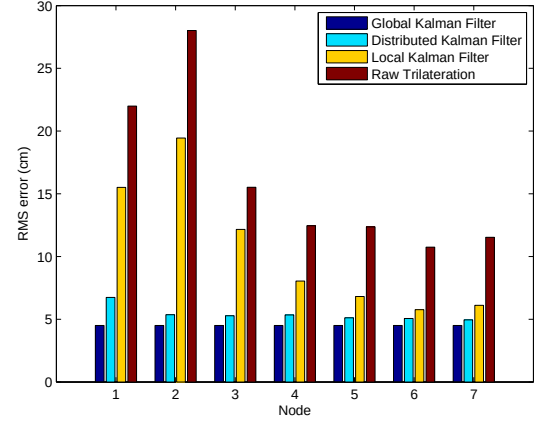


Fig. 5. RMS error ($\hat{p} - p^{cam}$) for different types of estimation schemes. Global refers to a Kalman filter that has access to all information without any delay. Distributed is the proposed distributed Kalman filter and Local refers to a Kalman filter that only uses local trilateration information. As a comparison raw trilateration is also plotted. The big difference in trilateration accuracy among nodes is due to the relative position of a node compared to the robot trajectory.

for estimates generated in Matlab compared to real experiments is approximately 1 cm. This error is mostly due to quantization effects in the logging of wheel speeds.

The main results of the experiments are summarized in Fig. 5 where the RMS error for different types of estimation schemes are shown. Global Kalman filter means that the filter has access to trilateration information from all nodes without any delay. Distributed denotes the proposed algorithm, whereas in the local case only local trilateration information is used. As a comparison the raw trilaterated estimate is also plotted. Note that even in the local case a node needs to communicate with two of its neighbors to be able to perform trilateration. The big difference in trilateration accuracy among nodes is due to the relative position of a node compared to the robot trajectory.

For both the global- and distributed Kalman filter to perform well it is crucial that the relation between the diagonal elements of R_e is correct. To achieve this, $\hat{R}_e$ was computed based on the same trajectory as the RMS errors in Fig. 5.

Examining the results, we can draw the conclusion that the distributed algorithm performs almost as good as a global solution. One can also note that the performance of the local estimator is very close to that of both the global and distributed schemes in e.g nodes 6 and 7 where the measurement accuracy is high. Using a permutation test [19] differences between the global- and distributed Kalman filter could only be verified at a 95% confidence level in node 1. Using the same test, differences between the local- and distributed Kalman filter could not be verified in nodes 6 and 7. The permutation test used here is somewhat conservative as it does not utilize the correlation between different estimates generated from the same data.

One critical issue for an algorithm that utilizes a wireless communication channel is sensitivity to packet loss. In the results presented in Fig. 5 the average packet loss probability

was approximately 10%. To further investigate how sensitive the proposed algorithm is to packet loss, a number of simulations were performed. In this study two different ways of handling lost packets were investigated: to use the last received packet and to use the local estimate. In Fig. 6 the average RMS error among the seven nodes is plotted as a function of packet loss for the two different methods. As a comparison the average RMS error for raw trilateration and local estimation are also shown. To isolate the errors caused by the distributed Kalman filter algorithm, packets used in the trilateration computations were left unaffected by the increased packet loss probability. Therefore, the errors for raw trilateration and local estimation is unaffected by the packet loss. From Fig. 6 it is apparent that using the

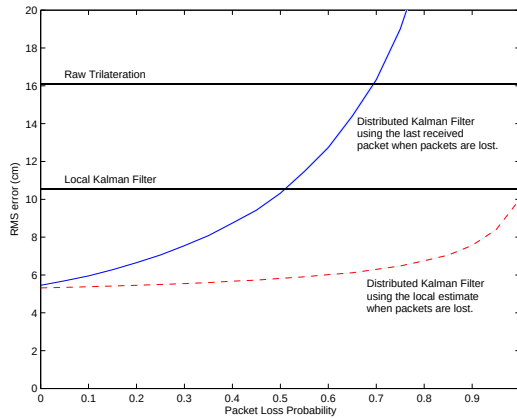


Fig. 6. Average RMS error ($\hat{p} - p^{cam}$) as a function of packet loss for the distributed Kalman filter when the last received packet is used when a packet is lost (solid) and when the local estimate is used (dashed). As a comparison the average RMS error for raw trilateration and local estimation is also shown.

local estimate when a packet is lost is preferable. If this approach is used the performance of the distributed solution approaches the local solution as the packet loss approaches one for this example.

VII. CONCLUSIONS

In this paper an optimization based algorithm for distributed estimation is evaluated experimentally. The algorithm is based on standard Kalman filtering results and then extended with one step where nodes merge their estimates. The estimates are merged by a weighted average approach.

The algorithm applies to a broad category of communication topologies, including graphs with loops. The weights are optimized off-line allowing only estimates to be communicated among the nodes. All communication is restricted to neighboring nodes, which allows the algorithm to scale.

An experimental evaluation was done to demonstrate how the proposed algorithm performs in an uncertain environment where, for example packets are lost and different nodes are not perfectly synchronized in time. The scenario chosen is one where seven nodes in a sensor network estimate the position of a mobile robot using ultrasound. It was concluded that the performance in RMS sense of the proposed algorithm

was very close to the performance of a optimal global solution. Also the distributed Kalman filter proved to be very insensitive to packet loss, which is of great importance when dealing with wireless communication links. As presented in Fig. 6 the performance degradation at for example 10% and 50% packet loss are only 1.1% and 9.4% respectively.

VIII. ACKNOWLEDGMENT

This work was partially funded by the RUNES Integrated Project contract IST-2004-004536.

REFERENCES

- [1] D. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.
- [2] R. Viswanathan and P. Varshney, "Distributed detection with multiple sensors part i. fundamentals," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 54–63, 1997.
- [3] H. Durrant-Whyte, B. Rao, and H. Hu, "Toward a fully decentralized architecture for multi-sensor data fusion," *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pp. 1331–1336 vol.2, 1990.
- [4] S. Grime, H. F. Durrant-Whyte, and P. Ho, "Communication in decentralized data-fusion systems," in *In Proc. American Control Conference*, 1992, pp. 3299–3303.
- [5] F. Bourgault and H. F. Durrant-Whyte, "Communication in general decentralized filters and the coordinated search strategy," in *The 7th International Conference on Information Fusion*, July 2004.
- [6] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [7] A. Speranzon, C. Fischione, and K. H. Johansson, "Distributed and collaborative estimation over wireless sensor networks," in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, December 2006, pp. 1025–1030.
- [8] A. Smith, H. Balakrishnan, M. Goraczko, and N. B. Priyantha, "Tracking Moving Devices with the Cricket Location System," in *2nd International Conference on Mobile Systems, Applications and Services (Mobisys 2004)*, Boston, MA, June 2004.
- [9] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The Active Badge Location System," *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, 1992.
- [10] J. Cadman, "Deploying commercial location-aware systems," in *Proc. Fifth International Conference on Ubiquitous Computing*, October 2003.
- [11] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket location-support system," in *Proc. Sixth ACM MOBICOM Conf.*, Boston, MA, August 2000.
- [12] P. Alriksson, "Distributed Kalman filtering using weighted averaging - theory and experiments," 2007, unpublished report. [Online]. Available: <http://www.control.lth.se/publications/>
- [13] Y. Bar-Shalom and L. Campo, "The effect of common process noise on two-sensor fused-track covariance," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 803–805, November 1986.
- [14] P. Alriksson and A. Rantzer, "Distributed Kalman filtering using weighted averaging," in *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, Kyoto, Japan, July 2006.
- [15] K.-E. Årzén, M. Ohlin, A. Cervin, P. Alriksson, and D. Henriksson, "Holistic simulation of mobile robot and sensor network applications using TrueTime," in *Proceedings of the European Control Conference*, Kos, Greece, July 2007.
- [16] D. E. Manolakis, "Efficient solution and performance analysis of 3-d position estimation by trilateration," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 4, pp. 1239–1249, 1996.
- [17] F. Thomas and L. Ros, "Revisiting trilateration for robot localization," *IEEE Transactions on Robotics*, vol. 21, no. 1, pp. 93–102, 2005.
- [18] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, 1988, pp. 147–151.
- [19] D. S. Moore and G. P. McCabe, *Introduction to the Practice of Statistics*, 5th ed. W H Freeman, 2006.