Distributed Receding Horizon Kalman Filter

J. M. Maestre, P. Giselsson and A. Rantzer

***Abstract*— In this paper a distributed version of the Kalman filter is proposed. In particular, the estimation problem is reduced to the optimization of a cost function that depends on the system dynamics and the latest output measurements and state estimates which is distributed among the agents by means of dual decomposition. The techniques presented in the paper are applied to estimate the position of mobile agents.**

I. Introduction

In recent years the proliferation of small microcontrollers with wireless communication capabilities has made possible the implementation of advanced control and estimation strategies in a distributed way. As a consequence, there has been a growing interest of the control community in the research of distributed systems to face the new challenges that appear [17], [9]. Issues such as the communicational burden of the different distributed architectures play a very important role. For example, the battery life of motes depend specially on the time the radio is on and on the number of messages sent [10].

Behind the concept of distributed systems there is a very basic idea: to divide the overall system into several smaller subsystems, each governed by a different agent which may or may not share information with the rest. Examples of these situations are large scale systems or networked systems such as traffic, water or power networks [11]. In this situation it becomes very important to have schemes that allow the distributed estimation of the state.

The most common approach to estimate the state of stochastic systems is the Kalman filter [7], developed in 1960 and named after his discoverer. The Kalman filter is the optimal state estimator for unconstrained linear systems subject to gaussian state and output noise. It is not possible to apply directly the centralized Kalman filter to a multiagent problem unless there is a node in the network that receives all the information. For example in [14], it can be seen how a central agent gathers the information from the moving devices and then distributes the position estimation back to them. An alternative is to calculate a decentralized version of the Kalman filter that takes into account the communications restrictions[6], [14].

In this paper we follow a different approach to solve the estimation problem in a distributed manner. First, the Kalman filter is posed as a dynamic programming problem [3]. Then, the resulting optimization problem is distributed among the agents using dual decomposition. This idea has been successfully applied to distributed control in [12] and [4]. Given that the observation problem is the dual of the control problem, it is natural to apply and enhance the techniques presented in these papers to deal with the state estimation problem.

In this context, the application of state estimation schemes to problems in which the state represents the position of an object is very attractive [1], [6]. The localization of moving entities, such as robots or people, is important for many applications. Military applications in which the goal is to track a target that moves in a distributed sensor environment are the typical examples. Other examples in which these techniques play an important role would be smart homes [8], in which it is basic to know where the inhabitants of the house are in order to control the heating and the lights and traffic and speed control. For this reason, they constitute good applications for the distributed state estimation algorithm that we present.

The outline of the paper is as follows. In section II the problem is formulated. Section III explains how dual decomposition can be used to distribute the problem among the agents involved. In section IV the techniques presented in the previous sections are applied in a simulation example. Finally, conclusions and future work are presented.

II. Problem Formulation

In this section we present a moving horizon estimation strategy that solves approximately the Kalman filter. Let us consider the following uncertain distributed linear system

$$\begin{aligned} x_i(\tau+1) &= A_{ii}x_i(\tau) + w_i(\tau) \\ y_i(\tau) &= \sum_{j=1}^{J} C_{ij}x_j(\tau) + v_i(\tau) \end{aligned} \tag{1}$$

where $x_i(\tau) \in \mathbb{R}^{n_i}$, $y_i(\tau) \in \mathbb{R}^{q_i}$, $w_i(\tau) \in \mathbb{R}^{n_i}$ and $v_i(\tau) \in \mathbb{R}^{q_i}$ are the state, measurable output, state noise and measurement noises of the i-th subbsytem respectively. The state and measurement noises are characterized by a normal distribution with zero mean and variances Q_i and R_i respectively; that is, $w_i(\tau)$ is a $N(0, Q_i)$ and $v_i(\tau)$ is a $N(0, R_i)$. From a centralized point of view the system is

$$\begin{aligned} x(\tau+1) &= Ax(\tau) + w(\tau) \\ y(\tau) &= Cx(\tau) + v(\tau) \end{aligned} \tag{2}$$

Financial support from DH Compliant project grant $MITC-09-TSI-020100-2009-359$, is gratefully acknowledged.

J. M. Maestre is with the department of Systems and Automation Engineering, University of Seville pepemaestre@cartuja.us.es.

Pontus Giselsson and A. Rantzer is with the LTH Automatic Control Department, University of Lund pontusg@control.lth.se, anders.rantzer@control.lth.se.

where

$$\begin{array}{l} x(\tau) = [x_1(\tau)\ x_2(\tau)\ \dots\ x_J(\tau)]^T \in \mathbb{R}^n \\ y(\tau) = [y_1(\tau)\ y_2(\tau)\ \dots\ y_J(\tau)]^T \in \mathbb{R}^q \\ w(\tau) = [w_1(\tau)\ w_2(\tau)\ \dots\ w_J(\tau)]^T \in \mathbb{R}^n \\ v(\tau) = [v_1(\tau)\ v_2(\tau)\ \dots\ v_J(\tau)]^T \in \mathbb{R}^q \\ n = \sum_i n_i,\ q = \sum_i q_i. \end{array}$$

Note that $w(\tau)$ is a $N(0, Q)$ with $Q = diag(Q_i)$ for $i = 1, \dots, J$ and $v(\tau)$ is a $N(0, R)$ with $R = diag(R_i)$ for $i = 1, \dots, J$.

From the point of view of probability theory, a state estimator attempts to reconstruct the a posteriori distribution $p(\hat{x}(\tau)|Y(0 : \tau))$, which is the probability that the state of the system is $\hat{x}(\tau)$ given measurements $Y(0 : \tau) = \{y(0), \dots, y(\tau)\}$. It is also possible to calculate the joint probability for a trajectory of state values, for example $p(\hat{X}(0 : \tau)|Y(0 : \tau))$. It is clear that if the distribution can be calculated then it is possible to obtain an estimate that maximizes it. The purpose of this paper is to build a distributed version of the Kalman filter, which is the optimal state estimator for unconstrained, linear systems subject to normally distributed state and measurement noise. In this case, the problem of obtaining an estimate that maximizes these probability density functions can be reduced to a dynamical programming problem. See [1] or [13] to obtain more details. In particular, the maximization of $p(\hat{X}(\tau - N : \tau)|Y(\tau - N : \tau))$ is equivalent to the minimization of a quadratic programming problem that will be presented in this section, but first let us define the quadratic function:

$$\begin{array}{l} V^N(\hat{X}(\tau - N : \tau)) = \\ \sum\limits_{k=\tau-N+1}^{\tau} \frac{1}{2}(y(k) - C\hat{x}(k))^T R^{-1}(y(k) - C\hat{x}(k)) \\ + \sum\limits_{k=\tau-N}^{\tau-1} \frac{1}{2}(\hat{x}(k+1) - A\hat{x}(k))^T Q^{-1}(\hat{x}(k+1) - A\hat{x}(k)) \\ + \Phi(\hat{x}(\tau - N)), \end{array} \tag{3}$$

where $\Phi(\hat{x}(\tau - N))$ is a term to weight the uncertainty of the first state estimated in the window. Note that, with a small abuse of notation, V^N is a function only of $\hat{X}(\tau - N : \tau)$ because the terms given by $y(k)$ are measurements whose values are available and not variables that have to be calculated.

Remark 1: Note that equation (3) can expressed as the sum of a stage cost for each estimate but the last one, which value is calculated through the terminal cost. According to this,

$$V^N(\hat{X}(\tau - N : \tau)) = \sum_{k=\tau-N+1}^{\tau} l(\hat{x}(k)) + \Phi(\hat{x}(\tau - N)). \tag{4}$$

Remark 2: The terminal cost in equation (3) is commonly referred as the arrival cost. This term summarizes the information not considered in the horizon at time τ. In the case that we have, that is, linear model and gaussian noises, this term would simply become $\Phi(\hat{x}(\tau - N)) = \|\hat{x}(\tau - N) - m\|^2_{P^{-1}(\tau-N)}$ [13], where $P^{-1}(\tau - N)$ is the inverse of the covariance matrix of the estimation error and m is the mean of $x(\tau - N)$. Nevertheless, it is not practical in a distributed dynamic programming problem to keep track of $P^{-1}(\tau - N)$ and approximations are needed. One possible choice is to use the steady state covariance matrix to weight the estimation at the beginning of the window. In this paper the problem will be relaxed assuming that $x(\tau - N)$ takes the value calculated in its latest estimation $\hat{x}(\tau - N)$. This assumption works well as long as the previous estimates are correctly estimated. Actually, in the case that the trajectory of estimated states out of the estimation window were all exact (which, of course, is highly improbable) then this approximation would become just an application of Bellman's principle of optimality [2].

The optimal estimation for the trajectory of states $\hat{X}^*(0 : \tau) = \{\hat{x}^*(0), \dots, \hat{x}^*(\tau)\}$ is obtained solving the following minimization problem

$$\hat{X}^*(0 : \tau) = arg \min_{\hat{X}(0:\tau)} V^\tau(\hat{X}(0 : \tau)) \tag{5}$$

subject to (2) and taking $\Phi(x(0)) = \|x(0) - m(0)\|^2_{P^{-1}(0)}$. This problem is equivalent to the Kalman filter [3] but it has a major drawback: the computational burden of (5) grows with τ as more measurements become available. We use an approximate moving horizon estimation approach [5] to fix the computational cost. The estimation we make is $\hat{X}(\tau - N : \tau) = \{\hat{x}(\tau - N), \dots, \hat{x}(\tau)\}$ and can be calculated solving the following QP problem:

$$\hat{X}^*(\tau - N : \tau) = arg \min_{\hat{X}(\tau-N:\tau)} V^N(\hat{X}(\tau - N : \tau)) \tag{6}$$

subject to (2) and $x(\tau - N) = \hat{x}(\tau - N)$.

Remark 3: Note that the state equation in (2) allows to determine the noise trajectory once the state trajectory has been calculated. This relationship can be used in the opposite way so that the QP problem can also be solved minimizing with respect the noise trajectory $w(\tau - N), .., w(\tau - 1)$. Taking into account the duality between the control and observation problem, a possible interpretation for the minimization alternative is that the term $w_i(\tau)$ is used to *control* the estimation.

III. Dual decomposition

The ultimate goal of the paper is to distribute the estimation problem between all the agents present in the system. Under certain assumptions, in [4] dual decomposition was used to distribute the optimization problem corresponding to a MPC controller between several agents. As the problem of estimation is the dual of the control problem, and we have reduced the estimation to the optimization of a cost function, the same methodology will be applied.

It can be seen in equation (1) that the outputs of the subsystems are coupled through the states. The coupling term represents the effect of the rest of the subsystems in the measurements of agent i. We will define $d_i(\tau) =$

$\sum_{i\neq j} C_{ij}x_j(\tau)$ to denote this effect. This allow to rewrite the subsystem model as

$$\begin{array}{l} x_i(\tau+1) = A_{ii}x_i(\tau) + w_i(\tau) \\ y_i(\tau) = C_{ii}x_i(\tau) - d_i(\tau) + v_i(\tau) \end{array} \qquad (7)$$

subject to the constraint $d_i(\tau) = -\sum_{i\neq j} C_{ij}x_j(\tau)$.

Dual decomposition can be used to distribute the centralized problem (6) between the agents. The introduction of Lagrange multipliers p_i in the cost function allows the distribution of the cost function (3). First, we define the Lagrange extended cost function as

$$\begin{array}{l} V^{N,p}(\hat{X}(\tau-N:\tau), D(\tau-N:\tau), P(\tau-N:\tau)) = \\ \sum\limits_{i=1}^{J}\{ \sum\limits_{k=\tau-N}^{\tau-1} \|\hat{x}_i(k+1) - A\hat{x}_i(k)\|^2_{Q_i^{-1}(k)} \\ + \sum\limits_{k=\tau-N}^{\tau} \|-C_{ii}\hat{x}_i(k) + y_i(k) + d_i(k)\|^2_{R_i^{-1}(k)} \\ + \sum\limits_{k=\tau-N}^{\tau} p_i^T(k)(d_i(k) + \sum_{i\neq j} C_{ij}\hat{x}_j(k))\} \end{array} \qquad (8)$$

where $p_i(\tau) \in \mathbb{R}^{q_i}$ is the lagrange multiplier corresponding to the constraint induced by $d_i(\tau) \in \mathbb{R}^{q_i}$, which is now a free variable. Their corresponding centralized vectors are respectively $p(\tau) = [p_1(\tau)\ p_2(\tau)\ \ldots\ p_J(\tau)]^T \in \mathbb{R}^q$ and $d(\tau) = [d_1(\tau)\ d_2(\tau)\ \ldots\ d_J(\tau)]^T \in \mathbb{R}^q$. Finally, we denote the sequences of these vectors in time as $P(\tau-N:\tau) = \{p(\tau-N),\ldots,p(\tau)\}$ and $D(\tau-N:\tau) = \{d(\tau-N),\ldots,d(\tau)\}$.

If we take $Q_i^{-1}(\tau) = 0$ in 8 we can reduce the two summations to one. Then, if we rearrange the lagrangian multipliers it is possible to rewrite the extended cost function as:

$$\begin{array}{l} V^{N,p}(\hat{X}(\tau-N:\tau), D(\tau-N:\tau), P(\tau-N:\tau)) = \\ \sum\limits_{i=1}^{J}\sum\limits_{k=\tau-N}^{\tau} [\|\hat{x}_i(k+1) - A\hat{x}_i(k)\|^2_{Q_i^{-1}(k)} \\ + \|-C_{ii}\hat{x}_i(k) + y_i(k) + d_i(k)\|^2_{R_i^{-1}(k)} \\ + p_i^T(k)d_i(k) + \hat{x}_i(k)^T \sum_{i\neq j} C_{ji}^T p_j(k)] \\ = \sum\limits_{i=1}^{J} V_i^{N,p}(\hat{X}_i(\tau-N:\tau), D_i(\tau-N:\tau), P(\tau-N:\tau)) \end{array}$$

The quadratic problem can be distributed among the agents because the local extended cost functions $V_i^{N,p}(\hat{X}_i(\tau-N:\tau), D_i(\tau-N:\tau), P(\tau-N:\tau))$ are decoupled. From a centralized point of view the problem that is solved at each time sample is

$$\max_{P(\tau-N:\tau)} \sum_{i=1}^{J} \min_{\substack{\hat{X}_i(\tau-N:\tau),\\ D_i(\tau-N:\tau)}} V_i^{N,p} \begin{pmatrix} \hat{X}_i(\tau-N:\tau), \\ D_i(\tau-N:\tau), \\ P(\tau-N:\tau) \end{pmatrix}$$

Remark 4: If we define the stage cost at the time sample k as

$$\begin{array}{l} l_i(\hat{x}_i(k), d_i(k)) = \|-C_{ii}\hat{x}_i(k) + y_i(k) + d_i(k)\|^2_{R_i^{-1}(k)} \\ + \|\hat{x}_i(k+1) - A\hat{x}_i(k)\|^2_{Q_i^{-1}(k)} \end{array}$$

Then, the extended local cost function can be posed as

$$\begin{array}{l} V_i^{N,p}(\hat{X}_i(\tau-N:\tau), P(\tau-N:\tau)) = \sum\limits_{k=\tau-N}^{\tau} [l_i(\hat{x}_i(k), d_i(k)) \\ + p_i^T(k)d_i(k) + \hat{x}_i(k)^T \sum_{i\neq j} C_{ji}^T p_j(k)] \end{array}$$

The local stage cost can also be extended to include the terms due to the lagrangian prices $l_i^p(\hat{x}_i(k), d_i(k), P(k)) = l_i(\hat{x}_i(k), d_i(k)) + p_i^T(k)d_i(k) + \hat{x}_i(k)^T \sum_{i\neq j} C_{ji}^T p_j(k)$, which allows to write the extended local cost function as:

$$V_i^{N,p}(\tau) = \sum_{k=\tau-N}^{\tau} l_i^p(\hat{x}_i(k), d_i(k), P(k)) \qquad (9)$$

Remark 5: After the introduction of dual variables, and assuming that the prices of the neighbors are given, it is possible to interpret the distributed optimization procedure in economic terms. Each agent behavior can be represented as a two player game. The first player objective is to minimize the price-extended stage cost

$$\sum_{k=\tau-N}^{\tau} l_i^p(\hat{x}_i(k), d_i(k), P(k)),$$

which is composed of three elements that are interpretable as

$$\begin{array}{l} l_i^p(\hat{x}_i(k), d_i(k), P(k)) = \\ \underbrace{l_i(\hat{x}_i(k), d_i(k))}_{local\ cost} + \overbrace{p_i^T(k)d_i(k)}^{neighbor\ help\ cost} + \underbrace{\hat{x}_i(k)^T \sum_{i\neq j} C_{ji}^T p_j(k)}_{incomes\ due\ to\ required\ help} . \end{array}$$

The second player chooses the prices $p_i(\tau-N),\ldots,p_i(\tau)$ to maximize

$$p_i^T(k)(d_i(k) + \sum_{i\neq j} C_{ij}\hat{x}_j(k)).$$

This game is repeated iteratively. First, an estimate is calculated according to the given prices. Then, the prices are updated and the cycle starts again. As a result of the repeated interaction of both players in each node the prices evolve until a maximum is reached. The consequence of this standard Lagrangian optimization procedure is that the minimum for the cost function (6) is attained and the constraints are satisfied when the price gradient is zero.

The algorithm that is followed by the agents in the system can be summarized as:

- Step 1: Each agent i estimates his own current state trajectory $\{\hat{x}_i(\tau-N), \hat{x}_i(\tau-N+1), .., \hat{x}_i(\tau)\}$ solving the optimization problem given in (9) for a set of given prices p_i $i = 0,\ldots,J$.
- Step 2: Once the state trajectory has been calculated then the prices of agent i are updated by a gradient step as follows.

$$p_i^{k+1}(\tau) = p_i^k(\tau) + \gamma_i^k[d_i(\tau) + \sum_{i\neq j} C_{ij}\hat{x}_j(k)] \qquad (10)$$

Convergence of such gradient algorithms has been proved under different type of assumptions on the step size sequence γ_i^k. See for example [15]. Note that in order to update the prices the agents must communicate.

- Step 3: If the precision obtained with the estimation is enough then there is no need to continue iterating. In the next section precise conditions are given. If enough precision is not attained and the number of iterations K exceeds a given threshold *maxiter*, then the algorithm also stops. In other case then the process is repeated from step 1 for $K = K + 1$.

A. Coordination alternatives for the price update

It can be seen that the calculation of the estimate $\hat{x}_i(t)$ for $t = \tau - N, \dots, \tau$ is completely decentralized once that prices are given. Therefore it is mandatory for an agent to keep the track of its neighbor prices. Nevertheless, in order to update the prices, coordination among the agents is necessary. The agents send their estimates $\hat{x}_i(\tau)$ to their neighbors so that equation (10) can be applied. For some systems it could be desirable not to share the state information with their neighbors. To avoid the exchange of the state estimates we propose two alternatives:

- Decentralized approach: The need for the shared information comes from term $\sum_{i \neq j} C_{ij}\hat{x}_j(k)$ in equation equation (10). According to the dynamics of the subsystems $\sum\limits_{j \neq i} C_{ij}x_i(\tau) = y_i(\tau) - C_{ii}x_i(\tau) - v_i(\tau)$, and thus it could be approximated by $\sum\limits_{j \neq i} C_{ij}\hat{x}_i(\tau) \approx y_i(\tau) - C_{ii}\hat{x}_i(\tau)$.
- Market approach: This alternative consists on changing the way in which prices are updated. To understand better this approach it is convenient to use the behavior model that represents each agent as a two player game. Then, it is possible to think on the centralized problem as a game with $2J$ players. The objective of the first player in each node is to minimize his own cost according to the given prices. However, the second player in each node bargains with the the rest of the second players to maximize (9) with respect to the prices. The second players can be seen as market makers that fix the prices of the help services that the agents provide each other according to the offer and demand of such services. To do so, a gradient optimization of the cost function (9) is implemented. Each update is based on the addition of contributions of the different agents. The contribution of agent i is

$$\nabla p_i^k(\tau) = \begin{bmatrix} \hat{x}_i(k)^T C_{1i}^T p_1(k) \\ \vdots \\ \hat{x}_i(k)^T C_{i-1,i}^T p_{i-1}(k) \\ d_i(k) \\ \hat{x}_i(k)^T C_{i+1,i}^T p_{i+1}(k) \\ \vdots \\ \hat{x}_i(k)^T C_{Ji}^T p_J(k) \end{bmatrix}$$

Theorem 1: The price update mechanism defined in the market approach provides the same results than the one presented in equation (10).

Proof:
It is straight forward to check that both methods provide the same centralized price vector. It is enough to sum the contribution $\nabla p_i^k(\tau)$ for all i

$$p^{k+1}(\tau) = p^k(\tau) + \gamma \sum_i \nabla p_i^k(\tau).$$

Then it can be seen that the price for agent i is just

$$p_i^{k+1}(\tau) = p_i^k(\tau) + \gamma_i^k[d_i(\tau) + \sum_{i \neq j} C_{ij}\hat{x}_j(k)]$$

■

If we move back to the agents and forget the game theory interpretation, it can be seen that under the market approach agents update their prices and also the prices of their neighbors and therefore there is no need to exchange the state estimate. All the public information needed are the prices and their updates. The estimation of the agents through the different iterations bring increments or decrements in the prices until equilibrium prices are reached. However, there is a price to pay in terms on the amount of model information that agents have. With this price mechanism it is needed that agent i has knowledge of the terms C_{ji}. In other words, agents have knowledge of the collateral effects they induce in their neighbors.

Remark 6: From an economic point of view, the situation can be interpreted as a market of help services. The price $p_i(\tau)$ is the unit price that agent i has to pay to his neighbors for them to change their current contribution to his output. The fact that neighbors of an agent i change their estimates affects to his price in such a way that it reflects how costy is for his neighborhood to help him after the estimate update. On the other hand helping his neighbors is rewarded in (9). Taking all of this into account, agents are both service offerers and demanders. All of them behave selfishly according to the prices fixed by the *market*, that is, the distributed price mechanism proposed in this approach.

Remark 7: In welfare economics, under certain assumptions such as the absence of externalities in transactions, it is proved that market prices guarantee that, despite of agents selfish behavior, a Pareto optimum is achieved [16]. In the optimization problem that we have, unfortunately we have to deal with the presence of externalities, taking this term in a wide sense. That is, decisions taken by agent i also affect other agents. In order to overcome this problem and still reach a Pareto optimum while keeping selfish, i.e. decentralized, behavior in the agents, some modifications have to be introduced in the market: first, all the agents behave as price takers as they were in a competitive market when they really have power to modify the prices and, second, prices are updated globally according to the proposed mechanism.

IV. Examples

The problem of estimating the position of a moving object can be faced using different approaches. For outdoor applications in which the precision requirements are low GPS estimation is the most used choice. Radar measurements help to improve the quality of the estimation. When it comes to indoor applications the problem of localization is normally solved by means of a sensor network. In cases in which low precision is needed some it may be enough with a network of presence detectors. If more precision is required then more sophisticated techniques have to be used. In the case that the application is executed in a very controlled scenario, it is possible to use cameras to estimate the position. Infrared or ultrasonic sensors also provide a greater accuracy than the presence detectors. In the last years the use of the link quality between wireless transceivers has been used too for this kind of applications [8].

A. Application to mobile robot localization

This subsection is based on the simulation scenario proposed in [6].

Let us consider a system consisting a set of $\mu = \{1, .., M\}$ reference nodes or beacons and a set $\eta = \{1, .., J\}$ of mobile devices. In this example we will consider $M = 6$ beacons and $J = 8$ mobile devices, which are located in the positions depicted in figure 1.

The goal is to estimate the position of the moving devices. If the sample time is assumed to be low enough, it is possible to simplify the dynamics considering that the devices move at every sample time a bit with respect their position. The equations for each device are:

$$x_i(\tau + 1) = x_i(\tau) + \Delta x_i(\tau) \ \forall i \in \eta = \{1, .., J\}$$

with $x_i(0) = x_i^0$. The beacon position is fixed so that $x_i(K + 1) = x_i(0) \ \forall i \in \mu = \{1, .., M\}$. The distance between the nodes and the mobile devices can be calculated using

$$d_{ij}^2 = (x_i - x_j)^T (x_i - x_j) \ \forall i, j \in \eta, \mu.$$

The distance can be linearized around the steady state positions $\overline{x}_i$ using a first order Taylor approximation, which leads to

$$d_{ij}^2 = \overline{d}_{ij}^2 + 2(\overline{x}_i - \overline{x}_j)^T (x_i - \overline{x}_j) + 2(\overline{x}_i - \overline{x}_j)^T (\overline{x}_i - x_j)$$

with $\overline{d}_{ij}^2 = d_{ij}^2(\overline{x}_i, \overline{x}_j)$. Now, system variables can be introduced for all the mobile devices such that:

$$\begin{aligned} x_i(\tau) &= x_i(\tau) - \overline{x}_i \ \forall i \in \eta \\ y_{ji}(\tau) &= d_{ij}^2 - \overline{d}_{ij}^2 \ \forall i \in \eta, \forall j \in \eta, \mu \\ C_{ji} &= 2(\overline{x}_i - \overline{x}_j) \ \forall i \in \eta, \forall j \in \eta, \mu. \end{aligned}$$

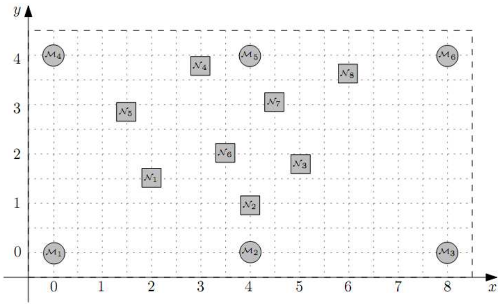

Fig. 1: Initial situation of the devices.

So each moving device's output provides information about the distance with respect the other moving devices and the beacons. If white gaussian additive noise is assumed in the state and output then each device can be modeled according to equation (1).

In order to make the situation more realistic it can be assumed that only devices and beacons within a range can communicate. Thus a communication radius ρ is defined. In general two devices i and j can communicate if $d_{ij} < \rho$. A communication graph can be defined to reflect what devices can communicate at each sample time. The communication graph at initial time is given by the following matrices:

$$A_0^\eta = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$A_0^\mu = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

where $A_0^\eta(i, j) = 1$ if the mobile device i is able to communicate with the mobile device j and $A_0^\mu(i, j) = 1$ if the mobile robot i is able to communicate with the beacon j.

The simulations have been done considering a dynamic graph, that is, a situation where the movement of the devices is big enough to guarantee that the communication graph changes with the relinearization of the system. At each time in which the system is relinearized it is necessary not only to update the equations but the information about the last samples that is kept in the agents. Let us assume that in time τ there is a change of linearization point of the system. Then,

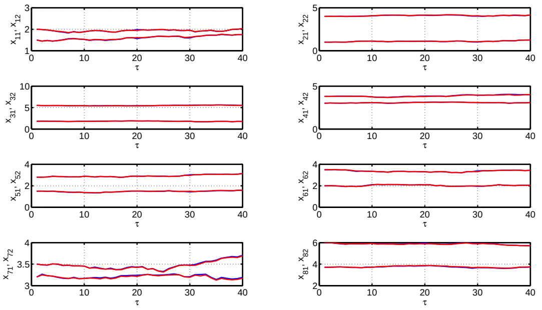

Fig. 2: Robots' state evolution with noise model mismatch.

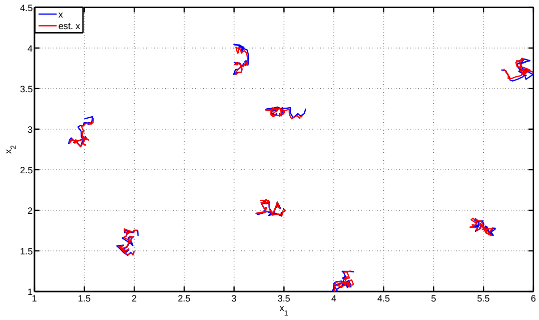

Fig. 3: Robots' trajectories with noise model mismatch.

$$\begin{aligned} x_i(t) &= \bar{x}_i(t) + x_i(t) - \bar{x}_i(\tau)\ \forall t \in [\tau - N, \tau] \\ y_i(t) &= C_i(\tau)\hat{x}_i(t)\ \forall t \in [\tau - N, \tau] \end{aligned}$$

This change of coordinates in the state estimates and the outputs allow to compute the distributed problem without suffering estimation disturbances after the change of linearization point.

The system has been simulated for 40 time samples with a state noise stronger than the original one. The first 10 samples a centralized Kalman filter is working and the second 20 the distributed strategy. In $t = 20$ and $t = 30$ the system is relinearized. The window size used for the estimation was 4. In blue it is depicted the real trajectory and in red the estimation.

The results for the estimation of the position of the mobile devices can be seen in figure 2.

The overall picture is shown in figure 3. The quality of the estimation depends on several parameters. For example, the more iterations are made the better the estimation gets. In this figure it can be seen that the estimation is very precise for most agents.

V. Conclusions

A distributed version of the Kalman filter based on dynamic programming has been developed in this paper. The use of dual decomposition allowed the problem distribution. In the simulations presented promising results of the future applications of these techniques are shown.

It will be important for future work some kind of suboptimality bounds to determine the precision obtained in the estimation after a number of iterations. Practical experiments will be developed too to see how the distributed estimation works in real application.

References

[1] Peter Alriksson. *State Estimation for Distributed and Hybrid Systems*. PhD thesis, Department of Automatic Control, Lund University, September 2008.

[2] R.E. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[3] Henry Cox. On the estimation of state variables and parameters for noisy dynamic systems. *IEEE Transactions on automatic control*, 9:5–12, 1964.

[4] P. Giselsson and A. Rantzer. Distributed model predictive control with suboptimality and stability guarantees. In *Proceedings of the 2010 Control and Decision Conference*, 2010.

[5] Eric L. Haseltine and James B. Rawlings. Critical evaluation of extended kalman filter and moving horizon estimation. *Industrial and Engineering Chemistry Research*, 44:2451–2460, 2005.

[6] Anne-Kathrin Hess. A distributed kalman filter algorithm for self-localization of mobile devices. Technical report, Department of Automatic Control, Lund University, Sweden, 2009.

[7] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering*, 82: Series D:35–45, 1960.

[8] J. M. Maestre and E. F. Camacho. Smart home interoperability: the domoesi project approach. *International Journal of Smart Home*, 3:31–44, 2009.

[9] P. Neumann. Communication in industrial automation - what is going on? *Control Engineering Practice*, 15:1332–1347.

[10] Joseph Polastre, Robert Szewczyk, and David Culler. Telos: enabling ultra-low power wireless research. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 48. IEEE Press, 2005.

[11] B. De Schutter R. R. Negenborn and H. Hellendoorn. Multi-agent model predictive control of transportation networks. In *Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control (ICNSC 2006)*, pages pp. 296–301, Ft. Lauderdale, Florida, April 2006.

[12] A. Rantzer. Dynamic dual decomposition for distributed control. In *American Control Conference 2009*, 2009.

[13] J.B. Rawlings and B.R. Bakshi. Particle filtering and moving horizon estimation. *Computers and Chemical Engineering*, 30:1529–1541, 2006.

[14] S.I. Roumeliotis and G.A. Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, 2002.

[15] N. Z. Shor. *Minimization Methods for Nondifferentiable functions*. Springer, 1985.

[16] H. R. Varian. *Intermediate Microeconomics: A Modern Approach*. W.W. Norton & Co., 2005.

[17] T. C. Yang. Networked control systems: a brief survey. In *IEE Procedings- Control Theory and Applications*, volume 152, pages 403–412, 2006.