

MASTER

Optimal scalable decentralized Kalman filter

Oruc, S.

Award date:
2008

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

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

Optimal Scalable Decentralized Kalman filter

by

S. Oruč

Master of Science thesis

Project period: October 2008

Report Number: 08A/07

Commissioned by:

Prof.dr.ir. P.P.J. van den Bosch

Supervisors:

Dr. M. Lazar

Ir. J. Sijs

Dr.ir. Tj. Tjalkens

Optimal Scalable Decentralized Kalman Filter

Sertaç Oruç

Abstract—The Kalman filter is a powerful state estimation algorithm which incorporates noise models, process model and measurements to obtain an accurate estimate of the states of the process. Realization of conventional Kalman filter algorithm requires a central processor that harvests measurements from all the sensors on the field. Central systems have some drawbacks like reliability, central communication and high computation which result in a need for non-central algorithms.

The interest of this study is centered at optimality and scalability in Decentralized Kalman Filter (DKF). Current research on DKF focuses on "better" estimation of process states in terms of aspects like computation power, communication overhead, memory usage, robustness, reliability and scalability. This study takes first optimality and then scalability in DKF as its focus and looks for an answer to problem of obtaining an optimal yet scalable DKF. An optimal DKF for a given network topology is found and a mathematical problem formulation for an optimal scalable DKF is made.

Index Terms—Kalman filter, state estimation, consensus filter, covariance intersection, multi-sensor, decentralized/distributed Kalman filter

I. INTRODUCTION

A. Background

WHEN a control engineer considers system analysis or controller design, first thing he does is forming a deterministic model, using all scientific and engineering knowledge he has. With this model and knowledge of control theory the engineer can investigate the system behavior and build his controller. To do this, the engineer also needs to build measurement devices to measure the interested variables of the system. Processing these measurement data for proper control inputs forms a very important phase in control systems.

Deterministic models and control theory cannot provide a sufficient tool to make analysis and design for every physical system. First of all, no mathematical model can perfectly represent a physical behavior. Every model reveals main characteristics and variables of the system and ignores the rest. Secondly dynamic systems are driven by "indeterministic" noises and thirdly "the measurement devices" or sensors do not provide perfect data about the system but gives measurements which are corrupted by indeterministic noise.

Thus, as a proper and practical solution to this problem one uses stochastic models and estimation theory to estimate the system behavior using noisy measurements and a stochastic process model.

MSc student S.Oruç is with the Department of Electrical Engineering, TU Eindhoven, The Netherlands
e-mail: s.oruc@student.tue.nl
15 September 2008

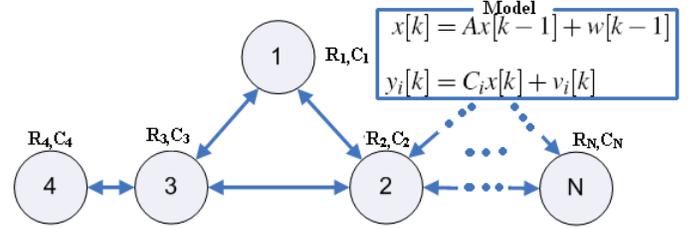


Fig. 1. System

In this project we deal with state estimation assuming a stochastic discrete time state-space model of the system and model measurement noises and process noises as zero mean Gaussian functions and known error covariances. We specifically deal with the Kalman filter as the estimator.

The Kalman filter, which is also known as Linear Quadratic Estimator (LQE) in Control Theory, was created by Rudolf Emil Kalman and published in 1960 in his famous paper "A New Approach to Linear Filtering and Prediction Problems" [1]. It was developed as a recursive solution to the discrete data linear filtering problem. It uses process model, noise models, measurements and finds an accurate estimate together with error covariance, namely uncertainty, of the estimate. It has been a very active research area and has broad applications in automated devices, navigation systems, weather forecast, econometrics, tracking systems and many other diverse fields.

Since it was created, many variants of the Kalman filter have been introduced, i.e. Information filter [2], The Extended Kalman filter [3], Fast Kalman filter [4], Unscented Kalman filter [5], Schmidt's extended filter and so on. Each of these methods attempts to solve another practical problem in applications of Kalman filter. Information filter brought more practical ease whereas Extended Kalman filter and Unscented Kalman filter provided applicability for non linear systems. In this paper we specifically consider Decentralized Kalman filter (DKF) and give a unique perspective to the problem of finding an optimal, scalable Kalman filter.

B. Problem Description

The system we consider in this study, which is shown in figure 1, is a network of *nodes* with their own sensors, hence own measurements. The states of a global process modeled by the equation $x[k] = Ax[k-1] + w[k-1]$ in discrete time is desired to be estimated at each node. This model is available to all the nodes. w is modeled as a zero mean Gaussian noise with an error covariance matrix of $Q = E[ww^T]$. In KF, modeling of process and measurement noises as zero mean Gaussian functions is an essential requirement for the formulation of

KF. The noise is always modeled as zero mean Gaussian. If the real noise is also zero mean Gaussian as modeled then KF estimate converges to zero.

All the nodes in the network measure some states of the global process. Measurements are modeled by measurement equations $y_i[k] = C_i x[k] + v_i[k]$. v_i is modeled as a zero mean Gaussian noise with a covariance matrix of $R_i = E[v_i v_i^T]$. So each node is characterized by respective C_i matrix and measurement covariance matrix R_i .

Each node estimates the state vector of the same global process. The sensor measurements are assumed not to be correlated.

In a decentralized network structure like in figure 1 each node makes its own estimation and measurement. It is also possible to transmit the estimate and/or measurement to the neighbors. Throughout this paper the concept *neighbor* is used for *communication neighbor* which is the node directly connected to the operating node. The nodes are allowed to transmit data to only their direct *neighbors* for better estimation.

”Decentralized Kalman Filter (DKF)” algorithm transmits both state estimates and measurements to the neighbors. Note that the measurements are fresh data which are not correlated with the current estimates or other measurements made at neighbors. The estimates however are found by incorporating the information from history, therefore they are correlated with the estimates from the other nodes. Also another algorithm which we call ”Local Kalman Filter (LKF)” is described in this paper for comparison purposes, which differs basically from DKF such that LKF does not transmit state estimates whereas DKF does.

In KF, trace of state error covariance matrix which is defined as $P[k] = E[(x[k] - \hat{x}[k])(x[k] - \hat{x}[k])^T]$ is minimized where \hat{x} is the estimate found by KF and x is the predicted real state according to the model in equation $x[k] = Ax[k-1] + w[k-1]$. If the model is accurately made this error is also equal to the error between the real state and estimate. The trace of $P[k]$ gives the sum of squared error between predicted real state and estimate which is minimized. Calculation of this $P[k]$ matrix is crucial for the solution of optimal Kalman filter. In central KF $P[k]$ is a function of A, Q, C, R and can be calculated by central processor. However in DKF calculation of $P_{ii}[k] = E[(x[k] - \hat{x}_i[k])(x[k] - \hat{x}_i[k])^T]$ is not straight forward since $\hat{x}_i[k]$ involves $\hat{x}_j[k-1] \quad j \in N_i$ terms because of state share between the nodes. The incorporation of neighbor states brings the need for calculation of cross-covariance terms which are defined as $P_{ij} = E[(x - \hat{x}_i)(x - \hat{x}_j)^T]$ into the picture. Calculation of these P_{ij} terms is the key challenge in finding an optimal scalable DKF algorithm.

In this study two results are achieved. Firstly an optimal decentralized Kalman filter depending on the network topology is obtained. This algorithm is built and discussed in section V. Secondly a problem formulation for a both optimal and scalable decentralized Kalman filter algorithm is proposed in section VI. Ultimately this research aims to find out an optimal scalable DKF algorithm with the following constraints:

- *Every* node of the network estimates the global state depending on the previous state estimates and measurements of themselves and neighbors. (Decentralization)
- The algorithm must solely utilize peer-to-peer communications, and should not involve global communications. (Decentralization)
- The algorithm must be scalable, which means adding or removing a node must not change the algorithm on all of the nodes, although it affect a limited number of nodes. (Scalability)
- The algorithm must be optimal such that the error between the estimated state and the true state must be minimized for an arbitrary network communication. (Optimality)

C. Approach

Many algorithms that have been suggested for a decentralized Kalman algorithm are based on the constraint of sending *only* measurements. In this situation an optimal algorithm is already known. In this paper we call this optimal decentralized algorithm ”Local Kalman Filter”, in which only measurements are communicated, to distinguish it from our decentralized Kalman filter (DKF). In this research we challenge the problem of finding an optimal and scalable DKF in which both state estimates and the measurements are transmitted. For this purpose first we consider optimality to find an optimal DKF and then we formulate the problem for both optimal and scalable algorithm.

Our approach is a simple yet creative one. In the first step to have an optimal algorithm we specifically take cross-covariances into account by considering the network as a whole. We consider all of the cross covariances between every two nodes in the network. For this purpose we introduce a new representation, called *Global System Representation*, for which we rebuild the Kalman filter equations. This representation made it possible to calculate all the cross covariances in the network and gave us a cost function for optimality in which the error between the individual estimates and predicted real state is minimized. Solving this formulation, the optimal state estimate at each individual node is obtained in terms of the measurements and previous estimates from *only* the neighbors and the node itself and also the network topology. Since the network topology must be made available to the nodes, this algorithm remains unscalable.

In the second step in section VI, we challenge the scalability constraint. We define scalability as ”Each node is allowed only to know the network connection of the first order neighborhood”. Using this statement a mathematical formulation of an Optimal Scalable Decentralized Kalman Filter is derived. The idea in this approach is using only local network topology instead of the whole network topology in the individual algorithms on each node. The rest of the terms in the algorithm regarding the network connections other than the local network topology is modeled as *uncertainty*.

The solution of this scalable formulation depends on the

modeling of this uncertainty mentioned above, about which we provide some ideas and propose this problem as a future work.

II. NOMENCLATURE

$x[k]$	The predicted real state according to the model at k^{th} iteration.*
$y_i[k]$	The measurement at i^{th} node at k^{th} iteration.
\hat{x}_i	The estimate of state x at i^{th} node
\hat{x}_i^-	The predicted estimate (or prediction) of state x at i^{th} node
P_{ij}	Error covariance matrix between state estimates at i^{th} and j^{th} node
M_{ij}	Kalman gain for state from j^{th} node on i^{th} node
K_{ij}	Kalman gain for measurement from j^{th} node on i^{th} node
N	The set of all nodes in the network
N_i	Neighborhood of i^{th} node
N_i^-	Neighborhood of i^{th} node without i^{th} node ($N_i - \{i\}$)
v_i	Measurement noise of the i^{th} sensor
w	Process noise
R_i	Measurement noise covariance of the i^{th} sensor
Q	Process noise covariance matrix
C_i	Measurement matrix in the measurement model, relating the state vector to the measurement (y_i) of i^{th} node

*For clarity time indices are dropped when there is no risk of confusion

III. KEY CONCEPTS

Throughout the paper some concepts and terms are used. For clarity these concepts and terms are explicitly stated to avoid confusion.

Sensor

Devices that measure a physical quantity and convert it into an analog signal. In this paper by sensor measurement we mean low-level interpreted digital data rather than raw analog data.

Node

Unit which has a sensor to make measurement and a processor to run an algorithm. Nodes also can communicate measurements and states. LKF and DKF algorithms are run on these nodes. In the case central KF nodes send their measurements to a central processor and the central processor does the estimate.

Neighbor

The node that is communication-wise adjacent to the operating node.

Neighborhood

The set of nodes that is adjacent to the operating node and the node itself.

2nd order neighborhood

The set of nodes that are neighbors to the neighbor nodes of the operating nodes.

Optimality

In this paper optimality refers to the minimization of the sum of squared errors between state estimate, \hat{x}_i and the predicted real state, x according to the model. Throughout this paper the adjective "optimal" is used in case this error is minimized for the given constraints, such as decentralization, transmission of states, scalability.

Scalability

Possibility to add/remove nodes such that the algorithms on only a finite number of nodes change.

Robustness

The performance of the system in case a node fails.

Globally Connected Network

The network structure in which every node has a direct communication to all other nodes.

Data Incest

The situation in which a data is used more than once in estimation. In case states are transmitted, the same state can be transmitted to a node from different nodes.

Covariance Matrix

A matrix of covariances between elements of the same vector

Cross Covariance Matrix

A matrix of covariances between elements of two different vectors

Decentralized System

A system which does not involve central activities like communication or processing but does these activities locally, in local processors.

Wireless Sensor Network (WSN)

A network consisting of spatially distributed autonomous devices(nodes) using sensors to cooperatively monitor physical or environmental conditions such as temperature, sound, vibration, pressure and motion

Residue

The discrepancy between estimate of the model on the node and the measurement taken in Kalman filtering

IV. KALMAN FILTER

Kalman filter [1] is an optimal recursive data processing algorithm. The term "filter" refers to the fact that it is a data processing algorithm on a processor which uses noisy measurements to find less noisy estimates. It gives the "optimal" estimate of the states of the process in least squares sense, combining model predictions and measurements [6]. KF processes all the measurements, uses the model of the system, the measurements and process noise to do the estimation without ignoring any of these information which makes Kalman Filter such a powerful estimator [6].

Conventional Kalman filtering is done in a centralized way, i.e. all measurements are gathered in a central processor and the algorithm is run only in this central processor. For this

reason we call this conventional Kalman filter central Kalman filter(CKF). This filter is proved to be optimal in least squares sense. On the other hand, LKF and DKF are done in a distributed fashion, i.e. measurements are processed locally in local processors(nodes) and each node runs its own algorithm. After the creation of KF till DKF there has been some research steps taken. In this chapter a related overview of evolution of Kalman filter is provided. First CKF is created and then the information filter(IF) has been found. IF made decentralization and LKF, in which only measurements are sent, possible. LKF has been followed by DKF which transmits state estimates to the neighbors and processes past state estimations of the neighbors for better estimation.

A. Central Kalman Filter (CKF)

Central Kalman filter is the conventional Kalman filter found by R. E. Kalman. The sensors provide measurement to a central processor and the algorithm is run in this central processor.

The observed process is described in state space as:

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

$$y[k] = Cx[k] + v[k] \quad (2)$$

The noise signals v and w are defined as measurement and process noises respectively which are modeled in CKF by zero mean Gaussians with covariance matrices R and Q respectively. If we consider the system in figure 1 as an example, all the measurements are sent to a central processor and the parameters are defined as follows;

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} C = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{bmatrix} v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} R = \begin{bmatrix} R_1 & 0 & \dots & 0 \\ 0 & R_2 & \dots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \dots & R_n \end{bmatrix} \quad (3)$$

The Kalman filter consists of two stages at each time step. At k^{th} time step KF equations are written as follows; [7]:

Time Update Equations:

$$\hat{x}^- [k] = A\hat{x}[k-1] \quad (4)$$

$$P^- [k] = AP[k-1]A^T + Q \quad (5)$$

Measurement Update Equations:

$$K[k] = P^- [k]C^T (CP^- [k]C^T + R)^{-1} \quad (6)$$

$$\hat{x}[k] = \hat{x}^- [k] + K[k](y[k] - C\hat{x}^- [k]) \quad (7)$$

$$P[k] = (I - K[k]C)P^- [k] \quad (8)$$

The time update stage propagates the past estimate in time to have a prediction state \hat{x}^- at k^{th} time step whereas measurement update stage updates the prediction with measurements by weighing measurements and prediction by a gain called Kalman gain, K which is calculated for minimum least squares error. The derivation of KF can be found in appendix A.

B. Information Filter (IF)

The information filter is a form of Kalman Filter that can be derived by rewriting the central Kalman filter equations. It has an important role in decentralization of the algorithm which makes it important to understand. The information filter is derived with following two definitions:

$$\text{Information Matrix: } J[k] = P^{-1}[k] \quad (9)$$

$$\text{Information State Vector: } j[k] = P^{-1}[k]x[k] \quad (10)$$

IF estimates j and J instead of x and P . Therefore their respective stages are found as follows;

Time Update Equations:

$$J[k]^- = AJ^{-1}[k-1]A^T + Q^{-1} \quad (11)$$

$$j[k]^- = L[k]j[k-1] \quad (12)$$

$$L[k] := J[k]^- AJ[k-1] \quad (13)$$

Measurement Update Equations:

$$J[k] = J[k]^- + C^T R^{-1} C \quad (14)$$

$$j[k] = j[k]^- + C^T R^{-1} y[k] \quad (15)$$

IF is beneficial compared to KF in terms of computation when the number of measurements are larger than the number of states since then it is less expensive to invert J in equation 11 and block diagonal matrix R in equations (14),(15), instead of inverting $(CP^- [k]C^T + R)$ in equation (6).

However more important added-value of IF is that it gives way to LKF and decentralization as explained in the next subsection.

C. Local Kalman Filter (LKF)

If we examine the IF equations we see that it is possible to write $C^T R^{-1} C$ and $C^T R^{-1} y$ terms in finite sums of C_i, R_i, y_i terms as;

$$C^T R^{-1} C = \sum_i C_i^T R_i^{-1} C_i \quad (16)$$

$$C^T R^{-1} y = \sum_i C_i^T R_i^{-1} y_i \quad (17)$$

using (3). Then the IF equations can be written as;

Time Update Equations:

$$J[k]^- = AJ^{-1}[k-1]A^T + Q^{-1} \quad (18)$$

$$j[k]^- = L[k]j[k-1] \quad (19)$$

$$L[k] := J[k]^- AJ[k-1] \quad (20)$$

Measurement Update Equations:

$$J[k] = J[k]^- + \sum_{j \in N_i} C_j^T R_j^{-1} C_j \quad (21)$$

$$j[k] = j[k]^- + \sum_{j \in N_i} C_j^T R_j^{-1} y_j \quad (22)$$

where N_i is the neighborhood of the i^{th} node. Then in a given network, like the network in figure 1, if only the measurements are sent, the above equations derived from IF can be used such that a node harvests $C_j^T R_j^{-1} y_j$ and $C_j^T R_j^{-1} C_j$ terms from the neighbors and processes these terms in its local algorithm. We call this decentralized algorithm Local Kalman Filter (LKF), in which only measurements are sent to the neighbors. This algorithm is proven to give locally optimal estimate for i^{th} node [2]. We call this filter "local" since it incorporates only the knowledge from its neighborhood. The measurement made at a second order neighbor can never have an effect on the operating node. However if the nodes send also the states as well as the measurements then inherently a measurement made at a second order neighbor reaches the operating node in two time steps. Importance of sending the states as well as measurements is stressed in the next section.

D. Decentralized Kalman Filter (DKF)

After the creation of the Kalman filter for one measurement, it is realized that instead of using one sensor and one measurement of a plant it is more advantageous to use multiple sensors because of the issues like robustness, flexibility and the fact that a broader observation of the states can be measured with different types of sensors. The more sensors we use, the less the system performance is affected in case of failure of a sensor. The erroneous measurement of a single sensor affects the overall performance less. All in all it is wiser to use multi-sensor systems and "fuse" the information obtained from different sensors to obtain an optimal estimation. This idea results in multi-sensor systems with a central processor in which a Kalman filter is run and fuses information from different sensors. [8]

Later on, the idea of using more sensors has been extended to using also more processors since central processing involves some drawbacks. First of all, central systems are not reliable since the survival of the central processor is very critical. Secondly, for systems with large number of sensors, communication with the central processor becomes difficult. Another reason, for spatially distributed systems, is that communication with central processor can be very difficult. As a result decentralized structures have been demanded for Kalman filtering [8], [9].

Decentralized architectures have become highly interesting since they promise flexible systems in which nodes can be added or removed and programmed independent of each other. The system becomes more robust since it does not depend on one master processor. No node or processor is very critical as all the nodes work in coherence without any hierarchy. Also the processor computation each processor must make can be reduced for advanced algorithms, which is not the focus in this research.

In the previous subsection we already reached a decentralized algorithm which we call LKF. Different than the LKF, the DKF incorporates also the states of the neighbors into the estimate. Incorporation of states brings the opportunity to

propagate a measurement throughout the network inherently in the state estimate. A measurement is made at node i at k^{th} time step and transmitted to node j . By transmitting the state to node j , the measurement has an effect on the estimation at node j . Then at $k+1^{th}$ time step, node j transmits its estimation to node k which is a second order neighbor of node i but after one time step, node k also inherently incorporates the same measurement in its estimation.

Transmission of states is crucial in decentralized structures. However note that for an arbitrary network incorporation of neighbor states in DKF is not trivial. Data incest is likely to occur unless the network connection is taken into account. In the next section we provide an optimal solution for DKF which incorporates states considering correlations between the states and avoids also data incest.

V. OPTIMAL DKF

In this research our ultimate goal is to tackle the problem of having an optimal yet scalable DKF. The aimed algorithm has to fulfill the constraints stated in the introduction section. Optimality and scalability are the key constraints.

In this study trace of auto-covariance matrix, which is the sum of squared errors between estimated states and the predicted real state according to the model, is our cost function. Minimization of this cost function with respect to Kalman gains gives the Kalman gains for minimized cost function, hence the optimal algorithm.

Incorporation of the scalability constraint into the problem formulation is an issue that we challenge in this study. Intuitively for the optimal DKF, the local algorithms must depend on the whole network connection, because of the state estimate communication which causes data incest. Formulating scalability and incorporating it into the mathematical formulation of the problem can solve this issue.

In this paper, we first ignore scalability, formulate and solve optimal DKF problem given the network connection. Later on, we define scalability mathematically such that it can be incorporated in the mathematical formulation to obtain an Optimal Scalable DKF.

A. Global System Representation

For optimality we introduce a new representation in which we consider the network as a whole. In this representation at each node we consider the effect of measurements and state estimates of the neighborhood on estimation. The cross correlations between the estimates of each node are taken into account as well as the auto correlations. Different than the previous algorithms in which only auto covariance matrices like P_1, P_2, \dots are used, in this research P_{11}, P_{22}, \dots are used which are equal respectively. The cross covariance terms P_{12}, P_{13}, \dots are defined as:

$$P_{ij} = E[(x - \hat{x}_i)(x - \hat{x}_j)^T] \quad (23)$$

We have the following process model at node i :

$$x[k] = Ax[k-1] + w[k-1] \quad (24)$$

$$y_i[k] = C_i x[k] + v_i[k] \quad (25)$$

Here we take $w[k]$ as process noise and $v_i[k]$ as measurement noise which are expressed as Gaussian noises with zero mean and Q, R_i covariance matrices respectively. If we assume no communication between the nodes, the state estimate at node i is calculated as follows:

predict:

$$\hat{x}_i^-[k] = A\hat{x}_i[k-1] \quad (26)$$

$$P_i^-[k] = AP_i[k-1]A^T + Q \quad (27)$$

update:

$$\hat{x}_i[k] = \hat{x}_i^-[k] + K_i[k](y_i - C_i\hat{x}_i^-[k]) \quad (28)$$

$$P_i[k] = (I - K_i[k]C_i)P_i^-[k](I - K_i[k]C_i)^T + K_i[k]R_iK_i[k]^T \quad (29)$$

$$K_i[k] = P_i^-[k]C_i^T(C_iP_i^-[k]C_i^T + R_i)^{-1} \quad (30)$$

for optimal $K_i[k]$:

$$P_i[k] = (I - K_i[k]C_i)P_i^-[k] \quad (31)$$

This brings us to the conventional Kalman filter equations without communication. $Tr(P_i[k])$ is taken as the cost function. In case we have communication the best estimate calculated at a node does not depend on only the measurements and estimations made at that node but also the measurements and estimates of the neighbor nodes according to the communication topology. Before going on with the problem formulation for DKF in which both states and measurements are communicated with the neighbors, *Global System Representation (GSR)*, in which we take the network communication topology explicitly, is constructed with the following definitions:

$$\begin{aligned} X &= \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \end{bmatrix} & Y &= \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{bmatrix} & \Delta &= \begin{bmatrix} A & 0 & 0 & \dots \\ 0 & A & 0 & \dots \\ 0 & 0 & A & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \\ \Gamma &= \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ \vdots \end{bmatrix} & V &= \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \end{bmatrix} & \Phi &= \begin{bmatrix} Q & Q & Q & \dots \\ Q & Q & Q & \dots \\ Q & Q & Q & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \\ \Omega &= \begin{bmatrix} R_1 & 0 & 0 & \dots \\ 0 & R_2 & 0 & \dots \\ 0 & 0 & R_3 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} & \Pi &= \begin{bmatrix} P_{11} & P_{12} & P_{13} & \dots \\ P_{21} & P_{22} & P_{23} & \dots \\ P_{31} & P_{32} & P_{33} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \\ \kappa &= \begin{bmatrix} K_{11} & K_{12} & K_{13} & \dots \\ K_{21} & K_{22} & K_{23} & \dots \\ K_{31} & K_{32} & K_{33} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} & \mu &= \begin{bmatrix} M_{11} & M_{12} & M_{13} & \dots \\ M_{21} & M_{22} & M_{23} & \dots \\ M_{31} & M_{32} & M_{33} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \end{aligned}$$

$$\begin{aligned} \kappa_i &= [K_{i1} \ K_{i2} \ K_{i3} \ \dots \ K_{ij} \ \dots] \quad \text{where } j \in N_i \\ \mu_i &= [M_{i1} \ M_{i2} \ \dots \ M_{ij} \ \dots] \quad \text{where } j \in N_i \\ \mu_i^- &= [M_{i1} \ M_{i2} \ \dots \ M_{ij} \ \dots] \quad \text{where } j \in N_i^- \end{aligned}$$

$$\Gamma_i = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_j \\ \vdots \end{bmatrix} \quad \text{where } j \in N_i$$

So our process in equations (24),(25) can be expressed as follows in *Global System Representation*:

$$x[k] = Ax[k-1] + w[k-1]$$

$$Y[k] = \Gamma X[k] + V[k]$$

GSR results is a new problem formulation by considering network topology and all the estimations in all nodes that is explained in the next section.

B. Problem Formulation

The problem formulation is based on the idea of correcting the predicted state using the state estimates and measurements from the neighbors and the measurement of operating node, just as in the conventional Kalman filter formulation. For correction the residue terms between each measurement or state estimate and the predicted state is weighed. For weighing each measurement or state estimate, a Kalman gain is used which is either Measurement Kalman gain K_{ij} or State Kalman gain M_{ij} .

Mathematically speaking the following formulation is found for i^{th} node:

predict:

$$\hat{x}_i^-[k] = A\hat{x}_i[k-1] \quad (32)$$

$$P_{ij}^-[k] = AP_{ij}[k-1]A^T + Q \quad (33)$$

update:

$$\begin{aligned} \hat{x}_i[k] &= \hat{x}_i^-[k] + \sum_{j \in N_i} K_{ij}[k](y_j[k] - C_j\hat{x}_i^-[k]) \\ &\quad + \sum_{j \in N_i^-} M_{ij}[k](\hat{x}_j^-[k] - \hat{x}_i^-[k]) \end{aligned} \quad (34)$$

$$= \sum_{j \in N_i} M_{ij}[k]\hat{x}_j^-[k] + \sum_{j \in N_i} K_{ij}[k]y_j[k] \quad (35)$$

$$\text{where } M_{ii} := I - \sum_{j \in N_i^-} M_{ij} - \sum_{j \in N_i} K_{ij}C_j \quad (36)$$

$$(37)$$

Here the predicted state $\hat{x}_i^-[k]$ is corrected with measurement and state residue terms $y_j - C_j\hat{x}_i^-[k]$ and $\hat{x}_j^-[k] - \hat{x}_i^-[k]$. These residue terms are weighed with corresponding Kalman gains K_{ij} and M_{ij} . These Kalman gains are found for minimized cost

function. The cost function is designated as sum of traces of all auto-covariance matrices P_{ii} as explained in the following paragraphs.

For the calculation of Kalman gains we need error covariance P_{ii} . For this purpose we use the estimation error:

$$\begin{aligned} \text{since } y_i &= C_i x + v_i \\ e_i &= x - \hat{x}_i = \sum_{j \in N_i} M_{ij} e_j^- - \sum_{j \in N_i} K_{ij} v_j \end{aligned} \quad (38)$$

In matrix form:

$$e_i = \mu_i \begin{bmatrix} e_1^- \\ e_2^- \\ e_3^- \\ \vdots \end{bmatrix} - \kappa_i \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \end{bmatrix} \quad (39)$$

since $E[e_j v_k^T] = 0 \quad j, k \in N_i$

$$P_{ii} = E[e_i e_i^T] = \mu_i \begin{bmatrix} P_{11}^- & P_{12}^- & \cdots \\ P_{21}^- & P_{22}^- & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \mu_i^T + \kappa_i \begin{bmatrix} R_1 & 0 & \cdots \\ 0 & R_2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \kappa_i^T \quad (40)$$

Thus it is shown that for the calculation of P_{ii} , calculation of all the cross covariance matrices P_{ij} is needed. Further more auto- and cross-covariance matrices of the neighbors are also needed. The real challenge in this research is finding these cross covariances in a decentralized fashion.

Thus using *Global System Representation* Kalman filter formulation can be made globally as follows;

predict:

$$\hat{X}^-[k] = \Delta \hat{X}[k-1] \quad (41)$$

$$\Pi^-[k] = \Delta \Pi[k-1] \Delta^T + \Phi \quad (42)$$

update:

$$\hat{X}[k] = \mu[k] \hat{X}^-[k] + \kappa[k] Y[k] \quad (43)$$

$$(Er[k] = X[k] - \hat{X}[k] = \mu[k] Err^-[k] - \kappa[k] V[k]) \quad (44)$$

$$\Pi[k] = E(Er[k] Er[k]^T) = \mu[k] \Pi[k]^- \mu[k]^T + \kappa[k] \Omega \kappa[k]^T \quad (45)$$

Trace of $\Pi[k]$ is the cost function in our optimization problem. Both state estimate and measurement sharing is done between neighboring nodes and the network connections are explicitly taken into account in the cost function which will be shown in the next two sections. It will also be shown that this formulation gives the same optimal estimate as conventional Kalman filter in case of global communication.

The communication topology is modeled such that K_{ij} and M_{ij} values must be zero in case there is no communication between i^{th} and j^{th} node. This means that the network communication must be known.

Note that in prediction step it is possible to decouple x_i terms in X and P_{ij}^- terms in $\Pi^-[k]$ as in equations (32) and (33).

It is also worth to note that M_{ij} where $i \neq j$ and K_{ij} values are independent variables where M_{ii} values depend on these variables.

After this step it's crucial to find κ and μ which minimizes the trace of equation (45). It is also important to note that

$$Tr(\Pi[k]) = Tr(P_{11}) + Tr(P_{22}) + \dots \quad (46)$$

1) *Solution for Global Communication:* This section shows a solution in case of global communication. It is shown that this solution gives the same result as central Kalman filter.

This means the estimated states at each node are identical since all of the nodes are fed with the same measurements from the field if the initial values are given the same. In this case the state transition is needless so we remove M_{ij} terms where $i \neq j$ since $\hat{x}_i - \hat{x}_j = 0$. With this reasoning, analysis for measurement sharing can be made accordingly. Throughout this paper, the matrix algebra derivations can be made with the help of "Derivatives of traces" list given in Appendix C [10].

$$Tr(\Pi[k]) = \sum_i^N Tr(P_{ii}) \quad \text{where} \quad (47)$$

$$Tr(P_{ii}) = Tr(M_{ii} P_{ii}^- M_{ii}^T) + Tr\left(\sum_j^N K_{ij} R_j K_{ij}^T\right) \quad \text{from (45)}$$

$$M_{ii} = I - \sum_{j=1}^N K_{ij} C_j = I - \kappa_i \Gamma$$

$$\text{then } Tr(P_{ii}) = Tr((I - \kappa_i \Gamma) P_{ii}^- (I - \kappa_i \Gamma)^T) + Tr(\kappa_i \Omega \kappa_i^T) \quad (48)$$

$$\frac{\partial}{\partial \kappa_i} Tr(\Pi[k]) = \frac{\partial}{\partial \kappa_i} Tr(P_{ii}) \quad \text{since only } P_{ii} \text{ depends on } \kappa_i \quad (49)$$

$$= -2P_{ii}^- \Gamma^T + 2\kappa_i \Gamma P_{ii}^- \Gamma^T + 2\kappa_i \Omega = 0 \quad (50)$$

$$\kappa_i = P_{ii}^- \Gamma^T (\Gamma P_{ii}^- \Gamma^T + \Omega)^{-1} \quad (51)$$

But this is nothing but the same Kalman gain obtained using conventional Kalman filter as in equation (6). Thus GSR approach gives the same result as central Kalman filter in case of global communication which is expected. In the next section solution for "Local Communication" in an arbitrary network is given.

2) *Solution for Local Communication:* In equation (40) it is shown that P_{ii} depends on only κ_i and μ_i . So also using the equation (46) we can make the decentralization:

$$\frac{\partial}{\partial \kappa_i} Tr(\Pi[k]) = \frac{\partial}{\partial \kappa_i} Tr(P_{ii}) \quad (52)$$

$$\frac{\partial}{\partial \mu_i^-} Tr(\Pi[k]) = \frac{\partial}{\partial \mu_i^-} Tr(P_{ii}) \quad (53)$$

where

$$\Pi = \mu_i \Pi^- \mu_i^T + \kappa_i \Omega \kappa_i^T \quad (54)$$

$$P_{ii} = \sum_{j \in N_i} \sum_{l \in N_i} M_{il} P_{lj} M_{il}^T + \sum_{j \in N_i} K_{ij} R_j K_{ij}^T \quad (55)$$

Then for the gains K_{ij} and M_{ij} at i^{th} node the following equations are derived;

$$\frac{\partial}{\partial K_{ij}} Tr(P_{ii}) = -2 \sum_{l \in N_i} M_{il} P_{li}^- C_j^T + 2K_{ij} R_j = 0 \quad \text{where } j \in N_i \quad (56)$$

$$\frac{\partial}{\partial M_{ij}} Tr(P_{ii}) = -2 \sum_{l \in N_i} M_{il} (P_{li}^- - P_{lj}^-) = 0 \quad \text{where } j \in N_i^- \quad (57)$$

rewriting equations (56), (57) and using equation (36):

$$P_{ii}^- C_j^T = \sum_{l \in N_i^-} M_{il} (P_{li}^- - P_{lj}^-) C_j^T + \sum_{l \in N_i} K_{il} C_l P_{ii}^- C_j^T + K_{ij} R_j \quad \text{where } j \in N_i \quad (58)$$

$$(P_{ii}^- - P_{ij}^-) = \sum_{l \in N_i^-} M_{il} (P_{li}^- - P_{lj}^-) + \sum_{l \in N_i} K_{il} C_l (P_{ii}^- - P_{ij}^-) - \sum_{l \in N_i^-} M_{il} (P_{li}^- - P_{lj}^-) \quad \text{where } j \in N_i^- \quad (59)$$

So this set of equations gives $2n - 1$ matrix equations from which the Kalman gains are calculated, where n is the number of nodes in N_i . These equations can be written in matrix form as;

$$[\kappa_i \quad \mu_i] \begin{bmatrix} \mathfrak{A}_i & \mathfrak{B}_i \\ \mathfrak{B}_i^T & \mathfrak{C}_i \end{bmatrix} = [P_{ii} \Gamma_i^T \quad \mathbb{P}_{ii}] \quad (60)$$

$$(61)$$

where

$$\mathfrak{A}_i = \Gamma_i P_{ii}^- \Gamma_i^T + \Omega_i \quad (62)$$

$$\mathfrak{B}_i = \Gamma_i \mathbb{P}_{ii} \quad (63)$$

$$\mathfrak{C}_i = \begin{bmatrix} \mathbb{P}_{ii} \\ \mathbb{P}_{ii} \\ \vdots \end{bmatrix} - \begin{bmatrix} \mathbb{P}_{1i} \\ \mathbb{P}_{2i} \\ \vdots \end{bmatrix} \quad \text{omitting } \mathbb{P}_{ii} - \mathbb{P}_{ii} \text{ term} \quad (64)$$

$$\mathbb{P}_{ji} = [P_{ji} \quad P_{ji} \cdots P_{ji} \cdots] - [P_{j1} \quad P_{j2} \cdots P_{jk} \cdots] \quad (65)$$

$$\text{where } j \in N_i \text{ and } k \in N_i^- \quad (66)$$

Hence the optimal DKF at i^{th} node is obtained with the following equations:

predict:

$$\hat{x}_i^- [k] = A \hat{x}_i [k-1] \quad (67)$$

$$P_{ij}^- [k] = A P_{ij} [k-1] A^T + Q \quad (68)$$

update:

$$\begin{aligned} \hat{x}_i [k] &= \hat{x}_i^- [k] + \sum_{j \in N_i} K_{ij} [k] (y_j [k] - C_j \hat{x}_i^- [k]) \\ &+ \sum_{j \in N_i} M_{ij} [k] (\hat{x}_j^- [k] - \hat{x}_i^- [k]) \end{aligned} \quad (69)$$

$$[\kappa_i \quad \mu_i] = [P_{ii} \Gamma_i^T \quad \mathfrak{B}_i] \begin{bmatrix} \mathfrak{A}_i & \mathfrak{B}_i \\ \mathfrak{B}_i^T & \mathfrak{C}_i \end{bmatrix}^{-1} \quad (70)$$

$$\Pi [k] = \mu [k] \Pi [k]^- \mu [k]^T + \kappa [k] \Omega \kappa [k]^T \quad (71)$$

Covariance matrices are required to be found for calculations of \mathfrak{A}_i , \mathfrak{B}_i and \mathfrak{C}_i . The governing equation for covariance

	CKF	LKF (@ i^{th} node)	Optimal DKF (@ i^{th} node)
Structure	Globally Connected Network – Central Processor	Arbitrary Network-Local Processors	Arbitrary Network-Local Processors
Required Information	A, C _j , R _j , Q, Y _j j ∈ N	A, C _i , R _i , Q, Y _i	A, C _j , R _j , Q, Y _j j ∈ N Y _k k ∈ N _i
Communicated Information	N/A	-C _i ^T R _i ⁻¹ C _i -C _i ^T R _i ⁻¹ Y _i	- \hat{x}_i -Y _i

Fig. 2. Comparison of Algorithms

matrices is equation (71). It involves all of the cross covariance matrices as well as all the Kalman gains in the network. Calculation of covariance matrices and the Kalman gains is independent of the measurements. So these values can be found by each node using the equations (70) and (71). So each node can calculate Π off-line although this calculation may cost too much and become impractical in case of a large number of nodes is used. Moreover, although this algorithm is optimal, it is *not* scalable. Note that each node has to know the global network connection to calculate Kalman gains regarding the connections.

The scalability issue is taken into consideration in the next section. Given the scalability constraint the problem formulation for a scalable and optimal algorithm is stated.

Another important characteristic of this method is the fact that as the states converge, cross covariances approach to each other. As a result \mathfrak{B}_i and \mathfrak{C}_i becomes singular and the inversion in equation (70) cannot be made. This is an inherent point that stems from the formulation of Π . As the states converge, Π becomes singular. Occurrence of singularity depends on the precision of the processor (because of rounding off to zero in calculation of rank of Π). The singularity can be expected by examining the equation (34) in problem formulation as well. Since the terms $\hat{x}_j - \hat{x}_i$ goes to zero, M_{ij} can go to any value.

Figure 2 shows a table in which some key differences like network structure, required and communicated information between the algorithms CKF, LKF and DKF are listed. In this table N is the set of all nodes in the network and N_i is the neighborhood of i^{th} node.

The Heat Flow Bar experiment illustrates the comparison between Optimal DKF, Local Kalman Filtering(LKF) and Central Kalman Filter as well as the singularity issue on an example.

Example: Heat Flow Bar

To illustrate the performance of Optimal DKF algorithm provided in this section, we simulate an experiment in which a bar is connected to two temperature reservoirs on both ends and heated from the middle. Our purpose is comparing DKF that we suggest in the previous section to CKF and LKF. Central Kalman filtering is done by a central processor which accesses all four measurements directly and processes them to make an estimate. LKF and DKF are run in the local processors(nodes). For LKF nodes transmit only their

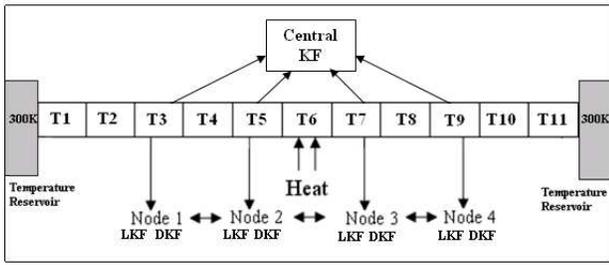


Fig. 3. Heat Flow Bar

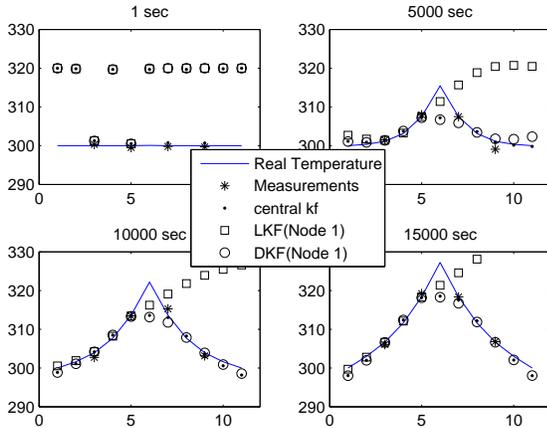


Fig. 4. Temperature Profile of The Bar Over Time

measurements to their neighbors whereas in DKF both states and measurements are transmitted and used as described in the previous section.

The temperature profile of the bar is needed to be observed. The bar is modeled with 11 states each of which represents the temperature of one segment of the bar as shown in figure 3. Reservoirs provide a boundary condition for the experiment and keep the temperature of the end points at 300K. The

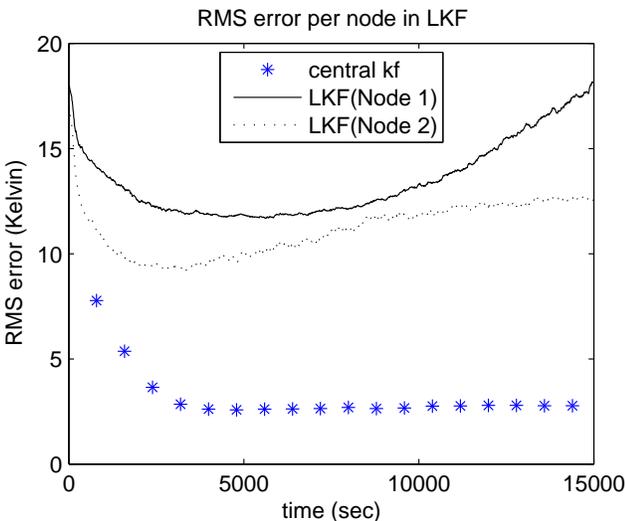


Fig. 5. Local Kalman Filter

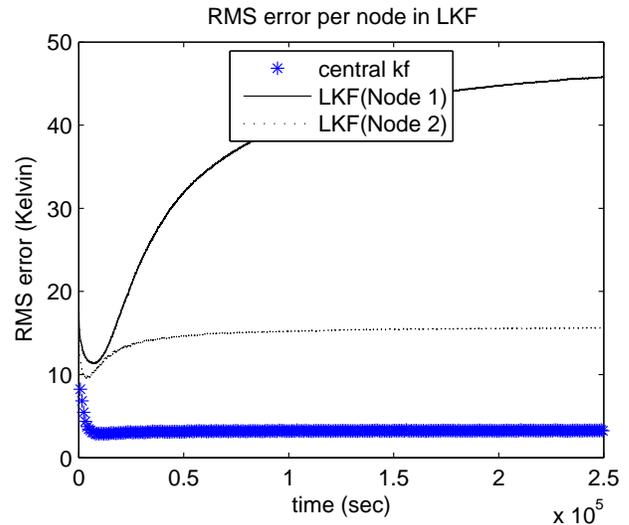


Fig. 6. Local Kalman Filter Convergence

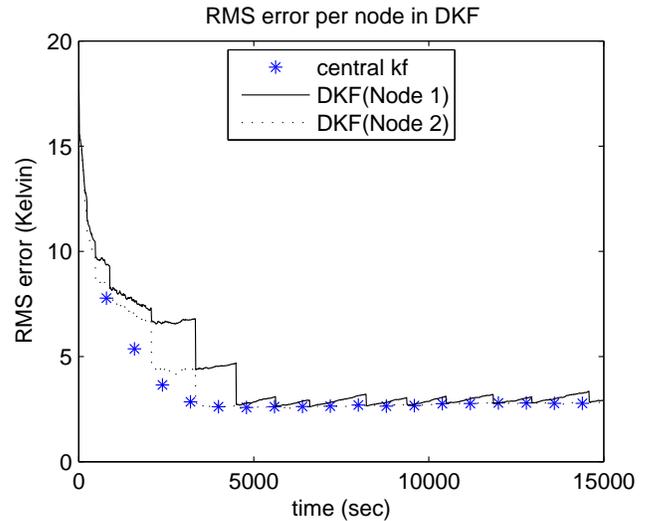


Fig. 7. Optimal Decentralized Kalman Filter

bar is heated from the 6th segment and temperature of only 3rd, 5th, 7th and 9th segments are measured. Each measurement is done by a node and the nodes can communicate with their neighbors to collaborate data. Initially the bar is kept at 300K which changes in time with the heat from the 6th segment. The temperature profile of the bar over time is shown in figure 4 with solid line.

The algorithms use the same state space model of the system as in figure 1, which is $x[k] = Ax[k-1] + w[k-1]$ and $y[k] = C_i x[k] + v_i[k]$. Here x denotes a vector consisting of 11 states each of which represents the temperature of one segment and w represents the heating/cooling noises on the states. Also C_i demotes measurement matrix and v_i measurement noise. For example $C_1 = [0010000000]$. Nodes do not know about the constant heat given to the bar. So in model we use a very high Q so unreliable model and a low R , reliable measurement.

It is crucial to note that the state space model that the

algorithms use does not know about the temperature reservoirs and the heat given to the bar. Because of that the model assumes a Gaussian process noise since this is the best it can do as it never knows about the amount, power and position of the heating or cooling being made. As mentioned before in first section, in KF process noise has to be modeled as zero mean Gaussian. This is also in accordance with the fact that the nodes do not know about the properties of the heat. This bad modeling leads to a bias in the estimate as seen in figure 6 since KF minimizes the sum of squared errors between the estimate and the mean value of "predicted real state" by the model. Since our KF models heat as zero mean Gaussian, this predicted real state differs from the real state considerably. The difference between "predicted real state" and real state creates a bias in the error in figure 6.

Although this is the case we still use this set up with a constant heat in the middle of the bar instead of setting up a scenario with a zero mean Gaussian heating/cooling process which could have been accurately described by the model. The reason for using this set up is that since the uncertainty in the process is very high and model is very uncertain, the measurements are very valuable for estimation and it is highly needed to reach far away nodes. In our system although we have 11 segments(states), we measure 4 of them, the rest is predicted by the model. If it would be possible to measure all of the segments then we could have been satisfied with only the measurements and do not even run any Kalman filter. However note that there are many unobserved segments.

As the process model is very inaccurate, we choose a very high Q , process noise covariance, whereas R , measurement noise covariance, is quite small. This makes measurements much more reliable than state predictions by the model in KF.

Since DKF incorporates states of the neighbors and states carry inherently measurements from all of the network, DKF proves to be very useful in this situation. On the other hand in LKF only measurements are sent which makes it impossible for example for first node to estimate accurately 7th segment because of the heat input at 6th segment.

Figure 5 shows performance of local Kalman filter. Root mean squared error between the real states and estimated states at 1st node and 2nd node are drawn versus time. Since the nodes are symmetric, the error of the 3rd and 4th nodes follow more or less the same trajectories as 2nd and 1st nodes respectively. For this reason these two nodes are not drawn for clarity. Also central KF is drawn for comparison. The steady state estimates are shown in figure 6. The 1st node have access to measurement from only 2nd node and itself. Because of this reason its estimation is worse than 2nd node. On the other hand 2nd node has access to the measurements from 1st and 3rd nodes, hence it achieves a better estimate by incorporating measurements from also the right side of the bar.

For the experiment which is shown in figure 7 a hybrid algorithm is used. In this experiment both states and measurements are sent and Optimal Decentralized Kalman filter algorithm together with LKF is applied. When Π matrix in section V-A

is full rank, Optimal DKF is used. In case Π matrix is singular LKF is applied. When LKF algorithm runs covariances also change and at some time points Π matrix again becomes full rank and DKF is used which causes the moments where we have "saw tooth" graph in the figure. When DKF works, even one time step proves to be sufficient to decrease the error significantly.

In figure 6 the error trajectories first tend to decrease but then begins to increase and settles at a high value. This behavior is related to the initial value attained in the algorithm. We chose initial temperature as 320K for each state. As measurements are more dominant for the measured states, for the first iterations of the algorithm the measured states quickly converges to the real value which makes the root mean squared error decrease quickly. After the quick convergence of the measured states, the other states are predicted by the model, using process noise model provided which is an inaccurate model. This causes the increase in the mean error. For example, the error between the steady state real state and the converged estimate of first node in LKF is found as $x_{lKF1} = [0.0, 0.0, 0.2, 0.1, 0.3, 0.0, 14.9, 29.8, 44.6, 59.5, 74.3]$. Since the first node accesses only the measurement from the second node it can estimate the left side very well, however for the right side it extrapolates the result for left side since it uses only the model for the estimation of the right side. This example shows the reason of the bias in error.

This experiment is a very clear example to show the importance of sending states in case the nodes measure different states of the plant and the process noise model is inaccurate. In our example the temperature profile differs significantly on left and right sides of the heating point and the model does not have direct information about the heating/cooling power or position. When only measurements are transmitted (LKF) first node cannot access to any information from the right side since it receives only the measurement of the second node. However for DKF case, first node accesses the measurements from also the right side inherently in the state received from the second node. Sending states provide the possibility of diffusing information from one node to the rest of the network. This makes it possible to converge to a better steady state error and DKF algorithm that is provided in the previous section provides the optimal solution for the given network topology and process model.

VI. OPTIMAL SCALABLE DKF

An important result of the previous section is the fact that it is not possible to find a scalable solution to Optimal DKF formulation made in section V-B2. The scalability issue in the Optimal DKF stems from the calculation of Π in equation (71). The idea of transmission of the covariance matrices as well as transmission of the estimates and measurements has been considered to see if it can solve this issue. It is realized that real problem is calculating cross-covariance, P_{ij} terms. Suppose we want to find P_{ii} term as in equation (40). Then it is needed to calculate also all the P_{ij} terms where i^{th} node

and j^{th} node can be at the farthest edges of the network. It is shown in the calculation of Π that even farthest nodes affect the algorithm of each other. Since distant nodes need information from each other to calculate their cross-covariance P_{ij} and local communication constraint is given, Optimal DKF algorithm derived in the previous section cannot be scalable.

The Covariance Intersection(CI) method [5] in which two correlated estimates are fused with an approximate covariance matrix is a practical, yet suboptimal solution for this problem. Like CI Method there are some studies which make similar approximations for data fusion for correlated data like Consensus Filter [11], [12]. However, in this research our motivation is to make a formulation for an optimal algorithm given the constraint of scalability.

A. Approach to the Scalability Problem

The first observation is that in the solution of this problem formulation in section V-B2, each node needs to know all the connections in the network which contradicts with the scalability constraint.

The second observation is that the solution of the problem depends on the formulation of the problem. If a formulation which takes scalability into account is made than a scalable solution can be found. In section V-B the problem formulation includes transmission of the state estimates as well as measurements which results in the correlation problem. In equation (40) cross correlations come into the picture.

A third observation is that "scalability" must be defined to fit into the scalable problem formulation.

Our approach to this problem depends on these three observations. Firstly we define scalability as "a node can see only its local network topology instead of whole network topology". This definition results in a problem formulation which involves only local network topology instead of global network topology of the network as in the optimal DKF. So any information regarding the non-neighbor nodes must not be involved in the algorithm of the operating node. The "local network topology" can be the node itself, first order neighborhood, second order neighborhood or n^{th} order neighborhood. In this study we take "local network topology" as first order neighborhood. So for example in the network in figure 8 the algorithm in the first node can involve cross covariance terms related to only second and third node. The rest of the cross covariances are modeled as "uncertainty" terms. The neighbors can help the first node to approximate this uncertainty terms as explained in the following sections.

B. Optimal Scalable DKF Problem Formulation

At this stage, we would like to introduce an example topology for which we will derive the optimal scalable DKF formulation without losing generality. Figure 8 shows our example topology and we are looking for an optimal scalable DKF algorithm specifically on the first node. We send

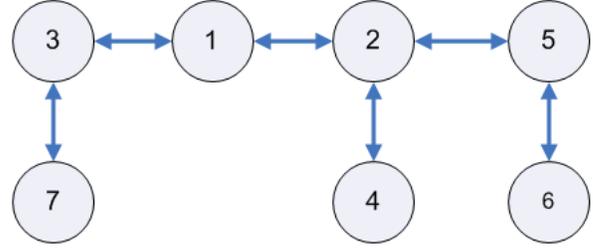


Fig. 8. Example Topology for OSDKF

both states and measurements and incorporate those into the estimation as in equation (35). Then from equation (39)

$$e_1 = [M_{11} \ M_{12} \ M_{13}] \begin{bmatrix} e_1^- \\ e_2^- \\ e_3^- \end{bmatrix} - [K_{11} \ K_{12} \ K_{13}] \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (72)$$

$$e_2 = [M_{21} \ M_{22} \ M_{24} \ M_{25}] \begin{bmatrix} e_1^- \\ e_2^- \\ e_4^- \\ e_5^- \end{bmatrix} - [K_{21} \ K_{22} \ K_{24} \ K_{25}] \begin{bmatrix} v_1 \\ v_2 \\ v_4 \\ v_5 \end{bmatrix} \quad (73)$$

$$e_3 = [M_{31} \ M_{33} \ M_{37}] \begin{bmatrix} e_1^- \\ e_3^- \\ e_7^- \end{bmatrix} - [K_{31} \ K_{33} \ K_{37}] \begin{bmatrix} v_1 \\ v_3 \\ v_7 \end{bmatrix} \quad (74)$$

However as the scalability condition implies, the algorithm in first node cannot involve e_4^- , e_5^- , e_7^- and v_4 , v_5 , v_7 terms. As a result e_2 and e_3 as seen at the first node can be written as;

$$e_2 = e_2^1 + \phi_2^1 \quad (75)$$

$$e_3 = e_3^1 + \phi_3^1 \quad (76)$$

where

$$e_2^1 = [M_{21} \ M_{22}] \begin{bmatrix} e_1^- \\ e_2^- \end{bmatrix} - [K_{21} \ K_{22}] \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (77)$$

$$e_3^1 = [M_{31} \ M_{33}] \begin{bmatrix} e_1^- \\ e_3^- \end{bmatrix} - [K_{31} \ K_{33}] \begin{bmatrix} v_1 \\ v_3 \end{bmatrix} \quad (78)$$

$$\phi_2^1 = [M_{24} \ M_{25}] \begin{bmatrix} e_4^- \\ e_5^- \end{bmatrix} - [K_{24} \ K_{25}] \begin{bmatrix} v_4 \\ v_5 \end{bmatrix} \quad (79)$$

$$\phi_3^1 = M_{37}e_7^- - K_{37}v_7 \quad (80)$$

Then

$$P_{12} = [M_{11} \ M_{12} \ M_{13}] \begin{bmatrix} P_{11}^- & P_{12}^- \\ P_{21}^- & P_{22}^- \\ P_{31}^- & P_{32}^- \end{bmatrix} \begin{bmatrix} M_{21}^T \\ M_{22}^T \end{bmatrix} + [K_{11} \ K_{12} \ K_{13}] \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} K_{24}^T \\ K_{25}^T \end{bmatrix} + S_{12}^1 \quad (81)$$

where

$$S_{12}^1 = E[e_1 \phi_2^T] = [M_{11} \quad M_{12} \quad M_{13}] \begin{bmatrix} P_{14}^- & P_{15}^- \\ P_{24}^- & P_{25}^- \\ P_{34}^- & P_{35}^- \end{bmatrix} \begin{bmatrix} M_{24}^T \\ M_{25}^T \end{bmatrix} \quad (82)$$

Similarly

$$P_{23} = [M_{21} \quad M_{22} \quad 0] \begin{bmatrix} P_{11}^- & P_{12}^- & P_{13}^- \\ P_{21}^- & P_{22}^- & P_{23}^- \\ P_{31}^- & P_{32}^- & P_{33}^- \end{bmatrix} \begin{bmatrix} M_{31}^T \\ 0 \\ M_{33}^T \end{bmatrix} \\ + [K_{21} \quad K_{22} \quad 0] \begin{bmatrix} R_1 & 0 & 0 \\ 0 & R_2 & 0 \\ 0 & 0 & R_3 \end{bmatrix} \begin{bmatrix} K_{31}^T \\ 0 \\ K_{33}^T \end{bmatrix} + S_{23}^1 \quad (83)$$

where

$$S_{23}^1 = E[e_2 \phi_3^T] + E[\phi_2 e_3^T] + E[\phi_2 \phi_3^T] \quad (84)$$

Generalizing these cross covariance calculations we get

$$\Pi_1 = \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix} \quad (85)$$

$$\mu_1 = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & 0 \\ M_{31} & 0 & M_{33} \end{bmatrix} \quad (86)$$

$$\kappa_1 = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & 0 \\ K_{31} & 0 & K_{33} \end{bmatrix} \quad (87)$$

$$\Omega_1 = \begin{bmatrix} R_1 & 0 & 0 \\ 0 & R_2 & 0 \\ 0 & 0 & R_3 \end{bmatrix} \quad (88)$$

$$\Sigma_1 = \begin{bmatrix} 0 & S_{12}^1 & S_{13}^1 \\ S_{21}^1 & S_{22}^1 & S_{23}^1 \\ S_{31}^1 & S_{32}^1 & S_{33}^1 \end{bmatrix} \quad (89)$$

$$\Pi_1 = \mu_1 \Pi_1 \mu_1^T + \kappa_1 \Omega_1 \kappa_1^T + \Sigma_1 \quad (90)$$

The solution of this formulation depends on the calculations of unknown S_{ij}^1 matrices in Σ_1 . An estimate or an educated approximation of these values can lead to the solution of the formulation. Estimation or approximation of these "uncertainty" terms is proposed as a future study to this research. The constraint for approximating or estimating S_{ij}^1 terms is that only local communication is allowed for all nodes. Some ideas about modeling these "uncertainty" terms is given in the next section.

C. Ideas for modeling uncertainty terms, Σ

Examining S_{12}^1 we see that every term in S_{12}^1 can be accessed by 2^{nd} node except for P_{34}^- and P_{35}^- terms. For this reason an approximation can be made by 2^{nd} node for calculation of P_{12} at 1^{st} node and then it can be sent to 1^{st} node. A possible approximation can be as follows by taking $P_{34} \approx P_{14}$ and $P_{35} \approx P_{15}$;

$$S_{12}^{1*} = [M_{11} + M_{13} \quad M_{12}] \begin{bmatrix} P_{14}^- & P_{15}^- \\ P_{24}^- & P_{25}^- \end{bmatrix} \begin{bmatrix} M_{24}^T \\ M_{25}^T \end{bmatrix} \quad (91)$$

Similar approaches can be made for the other terms of Σ_1 and these uncertainties can be approximated by neighbor nodes and sent to the operating node. This approximation would surely bring a bias in converged estimate. The stability and convergence analysis of the proposed solutions have to be made for the resulting algorithm.

VII. CONCLUSION

In this study we examined some optimality problems arising by the decentralization of the Kalman filter. The first contribution of this research is the introduction of a unique approach to DKF, which is called *Global System Representation*. Considering the network of nodes as a whole, we stated an problem formulation for optimality depending on the given network topology. Solving this problem formulation, we obtained an optimal DKF dependent on the given network graph. This solution provided us with optimal local algorithms which also takes the network graph into account. This result is very important since it proposes an algorithm which makes the best estimate that can be obtained, taking the network topology into account. So far in the literature the proposed algorithms for DKF have been compared with the central KF. However the central KF does not take the network topology into account whereas our Optimal DKF does and proves to be optimal in a sense that it minimizes the least squares error between the state estimate and predicted real state by the model.

Although this solution for DKF is proved to be optimal, since it depends on the network graph, it is still unscalable. So the next challenge was incorporating scalability condition into the problem formulation so that a scalable algorithm can be obtained.

A mathematical formulation of the scalable algorithm is made in the last section. The solution of the formulation remained as dependant on the modeling of some uncertainty terms which is proposed as a follow up to this research.

Lastly some comments, approximation tips are made for the modeling of these uncertainty terms. Still the formulation of these "uncertainty" terms, stability analysis of it remained as to be researched in the future.

APPENDIX A

DERIVATION OF KALMAN FILTER EQUATIONS

Predict:

$$\text{define } e = x - \hat{x} \quad \text{and} \quad e^- = x - \hat{x}^- \quad (92)$$

$$e[k] = Ae[k-1]^- + w[k-1] \quad \text{from (1), (4), (92)} \quad (93)$$

$$P[k] = E[e[k]e[k]^T] \quad (94)$$

$$P[k] = AP[k-1]A^T + Q \quad (95)$$

Update:

$$\hat{x}[k] = \hat{x}^-[k] + K[k](y[k] - C\hat{x}^-[k]) \quad (96)$$

$y[k] - C\hat{x}^-[k]$ term is known as "residual" or "innovation". Innovation represents the discrepancy between the prediction $C\hat{x}^-[k]$ and the measurement $y[k]$. The idea is improving the prediction towards the measurement by using a weight of $K[k]$, which is called Kalman gain. This Kalman gain can be calculated optimally as a function of the measurement covariance and predicted state covariance.

$$e[k] = (x[k] - \hat{x}^-[k]) - K[k](Cx[k] - C\hat{x}^-[k] + v[k]) \quad (97)$$

from (2), (96) and (92)

$$e[k] = (I - K[k]C)e^-[k] - K[k]v[k] \quad (98)$$

$$P[k] = E[e[k]e[k]^T] \quad (99)$$

$$P[k] = (I - K[k]C)P^-[k](I - K[k]C)^T + KRK^T \quad (100)$$

Here the trace of $P[k]$, which is the sum of the variances of each state, should be minimized with respect to $K[k]$ to find an optimal $K[k]$.

since $Tr(A) = Tr(A^T)$ and $P = P^T$

$$Tr(P[k]) = Tr(P^-[k]) - 2Tr(P^-[k]C^TK[k]^T) \quad (101)$$

$$+ Tr(K[k](CP^-[k]C^T + R)K[k]^T) \quad (102)$$

$$\frac{d}{dK[k]} Tr(P[k]) = -2P^-[k]C^T + 2K[k](CP^-[k]C^T + R) = 0 \quad (103)$$

$$K[k] = P^-[k]C^T(CP^-[k]C^T + R)^{-1} \quad (104)$$

using (104) in (100)

$$P[k] = (I - K[k]C)P^-[k] \quad (105)$$

APPENDIX B

DERIVATION OF INFORMATION FILTER

Predict step is found by direct substitution of Information filter and Information State in equations (4) and (5). Update step is found eliminating $K[k]$ by rewriting it in terms of $P[k]$ and substituting into the equation (8).

$$K[k](CP^-[k]C^T + R) = P^-[k]C^T \quad \text{from equation (6)} \quad (106)$$

$$K[k]R = P^-[k]C^T - K[k]CP^-[k]C^T \quad (107)$$

$$K[k]R = (I - K[k]C)P^-[k]C^T \quad (108)$$

$$K[k] = P[k]C^TR^{-1} \quad \text{from equation (8)} \quad (109)$$

Using (8) and (109)

$$P^-[k]^{-1} = P[k]^{-1}(I - K[k]C) \quad (110)$$

$$P[k]^{-1} = P^-[k]^{-1} + C^TR^{-1}C \quad (111)$$

$$J[k] = J^-[k] + C^TR^{-1}C \quad (112)$$

Also using (109) and (110) in (7) we find the expression for Information state vector

$$\hat{x}[k] = (I - K[k]C)\hat{x}^-[k] + K[k]y[k] \quad (113)$$

$$P[k]^{-1}\hat{x}[k] = P[k]^{-1}(I - K[k]C)\hat{x}^-[k] + P[k]^{-1}K[k]y[k] \quad (114)$$

$$P[k]^{-1}\hat{x}[k] = P^-[k]^{-1}\hat{x}^-[k] + C^TR^{-1}y[k] \quad (115)$$

$$j[k] = j^-[k] + C^TR^{-1}y[k] \quad (116)$$

APPENDIX C

DERIVATIVES OF TRACES

A. First Order

$$\begin{aligned} \frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{X}) &= \mathbf{I} \\ \frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{X}\mathbf{A}) &= \mathbf{A}^T \\ \frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{A}\mathbf{X}\mathbf{B}) &= \mathbf{A}^T\mathbf{B}^T \\ \frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{A}\mathbf{X}^T\mathbf{B}) &= \mathbf{B}\mathbf{A} \\ \frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{X}^T\mathbf{A}) &= \mathbf{A} \\ \frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{A}\mathbf{X}^T) &= \mathbf{A} \\ \frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{A} \otimes \mathbf{X}) &= \text{Tr}(\mathbf{A})\mathbf{I} \end{aligned}$$

B. Second Order

$$\begin{aligned} \frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{X}^2) &= 2\mathbf{X}^T \\ \frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{X}^2\mathbf{B}) &= (\mathbf{X}\mathbf{B} + \mathbf{B}\mathbf{X})^T \\ \frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{X}^T\mathbf{B}\mathbf{X}) &= \mathbf{B}\mathbf{X} + \mathbf{B}^T\mathbf{X} \\ \frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{X}\mathbf{B}\mathbf{X}^T) &= \mathbf{X}\mathbf{B}^T + \mathbf{X}\mathbf{B} \\ \frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{A}\mathbf{X}\mathbf{B}\mathbf{X}) &= \mathbf{A}^T\mathbf{X}^T\mathbf{B}^T + \mathbf{B}^T\mathbf{X}^T\mathbf{A}^T \\ \frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{X}^T\mathbf{X}) &= 2\mathbf{X} \\ \frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{B}\mathbf{X}\mathbf{X}^T) &= (\mathbf{B} + \mathbf{B}^T)\mathbf{X} \\ \frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{B}^T\mathbf{X}^T\mathbf{C}\mathbf{X}\mathbf{B}) &= \mathbf{C}^T\mathbf{X}\mathbf{B}\mathbf{B}^T + \mathbf{C}\mathbf{X}\mathbf{B}\mathbf{B}^T \\ \frac{\partial}{\partial \mathbf{X}} \text{Tr}[\mathbf{X}^T\mathbf{B}\mathbf{X}\mathbf{C}] &= \mathbf{B}\mathbf{X}\mathbf{C} + \mathbf{B}^T\mathbf{X}\mathbf{C}^T \\ \frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{A}\mathbf{X}\mathbf{B}\mathbf{X}^T\mathbf{C}) &= \mathbf{A}^T\mathbf{C}^T\mathbf{X}\mathbf{B}^T + \mathbf{C}\mathbf{A}\mathbf{X}\mathbf{B} \\ \frac{\partial}{\partial \mathbf{X}} \text{Tr}[(\mathbf{A}\mathbf{X}\mathbf{b} + \mathbf{c})(\mathbf{A}\mathbf{X}\mathbf{b} + \mathbf{c})^T] &= 2\mathbf{A}^T(\mathbf{A}\mathbf{X}\mathbf{b} + \mathbf{c})\mathbf{b}^T \\ \frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{X} \otimes \mathbf{X}) &= \frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{X})\text{Tr}(\mathbf{X}) = 2\text{Tr}(\mathbf{X})\mathbf{I} \end{aligned}$$

REFERENCES

- [1] R. Kalman, "A new approach to linear filtering and prediction problems," *Transaction of the ASME Journal of Basic Engineering*, vol. 82, no. D, pp. 35-42, 1960.
- [2] A. Mutambara and H. Durrant-Whyte, "A formally verified modular decentralized robot control system," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1993.
- [3] B. Quine, J. Uhlmann, and H. Durrant-Whyte, "Implicit jacobians for linearised state estimation in nonlinear systems," *Proceedings of the 1995 American Control Conference*, IEEE Press, 1995.
- [4] A. A. Lange, "An algorithmic approach for improving and controlling the quality of upper-air data," *WMO Instruments and Observing Methods Report, No. 35*, World Meteorological Organization, Geneva, Switzerland, 1989, 1989.
- [5] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," *Proceedings of the American Control Conference*, New Mexico, 1997.

- [6] P. S. Maybeck, *Stochastic models, estimation, and Control*. Academic Press, Harcourt Brace Jovanovich, 1979, ch. Chapter 1.
- [7] G. Welch and G. Bishop, "An introduction to kalman filter," 2006, uNC-Chapel Hill, TR-95-041.
- [8] H. Hashmipour, S. Roy, and A. Laub, "Decentralized structures for parallel kalman filtering," *IEEE Transactions on Automatic Control*, vol. 33, no. 1, pp. 88–93, 1988.
- [9] H. Durant-Whyte, B. Rao, and H. Hu, "Towards a fully decentralized architecture for multi-sensor data fusion," in *1990 IEEE Int. Conf. on Robotics and Automation*, Cincinnati, Ohio, USA, 1990, pp. 1331–1336.
- [10] K. Petersen and P. M.S., *The Matrix Cookbook*, 2008.
- [11] M. Alighanbari and J. P. How, "An unbiased kalman consensus algorithm," *Proceedings of the 2006 American Control Conference, Minneapolis, Minnesota, USA*, 2006.
- [12] R. Olfati-Saber, "Distributed kalman filtering for sensor networks," *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007.