Journal of Scientific Research & Reports
13(5): 1-10, 2017; Article no.JSRR.27886
ISSN: 2320-0227

SCIENCEDOMAIN ***international***
www.sciencedomain.org

Decentralized and Distributed Information Filter for Autonomous Intelligent Multisensor Systems

A. Olatunbosun[1] and I. O. Lawal[2*]

[1]Department of Electrical and Electronic, Faculty of Technology, University of Ibadan, Ibadan, Oyo State, Nigeria.
[2]Centre for Space Transport and Propulsion, National Space Research and Development Agency, Federal Ministry of Science and Technology, Abuja, Nigeria.

Authors' contributions

This work was carried out in collaboration between both authors. Author IOL designed the study, wrote the first and second drafts of the manuscript and did the simulations in MATLAB/Simulink. Author AO edited the second draft of the manuscript, verified all the simulation results and make necessary corrections. Both authors read and approved the final manuscript.

Article Information

DOI: 10.9734/JSRR/2017/27886
Editor(s):
(1) Grigorios L. Kyriakopoulos, School of Electrical and Computer Engineering, National Technical University of Athens (NTUA), Greece.
Reviewers:
(1) Sherin Zafar, Jamia Hamdard University, India.
(2) Grienggrai Rajchakit, Mae Jo University, Thailand.
Complete Peer review History: http://www.sciencedomain.org/review-history/18642

Original Research Article

Received 23rd Jun 2016
Accepted 17th July 2016
Published 14th April 2017

ABSTRACT

This research paper is on development of distributed and decentralized multisensor data estimation and fusion algorithm with linear information filter for fusing the information from these various sensors and embedded in the developed system. The estimation technique is modified Kalman filter that provides estimates of the information about a certain state. Consequently the developed system is Radar Tracking System (RTS); comprising of array of antennae, and GPS mounted on a Digital Rate Gyroscope (DRG), for accurate and effective Information Gathering and Processing (IGP). The system was simulated to determine the stability and drift error rate of the trajectory over a period of 300secs. The results are also derived under usual methods and proof procedure of some existing literature. The advantage of this algorithm is that the transmission of new information can be done asynchronously only when the amount of information, measure with its randomness, is larger than a fixed threshold level.

**Corresponding author: E-mail: lawal_segunmail@yahoo.com;*

Keywords: Decentralized; distributed; information filter; Autonomous Intelligent Multisensor Systems (AMISS).

1. INTRODUCTION

Since the early-1989, work on decentralized systems began as part of the ESPRIT project SKIDS. In the SKIDS project, a fully decentralized surveillance system was implemented using four cameras and a Transputer based architecture. The network was a fully-connected point-to- point topology. The system was capable of tracking multiple targets (humans and robots) and addressed such issues as decentralized data association and decentralized identification [1]. The SKIDS demonstrator, which continued to be refined and operated for almost 10 years, laid the basis for all subsequent work on decentralized data fusion.

Practical estimation and control applications generally entail the difficulty that the measured data is corrupted by noise. Thus, the derivation of specific estimates from noisy measurements is of little value, if the involved uncertainties are not considered appropriately. A common approach consists in modeling uncertain quantities stochastically by calculating mean and estimate variance. These parameters correspond to a Gaussian density characterizing the uncertainty about the state. For a linear evolution of the state variable and linear observation models, the Kalman filter [2] formulas represent an optimal closed-form solution to the estimation problem. In linear/nonlinear situations, mean and estimate variance often do not suffice to describe the underlying uncertainty and they can even be deceptive, in particular when the true probability density of the state estimate is multi-modal. Since a closed- form computation of the actual density is generally not possible, a lot of effort has been focused on approximate solutions to nonlinear Bayesian state estimation. For this purpose, either the underlying system and measurement mappings or the underlying probability densities are approximated.

In the former case, the nonlinearities are generally linearized by first-order Taylor series approximations, which are performed within the extended Kalman filter, or by a linear regression analysis, of which the unscented Kalman filter [3] and [4] is a well-known example. Of course, due to the Gaussian assumption, they only provide very limited capabilities for capturing multi-modalities. This can be better achieved through density approximations such as particle filters or finite-dimensional representations via orthonormal bases, e.g., truncated Fourier, or wavelet series. All these approaches are intended to provide finite and implementable parameterizations of the state estimates.

In the recent past, the information filter [5] and [6] has been derived as an inverse covariance formulation of the Kalman filter, with the benefit that the fusion of multiple sensor data can easily be distributed. This reformulation has widely been applied to sensor networks [6]. The information filter herein calculates estimates on the information about the state and not on the state itself, which simplifies the fusion significantly. Also, the elimination of double-counted information between two sensors nodes becomes very simple in the information space.

The extended information filter comes with the same drawbacks as the extended Kalman filter and developing distributed fusion structures for arbitrary probability densities is elaborate.

In order to solve these problems, the aims of this paper is to lay down the theoretical foundation for tractable linear/nonlinear decentralized and distributed data fusion. For this purpose, Section 2 briefly describes decentralized and distributed data fusion (DDDF); follow by Implementation of Control Architecture for AIMSS, The algorithmic structures of a decentralized sensing node for AIMSS were fully analysis. Section 3 provides numerical simulation example. Section 4 the result was discussed. Section 5 performance evaluation decentralized and distributed linear Information filter and concluded in section 6 of this paper.

2. FULLY DECENTRALIZED AND DISTRIBUTED MULTISENSOR NETWORK ARCHITECTURE

A decentralized system should not be confused with a distributed system. Distributed systems typically contain some form of centralized resource and as a result will never be scalable.

A distributed sensor network (DSN): can be defined as a set of spatially scattered intelligent sensors designed to obtain measurements from the environment, abstract relevant information from the data gathered, and to derive appropriate inferences from the information gained. Distributed sensor networks depend on multiple processors to simultaneously gather and process information from many sources.

Fully decentralized processing architecture: a decentralized data fusion system consists of a network of sensor nodes, each with its own processing facility, which together do not require any central fusion or central communication facility. In such a system, fusion occurs locally at each node on the basis of local observations and the information communicated from neighboring nodes. At no point is there a common place where fusion or global decisions are made. This is because a computational or communication bottleneck will always be associated with the central resource. This central resource is also a potential weakness because its failure will render the entire system unusable.

A decentralized data fusion system is characterized by three constraints:

1. There is no single central fusion center; no one node should be central to the successful operation of the network.
2. There is no common communication facility; nodes cannot broadcast results and communication must be kept on a strictly node-to-node basis.
3. Sensor nodes do not have any global knowledge of sensor network topology; nodes should only know about connections in their own neighborhood.

The constraints imposed provide a number of important characteristics for decentralized data fusion systems:

2.1 Implementation of Control Architecture for AIMSS

The control architecture of this paper have been develop in [7], were we comes up with decentralized, distributed, local interaction and heterarchical structure for AIMSS. The system described in this paper employs *a fully distributed and decentralized* architecture for multisensor data fusion. The motivation behind all these organizations is to hide complexity and make each sensor function as modular as possible. Typically, in these organizations, the architecture comes first, and the data fusion algorithm is then “designed to fit”.

2.1.1 Mathematical structure Distributed Decentralized Data Fusion (DDDF)

The problem is to use appropriate models and sensors to control the state of a dynamic system (e.g., industrial robot, mobile robot, autonomous vehicle, surgical robot, etc.). Usually such systems involve real-time feedback control loops in addition to state estimation, uncertainty models. In spite of this Distributed Decentralized Data Fusion (DDDF) methods were initially motivated by the insight that the information form of the Kalman filter data fusion algorithm could be implemented by simply adding information contributions from observations.

The mathematical structure of a DDDF sensor node is shown in Fig. 1 the sensor is modeled directly in the form of a likelihood function. Since, Kalman filter does not design sensor systems,

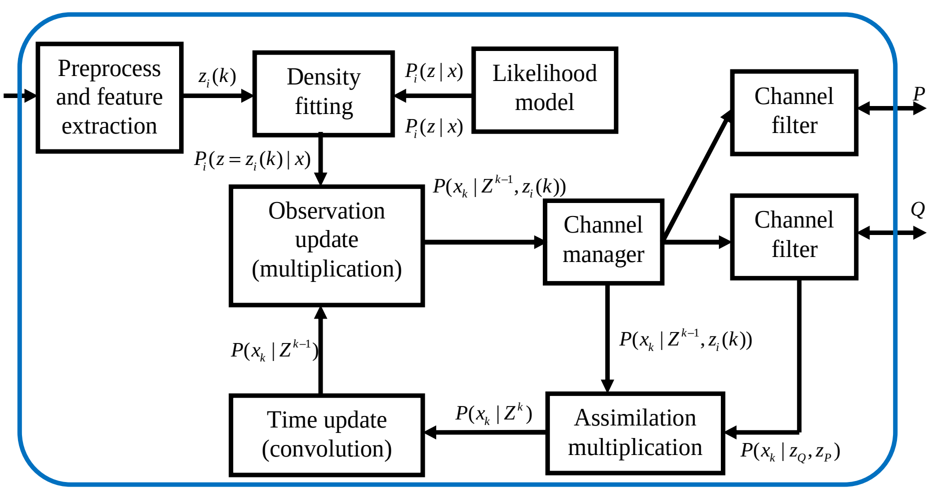

Fig. 1. Mathematical structure of decentralized distributed data fusion node
(Durrant-Whyte and Henderson, 2009)

but it provides the tool for doing it defensibly. That tool is the model for estimation uncertainty. The covariance propagation equations derived from the model can be used in characterizing estimation uncertainty as a function of the parameters of the design. Some of these parameters are statistical, such as the noise models of the sensors under consideration.

2.1.2 Information filtering for decentralized and distributed multisensor data fusion

Practical systems are in general affected by perturbations and inaccuracies, which have to be dealt with. For uncertain linear discrete-time systems and linear observation models

$$x(k) = F(k)x(k-1) + B(k)u(k) + G(k)w(k), \quad (1)$$

$$z(k) = H(k)x(k) + D(k)v(k). \quad (2)$$

The Kalman filter [8] and its derivatives provide estimates on the uncertain state $x(k)$ by computing mean $\hat{x}(k)$ and estimate covariance matrix $P(k)$ at each time instant k, the function $u(k)$ usually represents a known control input. For the rest of the discussion in this paper, we will assume that $u(k) = 0$. A basic assumption in the derivation of the Kalman filter is that the random sequences $w(k)$ and $v(k)$ describing process and observation noise are all Gaussian, temporally uncorrelated and zero-mean.

The ability to construct decentralized data fusion architecture clearly depends on whether it is possible to efficiently decentralize existing centralized data fusion algorithms [9]. For most common data fusion algorithms, this turns out to be possible, and indeed many decentralized data fusion algorithms are, surprisingly, more efficient, in terms of both computation and communication, than conventional distributed, federated or hierarchical data fusion algorithms.

The initial impetus for decentralized systems was the development of a decentralized form of the Kalman filter algorithm. This is achieved by first recasting the usual Kalman filter state estimation problem in information form. Briefly, consider a state $x(k)$ with discrete-time index k, a sequence of observations $Z^k = \{z(1), ..., z(k)\}$, the estimate of this state (conditional mean)

$$\hat{x}(i \mid j) = E\{x(i) \mid Z^j\} \quad (3)$$

together with estimate covariance

$$P(i \mid j) = E\{\tilde{x}(i \mid j)\tilde{x}^T(i \mid j) \mid Z^j\}. \quad (4)$$

The information form of the Kalman filter is obtained by re-writing the state estimate and covariance in terms of two new variables

$$\hat{y}(i \mid j) = P^{-1}(i \mid j)\hat{x}(i \mid j), \quad (5)$$

$$Y(i \mid j) = P^{-1}(i \mid j), \quad (6)$$

and, assuming observations in the form

$$z(k) = H(k)x(k) + v(k) \quad (7)$$

with

$$E\{w(i)w^T(j)\} = \delta ijR(i), \quad (8)$$

the information associated with an observation may be written in the form

$$i(k) \triangleq H^T(k)R^{-1}(k)z(k), \quad (9)$$

$$I(k) \triangleq H^T(k)R^{-1}(k)H(k), \quad (10)$$

with these definitions, the update stage of the Kalman filter becomes (Information Measurement Update)

$$\hat{y}(k \mid k) = \hat{y}(k \mid k-1) + i(k) \quad (11)$$

$$Y(k \mid k) = Y(k \mid k-1) + I(k) \quad (12)$$

This simple update form comes at the cost of complexity in the prediction stage which is dual to the update stage for the conventional Kalman filter [10,11,12]. The information form of the Kalman filter, while widely known, is not commonly used because the update terms are of dimension the state, whereas in the distributed Kalman filter updates are of dimension the observation. For single sensor estimation problems, this argues for the use of the Kalman filter over the information filter. However, in multiple sensor problems, the opposite is true.

The reason is that with multiple sensor observations

$$z_i(k) = H_i(k)x(k) + v_i(k), \quad i = 1, ..., N. \tag{13}$$

The estimate cannot be constructed from a simple linear combination of contributions from individual sensors

$$\hat{x}(k \mid k) \neq \hat{x}(k \mid k-1) + \sum_{i=1}^{N} S_i(k)[z_i(k) - H_i(k)\hat{x}(k \mid k-1)], \tag{14}$$

(with $S_i(k)$ independent gain matrices) as the innovation generated from each sensor is correlated because they share common information through the prediction $\hat{x}(k \mid k-1)$. However, in information form, estimates can be constructed from linear combinations of observation information

$$\hat{y}(k \mid k) = \hat{y}(k \mid k-1) + \sum_{i=1}^{N} i_i(k), \tag{15}$$

as the information terms $i_i(k)$ from each sensor are uncorrelated. Once the update equations have been written in this simple additive form, it is straightforward to distribute the data fusion problem (unlike for a Kalman filter); each sensor node simply generates the information terms $i_i(k)$, and these are summed at the fusion center to produce a global information estimate.

2.1.3 Decentralize information filter

The review shows that [3] and [7,10] as done a critical work on information filter and decentralize and distributed information filter. To decentralize the information filter all that is necessary is to replicate the central fusion algorithm (summation) at each sensor node and simplify the result. This yields a surprisingly simple nodal fusion algorithm. The algorithm is described graphically in Fig. 2 for a typical sensor node, i, in a decentralized data fusion system. The node generates information measures $\hat{y}_i(k \mid k)$ at a time k given observations made locally and information communicated to the node up to time k.

The node implements a local prediction stage to produce information measure predictions $\hat{y}_i(k \mid k-1)$) at time k given all local and communicated data up to time $k-1$ (this prediction stage is often the same on each node and may, for example, correspond to the path predictions of a number of common targets). At this time, local observations produce local information measures $i(k)$ on the basis of local observations.

The prediction and local information measures are combined, by simple addition, into total local information measure $\tilde{y}_i(k \mid k)$ at time k. This measure is handed down to the communication channels for subsequent communication to other nodes in the decentralized network. Incoming information from other nodes $\hat{y}_{ji}(k \mid k)$ is extracted from appropriate channels and is assimilated with the total local information by simple addition. The result of this fusion is a locally available global information measure $\hat{y}_i(k \mid k)$. The algorithm then repeats recursively.

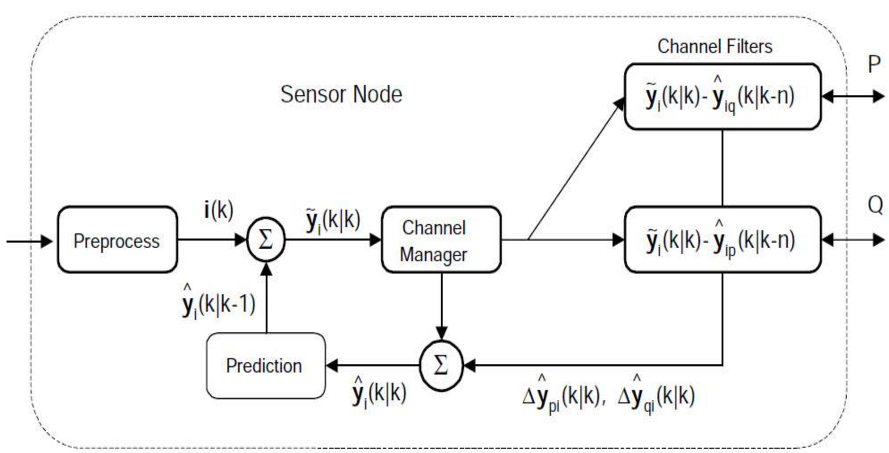

Fig. 2. Algorithmic structure of a decentralized sensing node

The communication channels exploit the associative property of information measures. The channels take the total local information $\tilde{y}_i(k|k)$ and subtract out all information that has previously been communicated down the channel, $\hat{y}_{ji}(k|k)$, thus transmitting only new information obtained by node i since the last communication. Intuitively, communicated from node i thus consists only of information not previously transmitted to a node j; because common data has already been removed from the communication, node j can simply assimilate incoming information measures by addition. As these channels essentially act as information assimilation processes, they are usually referred to as channel filters [13].

3. NUMERICAL SIMULATION EXAMPLE

The Decentralized Kalman Tracking System which consists of GPS mounted on digital rate gyro with cameras as a tracking system, this illustrative example is to demonstrate the effectiveness and accuracy of the proposed modified Kalman filter MSDDF-AIMSS. Consider the following linear system, which corresponds to a vehicle moving in three dimensional co-ordinate spaces.

DRG use to measure the angular rate $\dot{\theta}$. Use the data provided in data specification sheet to determine a model for the noise process and design a Kalman filter to determine the position and velocity of the trajectory of the vehicle.

The derivation of the Kalman filter assumes that the disturbances and noise are independent and white. Removing the assumption of independence is straightforward and simply results in a cross term $(E\{W(t)V(s)\} = R_{wv}\delta(s-t))$ being carried through all calculations.

The error state vector $\delta\vec{x}(t)$ is given by

$$\delta\vec{x}(t) = \begin{bmatrix} \delta\psi & \delta\theta & \delta\phi & \delta b_p & \delta b_q & \delta b_r \end{bmatrix}^T . \quad (16)$$

The first three entries of $\delta\vec{x}(t)$ represent Euler angle errors while $\delta b_p,\ \delta b_q,\ and\ \delta b_r$ represent errors in our knowledge of the rate gyro biases.

3.1 Time Update Equations

The dynamic matrix $A(t)$ is given by

$$A(t) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -10 & -0.13 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.13 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (17)$$

The process mapping matrix $G(t)$ is given by

$$G(t) = \left[0\ 0\ 0;0\ 0\ 0;0\ 0\ 0;0.7\ 0\ 0;0\ 0.7\ 0;\ 5.3\ 0\ 0\right]. \quad (18)$$

and relates the process noise vector $\vec{w}$, to $\delta\vec{x}$. The process noise vector $\vec{w}$ is given by

$$\vec{w} = \begin{bmatrix} n_p & n_q & n_r & w_p & w_q & w_r \end{bmatrix}^T . \quad (19)$$

The first three entries of $\vec{w}$ represent the wideband measurement errors on the p, q, and r gyro outputs. The last three entries of $\vec{w}$ are the driving noise terms for the stochastic gyro biases (i.e., the equivalent of $\vec{w}_{b_1}$ is $0.05 ⁄_{\text{sec}}$) In the actual implementation of the estimator, however, the process noise vector, $\vec{w}$, itself is not used. Instead, what is used is the process noise covariance matrix, R_w, and its associated power spectral density matrix, Q_w. The matrix R_w is defined as:

$$R_w = \varepsilon\{\vec{w}\vec{w}^T\}. \quad (20)$$

The symbol ε represents the expectation operator. Thus, the power spectral density matrix for $\vec{w}$ is denoted Q_w and is given by

$$Q_w = \begin{bmatrix} R_n & 0_{3x3} \\ 0_{3x3} & R_b \end{bmatrix} \quad (21)$$

The variables R_n and R_b are the Euler angle and gyro bias process noise matrices respectively. The matrix R_n is given by:

$$R_n = \begin{bmatrix} \sigma_p^2 & 0 & 0 \\ 0 & \sigma_q^2 & 0 \\ 0 & 0 & \sigma_r^2 \end{bmatrix} \quad (22)$$

Numerical values for $\sigma_p,\ \sigma_q\ and\ \sigma_r$ depend on the type of gyro being used and can be found in data sheet. The variables $\sigma_p^2,\ \sigma_q^2\ and\ \sigma_r^2$ are the variances of the wide-band noise on the three rate gyros. Similarly, numerical values for

σ_{w_p}, σ_{w_q}, *and* σ_{w_r}, are also found in the data sheet.

For the gyro bias process noise matrices when using ADIS16250/ADIS16255 rate gyro, the matrix R_b is given by:

$$R_b = \frac{2\sigma_\omega^2}{\tau_\omega}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (23)$$

Equation (23) we now equal,

$$Q_w = \text{dag}([\sigma_p^2;\sigma_q^2;\ \sigma_r^2;\frac{2\sigma_p^2}{\tau_\omega};\frac{2\sigma_q^2}{\tau_\omega};\frac{2\sigma_r^2}{\tau_\omega}]). \quad (24)$$

In this case matrix R_w is used in the equations for propagating the state error covariance matrix, P. Propagation of P forward in time is accomplished by using the solution to the discrete Riccati equation. Given the state error covariance matrix, $P(k)$, at time step k, then the covariance at time step $k+1$ is given by:

$$P(k \mid k) = A(k)P(k \mid k)A^T(k) + GR_wG^T. \quad (25)$$

this gives them a priori value of the covariance matrix of estimation uncertainty as a function of the previous a posteriori.

Updated state covariance equation

$$P(k \mid k) = P(k \mid k-1) + K(k)S(k)K^T(k) \quad (26)$$

3.2 Measurement (Update or Correction) Equations

At time k an observation $z(k)$ is made and the updated estimate $\hat{x}(k \mid k)$ of the state $x(k)$, together with the updated estimate covariance $P(k \mid k)$ is computed from the state prediction and observation according to

$$\hat{x}(k \mid k) = \hat{x}(k \mid k-1) + K(k)[z(k) - C(k)\hat{x}(k \mid k-1)], \quad (27)$$

$$P(k \mid k) = P(k \mid k-1) - K(k)S(k)K^T(k), \quad (28)$$

The measurement vector $\hat{y}_k$, is given by:

$$\hat{y}_k = [\psi_b\ \ \theta_b\ \ \phi_b]_k^T. \quad (29)$$

The measurement matrix C, is defined as:

$$C = [I_{3\times3}\ \ 0_{3\times3}] \quad (30)$$

where the gain matrix K is given by

$$K(k) = P(k \mid k-1)C(k)S^{-1}(k), \quad (31)$$

where

$$S(k) = R(k) + C'(k)P(k \mid k-1)C(k), \quad (32)$$

is the innovation covariance. The difference between the observation $z(k)$ and the predicted observation $C(k)\hat{x}(k \mid k-1)$ is termed the *innovation* or residual $v(k)$:

$$v(k) = z(k) - C(k)\hat{x}(k \mid k-1). \quad (33)$$

The innovation is an important measure of the deviation between the filter estimates and the observation sequence [14,15]. Indeed, because the true states are not usually available for comparison with the estimated states, the innovation is often the only measure of how well the estimator is performing. The innovation is particularly important in data association.

4. ANALYSIS OF THE RESULTS

4.1 Predicting Behavior of the System

It is important to predict performance of the system, if its characteristic values are growing without bound, then the theoretical performance of the Kalman Filter is said to be diverging. This can happen whenever the system state is unobservable and unstable. The observability, controllability and stability of the dynamics system can be modeled mathematically and there are in-built Matlab m-file functions for their implementations.

One can, for example, use eigenvalues and eigenvector decomposition of solutions to test their characteristic roots (they should be positive) and condition numbers.

4.1.1 Simulation result 1

$$\text{Rank\_Contr} = 6 \text{ and } \text{Rank\_Obsv} = 6\,.$$

Form these results the filter is uniformly asymptotically stable if the system model is stochastically controllable and observable.

4.1.2 Simulation result 2

Matlab/Simulink: Using Matlab/Simulink to obtain the solution for the design of the linear quadratic regulator controller. Where, K is the controller gain, S is the associated solution to the

Algebraic Riccati Equation of the controller design (Innovation Covariance matrix), E the open-loop eigenvalues of the plant dynamics and E_v , the open-loop eigenvector root of the plant dynamics. Note, that for LQR design the pair (A, H) must be *controllable*.

$$K = \begin{bmatrix} -0.0500 & -0.0000 & 0.9846 & -0.0905 & 0.0000 & 0.6199 \\ -0.0000 & -0.0500 & 0.0000 & -0.0000 & 0.2354 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \end{bmatrix}$$

optimal gain matrix K, called Kalman gain

$$S = \begin{bmatrix} 0.0052 & -0.0000 & -0.0310 & 0.0049 & -0.0000 & -0.0101 \\ -0.0000 & 0.0211 & 0.0000 & -0.0000 & 0.0714 & -0.0000 \\ -0.0310 & 0.0000 & 0.4311 & -0.0483 & 0.0000 & 0.191 \\ 0.0049 & -0.0000 & -0.0483 & 0.0064 & -0.0000 & -0.0179 \\ -0.0000 & 0.0714 & 0.0000 & -0.0000 & 0.3363 & 0.0000 \\ -0.0101 & -0.0000 & 0.1922 & -0.0179 & 0.0000 & 0.1193 \end{bmatrix}$$

Innovation Covariance matrix

$$E = \begin{bmatrix} -0.5008 + 1.1789i \\ -0.5008 - 1.1789i \\ -1.1754 + 0.4836i \\ -1.1754 - 0.4836i \\ -0.1474 + 0.1152i \\ -0.1474 + 0.1152i \end{bmatrix} \text{ Eigen values E.}$$

<u>4.1.3 Simulation result 3</u>

Returns the observer gain matrix L such that the Kalman filter produces an optimal state estimate $\hat{x}(k)$ of x using the sensor measurements $\hat{y}(k)$. Also returned solution of $\dot{P}(k \mid k)$ as associated updated state covariance of the Riccati equation.

$$L = \begin{bmatrix} 0.4123 & -0.0000 \\ -0.0000 & 0.0018 \\ -0.0018 & 0.0000 \\ 0.0850 & -0.0000 \\ 0.0000 & 0.0000 \\ -0.0000 & 0.0000 \end{bmatrix} \text{ Observer Gain Matrix L}$$

$$P = \begin{bmatrix} 0.0412 & -0.0000 & -0.0002 & 0.0008 & 0.0000 & -0.0000 \\ -0.0000 & 0.0002 & 0.0000 & -0.0000 & 0.0000 & 0.0000 \\ -0.0002 & 0.0000 & 0.0000 & -0.0001 & -0.0000 & 0.0000 \\ 0.0085 & -0.0000 & -0.0001 & 0.0028 & -0.0000 & -0.0000 \\ 0.0000 & 0.0000 & -0.0000 & -0.0000 & 0.0000 & 0.0000 \\ -0.0000 & 0.0000 & 0.0000 & -0.0000 & 0.0000 & 0.0000 \end{bmatrix}$$

Updated State Covariance

$$E = \begin{bmatrix} -0.1350 + 0.2163i \\ -0.1350 - 0.2163i \\ -0.2720 \\ -0.0003 \\ -0.0018 \\ -0.1300 \end{bmatrix} \text{ Estimator Poles}$$

5. PERFORMANCE EVALUATION DECENTRALIZED AND DISTRIBUTED LINEAR INFORMATION FILTER

Performance evaluation of a data fusion architecture does not pertain only on algorithms accuracy in localization but several other aspects must be considered as the data communication between sensors and fusion nodes, the computational complexity and the memory used. Then, in order to define a procedure to assess the performances of this kind of systems a general model of multisensor and of fusion nodes has been studied that takes into account the elements involved in a multisensor tracking process.

5.1 Performance Evaluation of Tracking Control System Algorithm

By applying the distributed and decentralized fusion algorithm in Chapter three and verified the result. The innovation $\hat{x}_0(t \mid t)$ i.e. the difference between the observation and the predicted observation and corresponding variance $P_0(t \mid t)$. The result compare the deviation between the filter estimates of different fusion algorithm and the observation sequence. We select appropriate coordinate to show their differences clearly. We denote position and velocity as $s(t)$ and $\dot{s}(t)$ respectively and decentralized and centralized system denoted as $s_0(t \mid t)$ and $s_c(t \mid t)$ respectively.

Fig. 3 a and b: The tracking performance comparison of the decentralized and centralized Tracking Control System:

(a) Predefined target position $s(t)$ and centralized estimated target position $\hat{s}_c(t \mid t)$; decentralized estimated target position $\hat{s}_0(t \mid t)$, (b) Predefined target velocity $\dot{s}(t)$ and decentralized estimated target velocity $\hat{\dot{s}}_0(t \mid t)$, centralized estimated target velocity $\hat{\dot{s}}_c(t \mid t)$.

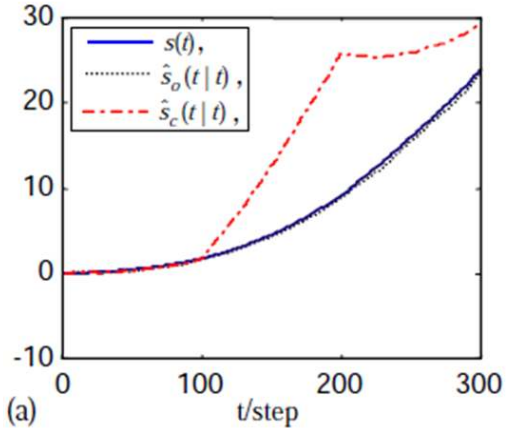

Fig. 3 a. from the simulation results of position tracking system, we see that the centralized estimated target position tracking system diverges at 100 seconds $(100 \le t < 200)$. But the decentralized estimated target position tracking can still track the target throughout the period of the flight

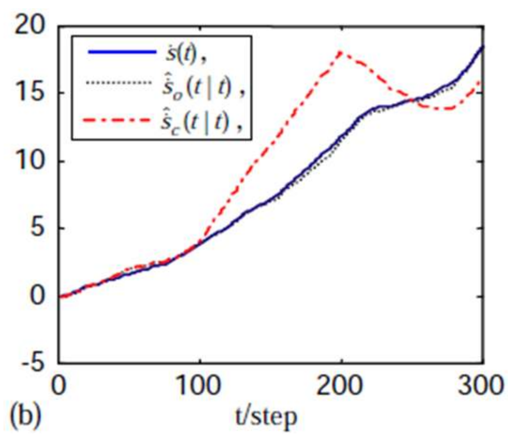

Fig. 3b. from the simulation results of velocity tracking system, we see that the centralized estimated target velocity tracking system diverges at 100 seconds $(100 \le t < 200)$ and at the 200 seconds $(t \ge 200)$ it started converging back. The decentralized estimated target velocity tracking system track the target throughout the flight period

It shows that the decentralized and distributed multi-target tracking control has better fault tolerance and robustness properties.

6. DISCUSSION AND CONCLUSION

The objective of the analysis is to evaluate the feasibility of an estimation system design for meeting some pre-specified acceptable level of uncertainty in the estimates that will be obtained. This can be achieved by adopting algorithm introduced in this paper, which is based on Decentralized, Distributed, Local interaction and Heterarchical Architecture requiring no central processing facility to perform data fusion. Each node communicates local information to each other node so each one arrives at the common global consensus. It was found that the information that each node needs to broadcast to each other node is simple and that the data assimilation equations that each node must perform are no more complex than the equations for the Kalman filter.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Mutambara AGO. Decentralized estimation and control for multisensor systems, Boca Raton, FL, USA: CRC Press, Inc.; 1998.
2. Kalman RE. A new approach to linear filtering and prediction problems. Transaction of the ASME-Journal of Basic Engineering, 1960;3:35-45.
3. Rao BS, Durrant-Whyte HF. Fully decentralized algorithm for multisensor Kalman filtering. IEE Proceedings-D. 1991; 138;5.
4. Hugh Durrant-Whyte, Mike Stevens. Data fusion in decentralized sensing networks. Australian Centre for Field Robotics The University of Sydney NSW; 2006. Available:http://www.acfr.usyd.edu.au
5. Hekler A, Kiefel M, Hanebeck UD. Nonlinear bayesian estimation with compactly supported wavelets. In Proceedings of the IEEE Conference on Decision and Control, Atlanta, Georgia. 2010;12.
6. Rosencrantz M, Gordon G, Thrun S. Decentralized sensor fusion with distributed particle filters. Proceedings of the 19 Conference on Uncertainty in Artificial Intelligence. 2003;(7):7-10.
7. Benjamin Noack, Daniel Lyons, Matthias Nagel, Uwe HD. Nonlinear information filtering for distributed multisensor data

fusion. American Control Conference on O'Farrell Street, San Francisco, CA, USA, 2011;29.
8. Hall D, Llinas J. Handbook of multisensor data fusion. CRC, Boca Raton; 2001.
9. Gasparri A, Pascucci F, Ulivi G. A distributed extended information filter for self-localization in sensor networks. In Proceedings of IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC. 2008;8:1–5.
10. Philip Ferguson, Jonathan How. Decentralized estimation algorithms for formation flying spacecraft. Massachusetts Institute of Technology, AIAA GNC. 2012; 8.
11. Nadig D, Iyengar S, Jayasimha D. New architecture for distributed sensor integration. IEEE Southeastcon Proc; 1993.
12. Bruno Siciliano, Oussama Khatib. (Eds.) Handbook of Robotics Springer-Verlag Berlin Heidelberg Springer; 2008.
13. Prasad L, Iyengar S, Kashyap RL, Madan RN. Functional characterization of fault tolerant integration in distributed sensor networks. IEEE Trans. Syst. Man Cybern. 1991;25:1082–1087.
14. Waltz EL, Llinas J. Sensor Fusion Artec House, Norwood; 1991.
15. Henderson TC, Allen PK, Mitiche A, Durrant-Whyte HF, Snyder W. Workshop on Multisensor Integration in Manufacturing Automation. Tech. Rep. UUCS-87406. University of Utah, Snowbird: Dept. of Computer Science; 1987.

Peer-review history:
The peer review history for this paper can be accessed here:
http://sciencedomain.org/review-history/18642