Development of Multi-Functional Control Architecture for Multisensor Surveillance Systems

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

ABSTRACT

Multisensor approach is often utilized in modern surveillance systems because of its abilities to provide complementary and overlapping coverage on targets. In order to generate target tracks and estimates, the sensor data need to be fused. While a centralized and hierarchical processing approach is theoretically optimal, there are significant advantages in decentralizing the fusion operations over multiple processing nodes. This paper discusses decentralized and heterarchical control architectures, whereby each node processes the data from its own set of sensors and communicates with other nodes to improve on fusions and estimates. A decentralized multisensor data fusion and estimation algorithm with nonlinear information filter were developed for each sensor node for effective information gathering, filtering and estimation along the desired trajectory. The dynamic systems were mathematically modelled and simulated. The simulation results show that the developed architecture satisfies stochastic stability criteria, manifests excellent tracking and filtering properties than the convectional architecture.

1. INTRODUCTION

In most cases a single sensor cannot provide all the information required about the system under consideration (restricted), the limitation is with the communication and available distributed and decentralized computing information [1]. Access to raw or possibly minimally preprocessed data from other sensors opens the opportunity for more effective exploitation in process system. In addition, capability and reliability of a single sensor is restricted, the possible failure of the sensor will result in complete system failure; for example, radar provides accurate range but poor angle data while infrared provides accurate angle but poor range data. Even for similar sensor types, the different viewing angles from multiple physically distributed sensors can be exploited to provide better location data. Furthermore, multiple sensors provide more robust performance due to the inherent redundancy.
The availability of wire/wireless networks and a wide range of sensor devices, with more computational capabilities allow the implementation of more sophisticated tracking and surveillance systems. These systems consist of networks of sensors e.g. video cameras, microphones, detectors, radar, etc., which are able to work in omnidirectional and directional (orient-table in three dimensions) modes, and can be mounted on mobile platforms (motorized artifacts that allow movement around facilities under tracking and surveillance) or fixed ones (anchored at a particular point of the facilities); all these parts form their control systems.

In the literature, various approaches have been proposed for architecture modeling. In [2], a twofold architecture description is defined on physical and logical level that we take as an inspiration: the physical architecture described as a graph constituted of nodes and links, while logical architecture is defined outlining a task decomposition (i.e. tracking) into subtasks (i.e. data acquisition, filtering etc.). In [3], a hierarchical architecture is proposed for video surveillance and tracking; the camera network sends the data gathered to the upper level nodes in which tracking data are fused in a centralized manner.

The vast majority of the works found in literature defines metrics related to specific parts of the architecture missing a more generic analysis. However, during the design of complex data fusion based systems, it is important to consider the influence of the architectural aspects of the system on the overall performances. In [4, 5], an evaluation benchmark is presented to compare different visual surveillance algorithms developed for PETS (Performance Evaluation of Tracking and Surveillance) workshops. PETS metrics are specifically developed to provide an automatic mechanism to quantitatively compare similar purpose algorithms operating on the same data.

The idea of the measurement conversion methods is to transform nonlinear measurements into a linear combination of the Cartesian coordinates, estimate the bias and covariance of the converted measurement noise, and then use the standard Kalman filter. This technique has been shown to outperform the EKF in general. One shortcoming of the EKF and the measurement conversion methods is the lack of a rigorous proof on the boundedness of the estimation errors. It is also well-known that the estimate produced by the EKF may diverge from the true state in practice [6].

The contributions of this paper include the development of a fully decentralized and heterarchical control architecture and robust nonlinear data fusion and estimation techniques for surveillance tracking. This filter is designed using ideas and methods from modern robust state estimation theory. The robust algorithm designed here permits efficient data fusion through a simple summation fusion structure. We discuss the process of multi-sensor information fusion, rather than multi-sensor data fusion. Data fusion is the process of integrating actual data measurements extracted from different sensors and combining them into one representation. Information fusion is the process of using information derived from multiple sensors and combining them at the information level. On this part nonlinear information filter (NIF) were also developed. The algorithm complexity is scalable with the number of sensors.

2. Modeling of Multisensor Architecture

Modeling multisensor systems is to define sensori-computational systems associated with each sensor to allow design, comparison, transformation, and reduction of any sensory system. Each sensor type has different characteristics and functional descriptions. Consequently, some approaches aim to develop general methods of modeling sensor systems in a manner that is independent of the physical sensors used. In turn, this enables the performance and robustness of multisensor systems to be studied in a general way [1].

There have been many attempts to provide the general model, along with its mathematical basis and description. Some of these modeling techniques concern error analysis and fault tolerance of multisensor systems [7,8]. Other techniques are model based, and require a priori knowledge of the sensed object and its environment.

There are also technical issues on system architecture and algorithms that need to be addressed before high performance systems can be developed. Some important technical issues include the following items.

- *Architecture*: how the nodes should share the fusion responsibility, e.g., which sources or sensors should report to each node, and the targets that each node should be responsible for.
- *Communication*: how the nodes should communicate, e.g., connectivity and bandwidth of the communication network, information push or pull, and communicating raw data versus processing results.
- *Algorithms*: how the nodes should fuse data for high performance results and select their communication actions (who, when, what, and how).

Modelling of multisensor network architecture framework were illustrated in figure 1: [9] architecture, algorithms, and implementation. All three aspects must be tailored to fulfill the requirements of the system.

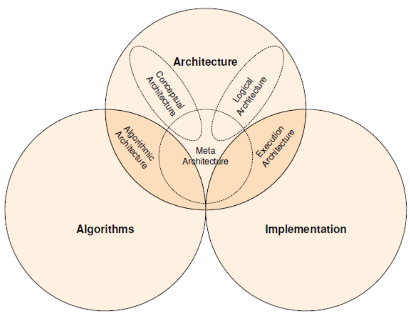


Fig. 1: Three aspects of multisensor network architecture framework. The importance of Meta architecture is indicated by its central position in the diagram.

2.1. Meta-architecture

This section describes and justifies the architectural choices and shows how these choices lead to the fulfillment of the multi-functional and non-functional requirements.

Meta-architecture: a set of high level decisions that strongly influence the structure of the system. The style of this multisensor network architecture is established by listing its distinguishing features: decentralization, distributed, modularity, and the use of strictly-local interactions. The rationale for these choices is explained in terms of their contribution to the fulfillment of the multisensor network requirements, namely large team size, active heterogeneous platforms, and long mission duration. The influence of the meta-architecture is far-reaching: the communication and structural patterns defined here are repeatedly applied across the architecture ensuring a consistent approach.

The distinguish features of this architecture are the reasons for considering it for the surveillance system. Decentralized means that no component is central to operation of the system, and the communication is peer to peer. Also, there are no central facilities or services (e.g., for communication, name and service lookup or timing). These features lead to a system that is scalable, fault tolerant, and reconfigurable. Local interactions mean that the number of communication links does not change with the network size. Moreover, the number of messages should also remain constant. This makes the system scalable as well as reconfigurable. Modularity leads to interoperability derived from interface protocols, re-configurability, and fault tolerance: failure may be confined to individual modules.

This type of architecture is the real-time control system (RCS). RCS is presented as a cognitive architecture for intelligent control, uses multisensor fusion to achieve complex control. The architecture focuses on task decomposition as the fundamental organizing principle. It defines a set of nodes, each comprised of a sensor processor, a world model and a behavior generation component. Nodes communicate with other nodes, generally in a heterarchical manner, i.e. heterogeneous communication channel, although across-layer connections are allowed. The system supports a wide variety of algorithmic architectures, from reactive behavior to semantic networks.

Moreover, it maintains signals, images, and maps and allows loose coupling. The architecture allows dynamic reconfiguration. It also maintains the static module connectivity structure of the specification.

3. Decentralized and Heterarchical Architectures and Approach

Multisensor data fusion problem is present in many different applications including surveillance, robotics, manufacturing automation, etc. In each application area, there are specific problem, even though the general goal is to utilize the available data to improve the understanding of some state. In this paper, we focus on the problem of control architecture and its effective use in surveillance, which is applicable to defense, air control, robotics etc., to determine the location, velocity, and other attributes of multiple moving objects from sensor data.

Many existing fusion systems have a centralized architecture with all data processed by a single fusion node. The availability of distributed computing and the need to deal with bigger problems, however, will imply decentralized fusion systems with multiple fusion nodes processing data from their own sensors and communicating with other nodes to improve upon the local results. The presence of multiple data sources and fusion nodes provides many choices in the architecture, i.e., how the sensors or data sources report to each fusion node and the connectivity among the nodes.

Figure 2 shows decentralized, distributed and heterarchical control architectures. In a fully decentralized architecture, there is no pre-determined superior/subordinate relationship, each node can communicate with any other node subject to connectivity constraints, and the communication can be asynchronous.

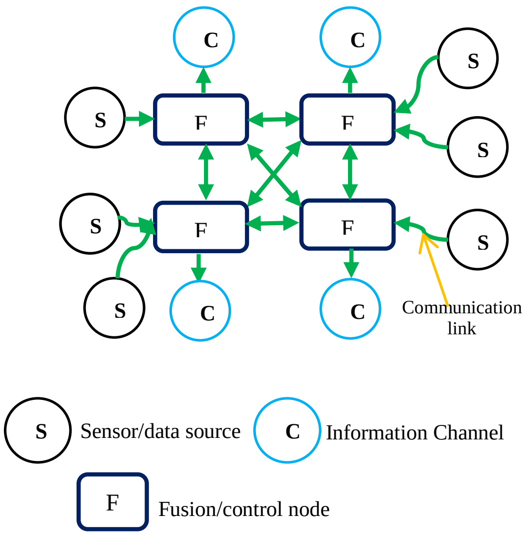


Fig. 2. Decentralized and heterarchical architectures.

3.1. Decentralized and Heterarchical Control Algorithm with nonlinear information filter.

Most of these sensor fusion algorithms discussed in literature are centralized and designed for linear systems, whereas most practical problems are nonlinear. The objective of this section is to outline a new estimation algorithm for nonlinear systems. Its novelty stems from the ease with which it can be decentralized, thus, making fully decentralized data fusion and control for nonlinear systems feasible.

3.1.1. Mathematical Models for Decentralized and Heterarchical Control Algorithm

The primary objective of surveillance tracking is to estimate the state trajectories of a moving object. Although a target is almost never really a point in the space and the information about its orientation is valuable for tracking. A target dynamic model or motion model describes the evolution of the target state with respect to time. Almost all surveillance tracking methods are model-based. They assume that the target motion and its observations can be represented by some known mathematical models accurately. The most commonly used such models are those known as state-space models, in the following form

Consider a nonlinear system

$$\dot{X} = f(X, U, W), \quad X \in R^m, u \in R^m \tag{1}$$

$$Y = CX + V, \quad Y \in R^p, \tag{2}$$

where w and v are Gaussian white noise processes with covariance matrices R_w and R_v. A nonlinear observer for the system can be constructed by using the process

$$\dot{\hat{X}} = f(\hat{X}, U, 0) + L(Y - C\hat{X}). \tag{3}$$

If we define the error as $E = X - \hat{X},$, the error dynamics are given by

$$\dot{E} = f(X, U, W) - f(\hat{X}, U, 0) - LC(X - \hat{X}), \tag{4}$$

$$= G(E, \hat{X}, U, W) - LCe \tag{5}$$

where

$$G(E, \hat{X}, U, W) = f(E, + \hat{X}, U, W) - f(\hat{X}, U, 0) \tag{6}$$

we can now linearize around current estimate $\hat{X}$:

$$\hat{E} = \frac{\partial G}{\partial E}E + \underbrace{G(0, \hat{X}, U, 0)}_{=0} + \frac{\partial G}{\partial W}W - \underbrace{LCe}_{\text{observer gain}} + \text{h.o.t}, \tag{7}$$

$$\approx \tilde{F}E + \tilde{G}W - LCE. \tag{8}$$

Depend on current estimate $\hat{X}$. We can now design an observer for the linearized system around the current estimate:

$$\dot{\hat{X}} = f(\hat{X}, U, 0) + L(Y - C\hat{X}), \tag{9}$$

$$L = PC^T R_v^{-1}, \tag{10}$$

$$\dot{P} = (\tilde{F} - LC)P + P(\tilde{F} - LC)^T + \tilde{G}R_w\tilde{G}^T + LR_vL^T, \tag{11}$$

$$P(t_0) = E\{X(t_0)X^T(t_0)\} \tag{12}$$

This is called the (Schmidt) extended Kalman filter (EKF).

The intuition in the Kalman filter is that we replace the prediction portion of the filter with the nonlinear modeling while using the instantaneous linearization to compute the observer gain. In this paper it is assumed that each local control node has a state space model identical to an equivalent centralized model.

- **State transition**

$$x(k) = F(k)x(k-1) + B(k)u_i(k) + G(k)w(k), \quad i = 1, \ldots, N \tag{13}$$

Prediction

$$\hat{y}_j(k \mid k-1) = L_j(k \mid k-1)\hat{y}_j(k-1 \mid k-1) \tag{14}$$

$$\hat{Y}_j(k \mid k-1) = [F(k)Y_j^{-1}(k-1 \mid k-1)F^T(k) + Q(k)]^{-1} \tag{15}$$

where

$$L_j(k \mid k-1) = Y_j(k \mid k-1)F(k)Y_j^{-1}(k-1 \mid k-1) \tag{16}$$

Observation

$$z_j(k) = H_j(k)x(k) + v_j(k), \quad j = 1, \ldots, N. \tag{17}$$

Estimation

$$\hat{y}_j(k \mid k) = \hat{y}_j(k \mid k-1) + i_j(k) \tag{18}$$

$$\hat{Y}_j(k \mid k) = Y_j(k \mid k-1) + I_j(k) \tag{19}$$

- **Communication**

These partial information state estimates are communicated to neighboring nodes where they are assimilated to produce the global information estimate:

$$\hat{y}_j(k|k) = \hat{y}_j(k|k-1) + \sum_{j=1}^{N}[\hat{y}_j(k|k) - \hat{y}_i(k|k-1)] \tag{20}$$

$$\hat{Y}_j(k \mid k) = Y_i(k \mid k-1) + \sum_{j=1}^{N}[\hat{Y}_j(k \mid k) - \hat{Y}_i(k \mid k-1)] \tag{21}$$

- **Control generation**

The global state estimate and control vector are then calculated

$$\hat{x}_i(k \mid k) = Y_i(k \mid k)^{-1}\hat{y}_i(k \mid k) \tag{22}$$

$$u_i(k) = -G_i(k)[\hat{x}_i(k \mid k) - \hat{x}_i(k \mid k)^{-1}] \tag{23}$$

- **Control law**

$$G_i(k) = [U(k) + B(k)^T K_i(k)B(k)]^{-1}[B(k)^T K_i(k)F(k)] \tag{24}$$

The Kalman gain $K_i(k)$:

$$K_i(k) = X(k) + [F(k)^T K_i(k-1)][F(k) - B(k)G_i(k)] \tag{25}$$

- **Conditions**

If each node begins with a common initial information state estimate; $\hat{y}_i(0 \mid 0) = 0 \text{ and } Y_j(0 \mid 0) = 0$ and also the network is fully connected, then the global

$\{\hat{y}_i(k \mid k)$ and $\hat{x}_j(k \mid k)\}$ and the control vector $\{u_i(k)\}$ obtained by each node will be identical.

$$i_j(k) = \hat{y}_j(k \mid k) - \hat{y}_j(k \mid k-1) \quad (26)$$

$$I_j(k) = \hat{Y}_j(k \mid k) - Y_j(k \mid k-1) \quad (27)$$

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

$$\hat{y}(i \mid j) \triangleq P^{-1}(i \mid j)\hat{x}(i \mid j), \quad (28)$$

$$Y(i \mid j) \triangleq P^{-1}(i \mid j) \quad (29)$$

with

$$E[w(i)w^T(j)] = \delta ijR(i), \quad (30)$$

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

$$i(k) \triangleq H^T(k)R^{-1}(k)z(k) \quad (31)$$

$$I(k) \triangleq H^T(k)R^{-1}(k)H(k) \quad (32)$$

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

3.1.2. The Nonlinear Information Filter.

The decentralized multisensor linear Information filter can now be extended to a linearized estimation algorithm for nonlinear systems by using principles from both the derivations of the information filter and the EKF [10]. This generates a filter that predicts and estimates information about nonlinear state parameters given nonlinear observations and nonlinear system dynamics. The new filter will be termed the Nonlinear Information filter (ENIF). This is because in the nonlinear case, the function operator h cannot be separated from $x(k)$ in the nonlinear observation equation, and yet the derivation of the Information filter depends on this separation. The NIF algorithm is summarized as follows;

Prediction

$$\hat{y}_j(k \mid k) = Y_j(k \mid k-1)f(k, \hat{x}(k-1 \mid k-1), u(k-1)) \quad (33)$$

$$Y_j(k \mid k-1) = [\nabla f_x(k)Y_j^{-1}(k-1 \mid k-1) \nabla f_x^T(k) + Q(k)]^{-1}. \quad (34)$$

Estimation

$$\hat{y}_j(k \mid k) = \hat{y}_j(k \mid k-1) + i_j(k) \quad (35)$$

$$\hat{Y}_j(k \mid k) = Y_j(k \mid k-1) + I_j(k) \quad (36)$$

Information State

The information state contribution and its associated information matrix are given, respectively, as

$$I_j(k) = \tilde{N}h_x^T(k)R^{-1}(k)\tilde{N}h_x(k) \quad (37)$$

$$i_j(k) = \tilde{N}h_x^T(k)R^{-1}(k)[v_j(k) + \tilde{N}h_x(k)\hat{x}_i(k \mid k-1)], \quad (38)$$

where $v_j(k)$ is the innovation covariance given by

$$v_j(k) = z_j(k) - h(\hat{x}_i(k \mid k-1)) \quad (39)$$

4. Description of the Model

Consider a network of N stationary sensors (e.g. radar, video camera, and leaser sensors) positioned in 3D and communicating over a complete graph. That is, a network of sensors with a heterogeneous communication topology $H_c = \{V, E\}$ where $V = \{1, \ldots, N\}$ represents the graph vertices, i.e. the sensors, and $E \subseteq V \times V$ is the set of inter-sensor links. Each sensor i communicates with a set of neighbors $N_i \subseteq V$ and $j \in N_i \Leftrightarrow i \in N_j \ \forall i, j \in V$. For notational simplicity it is always assumed that $i \in N_i$. If $N_i = V$ then the communication graph is complete. Considering the tracking surveillance problem in 3D, then the position of the i^{th} sensor is given by $s_i = [s_{i1}\ s_{i2}\ s_{i3}]^T$ where s_{i1}, s_{i2} and s_{i3} denote the traditionally denoted x, y and z positions of i^{th} sensor. Each sensor i knows its own position s_i in the global coordinate system.

5. The Dynamic Models

Consider the problem of estimating the position and velocity of an Autonomous Mobile Surveillance Vehicle (AMSV); comprising of array of sensor nodes and their sensors: Global Positioning System (GPS) and an inertial measurement unit (IMU) for accurate and effective Information Gathering and Processing (IGP). We assume that the vehicle is disturbance free, but that we have noisy measurements from the GPS receiver and IMU and an initial condition error. An IMU can be used to measure angular rates and linear acceleration. Assuming that we use a Digital Rate Gyroscope (ADIS16250) (DRG), to measure the angular rate $\dot{\theta}$.

A point moving in three-dimensional physical world can be described by its three-dimensional position and velocity vectors. For instance, $x = [x, \dot{x}, y, \dot{y}, z, \dot{z}]'$ can be used as a state vector of such a point in the Cartesian coordinate system, where (x, y, z) the position are coordinates along x, y and z axes, respectively, and $[\dot{x}, \dot{y}, \dot{z}]'$ are the velocity vector. When a target is treated as a point object, the non-maneuvering motion is thus described by the vector-valued equation $\dot{x}(t) = 0$, where $x = [\dot{x}, \dot{y}, \dot{z}]'$. The ideal equation is usually modified as $\dot{x}(t) = w(t) \approx 0$, where $w(t)$ is a white noise process, with a "small" effect on x, which accounts for unpredictable modeling errors due to turbulence, etc. The corresponding state-space model is given by, with state $x=[x,\dot{x},y,\dot{y},z,\dot{z}]'$. Use the datasheet to determine a model for the noise process and use the developed architecture to fuses and estimate the GPS and IMU information to determine the position and the velocity of the vehicle.

5.1. Problem Formulation

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

$$\delta\vec{x}(t)=\left[\delta\psi \quad \delta\theta \quad \delta\phi \quad \delta b_p \quad \delta b_q \quad \delta b_r\right]^T. \quad (40)$$

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

- **Time Update Equations**

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

$$F(t)=\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -10 & -0.13 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.13 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (41)$$

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

$$G(t)=\left[0\ 0\ 0;0\ 0\ 0;0\ 0\ 0;0.7\ 0\ 0;0\ 0.7\ 0;\ 5.3\ 0\ 0\right]. \quad (42)$$

The process noise vector $\vec{w}$ is given by

$$\vec{w}=\left[n_p \quad n_q \quad n_r \quad w_p \quad w_q \quad w_r\right]^T. \quad (43)$$

The process noise covariance matrix, R_w, and its associated power spectral density matrix, Q_w. The matrix R_w is defined as:

$$R_w=\varepsilon\{\vec{w}\vec{w}^T\}. \quad (44)$$

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

$$Q_w=\begin{bmatrix} R_n & 0_{3x3} \\ 0_{3x3} & R_b \end{bmatrix} \quad (45)$$

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

$$R_n=\begin{bmatrix} \sigma_p^2 & 0 & 0 \\ 0 & \sigma_q^2 & 0 \\ 0 & 0 & \sigma_r^2 \end{bmatrix} \quad (46)$$

Numerical values for σ_p, σ_q *and* σ_r depend on the type of IMU being used and can be found in data sheet. The variables σ_p^2, σ_q^2 *and* σ_r^2 are the variances of the wide-band noise on the three orthogonal IMU. Similarly, numerical values for σ_{w_p}, σ_{w_q} *and* σ_{w_r}, are also found in the data sheet.

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

$$R_b=\frac{2\sigma_\omega^2}{\tau_\omega}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (47)$$

Equation (35) now becomes,

$$Q_w=\mathrm{dag}([\sigma_p^2;\sigma_q^2;\ \sigma_r^2;\frac{2\sigma_p^2}{\tau_\omega};\frac{2\sigma_q^2}{\tau_\omega};\frac{2\sigma_r^2}{\tau_\omega}]). \quad (48)$$

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, $P_i(k)$ at time step k, then the covariance at time step $k+1$ is given by:

$$P_i(k\mid k)=F(k)P_i(k\mid k)F^T(k)+GR_wG^T. \quad (49)$$

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 is

$$P_i(k\mid k)=P_i(k\mid k-1)+K(k)S(k)K^T(k). \quad (50)$$

- **Measurement (update or correction) equations:**

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

$$\hat{x}(k\mid k)=\hat{x}(k\mid k-1)+K(k)[z_i(k)-C_i(k)\hat{x}(k\mid k-1)], \quad (51)$$

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

$$\hat{y}_k=[\psi_b\ \theta_b\ j_b]_k^T. \quad (52)$$

The measurement matrix C_i, is defined as:

$$C_i=\left[I_{3\times3}\ 0_{3\times3}\right] \quad (53)$$

where the gain matrix K is given by:

$$K(k)=P_i(k\mid k-1)C_i(k)S^{-1}(k), \quad (54)$$

and

$$S(k)=R(k)+C_i^{'}(k)P_i(k\mid k-1)C_i(k), \quad (55)$$

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

5.2. Analysis of the Results

It is important to predict performance of the developed architecture and compare with the conventional centralized system. This can happen whenever the system state is unobservable, controllable and unstable. The observability, controllability and stability of the dynamics system can be modeled mathematically and simulated. The consistence test can also be carried out on the developed architecture to authenticate the system performances.

- **Simulation result 1:**

The mathematical model resulted in a 6x6 state matrix and 6x3 measurement vector. The rank of the Gramian matrix resulting from the observability and controllability tests was six (6); which is equal to the number of columns in the state matrix and rows in the measurement vector. This indicates that the architecture is stochastically observable and controllable.

- **Simulation result 2:**

Matlab/Simulink: Using Matlab/Simulink to obtain the solution for the design of the linear quadratic regulator controller. Where S is the associated solution to the Algebraic Riccati Equation of the controller design (Innovation Covariance matrix). The innovation is an important measure of the deviation between the filter estimates and the observation sequence. 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.

$$S = \begin{bmatrix} 0.0152 & -0.0120 & -0.0310 & 0.1149 & -0.2130 & -0.0101 \\ -0.1230 & 0.1211 & 0.1010 & -0.0000 & 0.0714 & 0.6130 \\ -0.0310 & 0.2100 & 0.5311 & -0.0483 & 0.0000 & 0.1301 \\ 0.1049 & -0.0000 & 0.4830 & 0.1423 & -0.0000 & -0.2179 \\ -0.0200 & 0.0714 & 0.3400 & 0.0000 & 0.6734 & -0.0230 \\ 0.1101 & 0.0000 & 0.1422 & -0.0009 & 0.1000 & 0.2156 \end{bmatrix}$$

Innovation Covariance matrix of the convectional centralized architecture

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

Innovation Covariance matrix of the developed architecture

The diagonal elements of the innovation covariance matrices for the conventional and the developed architecture ranged from 0.1123 to 0.6734 and 0.0052 to 0.4311 respectively indicating that the developed architecture were more asymptotically stable.

5.3. Performance Evaluation of Developed Algorithm for Surveillance Tracking Systems

Figures 3 and 4 show the surveillance tracking performance comparison of the convectional centralized architecture and the developed architecture with the predefined system trajectory tracking. The position and velocity changes in the trajectory of the conventional architecture were 0.00; 16.22; 10.22m, and 0.04; 7.14; 5.89m/s; while that of the developed architecture were 0.00; 0.04; 0.45m, and 0.00; 0.05; 0.41m/s for time intervals: 0 to 100; 101 to 200 and 201 to 300s respectively. This indicates that there is significant difference between the tracking performance of the developed and the conventional architecture with a P-value equals to 0.0102.

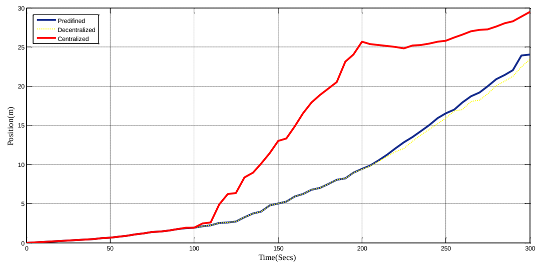
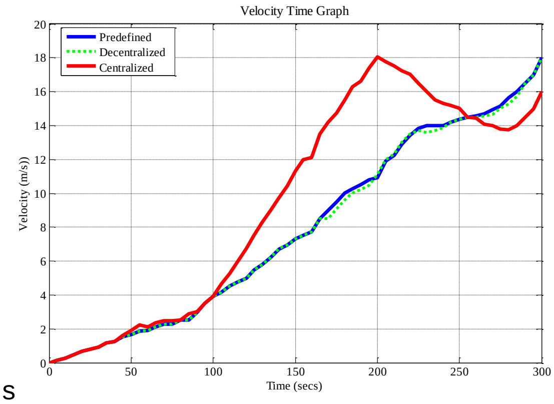


Fig. 3: Position/Time Graph of surveillance Tracking System.



s

Fig. 4: Velocity/Time Graph of surveillance Tracking System.

5.4. Performance Evaluation Decentralized and Distributed Nonlinear 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. A decentralized and heterarchical architecture with improved tracking performance has been developed. A distributed and decentralized multisensor data estimation and fusion algorithm with nonlinear information filter was developed and implemented, for fusing the information from these various sensors and embedded in the developed system. This algorithm may be run simultaneously on each node of a multisensor network to give a parallel, highly survivable multi-tracking system. The information that needs to be communicated between nodes is simple and the equations for data fusion are no more complex than for local estimate update. The fully distributed and decentralized nature ensures that it is ideal for implementation on a parallel processing array such as an autonomous intelligent multisensor network [1]. The system could have application in a real time high quality data gathering in navigation system.

6. Conclusion

In this paper a model of a system architecture for multisensor surveillance tracking has been developed. The aim of this model is to represent basic tasks performed by such a system in order to address the problem of performance evaluation. The extension of convectional centralized control architecture has been successfully extended to a decentralized and heterarchical control architecture. This successful extension is based on the formal verification of the design, and is hence consistent with information space ideas employed by the decentralized observer. The developed architecture involved heterogeneous connection of nodes and there sensors. In fact, the distributed and decentralized data fusion and nonlinear information filters structure tremendously improves the accuracy of the navigation systems. The algorithm produced reliable results even when presented with potentially very noisy data. Finally the control system will continued to function properly even when some of the sensor are isolated from the system when it was running and also shows all the advantages that were predicted.

Principle of this solution shown to be indeed viable by a simulated test implementation. The network satisfies stochastic stability criteria, manifests excellent filtering properties.

Future works will be devoted to applying the model to decentralize fusion architectures and to realize a more accurate model of different topologies of smart sensors.