Local/INS navigation for autonomous vehicle formation

Nuno M.S. Valverde
Institute for Systems and Robotics (ISR)
Instituto Superior Técnico
nvalverde@isr.ist.utl.pt

Abstract - This report presents the study of computation of the distance between vehicles of a formation. This work proposes to study several strategies in order to determine the distance between the constituent vehicles of a formation aided by a GPS system and using Kalman filtering.

The GPS system is studied in detail. It is also presented the errors that affect the GPS system. Based on this location system, is introduced, developed and implemented several strategies to obtain an estimation of the vehicles distance.

The centralized strategy consists in estimating the distance between vehicles using all the existing information. The decentralized strategy estimates the position of each vehicle and then computes the distance between the vehicles. The last strategy presented consists in trying to improve the estimation of the decentralized strategy using Covariance Intersection algorithm.

These strategies are also compared with an iterative algorithm to determine the position.

At the end, is presented the performance of the studied algorithms.

Keywords: GPS, Pseudo-distance, Vehicle Formation, Kalman Filtering, *Covariance Intersection*, *Posterior Cramer Rao*

I. Introduction

The autonomous vehicles have been a great area of study for the last decades. The will of improving capabilities and performance of autonomous vehicles turned this area into one of the most important areas of study.

The autonomous vehicles formation appeared from the idea of having several single autonomous vehicles interacting.

The concept of a vehicle formation has great advantages to a single operating vehicle. The formation can increase efficiency, performance and has new emerging capabilities that single autonomous vehicles does not have. However it is necessary to know where exactly each of the vehicles is and their relative positions.

In this paper it is introduced and compared several strategies for distance estimation for the vehicles of a formation equipped with a GPS system.

II. GPS System

The GPS system main objective is to give the precise 3-dimensional position of a user on earth.

The system is constituted by 24 Navstar GPS satellites, about 20200 km above the earth surface. They are uniformly distributed by 6 plans with an angle of 55º a spaced of 60º of latitude. Each satellite takes 12 hours to complete one turn around the earth. Each point at the earth's surface is seen, at least, by 4 satellites.

A. Basic function of GPS System

The position calculated by the GPS system is determined in reference to the ECEF referential [1].

In order to compute the coordinates of a GPS user, the system uses the distance from each available satellite to the user. This distance is determined by (1), and is named as Ideal Pseudo-Distance.

$$d_i^* = \sqrt{\left(X_i - x\right)^2 + \left(Y_i - y\right)^2 + \left(Z_i - z\right)^2} , \quad (1)$$

where:

- X_i, Y_i, Z_i are the satellite coordinates;
- x, y, z are the user coordinates.

Based on the ideal pseudo-distances introduced by (1), and using the trilateration method, the necessary numbers of satellites that are needed to compute the coordinates are 3 satellites. However, since the pseudo-distances are affected by errors, it is needed to consider at least 4 satellites.

B. Errors affecting GPS

The pseudo-distances are affected by several errors from several origins, which affect the precision and accuracy of the system.

The errors sources can be divided in 3 categories: satellites, GPS receptors and propagation mean.

The errors originated by the satellites are much smaller than the other errors that affect the system. Examples of this type of errors are the real position of the satellite and the satellites clock.

The propagation mean is one of the most severing errors categories. The atmospheric conditions [3] highly affect the GPS signal propagation velocity. The ionosphere and troposphere also affect the system measurements. This type of errors can be easily calculated for a particular geographical area and help to reduce this error effect.

The GPS receiver is the main error source for the GPS system, by adding some error to the measurements. Some examples are the multi-path of the signal and the inaccuracy of the receiver's clock.

The presented errors can be reorganized in 3 new categories:

- Errors induced by the clock;
- Correlated errors (Rc);
- Uncorrelated errors (Ri);

Based on this division, the pseudo-distance can be written as (2).

$$d_i = \sqrt{\left(X_i - x\right)^2 + \left(Y_i - y\right)^2 + \left(Z_i - z\right)^2} + b_u + \eta_c + \eta_{di} \quad (2)$$

where:

- b_u is the error induced by the GPS receiver's clock. This error can be determined.
- η_c is the correlated part of the error that affects the satellite's measurements.
- η_{di} is the uncorrelated part of the error that affects the satellite's measurements.

III. POSITIONAL ALGORITHMS

There are several algorithms [1][2][4][5] to calculate the coordinates of an user using the pseudo-distances (2) given from the available satellites. The iterative algorithm is one example. This algorithm is a basic recursive method to determine the user's coordinates.

It is also possible to determine the coordinates using Kalman filtering. The objective of this approach is to get a better estimation of the vehicles position.

A. Iterative algorithm [1]

Assuming that there are N visible satellites, there will be also N pseudo-distances available. The system to compute is given by (3).

$$\begin{cases} d_1 = \sqrt{\left(X_1 - x\right)^2 + \left(Y_1 - y\right)^2 + \left(Z_1 - z\right)^2} + b_u \\ d_2 = \sqrt{\left(X_2 - x\right)^2 + \left(Y_2 - y\right)^2 + \left(Z_2 - z\right)^2} + b_u \\ d_3 = \sqrt{\left(X_3 - x\right)^2 + \left(Y_3 - y\right)^2 + \left(Z_3 - z\right)^2} + b_u \\ \vdots \\ d_N = \sqrt{\left(X_N - x\right)^2 + \left(Y_N - y\right)^2 + \left(Z_N - z\right)^2} + b_u \end{cases} \quad .(3)$$

Computing the first order Taylor series, the matrix equation (4) is defined.

$$\delta d = \alpha.\delta c \Leftrightarrow$$
$$\begin{bmatrix} \delta d_1 \\ \delta d_2 \\ \delta d_3 \\ \vdots \\ \delta d_N \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_{N1} & \alpha_{N2} & \alpha_{N3} & \alpha_{N4} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta b_u \end{bmatrix}, \quad (4)$$

where:

$$\alpha_{j1} = -\frac{\left(X_j - x\right)}{\sqrt{\left(X_j - x\right)^2 + \left(Y_j - y\right)^2 + \left(Z_j - z\right)^2}}, \quad (5)$$

$$\alpha_{j2} = -\frac{\left(Y_j - y\right)}{\sqrt{\left(X_j - x\right)^2 + \left(Y_j - y\right)^2 + \left(Z_j - z\right)^2}}, \quad (6)$$

$$\alpha_{j3} = -\frac{\left(Z_j - z\right)}{\sqrt{\left(X_j - x\right)^2 + \left(Y_j - y\right)^2 + \left(Z_j - z\right)^2}}. \quad (7)$$

Manipulating (4), (8) is obtained and represents the differential increment to the original position guess.

$$\delta c = \left[\alpha^T.\alpha\right]^{-1} \alpha^T.\delta d, \quad (8)$$

where:

- δc is the differential increment to the original position guess;
- δd is the difference between the satellite's measured pseudo-distance and the pseudo-distance calculated using the current position guess.

Once the new position guess is obtained, equation (8) is applied until the error delta, computed by (9), is lower than an arbitrary $\varepsilon_{\delta v}$.

$$\delta v = \sqrt{\delta x^2 + \delta y^2 + \delta z^2 + \delta b_u^2}. \quad (9)$$

This algorithm diverges when the initial guess is far from the real position.

B. Algorithm using Kalman filtering

Building a state model and a model for the noise, with known covariance, the coordinates of an user can be determined using Kalman filtering.

State model

A 2-dimensional state model is built for each vehicle. The state vector is constituted by:

- ψ - angle between the referential of the body and the inertial referential;
- p_x - x vehicle position on the inertial referential;
- p_y - y vehicle position on the inertial referential;
- ω - vehicle's angular velocity;
- V - vehicle's linear velocity;
- b_u - error induced by the GPS receiver's clock.

Based on the state vector defined above, a nonlinear state model is presented in (10).

$$\begin{cases} \dot{\psi}(t) = \omega(t) \\ \dot{p}_x(t) = V(t)\cos(\psi(t)) \\ \dot{p}_y(t) = V(t)\sin(\psi(t)) \\ \dot{\omega}(t) = 0 \\ \dot{V}(t) = 0 \\ \dot{b}_u = 0 \end{cases}. \qquad (10)$$

The state model (10) assumes that the linear and angular velocities are constant. However, the model can be extended to allow variations on the velocities that characterize the trajectory, by defining $\xi_v(t)$ and $\xi_w(t)$ as random variables which characterize the changes in the linear and angular velocities [6]. Considering this fact, a new nonlinear state model is defined:

$$\begin{cases} \dot{\psi}(t) = \omega(t) \\ \dot{p}_x(t) = V(t)\cos(\psi(t)) \\ \dot{p}_y(t) = V(t)\sin(\psi(t)) \\ \dot{\omega}(t) = \xi_w(t) \\ \dot{V}(t) = \xi_v(t) \\ \dot{b}_u = 0 \end{cases}. \qquad (11)$$

Discretizing the state model and assuming the state evolution proportional to the sample time (h) square root [6], the discrete state model is obtained:

$$\begin{cases} \psi(t+h) = \psi(t) + h\omega(t) \\ p_x(t+h) = p_x(t) + V(t)\cos(\psi(t)).h \\ p_y(t+h) = p_y(t) + V(t)\sin(\psi(t)).h \\ \omega(t+h) = \omega(t) + \sqrt{h}\xi_\omega(t) \\ V(t+h) = V(t) + \sqrt{h}\xi_v(t) \\ b_u(t+h) = b_u(t) \end{cases}. \qquad (12)$$

Writing the system (12) in a matrix form, (13) is obtained.

$$\mathbf{x_{k+1}} = \boldsymbol{\varphi}(\mathbf{x_k}, k) + \mathbf{G}\boldsymbol{\xi}_\mathbf{k}, \qquad (13)$$

where:

$$\mathbf{x_{k+1}} = \begin{bmatrix} \psi_{k+1} & (p_x)_{k+1} & (p_y)_{k+1} & \omega_{k+1} & V_{k+1} & b_{u_{k+1}} \end{bmatrix}^T; (14)$$

$$\boldsymbol{\varphi}(\mathbf{x_k}, k) = \begin{bmatrix} 1 & 0 & 0 & h & 0 & 0 \\ 0 & 1 & 0 & 0 & \cos(\psi_k)h & 0 \\ 0 & 0 & 1 & 0 & \sin(\psi_k)h & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}; (15)$$

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & \sqrt{h} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{h} & 0 \end{bmatrix}^T; \qquad (16)$$

$$\boldsymbol{\xi}_\mathbf{k} = \begin{bmatrix} \xi_w \\ \xi_v \end{bmatrix}. \qquad (17)$$

The linearization of the transition matrix is given by (18).

$$\mathbf{J}(\mathbf{x_k}, k) = \frac{\partial \boldsymbol{\varphi}(\mathbf{x_k}, k)}{\partial \mathbf{x_k}} = \\ = \begin{bmatrix} 1 & 0 & 0 & h & 0 & 0 \\ -V_k sen(\psi_k)h & 1 & 0 & 0 & \cos(\psi_k)h & 0 \\ V_k\cos(\psi_k)h & 0 & 1 & 0 & \sin(\psi_k)h & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (18)$$

The state noise covariance matrix is given by (19) [6].

$$\mathbf{Q}_k = E\left\{\mathbf{G}\xi\xi^T\mathbf{G}^T\right\} = \\ = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & h\sigma_\omega^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & h\sigma_v^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (19)$$

where:

- h is the sample time;
- σ_ω^2 e σ_v^2 are the noise variance affecting the angular and linear velocity.

Kalman observations

The Kalman filter observations are the pseudo-distances, defined in (2), given by each available satellite. Since the pseudo-distances are nonlinear equations, it is necessary to compute the Jacobian of the system (3), in order to relate the state vector to the Kalman observations. The linearized matrix is given by (20).

$$\frac{\partial h(x_k,k)}{\partial x_k} = \begin{bmatrix} 0 & \frac{\partial d_{1i}}{\partial x} & \frac{\partial d_{1i}}{\partial y} & 0 & 0 & 1 \\ 0 & \frac{\partial d_{2i}}{\partial x} & \frac{\partial d_{2i}}{\partial y} & 0 & 0 & 1 \\ 0 & \frac{\partial d_{3i}}{\partial x} & \frac{\partial d_{3i}}{\partial y} & 0 & 0 & 1 \\ 0 & \frac{\partial d_{4i}}{\partial x} & \frac{\partial d_{4i}}{\partial y} & 0 & 0 & 1 \end{bmatrix}, (20)$$

where:

$$\frac{\partial d_{ji}}{\partial x} = -\frac{(X_j - x)}{\sqrt{(X_j - x)^2 + (Y_j - y)^2 + (Z_j - z)^2}};$$

$$\frac{\partial d_{ji}}{\partial y} = -\frac{(Y_j - y)}{\sqrt{(X_j - x)^2 + (Y_j - y)^2 + (Z_j - z)^2}}.$$

The observation noise covariance matrix is given by (21).

$$\mathbf{R}_{ki} = E\left\{\boldsymbol{\eta}_i\boldsymbol{\eta}_i^T\right\} = \\ = \begin{bmatrix} \sigma_{d1i}^2 + \sigma_c^2 & \sigma_c^2 & \sigma_c^2 & \sigma_c^2 \\ \sigma_c^2 & \sigma_{d2i}^2 + \sigma_c^2 & \sigma_c^2 & \sigma_c^2 \\ \sigma_c^2 & \sigma_c^2 & \sigma_{d3i}^2 + \sigma_c^2 & \sigma_c^2 \\ \sigma_c^2 & \sigma_c^2 & \sigma_c^2 & \sigma_{d4i}^2 + \sigma_c^2 \end{bmatrix}, (21)$$

where:

$$\boldsymbol{\eta}_i = [(\eta_c + \eta_{d1i}) \quad (\eta_c + \eta_{d2i}) \quad (\eta_c + \eta_{d3i}) \quad (\eta_c + \eta_{d4i})]^T \quad (22)$$

IV. Studied Strategies

Three different strategies were developed, studied and implemented in order to estimate the distance between vehicles, using Kalman filtering. A centralized strategy, a decentralized strategy and a strategy using covariance intersection algorithm were studied, developed and implemented [8].

A. Centralized Strategy

The centralized strategy consists in a data processing unit that collects all the data/information available, as all the pseudo-distances for each vehicle, the cinematic of each vehicle and the cross-correlation between vehicle's pseudo-distances noise. The processing unit can be aboard on a vehicle or on an external processing unit.

This strategy has the disadvantage of having large matrixes, typically N variables per vehicle, which as a great computational cost and complexity.

Kalman filter

The state vector for the centralized strategy is defined by N state vectors (14), where N represents the number of vehicles. For a 2 vehicles formation, the state vector is defined as (23).

$$\mathbf{x_k} = \left[(\mathbf{x_k})_\mathbf{1}^T \quad (\mathbf{x_k})_\mathbf{2}^T\right]^T, \quad (23)$$

where $(\mathbf{x}_k)_i$ is the state vector (14).

Taking into account (23), the equation for the state model can be written:

$$\mathbf{x_{k+1}} = \boldsymbol{\varphi}(\mathbf{x_k}, k) + \mathbf{G}\boldsymbol{\xi}_\mathbf{k}. \quad (24)$$

The transition matrix $\boldsymbol{\varphi}(\mathbf{x_k}, k)$ and $\mathbf{G}$ matrix are given by (25) and (26), respectively.

$$\boldsymbol{\varphi}(\mathbf{x_k},k)=\begin{bmatrix}\boldsymbol{\varphi}((\mathbf{x_k})_1,k) & 0\\ 0 & \boldsymbol{\varphi}((\mathbf{x_k})_2,k)\end{bmatrix},\quad(25)$$

where $\boldsymbol{\varphi}((\mathbf{x_k})_\mathbf{i},k)$ is the transition matrix (15) associated at each vehicle.

$$\mathbf{G}=\begin{bmatrix}(\mathbf{G})_1^T & 0\\ 0 & (\mathbf{G})_2^T\end{bmatrix}^T,\quad(26)$$

where $(\mathbf{G})_\mathbf{i}$ is the (16) matrix, associated at each vehicle.

Once the transition matrix is nonlinear, it is necessary to linearize it. The linearized matrix is given by (27).

$$\mathbf{J_k}(\mathbf{x_k},k)=\frac{\partial\boldsymbol{\varphi}(\mathbf{x_k},k)}{\partial\mathbf{x_k}}=$$
$$=\begin{bmatrix}\frac{\partial\boldsymbol{\varphi}((\mathbf{x_k})_1,k)}{\partial(\mathbf{x_k})_1} & 0\\ 0 & \frac{\partial\boldsymbol{\varphi}((\mathbf{x_k})_2,k)}{\partial(\mathbf{x_k})_2}\end{bmatrix},\quad(27)$$

where $\frac{\partial\boldsymbol{\varphi}((\mathbf{x_k})_\mathbf{i},k)}{\partial(\mathbf{x_k})_i}$ is the Jacobian (18) for each vehicle.

The matrix of noise $\boldsymbol{\xi}_\mathbf{k}$ is given by (28) and has the covariance matrix (29) associated.

$$\boldsymbol{\xi}_\mathbf{k}=\begin{bmatrix}(\boldsymbol{\xi}_\mathbf{k})_1 & 0\\ 0 & (\boldsymbol{\xi}_\mathbf{k})_2\end{bmatrix},\quad(28)$$

where $(\boldsymbol{\xi}_\mathbf{k})_\mathbf{i}$ is the vector for the noise (17) affecting each vehicle.

$$\mathbf{Q_k}=\begin{bmatrix}(\mathbf{Q_k})_1 & 0\\ 0 & (\mathbf{Q_k})_2\end{bmatrix},\quad(29)$$

where $(\mathbf{Q}_k)_i$ is the covariance matrix associated at the state noise for each vehicle and defined by (19).

The matrix $\mathbf{H}$, that relates the state vector to the observations is given by (30)

$$\mathbf{H}=\frac{\partial h(\mathbf{x_k},k)}{\partial x_k}=$$
$$=\begin{bmatrix}\frac{\partial h((\mathbf{x_k})_1,k)}{\partial x_k} & 0\\ 0 & \frac{\partial h((\mathbf{x_k})_2,k)}{\partial x_k}\end{bmatrix},\quad(30)$$

where $\frac{\partial h((\mathbf{x_k})_\mathbf{i},k)}{\partial x_k}$ is the matrix, defined by (20), that relates the state vector to each vehicle observations,.

The covariance matrix associated to the observations errors is given by (31).

$$R_k=\begin{bmatrix}(R_k)_1 & \sigma_c^2\\ \sigma_c^2 & (R_k)_2\end{bmatrix},\quad(31)$$

where $(R_k)_i$ is the covariance matrix (21) associated to the observations of each vehicle, and σ_c^2 is the observations correlated noise variance.

B. Decentralized Strategy

The decentralized strategy consists in estimating each vehicle state locally, and then estimates the distance between vehicles. The vehicle state is obtained using Kalman filtering based on the available data, i.e., the pseudo-distance associated at each vehicle.

This strategy has the advantage of having lower computational complexity than the centralized strategy. However, it is expected to obtain worst results than the centralized strategy.

Kalman Filter

The state estimation of a vehicle is done independently of the others, so the equations already introduced in the chapter III can be used. Summarizing, the state model to consider for each vehicle is given by (13), and the covariance matrix for the state noise is (19). The transition matrix (15) linearization is given by the matrix (18). The matrix that relates the state vector to the observations is the matrix (20), and the observation noise covariance matrix associated is (21).

C. Decentralized strategy using Covariance Intersection

The decentralized strategy using Covariance Intersection [8][9] is based on the same idea as the decentralized strategy. Each vehicle estimates its position and then, using covariance intersection algorithm, the distance between vehicles is estimated.

The objective of this strategy is to improve the estimative of the distance between vehicles without increasing the computational complexity of the solution.

Kalman Filter

The Kalman filtering used for this strategy is the same for the decentralized strategy.

Covariance Intersection

Let **X** and **Y** be random variables, whose mean and variance are given by:

$$E\{\mathbf{X}\} = \overline{\mathbf{x}} = E\{\mathbf{Y}\} = \overline{\mathbf{y}}\,, \tag{32}$$

$$E\{\mathbf{X}.\mathbf{X}^T\} = \overline{\mathbf{P}}_{\mathbf{xx}}\,, \tag{33}$$

$$E\{\mathbf{Y}.\mathbf{Y}^T\} = \overline{\mathbf{P}}_{\mathbf{yy}}\,, \tag{34}$$

$$E\{\mathbf{X}.\mathbf{Y}^T\} = \overline{\mathbf{P}}_{\mathbf{xy}}\,. \tag{35}$$

Let $\hat{\mathbf{z}}$ be a linear combination of **X** and **Y**:

$$\hat{\mathbf{z}} = \mathbf{W}_{\mathbf{x}}.\mathbf{x} + \mathbf{W}_{\mathbf{y}}.\mathbf{y}\,, \tag{36}$$

where x and y are a priori estimations of $\overline{x}$ with a known variance. $\mathbf{W}_{\mathbf{x}}$ and $\mathbf{W}_{\mathbf{y}}$ are matrices that obey to $\mathbf{W}_{\mathbf{x}}+\mathbf{W}_{\mathbf{y}} = \mathbf{I}$.

The mean value of $\hat{\mathbf{z}}$ is given by (37), and it's covariance by (38).

$$E\{\hat{\mathbf{z}}\} = \overline{\mathbf{z}} = \mathbf{W}_{\mathbf{x}}.\mathbf{x} + \mathbf{W}_{\mathbf{y}}.\overline{\mathbf{y}} = \overline{\mathbf{x}} \tag{37}$$

$$\overline{\mathbf{P}}_{\mathbf{zz}} = \mathbf{W}_{\mathbf{x}}.\overline{\mathbf{P}}_{\mathbf{xx}}.\mathbf{W}_{\mathbf{x}}^T + \mathbf{W}_{\mathbf{x}}.\overline{\mathbf{P}}_{\mathbf{xy}}.\mathbf{W}_{\mathbf{y}}^T + \mathbf{W}_{\mathbf{y}}.\overline{\mathbf{P}}_{\mathbf{yx}}.\mathbf{W}_{\mathbf{x}}^T + \mathbf{W}_{\mathbf{y}}.\overline{\mathbf{P}}_{\mathbf{yy}}.\mathbf{W}_{\mathbf{y}}^T \tag{38}$$

If $\overline{\mathbf{P}}_{\mathbf{xy}}$ is unknown, the covariance algorithm permits to obtain an estimative of $\overline{\mathbf{z}}$ and a covariance matrix as $\mathbf{P}_{\mathbf{zz}}>\overline{\mathbf{P}}_{\mathbf{zz}}$. The equations that define the covariance algorithm are (39) and (40).

$$\mathbf{P}_{\mathbf{zz}}^{-1} = \omega.\mathbf{P}_{\mathbf{xx}}^{-1} + (1-\omega)\,\mathbf{P}_{\mathbf{yy}}^{-1}\,, \tag{39}$$

$$\hat{\mathbf{z}} = \mathbf{P}_{\mathbf{zz}}\left(\omega.\mathbf{P}_{\mathbf{xx}}^{-1}.\mathbf{x} + (1-\omega).\mathbf{P}_{\mathbf{yy}}^{-1}.\mathbf{y}\right), \tag{40}$$

where $\mathbf{P}_{\mathbf{xx}}$ and $\mathbf{P}_{\mathbf{yy}}$ are covariance matrices for x and y, whose $\mathbf{P}_{\mathbf{xx}} \geq \overline{\mathbf{P}}_{\mathbf{xx}}$ and $\mathbf{P}_{\mathbf{yy}} \geq \overline{\mathbf{P}}_{\mathbf{yy}}$. ω is required to be optimized at every step, for example by minimizing the trace of $\mathbf{P}_{\mathbf{zz}}$ [9].

To apply the covariance intersection algorithm, an estimation of the vehicles distance must be obtained:

$$\begin{aligned} \mathbf{d_1} &= 2\left(\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} - \begin{bmatrix} x_c \\ y_c \end{bmatrix}\right) \\ \mathbf{d_2} &= 2\left(\begin{bmatrix} x_c \\ y_c \end{bmatrix} - \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}\right) \end{aligned}. \tag{41}$$

The covariance from the Kalman filter, multiplied by four, is used as input of the covariance matrix.

$$\begin{aligned} &E\{d_i.d_i^T\} = \\ &= 4.E\left\{\begin{bmatrix} x_i \\ y_i \end{bmatrix}.\begin{bmatrix} x_i & y_i \end{bmatrix}\right\} = 4P \end{aligned}, \tag{42}$$

where P is the covariance matrix obtained from the Kalman filter covariance matrix.

D. Strategies comparison and conclusions

In order to compare the introduced strategies, MatLab/Simulink software is used to simulate the algorithms. For this simulation a 2 vehicles formation is used. They move in a circular trajectory at 3 m/s and 0.01 rad/s. The initial vehicles positions are x=0 m, x=51 m and y=0 m. The simulation is made during 1000 s at a sample rate of 1 s. The decentralized strategy has, for the state noise covariance matrix, the matrix (43). For the centralized strategy matrix (44) is used.

$$Q = E\{\xi_k \xi_k^T\} = \begin{bmatrix} 10^{-9} & 0 \\ 0 & 10^{-9} \end{bmatrix}, \tag{43}$$

$$Q_{cent} = E\{\xi_k \xi_k^T\} = \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix}. \tag{44}$$

The simulation results are presented for several pairs of variances of correlated and uncorrelated noise. The correlated noise standard deviation used in the simulation is in the interval [0; 90] m, and the uncorrelated noise in the interval from 1 m to 40 m.

Figures 1 and 2 show the estimation error variance for each of the studied strategies and for the coordinate x and coordinate y, respectively.

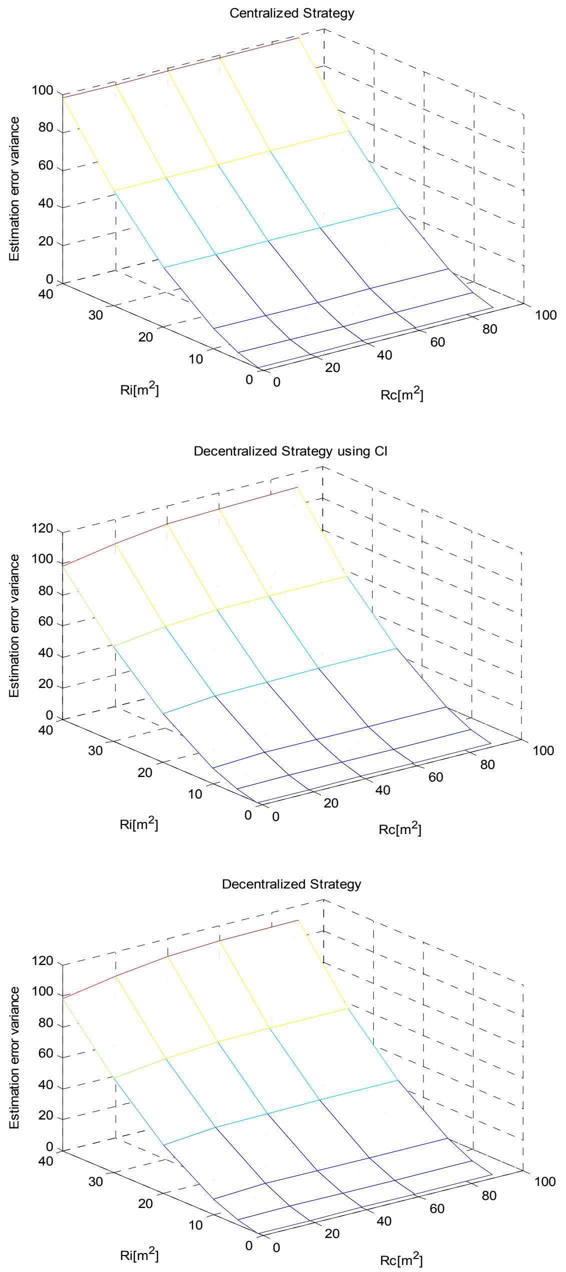
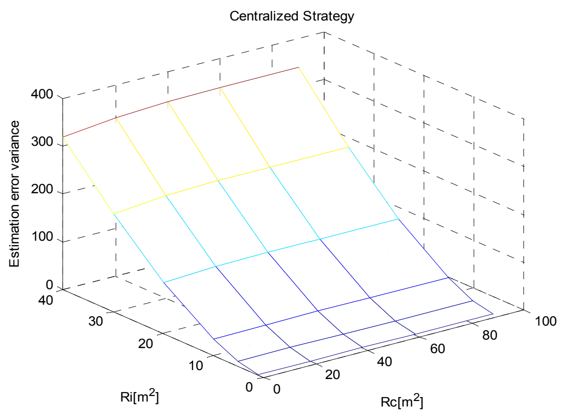


Fig. 1. Estimation error variance for the Centralized, Decentralized using CI and Decentralized strategies for the x coordinate.



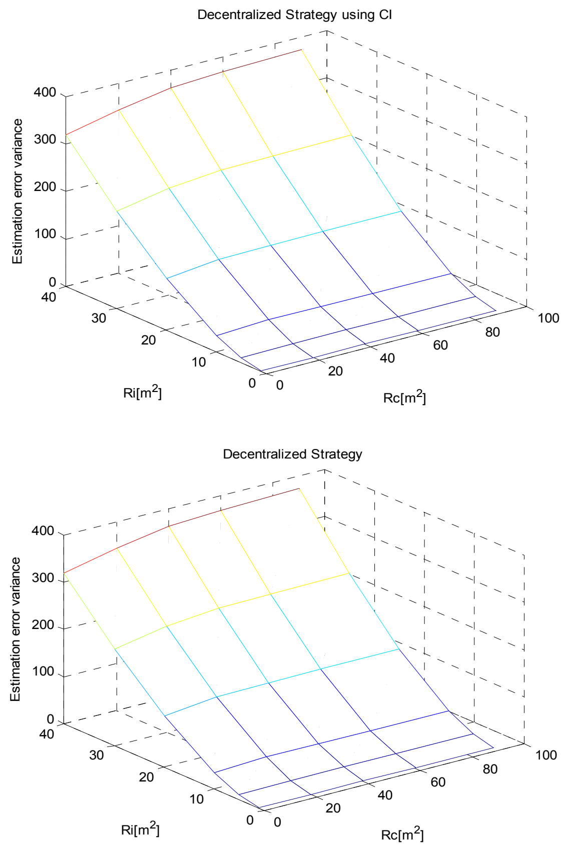


Fig. 2. Estimation error variance for the Centralized, Decentralized using CI and Decentralized strategies for the y coordinate.

Analyzing the results, it is possible to conclude that the centralized strategy has the best results among the 3 strategies. The decentralized strategy has the worst results, as expected, because it uses less data to estimate the distance between vehicles. The decentralized strategy using CI algorithm does not significantly improve the decentralized estimation.

From the Figures 1 and 2, it can be seen that the uncorrelated noise highly affects the estimation error variance. However, the estimation error variance is not much influenced by the increase of the uncorrelated noise variance.

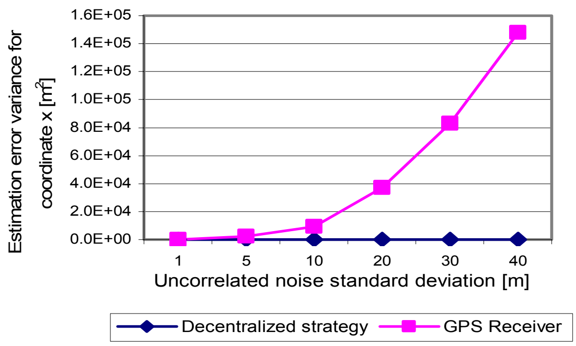


Fig. 3. Comparison of the estimation error variance for the Decentralized strategy and for the strategy using the iterative algorithm, for the coordinate x.

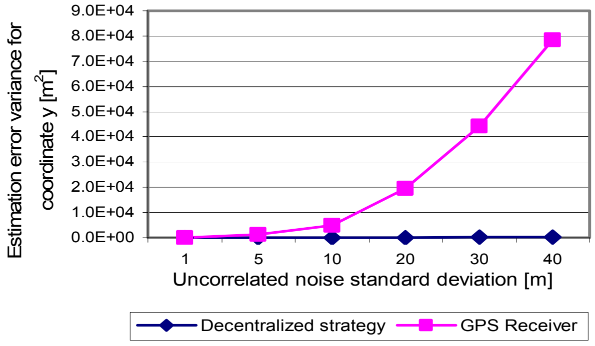


Fig. 4. Comparison of the estimation error variance for the Decentralized strategy and for the strategy using the iterative algorithm, for the coordinate y.

The variance obtained from the strategies using Kalman filtering is lower than the variance obtained from a basic positional algorithm as the iterative algorithm, as can be seen in Figure 3 and 4. This is justified by the fact that the iterative algorithm only uses the pseudo-distance to estimate the distance. On the other hand, the studied strategies use the pseudo-distance, the vehicles dynamic and the information about the noise that affect the pseudo-distances.

V. Algorithm performance

In order to analyze the studied algorithm performance, Posterior Cramér-Rao algorithm [10][11][12][13] is implemented for the centralized strategy and compared to the results obtained for the centralized strategy.

The Posterior Cramér-Rao permits to know what is the minimum square error matrix, P, possible to obtain for a nonlinear estimation problem (45). With this result it is possible to know how good the studied algorithm is.

$$\mathbf{C}-\mathbf{P}\geq\mathbf{0}, \tag{45}$$

where $\mathbf{C}$ is the matrix obtained from an estimation algorithm, and $\geq\mathbf{0}$ represents a positive semi defined matrix.

A. Posterior Cramér-Rao

Consider the nonlinear estimation problem (46).

$$\begin{cases}\mathbf{x}_{k+1}=f_k\left(\mathbf{x}_k,\mathbf{w}_k\right)\\ \mathbf{z}_k=h_k\left(\mathbf{x}_k,\mathbf{v}_k\right)\end{cases}, \tag{46}$$

where:

- $\mathbf{x}_k$ is the state vector (r x 1) for the time k;
- $\mathbf{z}_k$ are the Kalman observation vectors for the time k;
- $\mathbf{w}_k$ and $\mathbf{v}_k$ are uncorrelated white noise which affect the state and the observations of the system;
- f_k and h_k are nonlinear functions which relate the state for time k and k+1, and the observations to the state for time k;

It is assumed that the initial state probability $p\left(\mathbf{x}_0\right)$ is known. The system (46) defines, univocally, a probability density function for $\mathbf{X}_k=\left(\mathbf{x}_0,\ldots,\mathbf{x}_k\right)$ and $\mathbf{Z}_k=\left(\mathbf{z}_0,\ldots,\mathbf{z}_k\right)$:

$$p\left(\mathbf{X}_k,\mathbf{Z}_k\right)=p\left(\mathbf{x}_0\right)\prod_{j=1}^{k}p\left(\mathbf{z}_j\mid\mathbf{x}_j\right)\prod_{i=1}^{k}p\left(\mathbf{x}_{i+1}\mid\mathbf{x}_i\right), \tag{47}$$

where $p\left(\mathbf{z}_j\mid\mathbf{x}_j\right)$ and $p\left(\mathbf{x}_{i+1}\mid\mathbf{x}_i\right)$ are defined from (46).

For the studied strategies, $p\left(\mathbf{x}_{i+1}\mid\mathbf{x}_i\right)$ is not defined, because the noise state covariance matrix is singular (det = 0).

A possible solution for this problem is to redefine the filter (46).

Let the state vector $\mathbf{x}_k$ be written as (48).

$$\mathbf{x}_n=\begin{bmatrix}\mathbf{x}_k^{(1)}\\ \mathbf{x}_k^{(2)}\end{bmatrix}, \tag{48}$$

where $\mathbf{x}_k^{(1)}$ are the state variables that are affected by error.

The system can now be defined by the equations (49), (50) and (51).

$$\mathbf{x}_{k+1}^{(1)}=f_k\left(\mathbf{x}_k,\mathbf{w}_k\right), \tag{49}$$

$$\mathbf{x}_{k+1}^{(2)}=g_k\left(\mathbf{x}_k,\mathbf{x}_{k+1}^{(1)}\right), \tag{50}$$

$$\mathbf{z}_k=h_k\left(\mathbf{x}_k,\mathbf{v}_k\right). \tag{51}$$

The main idea to solve this problem is to add white noise, $\mathbf{w}_k^{(2)}$, with covariance matrix $\varepsilon\mathbf{I}$ ($\varepsilon\rightarrow 0$), to the equation (50):

$$\mathbf{x}_{k+1}^{(2)}=g_k\left(\mathbf{x}_k,\mathbf{x}_{k+1}^{(1)}\right)+\mathbf{w}_k^{(2)}. \tag{52}$$

Let $p_\varepsilon(.)$ and E_ε be the probability density function and the expected value of the perturbed system (49), (50), (51). For the new system, $p_\varepsilon\left(\mathbf{x}_{k+1}\mid\mathbf{x}_k\right)$ is now defined by (53).

$$p_\varepsilon\left(\mathbf{x}_{k+1}\mid\mathbf{x}_k\right)=p\left(\mathbf{x}_{k+1}^{(1)}\mid\mathbf{x}_k\right)p_\varepsilon\left(\mathbf{x}_{k+1}^{(2)}\mid\mathbf{x}_k,\mathbf{x}_{k+1}^{(1)}\right). \tag{53}$$

where $p\left(\mathbf{x}_{k+1}^{(1)} \mid \mathbf{x}_k\right)$ is the probability density function determined from (49), and $-\log\left(p_\varepsilon\left(\mathbf{x}_{k+1}^{(2)} \mid \mathbf{x}_k, \mathbf{x}_{k+1}^{(1)}\right)\right) = c_1 + \frac{1}{2\varepsilon}\left\|\mathbf{x}_{k+1}^{(2)} - g_k\left(\mathbf{x}_k, \mathbf{x}_{k+1}^{(1)}\right)\right\|^2$, where c_1 is a constant.

The solution for this problem can now be given by the recursive equation (54).

$$\mathbf{J}_{k+1} = \mathbf{D}_k^{22} - \mathbf{D}_k^{21}\left(\mathbf{J}_k + \mathbf{D}_n^{11}\right)^{-1}\mathbf{D}_k^{12}, \quad (54)$$

where $\mathbf{D}_{\varepsilon,k}^{ij} = \bar{\mathbf{D}}_{\varepsilon,k}^{ij} + \frac{1}{\varepsilon}\mathbf{K}_{\varepsilon,k}^{ij},\ i,j = 1;2$, and:

$$\mathbf{D}_k^{11} = E\left\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log\, p\left(\mathbf{x}_{k+1}^{(1)} \mid \mathbf{x}_k\right)\right\},$$

$$\mathbf{D}_k^{12} = E\left\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_{k+1}} \log\, p\left(\mathbf{x}_{k+1}^{(1)} \mid \mathbf{x}_k\right)\right\},$$

$$\mathbf{D}_k^{12} = E\left\{-\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_k} \log\, p\left(\mathbf{x}_{k+1}^{(1)} \mid \mathbf{x}_k\right)\right\} = \left[\mathbf{D}_k^{21}\right]^T,$$

$$\mathbf{D}_k^{22} = E\left\{-\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} \log\, p\left(\mathbf{x}_{k+1}^{(1)} \mid \mathbf{x}_k\right)\right\} + E\left\{-\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} \log\, p\left(\mathbf{z}_{k+1} \mid \mathbf{x}_k\right)\right\}$$

$$\mathbf{K}_{\varepsilon,k}^{11} = E_\varepsilon\left\{\left[\nabla_{x_k} g^T\right]\left[\nabla_{x_k} g^T\right]^T\right\},$$

$$\mathbf{K}_{\varepsilon,k}^{12} = \left[E_\varepsilon\left\{\left[\nabla_{x_k} g^T\right]\left[\nabla_{x_{k+1}^{(1)}} g^T\right]^T\right\} \quad E_\varepsilon\left\{\nabla_{x_k} g^T\right\}\right],$$

$$\mathbf{K}_{\varepsilon,k}^{22} = \begin{bmatrix} E_\varepsilon\left\{\left[\nabla_{x_{k+1}^{(1)}} g^T\right]\left[\nabla_{x_{k+1}^{(1)}} g^T\right]^T\right\} & -E_\varepsilon\left\{\nabla_{x_{k+1}^{(1)}} g^T\right\} \\ -E_\varepsilon\left\{\nabla_{x_{k+1}^{(1)}} g^T\right\}^T & \mathbf{I} \end{bmatrix},$$

$$\Delta_a^b = \nabla_a \nabla_b^T,$$

$$\nabla_a = \left[\frac{\partial}{\partial a_1} \quad \cdots \quad \frac{\partial}{\partial a_r}\right]^T.$$

According to [14], the equations presented before can be simplified to (55).

$$\mathbf{J}_{k+1} = \left(\frac{\partial \mathbf{h}_{k+1}}{\partial \mathbf{x}_k}\right)^{\mathbf{T}} \mathbf{R}_{k+1}^{-1} \frac{\partial \mathbf{h}_{k+1}}{\partial \mathbf{x}_k} + \left(\mathbf{Q}_k + \frac{\partial \boldsymbol{\varphi}_k}{\partial \mathbf{x}_k} \mathbf{J}_k^{-1} \left(\frac{\partial \boldsymbol{\varphi}_k}{\partial \mathbf{x}_k}\right)^T\right)^{-1}, (55)$$

where:

- $\left(\frac{\partial \mathbf{h}_{k+1}}{\partial \mathbf{x}_k}\right)$ is the linearization, on the nominal trajectory, of the matrix that relates the state vector to the observations;
- $\mathbf{R}_{k+1}$ is the observation noise covariance matrix for time k+1;
- $\mathbf{Q}_k$ is the state noise covariance matrix, for time k;
- $\frac{\partial \boldsymbol{\varphi}_k}{\partial \mathbf{x}_k}$ is the transition matrix linearization, on the nominal trajectory, for time k.

B. Results and conclusions

The results for Cramér-Rao for the centralized strategy are presented in Figure 3 and 4.

Figures 5 and 6 show the estimation error variance comparison for the coordinate x and y, for a given pair of correlated and uncorrelated noise variance.

Analyzing the results it is possible to conclude that the estimation error variance is highly dependent on the uncorrelated noise and suffers a minor variation when the correlated noise variance increases. Figures 5 and 6 also show that the results obtained for the centralized strategy are not far from the optimal solution given by the Posterior Cramér-Rao algorithm.

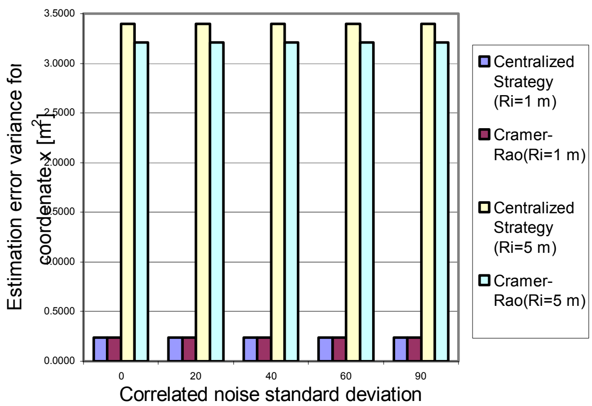


Fig. 5 .Comparison of the estimation error variance obtained by the centralized strategy and Posterior Cramér-Rao, for the coordinate x, for 1 m of correlated noise (Ri) standard deviation.

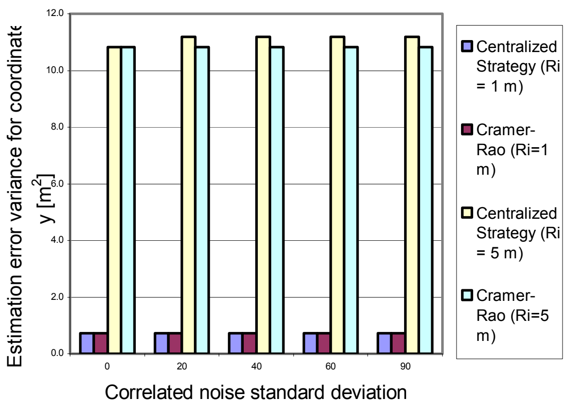


Fig. 6 .Comparison of the estimation error variance obtained by the centralized strategy and Posterior Cramér-Rao, for the

coordinate y, for 1 m of correlated noise (Ri) standard deviation.

VI. Conclusions

In this paper was presented and compared three different strategies (Centralized, Decentralized and Decentralized using Covariance Intersection algorithm), using Kalman filtering, to estimate the distance between vehicles of a formation. These results were also compared to a strategy using a basic absolute positional algorithm, like iterative algorithm.

It was concluded that the strategies using Kalman filtering had lower square mean error estimation variance than the iterative algorithm.

Comparing the Centralized, Decentralized and Decentralized using Covariance Intersection algorithm strategies, it was concluded that the Centralized strategy had the lowest estimation error variance.

In order to improve the Decentralized strategy results, it was applied Covariance Intersection algorithm to the Decentralized strategy output. However, the results couldn't be improved further.

The Centralized strategy performance was also analyzed, by comparing the results to the variance obtained by the Posterior Cramér-Rao. It was concluded that the estimation error variance resulting from the Centralized strategy are not far from the optimal estimation given by the Posterior Cramér-Rao algorithm.

It was also concluded that the uncorrelated noise affecting the pseudo-distances has great effect on the results. However the correlated noise that affects the pseudo-distances has minimal effect on the estimation error variance.

VII. References

[1] A. Gomes, R. Mendes. Técnicas Avançadas para Posicionamento Global Baseadas em GPS. Trabalho Final de Curso, Universidade Técnica de Lisboa, 2006.

[2] Bancroft, S., "An algebraic solution of the GPS equations", *IEEE Transactions on Aerospace and Electronics System*, 1985, 21(7), 56-59.

[3] Stein, V., Determination of the ionospheric electron density distribution from time delay measurements of VHF-satellite tracking data, European Space Agency, Paris, France, December 1976.

[4] Jonge, P.J., Tiberius, C.C.J.M., "*The LAMBDA method for integer ambiguity estimation: implementation aspects*", in LGR-Series publications of the Delft Geodetic Computing Center, No. 12, 1996.

[5] Bao, J., Tsui, Y., Fundamentals of Global Positioning System Receivers. John Wiley&Sons, Inc., 2000.

[6] P. Alves. Seguimento e Predição de Trajectória de Alvos Móveis em Duas Dimensões.. Tese de Mestrado, Universidade Técnica de Lisboa, 2004.

[7] Wolfgang Niehsen. *Information Fusion based on Fast Covariance Intersection Filtering.*

[8] Pablo O. Arambel, Constantino Rago, Raman K. Mehra. *Covariance Intersection Algorithm for Distributed Spacecraft State Estimation*. Proceedings of the American Control Conference, Arlington, VA June 25-27, 2001.

[9] Lingji Chen, Pablo O. Arambel, Raman K. Mehra. *Estimation Under Unknown Correlation: Covariance Intersection Revisited*. IEEE Transactions on Automatic Control, Vol 47, NO.11, November 2002.

[10] Petr Tichavský, Carlos H. Muravchik, Arye Nehorai. *Posterior Cramér-Rao Bounds for Discrete-Time Nonlinear Filtering*. IEEE Transactions on Signal Processing, Vol 46, NO. 5, May 1998.

[11] Thomas H. Kerr. *Status of CR-Like Lower Bounds for Nonlinear Filtering*. IEEE Transactions on Aerospace and Electronic Systems, Vol 25, NO.5, September 1989.

[12] James H. Taylor. *The Crámer-Rao Estimation Error Lower Bound Computation for Deterministic Nonlinear Systems*. IEEE Transaction on Automatic Control, Vol 24, NO. 2, April 1979

[13] Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1993

[14] William F. Leven. *Approximate Crámer-Rao Bounds for Multiple Target Tracking*. May 2006