

# Underwater vehicle tracking

Pedro M. S. Valverde  
 Instituto Superior Técnico  
 Institute for Systems and Robotics, ISR  
 Lisbon, Portugal  
 pvalverde@isr.ist.utl.pt

**Abstract** — The main objective of this work is the comparative study of tracking systems that are able to estimate the position and the attitude of an underwater vehicle.

The performance of several estimation algorithms will be studied in detail and compared with the Cramér Rao Lower Bound and the a posteriori Cramér-Rao Lower Bound in static and dynamic scenarios, respectively.

To achieve this goal several dynamic state models for target tracking are discussed, Extended Kalman filters are developed and implemented, based on the measurements available from a range and bearing sensor and their performance are compared resorting to simulation results under realistic conditions.

**Keywords**—Target Tracking; Kalman Filtering; Cramér-Rao Lower Bound; Posterior Cramér-Rao Lower Bound.

## I. INTRODUCTION

Nowadays, the Autonomous Underwater Vehicles and the Remotely Operated Vehicles are perhaps one of the most important and challenging engineering problems. Estimating the vehicle's position with accuracy is just one of the many problems related with this thematic. Since the practical estimation algorithms that are developed for target tracking are rarely optimal, a very concern is to determine how far the best that can be done is and compare it to a given algorithm's performance to see if seeking for a more effective algorithm is worthwhile. During this paper two estimation algorithms are evaluated in order to discuss which algorithm is the best estimator.

## II. DISCRETE-TIME KALMAN FILTER

The Kalman filtering (KF) algorithm is, perhaps, the best known estimation method for linear systems, since it converges when the system is stable and time-invariant [2]. Despite this the models' configuration considered in this approach obligates to use an extension of Kalman filter, the Extended Kalman Filter (EKF). The EKF, as KF, is a powerful suboptimal estimation method for non-linear systems. Although, the behavior of EKF is not as well known as the behavior of KF. Consider a dynamic system described by

$$\begin{cases} \mathbf{x}_{k+1} = \phi(\mathbf{x}_k, \mathbf{k}) + \mathbf{w}_k \\ \mathbf{z}_k = h(\mathbf{x}_k, \mathbf{k}) + \mathbf{v}_k \end{cases} \quad (1)$$

where  $\mathbf{x}_k \in \mathbb{R}^n$  is the state vector at time  $k$ ,  $\phi \in \mathbb{R}^{n \times n}$  is the state transition matrix,  $h \in \mathbb{R}^{m \times n}$  is the matrix that connects the measurements and the state vector in ideal conditions,  $\mathbf{z}_k \in \mathbb{R}^m$

is the noisy measurement vector,  $\mathbf{w}_k \in \mathbb{R}^n$  is the Gaussian white process noise and  $\mathbf{v}_k \in \mathbb{R}^m$  is the Gaussian measurement noise.

The recursive KF algorithm minimizes the covariance of the estimation error using two cycles, the *prediction cycle* that propagates the state mean using a model of the system dynamics and the *update cycle* that updates the predicted state with measurement information. The equations for each one of these cycles are described below.

*Prediction cycle:* In this cycle the predicted error state covariance matrix  $\mathbf{P}_k^-$  and the state estimate  $\hat{\mathbf{x}}_k^-$ , are computed as follows,

$$\hat{\mathbf{x}}_{k+1}^- = \phi(\hat{\mathbf{x}}_k, \mathbf{k}) \quad (2)$$

$$\mathbf{P}_{k+1}^- = \phi_{Lin} \mathbf{P}_k \phi_{Lin}^T + \mathbf{Q}_k \quad (3)$$

where  $\mathbf{P}_k$  is the updated state covariance matrix at time  $k$ ,  $\mathbf{Q}_k \in \mathbb{R}^{n \times n}$  is the covariance matrix of the Gaussian white process noise and  $\phi_{Lin}$  is the state transition matrix linearized about the estimated state

$$\phi_{Lin} = \left. \partial\phi / \partial\mathbf{x}_k \right|_{\mathbf{x}_k = \hat{\mathbf{x}}_k}. \quad (4)$$

*Update cycle:* The updated state estimate  $\hat{\mathbf{x}}_{k+1}$  and the updated error covariance matrix  $\mathbf{P}_{k+1}$  at time  $k+1$  are computed respectively as

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1}^- + \mathbf{K}_k \left[ \mathbf{z}_{k+1} - h(\hat{\mathbf{x}}_{k+1}^-) \right] \quad (5)$$

$$\mathbf{P}_k = \mathbf{P}_k^- \left[ \mathbf{I} - (\partial h / \partial \mathbf{x}_k)^T \mathbf{S}_{k+1}^{-1} (\partial h / \partial \mathbf{x}_k) \mathbf{P}_k^- \right] \quad (6)$$

The residual covariance matrix at time  $k+1$  is defined as

$$\mathbf{S}_{k+1} = cov[\mathbf{r}_{k+1}; \mathbf{r}_{k+1}] = (\partial h / \partial \mathbf{x}_k) \mathbf{P}_{k+1}^- (\partial h / \partial \mathbf{x}_k)^T + \mathbf{R}_k \quad (7)$$

where  $\mathbf{R}_k \in \mathbb{R}^{m \times m}$  is the covariance matrix of the measurement noise. The Kalman Filter gain matrix  $\mathbf{K}_k \in \mathbb{R}^{n \times m}$ , is defined as

$$\mathbf{K}_k = \mathbf{P}_k (\partial h / \partial \mathbf{x}_k)^T \mathbf{R}_k^{-1}. \quad (8)$$

The residual vector at time  $k+1$  is

$$\mathbf{r}_{k+1} = \mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1}^-, \quad (9)$$

where  $\hat{\mathbf{z}}_{k+1}^- = h(\mathbf{x}_{k+1}^-, \mathbf{k}) + \mathbf{v}_{k+1}$

### III. SENSORS

The system considered in this work uses measurements from a sensor. Since the system is working underwater, the sensor must be based on acoustic waves because the electromagnetic waves experience strong attenuation in aquatic environments. As matter of fact the Ultra-Short Base Line (USBL) sensor is the a good choice for this kind of application.

The sensor measurements are the distance from the vehicle and the bearing angle,  $d$  and  $\Theta$  respectively, as represented in Figure 1.

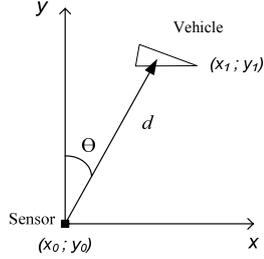


Figure 1. Sensor measures and sensor and vehicle positions.

The sensor works in polar coordinates that are related to Cartesian coordinates with the transformation  $g(x, y)$

$$\begin{bmatrix} d \\ \theta \end{bmatrix} = g(x, y) = \begin{bmatrix} \sqrt{x^2 + y^2} \\ \tan^{-1}(y/x) \end{bmatrix}. \quad (10)$$

The inverse transformation  $g^{-1}(d, \theta)$  exist everywhere, except when the distance is null

$$\begin{bmatrix} x \\ y \end{bmatrix} = g^{-1}(d, \theta) = \begin{bmatrix} d \cdot \cos(\theta) \\ d \cdot \sin(\theta) \end{bmatrix} \quad (11)$$

It is considered that the sensor measurements are disturbed by noise which can approximated by a Gaussian distribution. The measures are statistically characterized by

$$p(d, \theta | x, y) = \frac{1}{(2\pi|\mathbf{R}|)^{1/2}} \cdot \exp\left\{-\frac{1}{2}(\mathbf{z} - g(x, y))^T \mathbf{R}^{-1} (\mathbf{z} - g(x, y))\right\}, \quad (12)$$

where  $\mathbf{z} = [d, \theta]^T \in \mathbb{R}^m$  is the measurement vector and  $\mathbf{R} \in \mathbb{R}^{m \times m}$  is the measurements covariance matrix.

Applying the transformation (11) to the measurements, the new measurements have no longer a Gaussian distribution.

Applying linearization, the transformation represented in (10) leads to

$$J_g(x, y) = \begin{bmatrix} \frac{x}{\sqrt{x^2 + y^2}} & \frac{y}{\sqrt{x^2 + y^2}} \\ -\frac{y}{x^2 + y^2} & \frac{x}{x^2 + y^2} \end{bmatrix}. \quad (13)$$

Writing (13) in function of  $d$  and  $\Theta$  coordinates yields to

$$J_g(g^{-1}(d, \theta)) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\frac{\sin(\theta)}{d} & \frac{\cos(\theta)}{d} \end{bmatrix}. \quad (14)$$

After considering the linearization, if the measurements noises have a Gaussian distribution, the system can be approximated with a Gaussian distribution too.

### IV. TARGET TRACKING STATE MODELS

In this section two dynamic models for tracking purpose are presented, a circular motion dynamic model and a constant turn model with known turn rate. Both are implemented using an Extended Kalman Filter (EKF) for estimation of the target's position. The main objective of this analysis is to determinate which of the two state models presented is the best vehicle's position estimator. It is considered that the vehicle is describing a circular path during the period of simulation. Actually, all the paths that a vehicle can describe can be approximated by a circular or linear path (the linear case is considered a particular case of the circular case). The state models are presented in the next sections.

#### A. Circular motion state model

The model presented in this subsection is suited for vehicles performing circular paths with constant linear and angular velocities [6]. The target dynamics are driven by

$$\begin{cases} \dot{\psi} = \omega \\ \dot{\mathbf{p}} = \mathbf{R}(\psi) \cdot \begin{bmatrix} v \\ 0 \end{bmatrix} \end{cases} \quad (15)$$

where  $\psi$  is the angle that the vehicle does in relation to a direction of reference,  $\omega$  is the angular velocity,  $\mathbf{p} = [p_x \ p_y]^T$  is the vehicle's position in  $\mathbb{R}^2$ ,  $\mathbf{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix}$  is the rotational matrix in  $\mathbb{R}^2$  and  $v$  is the vehicle's linear velocity.

After some algebraic manipulation, (15) yields

$$\psi(t) = \psi_0 + \omega_0 \cdot (t - t_0), \quad (16)$$

$$p_x = v_0 \cos(\psi_0 + \omega_0 \cdot (t - t_0)), \quad (17)$$

$$p_y = v_0 \sin(\psi_0 + \omega_0 \cdot (t - t_0)), \quad (18)$$

$$\omega(t) = \omega_0, \quad (19)$$

$$\mathbf{v}(t) = \mathbf{v}_0. \quad (20)$$

It is possible to rewrite (17) and (18), taking in consideration the rotational matrix definition, as

$$\dot{\mathbf{p}} = \mathbf{R}(\omega_0 \cdot (t - t_0)) \mathbf{R}(\psi_0) \begin{bmatrix} \mathbf{v}_0 \\ 0 \end{bmatrix}. \quad (21)$$

Integrating (21), it simply becomes

$$\mathbf{p} = \mathbf{p}_0 + \mathbf{R}(\psi_0) \cdot [\mathbf{I} - \mathbf{R}(\omega_0 \cdot (t - t_0))]. \begin{bmatrix} 0 \\ \mathbf{v}_0 / \omega_0 \end{bmatrix}, \quad (22)$$

that is easily identified as the circumference's equation with center in  $\mathbf{p}_0 = [p_{x0} \ p_{y0}]$  and radius  $v_0 / \omega_0$ . Then, the state model becomes

$$\begin{bmatrix} \psi(t) \\ p_x(t) \\ p_y(t) \\ \omega(t) \\ v(t) \end{bmatrix} = \begin{bmatrix} \psi_0 \\ p_{x0} \\ p_{y0} \\ \omega_0 \\ v_0 \end{bmatrix} + \begin{bmatrix} \omega_0(t - t_0) \\ \mathbf{R}(\psi_0) \cdot \frac{\mathbf{I} - \mathbf{R}(\omega_0 \cdot (t - t_0))}{\omega_0} \cdot \begin{bmatrix} 0 \\ \mathbf{v}_0 \end{bmatrix} \\ 0 \\ 0 \end{bmatrix}. \quad (23)$$

It can be included in the model stochastic accelerations that can represent some uncertainty in some state variables and allow variations in the state. Following that, the model changes to

$$\begin{bmatrix} \psi(t) \\ p_x(t) \\ p_y(t) \\ \omega(t) \\ v(t) \end{bmatrix} = \begin{bmatrix} \psi_0 \\ p_{x0} \\ p_{y0} \\ \omega_0 \\ v_0 \end{bmatrix} + \begin{bmatrix} \omega_0 \Delta t \\ \mathbf{R}(\psi_0) \cdot \frac{\mathbf{I} - \mathbf{R}(\omega_0 \cdot \Delta t)}{\omega_0} \cdot \begin{bmatrix} 0 \\ \mathbf{v}_0 \end{bmatrix} \\ 0 \\ 0 \end{bmatrix} + \sqrt{\Delta t} \mathbf{B} \begin{bmatrix} \eta_\omega \\ \eta_v \end{bmatrix} \quad (24)$$

where  $\Delta t$  is the sampling period and the random variables  $\eta_\omega$  and  $\eta_v$  are introduced and represent the unknown acceleration that affects the angular and linear velocity, respectively. The

matrix  $\mathbf{B}$  is defined by  $\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T$ .

The covariance of the Gaussian white process noise is defined by

$$\mathbf{Q}_k = \text{blkdiag} \left( \mathbf{0}_{3 \times 3}, \Delta t \begin{bmatrix} \sigma_\omega^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix} \right), \quad (25)$$

where  $\text{blkdiag}$  represent a block diagonal matrix,  $\sigma_\omega^2$  and  $\sigma_v^2$  are the variance of the random variables  $\eta_\omega$  and  $\eta_v$ , respectively.

### 1) State Model linearization

The state model presented is non-linear, so the best approach is the linearization. The linearization is based on first-order Taylor series expansion around state estimates. The transition matrix's Jacobian becomes

$$\frac{\partial \phi(\mathbf{x}_k, k)}{\partial \mathbf{x}_k} = [\mathbf{J}_1 \mid \mathbf{J}_2 \mid \mathbf{J}_3], \quad (26)$$

where  $\mathbf{J}_1$ ,  $\mathbf{J}_2$ ,  $\mathbf{J}_3$  are defined as follows

$$\mathbf{J}_1 = \begin{bmatrix} 1 \\ \mathbf{R}(\psi) \cdot \frac{\mathbf{I} - \mathbf{R}(\omega \Delta t)}{\omega} \cdot \begin{bmatrix} -v \\ 0 \end{bmatrix} \\ \mathbf{0}_{2 \times 1} \end{bmatrix} \quad \mathbf{J}_2 = \begin{bmatrix} \mathbf{0}_{1 \times 2} \\ \mathbf{I}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} \end{bmatrix} \quad (27)$$

$$\mathbf{J}_3 = \begin{bmatrix} \Delta t \\ \mathbf{R}(\psi) \cdot \frac{\mathbf{I} - \mathbf{R}(\omega \Delta t)}{\omega} \cdot \begin{bmatrix} -\frac{1}{2} v \Delta t \\ -\frac{1}{2} + \frac{v \Delta t \sin(\omega \Delta t)}{\omega} \end{bmatrix} \\ 1 \\ 0 \end{bmatrix} \mathbf{R}(\psi) \cdot \frac{\mathbf{I} - \mathbf{R}(\omega \Delta t)}{\omega} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (28)$$

### 2) Circular motion state model with known angular velocity

In the cases where the angular velocity can be known apriori, as in the case of pre-programmed missions to be carried out by Autonomous Underwater Vehicles (AUVs), the angular velocity can be already known, so this variable is no longer required to be included in the dynamic model as a state variable. Then, the model described in (24) can be simplified to

$$\begin{bmatrix} \psi(t) \\ p_x(t) \\ p_y(t) \\ v(t) \end{bmatrix} = \begin{bmatrix} \psi_0 \\ p_{x0} \\ p_{y0} \\ v_0 \end{bmatrix} + \begin{bmatrix} \omega_0 \Delta t \\ \mathbf{R}(\psi_0) \cdot \frac{\mathbf{I} - \mathbf{R}(\omega_0 \cdot \Delta t)}{\omega_0} \cdot \begin{bmatrix} 0 \\ \mathbf{v}_0 \end{bmatrix} \\ 0 \end{bmatrix} + \eta_v \cdot \sqrt{\Delta t} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (29)$$

The covariance is defined by (25) and the transition matrix's Jacobian is defined by (26) and (28), deleting the rows and columns that correspond to the angular velocity.

### B. Constant turn model with known turn rate

This dynamic model is derived from the standard curvilinear-motion model kinematics of a target moving in the horizontal plane [8]

$$\begin{cases} \dot{\psi} = a_n(t)/v(t) \\ \dot{p}_x = v(t) \cos \psi(t) \\ \dot{p}_y = v(t) \sin \psi(t) \\ \dot{v}(t) = a_t(t) \end{cases}, \quad (30)$$

where  $p_x$  and  $p_y$  are the target position in Cartesian coordinates,  $v$  is the vehicle's linear velocity,  $\psi$  is the angle that the vehicle does in relation to a direction of reference and  $a_t$  and  $a_n$  are the target tangential and normal accelerations in the horizontal plane.

This model is fairly general; it takes in account along and cross accelerations. Thus, it can be reduced to some special cases. The interesting particular case in this application is a constant speed and constant turn rate motion. The normal and tangential acceleration are then constant and zero ( $a_n = a_{n0} = \text{cte}$ ,  $a_n \neq 0$ ), respectively. Then, the first equation of (30) can be simplified to

$$\dot{\psi} = \omega, \quad (31)$$

where  $\omega$  is the vehicle's angular velocity.

It follows from (30) and (31) that such circular motion is described by

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}(t) & -\omega \dot{y}(t) & \dot{y}(t) & \omega \dot{x}(t) \end{bmatrix}^T + \mathbf{G}\mathbf{w}(t) \quad (32)$$

where  $\mathbf{x} = \begin{bmatrix} p_x & \dot{p}_x & p_y & \dot{p}_y \end{bmatrix}^T$  and  $\mathbf{G}\mathbf{w}(t)$  are the stochastic accelerations that can represent some uncertainty in some state variables and allow variations in the state, defined by

$$\mathbf{G} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T \quad \mathbf{w}(t) = \begin{bmatrix} w_x & w_y \end{bmatrix}^T \quad (33)$$

The discrete-time equivalent written in the form

$$\mathbf{x}_{k+1} = \phi_k \mathbf{x}_k + \mathbf{w}_k, \quad (34)$$

yields to

$$\phi_k = e^{\mathbf{F}\Delta t} = \begin{bmatrix} 1 & \frac{\sin \omega \Delta t}{\omega} & 0 & -\frac{1 - \cos \omega \Delta t}{\omega} \\ 0 & \cos \omega \Delta t & 0 & -\sin \omega \Delta t \\ 0 & \frac{1 - \cos \omega \Delta t}{\omega} & 1 & \frac{\sin \omega \Delta t}{\omega} \\ 0 & \sin \omega \Delta t & 0 & \cos \omega \Delta t \end{bmatrix}, \quad (35)$$

$$\mathbf{w}_k = \begin{bmatrix} \frac{\sin \omega \Delta t}{\omega} & -\frac{1 - \cos \omega \Delta t}{\omega} \\ \cos \omega \Delta t & -\sin \omega \Delta t \\ \frac{1 - \cos \omega \Delta t}{\omega} & \frac{\sin \omega \Delta t}{\omega} \\ \sin \omega \Delta t & \cos \omega \Delta t \end{bmatrix} \begin{bmatrix} \eta_x \\ \eta_y \end{bmatrix}, \quad (36)$$

where  $\eta_x$  and  $\eta_y$  are random variables that represent unknown accelerations.

The covariance matrix can be calculated by

$$\mathbf{Q}_k = \int_{t_k}^{t_{k+1}} \int_{t_k}^{t_{k+1}} \phi(t_{k+1}, \xi) \mathbf{G}(\xi) E[\mathbf{w}(\xi) \mathbf{w}^T(\tau)] \mathbf{G}^T(\tau) \phi^T(t_{k+1}, \tau) d\xi d\tau, \quad (37)$$

and the resulting expression is omitted due to lack of space.

### C. Observability analysis

The system's observability is a very important characteristic, and only this can guarantee a good estimation performance. Thus it should be verified that this characteristic is satisfied by the tested models.

Observability is a measure for how well internal states of a system can be inferred by knowledge of its external measured quantities. Formally, a system is said to be observable if, for any possible sequence of state and control vectors, the current state can be determined in finite time using only the outputs. A convenient way to verify system's observability is calculating the *rank* of the observability matrix, defined in (38).

$$\mathbf{O}(k_0, k_f) = \begin{bmatrix} \mathbf{H}_{k_0} \\ \mathbf{H}_{k_0+1} \phi(k_0+1, k_0) \\ \dots \\ \mathbf{H}_{k_f-1} \phi(k_f-1, k_0) \end{bmatrix} \quad (38)$$

For a system with  $n$  states, the system is observable only if the rank of the observability matrix is also  $n$ .

#### 1) Conclusions

The models were tested for various paths described by the vehicle and conclusions about the models' observability were taken. The rank of the observability matrixes obtained are resumed in TABLE I. and TABLE II. for the various test conditions and for both models.

TABLE I. OBSERVABILITY ANALYSIS FOR CIRCULAR MOTION STATE MODEL

Test conditions	$rank \mathbf{O}(k_0, k_f)$
	<b>Circular motion state model</b>
$V=0 \text{ m.s}^{-1}$	3
$\omega=0,1 \text{ rad.s}^{-1}$   $v=0,01 \text{ m.s}^{-1}$	4
$\Delta t = 1 \times 10^{-3} \text{ s}$	4
$v=5 \text{ m.s}^{-1}, \omega=0,07 \text{ rad.s}^{-1}, \Delta t = 1 \text{ s}$	5

Analyzing the results presented on TABLE I. it is verified that the model losses observability in some tested situations. When the vehicle's linear velocity is null, only three state variables are observable. Those variables are the position and the linear velocity, while the angular velocity and the vehicle's direction don't cause any variation on the observations. Physically, the sensors couldn't detect any variation on the vehicle's state if it spinning. The vehicle's state can't be determined on these conditions. If the vehicle's turning rate is greater than the linear velocity, the same phenomenon happens.

If the sampling time converges to zero, the system losses observability. This happens because between two consecutive observations the state has a minor variation, so the system losses observability.

TABLE II. OBSERVABILITY ANALYSIS FOR CONSTANT TURN MODEL WITH KNOWN TURN RATE

Test conditions	$rank \mathbf{O}(k_0, k_f)$
	<b>Constant turn model with known Turn Rate</b>
$V=0 \text{ m.s}^{-1}$	4
$\omega=0,1 \text{ rad.s}^{-1}$   $v=0,0001 \text{ m.s}^{-1}$	4
$\Delta t = 1 \times 10^{-15} \text{ s}$	2
$v=5 \text{ m.s}^{-1}, \omega=0,07 \text{ rad.s}^{-1}, \Delta t = 1 \text{ s}$	4

Analyzing TABLE II. it is verified that the constant turn model with known turn rate doesn't lose observability when the vehicle's linear velocity is null or when the angular velocity is greater than the linear velocity. These facts are justified because the state variables are only the vehicle's position and velocity. The observability is lost when the sampling time converges to zero, as explained in the previous situation.

#### D. Testing models

Both models were simulated considering that the vehicle was describing a circular path as shown in Figure 2.

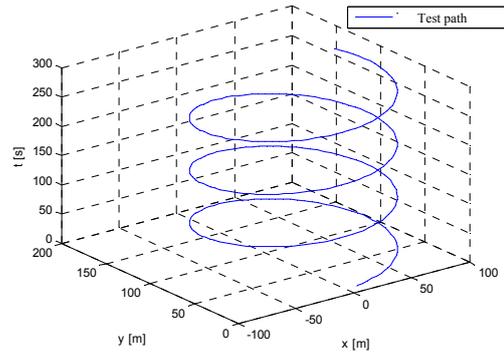


Figure 2. Test path used to perform the models simulation.

Monte Carlo simulations were performed in order to evaluate the performance of the algorithms' estimation. The square mean error for the considered noise standard deviations are summarized in TABLE III.

Analyzing the results presented in TABLE III. it can be concluded that the circular motion state model has better performance than the constant model with known turn rate for all noise conditions tested. The performance of the estimation algorithms became worse with higher noise standard deviation. It is interesting to notice that the square mean error of both estimation algorithms get closer for higher noise standard deviations.

TABLE III. SQUARE MEAN ERROR OBTAINED FOR THE CONSIDERED NOISE VARIANCE FOR BOTH STATE MODELS

Noise Variance		Circular motion state model		Constant model with known Turn Rate	
distance [m <sup>2</sup> ]	angle [(°) <sup>2</sup> ]	px error	py error	px error	py error
0,01 <sup>2</sup>	0,01 <sup>2</sup>	5.275E-03	4.102E-03	8.059E-02	6.366E-02
0,05 <sup>2</sup>	0,05 <sup>2</sup>	1.518E-02	1.345E-02	1.865E-01	1.420E-01
0,1 <sup>2</sup>	1 <sup>2</sup>	9.813E-02	6.894E-02	4.033E-01	3.196E-01
1 <sup>2</sup>	2 <sup>2</sup>	2.242E-01	1.727E-01	6.084E-01	5.563E-01
3 <sup>2</sup>	2 <sup>2</sup>	3.621E-01	4.414E-01	6.643E-01	6.233E-01
4 <sup>2</sup>	3 <sup>2</sup>	4.929E-01	5.273E-01	7.388E-01	6.774E-01
4 <sup>2</sup>	4 <sup>2</sup>	5.665E-01	5.280E-01	7.921E-01	7.049E-01
5 <sup>2</sup>	4 <sup>2</sup>	6.704E-01	6.435E-01	8.098E-01	7.275E-01
5 <sup>2</sup>	5 <sup>2</sup>	7.595E-01	6.721E-01	8.761E-01	7.656E-01
5 <sup>2</sup>	6 <sup>2</sup>	9.977E-01	8.526E-01	9.458E-01	8.065E-01
6 <sup>2</sup>	6 <sup>2</sup>	9.320E-01	8.383E-01	9.673E-01	8.337E-01

#### V. NON-LINEAR ESTIMATION PERFORMANCE

The problem of estimating the state variables from a discrete non-linear time-varying system with discrete non-linear observations in additive Gaussian white noise arises in many applications nowadays. Since the estimation algorithms developed for this purpose are rarely optimal, a very important concern is determining the lower bound of estimation and comparing with a given algorithm's performance to see if the filter is adequate or to determine if seeking a more effective

algorithm is worthwhile. A powerful result to determinate the best performance possible in estimation is the Cramér-Rao inequality. Defining  $\mathbf{P}$  to be the estimation error covariance matrix corresponding to any unbiased estimator, then the inequality can be stated as

$$\mathbf{P} \geq \mathbf{P}^* = J^{-1} \quad (39)$$

where  $J$  is the *Fisher* information matrix and  $\mathbf{P}^*$  defines the estimation lower bound.

#### A. Cramér-Rao Lower Bound

It is assumed that  $x$  is an unknown deterministic parameter and  $\mathbf{z}$  are the observations. The observations are corrupted with noise and have a probability density function (pdf)  $p(\mathbf{z}; x)$  known. If the pdf satisfies the regularity condition [7]

$$E \left[ \frac{\partial \ln p(\mathbf{z}; x)}{\partial x} \right] = 0, \quad \forall x \quad (40)$$

the variance of any unbiased estimator must satisfy

$$\text{var}(\hat{x}) \geq \frac{1}{-E \left[ \frac{\partial^2 \ln p(\mathbf{z}; x)}{\partial x^2} \right]} \quad (41)$$

where the denominator is defined as the *Fisher* matrix, (42).

$$J(x) = -E \left[ \frac{\partial^2 \ln p(\mathbf{z}; x)}{\partial x^2} \right] \quad (42)$$

Thus, the best performance for an estimator  $\hat{x}$  is given by

$$\text{var}(\hat{x}) = J(x)^{-1}. \quad (43)$$

The Cramér-Rao theory presented until here is almost a general case. For the specific case studied here a deeper analysis will be performed based in Taylor's method. Assuming that the system is written in the form presented in (1) and the system is not corrupted with any process noise  $\mathbf{w}_k = \mathbf{0}$ , the pdf  $p(\mathbf{Z}_k | \mathbf{X}_k)$  can be defined as [10], [11], [13]

$$p(\mathbf{Z}_k | \mathbf{X}_k) = \frac{1}{(2\pi)^{n/2} |\mathbf{S}_0|^{n/2}} \exp \left\{ -1/2 (\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T \mathbf{S}_0^{-1} (\mathbf{x}_0 - \hat{\mathbf{x}}_0) \right\} \cdot \prod_{k=1}^K \frac{1}{(2\pi)^{m/2} |\mathbf{R}_k|^{m/2}} \exp \left\{ -1/2 (\mathbf{z}_k - h(\mathbf{x}_k))^T \mathbf{R}_k^{-1} (\mathbf{z}_k - h(\mathbf{x}_k)) \right\} \quad (44)$$

where  $\hat{\mathbf{x}}_0 \sim N(\mathbf{x}_0, \mathbf{S}_0)$ ,  $\mathbf{Z}_k = \{\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_k\}$  and  $\mathbf{X}_k = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k\}$ . Taking the logarithm of (44) yields

$$-\ln p(\mathbf{Z}_k | \mathbf{X}_k) = cte + \frac{1}{2} (\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T \mathbf{S}_0^{-1} (\mathbf{x}_0 - \hat{\mathbf{x}}_0) + \frac{1}{2} \sum_{k=1}^K (\mathbf{z}_k - h(\mathbf{x}_k))^T \mathbf{R}_k^{-1} (\mathbf{z}_k - h(\mathbf{x}_k)) \quad (45)$$

Taking the expectation of the second partial of (45) in accordance with (42) yields

$$J_k = \left( \frac{\partial \mathbf{x}_0}{\partial \mathbf{x}_k} \right)^T \mathbf{S}_0^{-1} \left( \frac{\partial \mathbf{x}_0}{\partial \mathbf{x}_k} \right) + \sum_{k=1}^K \left( \frac{\partial h_k}{\partial \mathbf{x}_k} \right)^T \mathbf{R}_k^{-1} \left( \frac{\partial h_k}{\partial \mathbf{x}_k} \right). \quad (46)$$

Defining  $\mathbf{H}_k$  as described below

$$\mathbf{H}_k \triangleq \left. \frac{\partial h(\mathbf{x}_k, k)}{\partial \mathbf{x}_k} \right|_{\mathbf{x}_k}, \quad (47)$$

equation (46), by the chain rule of partial differentiation, yields

$$J_k = \left( \frac{\partial \mathbf{x}_0}{\partial \mathbf{x}_k} \right)^T \mathbf{S}_0^{-1} \left( \frac{\partial \mathbf{x}_0}{\partial \mathbf{x}_k} \right) + \sum_{k=1}^K \left( \frac{\partial \mathbf{x}_k}{\partial \mathbf{x}_k} \right)^T \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \left( \frac{\partial \mathbf{x}_k}{\partial \mathbf{x}_k} \right). \quad (48)$$

Defining

$$\frac{\partial \mathbf{x}_k}{\partial \mathbf{x}} = \frac{\partial \boldsymbol{\phi}_k}{\partial \mathbf{x}}, \quad (49)$$

equation (48) can be written in a recursive form

$$J_k = \left[ \left( \frac{\partial \boldsymbol{\phi}_k}{\partial \mathbf{x}} \Big|_{\mathbf{x}_k} \right) J_{k-1} \left( \frac{\partial \boldsymbol{\phi}_k}{\partial \mathbf{x}} \Big|_{\mathbf{x}_k} \right)^T \right]^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k. \quad (50)$$

In terms of  $\mathbf{P}^*$  is obtained

$$\left( \mathbf{P}_k^* \right)^{-1} = \left[ \left( \frac{\partial \boldsymbol{\phi}_k}{\partial \mathbf{x}} \Big|_{\mathbf{x}_k} \right) \mathbf{P}_{k-1}^* \left( \frac{\partial \boldsymbol{\phi}_k}{\partial \mathbf{x}} \Big|_{\mathbf{x}_k} \right)^T \right]^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k. \quad (51)$$

In a concluding remark, it can be said that

$$\mathbf{P}_{\text{estimator}, k} \geq \mathbf{P}_k^* \quad (52)$$

#### B. Posterior Cramér-Rao Bound

The Posterior Cramér-Rao Bound (PCRB) can provide a mean-square error lower bound for the discrete non-linear filtering problem. This lower bound is applicable to multidimensional non-linear, possibly non-Gaussian, dynamical systems and is more general than other approaches described in literature [12]. This approach can deal with process noise, so the system must be written in the form described by (1).

The *Fisher* matrix defined as

$$J = E \left[ \left( \frac{\partial \ln p(\mathbf{X}_k, \mathbf{Z}_k)}{\partial \mathbf{x}} \right) \left( \frac{\partial \ln p(\mathbf{X}_k, \mathbf{Z}_k)}{\partial \mathbf{x}} \right)^T \right] \quad (53)$$

that is, simplifying the notation, modified to

$$J = -E \left[ \nabla_{\mathbf{x}} \left\{ \left( \nabla_{\mathbf{x}} \ln p(\mathbf{X}_k, \mathbf{Z}_k) \right)^T \right\} \right], \quad (54)$$

where  $\nabla_{\mathbf{x}} = \left[ \frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_k} \right]$ . In this case both  $\mathbf{Z}_k$  and  $\mathbf{X}_k$  are

random variables, instead of  $\mathbf{X}_k$  being deterministic as happened in Cramér-Rao Lower Bound (CRLB) theory. So, the pdf is defined as

$$p(\mathbf{X}_k, \mathbf{Z}_k) = p(\mathbf{x}_0) \prod_{i=1}^k p(\mathbf{z}_i | \mathbf{x}_i) \prod_{j=1}^k p(\mathbf{x}_j | \mathbf{x}_{j-1}). \quad (55)$$

In [12] is shown that the *Fisher* matrix in a recursive form is given by

$$J_{k+1} = D_k^{22} - D_k^{21} (J_k + D_k^{11})^{-1} D_k^{12}, \quad (56)$$

where, in the general case,

$$D_k^{11} = E \left[ -\nabla_{\mathbf{x}_k} \nabla_{\mathbf{x}_k}^T \ln p(\mathbf{x}_{k+1} | \mathbf{x}_k) \right], \quad (57)$$

$$D_k^{12} = E \left[ -\nabla_{\mathbf{x}_k} \nabla_{\mathbf{z}_{k+1}}^T \ln p(\mathbf{x}_{k+1} | \mathbf{x}_k) \right] = \left[ D_k^{21} \right]^T, \quad (58)$$

$$D_k^{22} = E \left[ -\nabla_{\mathbf{z}_{k+1}} \nabla_{\mathbf{z}_{k+1}}^T \ln p(\mathbf{x}_{k+1} | \mathbf{x}_k) \right] + E \left[ -\nabla_{\mathbf{z}_{k+1}} \nabla_{\mathbf{z}_{k+1}}^T \ln p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}) \right]. \quad (59)$$

Simplifying with the approximation of white Gaussian noise, the pdfs' general expression defined in (55), taking the logarithm, are given by

$$-\ln p(\mathbf{x}_{k+1} | \mathbf{x}_k) = cte_1 + \frac{1}{2} (\mathbf{x}_{k+1} - \phi_k(\mathbf{x}_k))^T \mathbf{Q}_k^{-1} (\mathbf{x}_{k+1} - \phi_k(\mathbf{x}_k)), \quad (60)$$

$$-\ln p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}) = cte_2 + \frac{1}{2} (\mathbf{z}_{k+1} - h(\mathbf{x}_{k+1}))^T \mathbf{R}_{k+1}^{-1} (\mathbf{z}_{k+1} - h(\mathbf{x}_{k+1})). \quad (61)$$

The equations given by (57), (58), and (59) are then simplified to

$$D_k^{11} = \left[ \nabla_{\mathbf{x}_k} \phi_k(\mathbf{x}_k) \right]^T \mathbf{Q}_k^{-1} \left[ \nabla_{\mathbf{x}_k} \phi_k(\mathbf{x}_k) \right], \quad (62)$$

$$D_k^{12} = - \left[ \nabla_{\mathbf{x}_k} \phi_k(\mathbf{x}_k) \right]^T \mathbf{Q}_k^{-1} = \left[ D_k^{21} \right]^T, \quad (63)$$

$$D_k^{22} = \mathbf{Q}_k^{-1} + \left[ \nabla_{\mathbf{x}_{k+1}} h(\mathbf{x}_{k+1}) \right]^T \mathbf{R}_{k+1}^{-1} \left[ \nabla_{\mathbf{x}_{k+1}} h(\mathbf{x}_{k+1}) \right]. \quad (64)$$

The *Fisher* matrix previously given by (56) is simplified to

$$J_{k+1} = \left( \left[ \nabla_{\mathbf{x}_k} \phi_k(\mathbf{x}_k) \right]^T \mathbf{Q}_k^{-1} \left[ \nabla_{\mathbf{x}_k} \phi_k(\mathbf{x}_k) \right] + \mathbf{Q}_k \right)^{-1} + \left[ \nabla_{\mathbf{x}_{k+1}} h(\mathbf{x}_{k+1}) \right]^T \mathbf{R}_{k+1}^{-1} \left[ \nabla_{\mathbf{x}_{k+1}} h(\mathbf{x}_{k+1}) \right]. \quad (65)$$

Comparing (65) to (50), it is apparent that these two equations are identical, which leads to an interesting conclusion: Taylor's method for computing the CRLB can be easily made to compute an approximate PCRB by simply including the process noise covariance. It is important to notice that the

linearization in (47), (50), (51) and (65) are along the nominal path considered in simulation.

### C. Models performance – Results and conclusions

In order to analyze the performance of both estimation algorithms and compare it with the best that can be done using the optimal estimator, an analysis based on CRLB and PCRB was performed.

Firstly was performed a CRLB analysis, where the system was not corrupted with process noise. The results are summarized in TABLE IV. and represented graphically for a easier interpretation in Figure 3. and Figure 4.

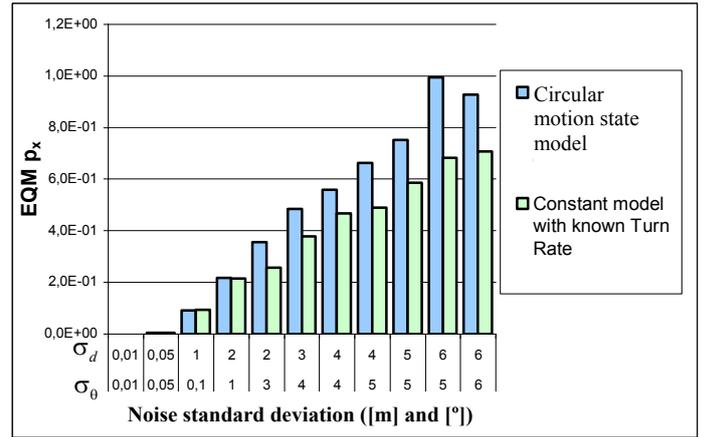


Figure 3. Estimation square mean error (EQM) for the coordinate x for both models.

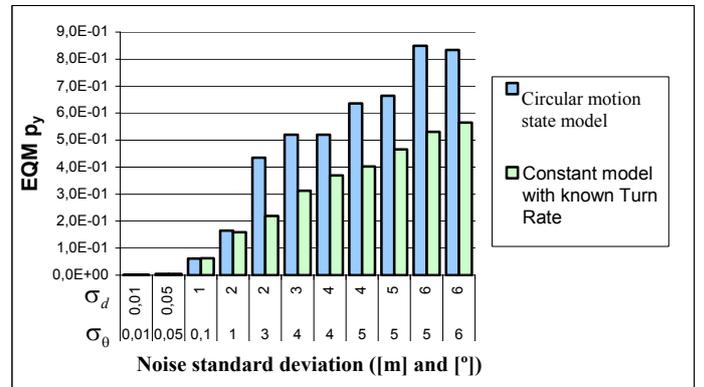


Figure 4. Estimation square mean error (EQM) for the coordinate y for both models.

It can be easily seen in TABLE IV. that the results can be improved further using a better estimator for both models. Although for the bigger noise variances considered in the analysis, the performance of the estimator appear to become closer to the optimal situation.

Analyzing Figure 3. and Figure 4. it can be concluded that the constant model with known turn rate has a better performance than the circular motion state model. It is also interesting to notice that the estimation performance is quite similar for both models for smaller noise variances. As the noise variance increases the estimation performance degrades.

Secondly was performed the PCRLB analysis. The results are summarized in TABLE V. and in Figure 5. and Figure 6.

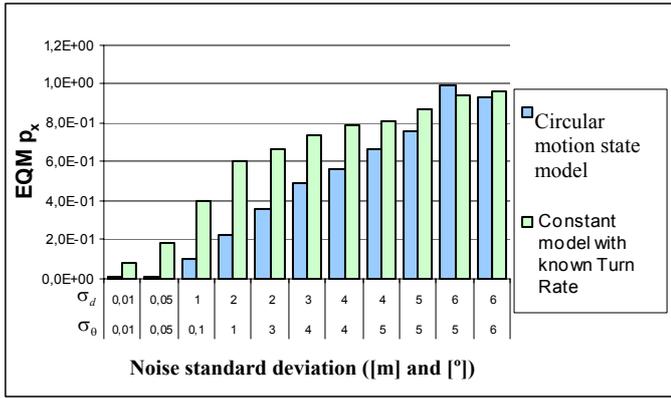


Figure 5. Estimation square mean error (EQM) for the coordinate x for both models.

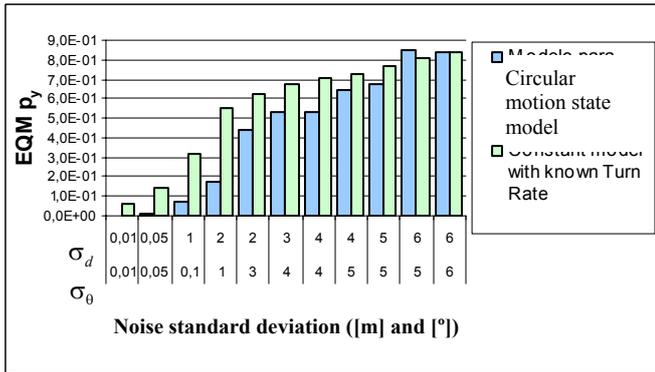


Figure 6. Estimation square mean error (EQM) for the coordinate y for both models.

Analyzing the results in TABLE V, it can be concluded that the algorithms' performance can be improved further with the help of a better estimator.

The Figure 5. and Figure 6. shows that in this situation, the circular motion state model has a better performance than the constant model with known turn rate almost for all noise variances considered in the simulation. It is also easily seen that the algorithms' performance become closer for higher noise variances. As happened before the performance of both algorithms degrades with the noise variance increase.

It is also interesting to see the periodic evolution of the square mean error in the optimal situation represented in Figure 7. and Figure 8. This evolution is due to the circular motion performed by the vehicle, see Figure 2.

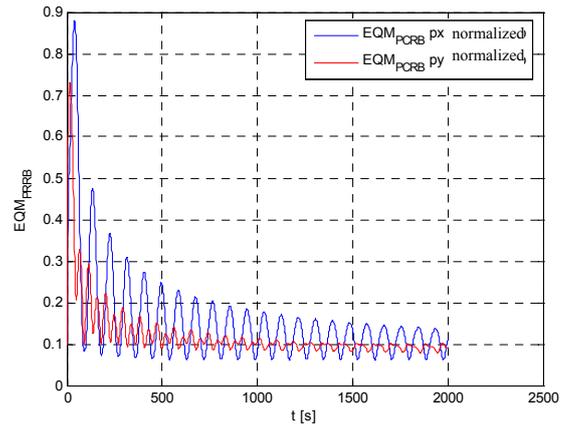


Figure 7. Square mean error evolution for the circular motion state model using the optimal estimator.

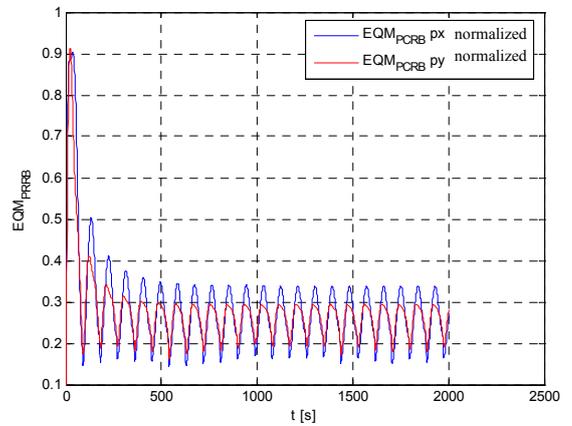


Figure 8. Square mean error evolution for the constant model with known turn rate using the optimal estimator.

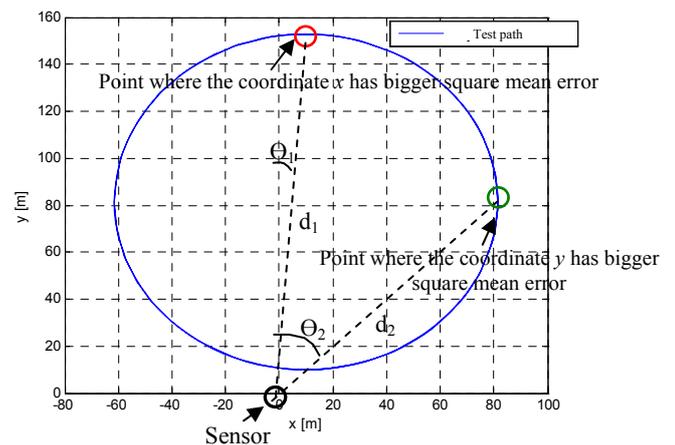


Figure 9. Circular path performed by the vehicle and points where the estimator has bigger uncertainty in estimating the vehicle's position.

In Figure 9. is represented the circular path performed by the vehicle and the points where the optimal estimator has bigger square mean error in determining the vehicle's position. The coordinate x has bigger uncertainty where vehicle is far from the sensor in the coordinate y. Similar happens for the coordinate x. The bigger uncertainty for coordinate x happens

when the vehicle is far from the sensor in coordinate  $y$ . This fact is justified due to the noise that corrupts the bearing angle. The noise in the bearing angle provokes, for bigger distances, a bigger uncertainty in the vehicle's position estimation.

## VI. CONCLUSIONS

In this paper two different dynamic models for target tracking were presented. The performance of the estimation algorithms using both models presented taking advantage from an extended Kalman filter were studied. The results obtained show that the proposed algorithms can be improved further because their estimation square mean errors are relatively far from the optimal solution.

Comparatively, the circular motion state model has better performance than the constant model with known turn rate in the case where it was considered process noise in the system. In the case where the system did not have process noise, the constant model with known turn rate had better performance. Although it was interesting to see that the estimation performance of both model become closer as the noise variance increases in both situation referred in the previous sentence. The performance of both algorithms become worse as the noise variance increased. The analyses performed lead us to the conclusion that the uncertainty affecting the bearing angle had great influence in the algorithms' performance.

## REFERENCES

[1] M. Morgado. Sistema de Navegação Inercial com ajuda USBL. Relatório de Trabalho Final de Curso, Universidade Técnica de Lisboa – Instituto Superior Técnico, Setembro 2005.

[2] R.G. Brown and P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, third edition, 1997.

[3] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, third edition, 1991.

[4] J. Craig. *Introduction to Robotics, Mechanics and Control*. Addison-Wesley, New York, second edition, 1989.

[5] M. Athans. *Multiple-Model Adaptive Estimation (MMAE)*. MIT & IST October 17, 2001.

[6] P. M. Alves. Seguimento e predição da trajectória de alvos móveis em duas dimensões. Dissertação para obtenção do Grau de Mestre

Engenharia Electrotécnica e de Computadores, Universidade Técnica de Lisboa – Instituto Superior Técnico, Outubro de 2004.

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

[8] X. Rong Li, Vesselin P. Jilkov. Survey *Maneuvering Target Tracking*. IEEE Trans. on Aerospace and Electronic Systems, vol. 39, pp 1333-1364, October 2003.

[9] Wilson J. Rugh. *Linear System Theory*. Prentice-Hall, second edition, USA 1996.

[10] James H. Taylor. *The Cramér-Rao Estimation Error Lower Bound Computation for Deterministic Nonlinear Systems*. IEEE Trans. on Automatic Control, vol. AC-24, pp 343-344 April 1979.

[11] Thomas H. Kerr. *Status of CR-Like Lower Bounds for Nonlinear Filtering*. IEEE Trans. on Aerospace and Electronic Systems, vol. 25, pp 590-597, September 1989.

[12] Petr Tichavský, Carlos H. Muravchik, Arye Nehorai. *Posterior Cramér-Rao Bounds for Discrete-Time Nonlinear Filtering*. IEEE Trans. on Signal Processing, vol. 46, pp 1386-1396 May 1998.

[13] William F. Level. *Approximate Cramér-Rao Bound for Multiple Target Tracking*. In Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy, School of Electrical and Computer Engineering Georgia Institute of Technology, May 2006.

TABLE IV. SQUARE MEAN ERROR AND RESPECTIVE CRAMÉR-RAO LOWER BOUND FOR BOTH MODELS AND FOR THE CONSIDERED NOISE VARIANCE

Noise Variance		Circular motion state model				Constant model with known Turn Rate			
		Square mean error		Crámer-Rao Lower Bound		Square mean error		Crámer-Rao Lower Bound	
distance [m <sup>2</sup> ]	angle [(°) <sup>2</sup> ]	P <sub>x</sub> error	P <sub>y</sub> error	P <sub>x</sub> error	P <sub>y</sub> error	P <sub>x</sub> error	P <sub>y</sub> error	P <sub>x</sub> error	P <sub>y</sub> error
0,01 <sup>2</sup>	0,01 <sup>2</sup>	1.177E-03	9.366E-04	3.979E-04	2.927E-04	1.177E-03	9.366E-04	8.897E-04	6.546E-04
0,05 <sup>2</sup>	0,05 <sup>2</sup>	5.886E-03	4.684E-03	1.989E-03	1.464E-03	5.887E-03	4.684E-03	4.448E-03	3.273E-03
0,1 <sup>2</sup>	1 <sup>2</sup>	9.224E-02	6.181E-02	3.668E-02	1.731E-02	9.298E-02	6.201E-02	8.196E-02	3.868E-02
1 <sup>2</sup>	2 <sup>2</sup>	2.175E-01	1.647E-01	7.623E-02	4.393E-02	2.142E-01	1.593E-01	1.700E-01	9.800E-02
3 <sup>2</sup>	2 <sup>2</sup>	3.557E-01	4.352E-01	8.170E-02	7.185E-02	2.566E-01	2.192E-01	1.822E-01	1.604E-01
4 <sup>2</sup>	3 <sup>2</sup>	4.859E-01	5.206E-01	1.205E-01	9.895E-02	3.784E-01	3.131E-01	2.680E-01	2.205E-01
4 <sup>2</sup>	4 <sup>2</sup>	5.590E-01	5.202E-01	1.564E-01	1.120E-01	4.686E-01	3.697E-01	3.460E-01	2.487E-01
5 <sup>2</sup>	4 <sup>2</sup>	6.630E-01	6.361E-01	1.592E-01	1.259E-01	4.902E-01	4.025E-01	3.524E-01	2.797E-01
5 <sup>2</sup>	5 <sup>2</sup>	7.520E-01	6.645E-01	1.950E-01	1.391E-01	5.871E-01	4.663E-01	4.290E-01	3.076E-01
5 <sup>2</sup>	6 <sup>2</sup>	9.956E-01	8.498E-01	2.309E-01	1.534E-01	6.837E-01	5.306E-01	5.041E-01	3.370E-01
6 <sup>2</sup>	6 <sup>2</sup>	9.272E-01	8.335E-01	2.336E-01	1.661E-01	7.074E-01	5.654E-01	5.102E-01	3.658E-01

TABLE V. SQUARE MEAN ERROR AND RESPECTIVE POSTERIOR CRAMÉR-RAO BOUND FOR BOTH MODELS AND FOR THE CONSIDERED NOISE VARIANCE

Noise Variance		Circular motion state model				Constant model with known Turn Rate			
		Square mean error		Posterior Crámer- Rao Bound		Square mean error		Posterior Crámer- Rao Bound	
distance [m <sup>2</sup> ]	angle [(°) <sup>2</sup> ]	P <sub>x</sub> error	P <sub>y</sub> error	P <sub>x</sub> error	P <sub>y</sub> error	P <sub>x</sub> error	P <sub>y</sub> error	P <sub>x</sub> error	P <sub>y</sub> error
0,01 <sup>2</sup>	0,01 <sup>2</sup>	5.275E-03	4.102E-03	3.297E-04	3.482E-04	8.059E-02	6.366E-02	1.312E-02	9.592E-03
0,05 <sup>2</sup>	0,05 <sup>2</sup>	1.518E-02	1.345E-02	1.649E-03	1.741E-03	1.865E-01	1.420E-01	3.939E-02	2.908E-02
0,1 <sup>2</sup>	1 <sup>2</sup>	9.813E-02	6.894E-02	6.556E-03	5.180E-03	4.033E-01	3.196E-01	1.899E-01	1.070E-01
1 <sup>2</sup>	2 <sup>2</sup>	2.242E-01	1.727E-01	4.989E-02	4.538E-02	6.084E-01	5.563E-01	2.743E-01	2.042E-01
3 <sup>2</sup>	2 <sup>2</sup>	3.621E-01	4.414E-01	7.093E-02	8.029E-02	6.643E-01	6.233E-01	2.941E-01	2.765E-01
4 <sup>2</sup>	3 <sup>2</sup>	4.929E-01	5.273E-01	1.043E-01	1.159E-01	7.388E-01	6.774E-01	3.628E-01	3.301E-01
4 <sup>2</sup>	4 <sup>2</sup>	5.665E-01	5.280E-01	1.308E-01	1.380E-01	7.921E-01	7.049E-01	4.255E-01	3.571E-01
5 <sup>2</sup>	4 <sup>2</sup>	6.704E-01	6.435E-01	1.370E-01	1.505E-01	8.098E-01	7.275E-01	4.321E-01	3.827E-01
5 <sup>2</sup>	5 <sup>2</sup>	7.595E-01	6.721E-01	1.630E-01	1.678E-01	8.761E-01	7.656E-01	4.966E-01	4.104E-01
5 <sup>2</sup>	6 <sup>2</sup>	9.977E-01	8.526E-01	1.846E-01	1.874E-01	9.458E-01	8.065E-01	5.622E-01	4.390E-01
6 <sup>2</sup>	6 <sup>2</sup>	9.320E-01	8.383E-01	1.941E-01	2.045E-01	9.673E-01	8.337E-01	5.681E-01	4.636E-01