INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

An Overview on Target Tracking Using Multiple Model Methods

Joana Barbosa Bastos Gomes

Dissertation submitted for obtaining the degree of Master in

Electrical and Computer Engineering

Jury

Supervisor: Paulo Jorge Coelho Ramalho Oliveira
Co-Supervisor: Carlos Jorge Ferreira Silvestre

September 2008

Acknowledgements

First, I would like to thank Professor Paulo Oliveira and Professor Carlos Silvestre for their guidance, support and motivation throughout this work.

To all my friends from Instituto Superior Técnico, for all the moments spent during the academic life, especially Clarita, Beta and Jerónimo for all the help and encouragement during the last five years and to the Algarve group for making my summers even better.

To my girlfriends from Mira Rio, for all their unconditional friendship and ever-presence in my life.

To my boyfriend João Pedro who was always there, helping me not to stress myself, and for choosing the best spots for us to go.

To my big brother Alexandre whose path I have followed and whose advice I've always sought.

And, finally, to my mother and father for their unconditional love and their believe that I can accomplish anything I purpose myself to do.

Resumo

Esta dissertação tem como objectivo apresentar uma colecção de algoritmos multi-modelo (MM) capazes de efectuar o seguimento de um ou múltiplos objectos na presença de incertezas no movimento do objecto e na origem das observações. Ademais, esta dissertação apresenta um novo algoritmo MM para seguimento de múltiplos objectos.

A estimativa da posição de um objecto é baseada em estimadores MM (formados por bancos de filtros de Kalman) com a ajuda de observações de um sensor. São apresentadas as três gerações de estimadores MM juntamente com a descrição e simulação de um algoritmo representativo. Estas gerações são i) Autónomos, ii) Cooperativos e iii) Estrutura Variável. Os correspondentes algoritmos representativos descritos são o Minimum Mean Square Error Autonomous Multiple Model (MMSE-AMM), o Interacting Multiple Model (IMM) e o Likely Model-Set (LMS). Relativamente ao seguimento de múltiplos objectos, dois algoritmos são descritos e simulados. O primeiro combina o algoritmo IMM com o Joint Probabilistic Data Association Filter (JPDAF); o segundo é um algoritmo novo baseado nos algoritmos LMS e JPDAF.

O MMSE-AMM é o mais simples e fraco dos algoritmos para seguimento de um objecto; a complexidade e desempenho do IMM e LMS são semelhantes, mas o LMS tem mais espaço para melhorar e provavelmente irá suplantar o IMM (e o MMSE-AMM). No exemplo simulado na presença de múltiplos objectos, o algoritmo proposto LMS-JPDAF suplanta o desempenho do IMM-JPDAF na presença de incerteza na origem das observações, apesar de mostrar um pior seguimento dos objectos quando estes estão afastados.

Palavras-chave: Seguimento de Objectos, Estimação multi-modelo, Data Association

Abstract

The aim of this thesis is to present a collection of multiple model (MM) algorithms capable of single or multiple target tracking by solving one or both target motion and measurement origin uncertainties. Furthermore, this thesis introduces a new MM algorithm for multiple target tracking.

A single target position estimation is based on MM estimators (composed of Kalman filters banks) with the help of sensor measurements. The three MM generations are presented along with a description and simulation of a representative algorithm. These generations are: i) the Autonomous, ii) the Co-operative and iii) the Variable Structure. The correspondent representative algorithms described are the Minimum Mean Square Error Autonomous Multiple Model (MMSE-AMM), the Interacting Multiple Model (IMM) and the Likely Model-Set (LMS). Regarding the tracking in the presence of the multiple targets two algortihms are described and simulated. The first combines the IMM and Joint Probabilistic Data Association Filter (JPDAF) algorithm, the second is a new algorithm based on the LMS and on the JPDAF.

The MMSE-AMM is the simplest and poorer single target tracker; the IMM and LMS complexity and responses are similar but the LMS has further room for improvement and will probabilly surpass the IMM (and the MMSE-AMM) performance. On the multiple target example simulated, the proposed LMS-JPDAF surpassed the IMM-JPDAF tracking performance under measurement origin uncertainty, although provided a worse overall single target tracking when the targets are far apart.

Keywords: Target Tracking, Multiple Model Estimation, Data Association

Acronyms

2D	Two Dimensions
ATC	Air Traffic Control
AMM	Autonomous Multiple Model
CMM	Cooperating Multiple Model
CV	(nearly) Constant Velocity
CT	(nearly) Constant Turn
DA	Data Association
FSMM	Fixed Structure Multiple Model
IMM	Interacting Multiple Model
IMM-JPDAF	Interacting Multiple Model Joint Probabilistic Data Association Filter
JPDAF	Joint Probabilistic Data Association Filter
KF	Kalman Filter
LMS	Likely Model-Set
LMS-JPDAF	Likely Model-Set Joint Probabilistic Data Association Filter
MGS	Model-Group Switching
MJLS	Markov Jump-Linear System
MMSE	Minimum Mean Squared Error
MMSE-AMM	Minimum Mean Square Error Autonomous Multiple Model
MSE	Mean Squared Error
MTT	Multiple Target Tracking
RADAR	RAdio Detection And Ranging
RAMS	Recursive Adaptive Model-Set
SM	Single Model
SONAR	SOund Navigation and Ranging
STT	Single Target Tracking
VSIMM	Variable Structure Interacting Multiple Model
VSIMM-JPDAF	Variable Structure Interacting Multiple Model Joint Probabilistic Data Association Filter
VSMM	Variable Structure Multiple Model

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter gives a brief overview of the work developed in this thesis, beginning by describing the motivations behind it, section 1.1.

The subject of target tracking is not new but current advances in technology, namely the evolution on the sensory systems, as provided new means that allow the development of better and more robust algorithms. The current State-of-the-Art in relation to target tracking is presented in section 1.2, beginning by the ever-present single tracker Kalman Filter and ending with the recent proposed multiple target trackers.

An overview of this work scope is also presented in section 1.2. At the end of the chapter, in section 1.4, the thesis structure is provided.

1.1 Motivation

The problem of target tracking dates back as far as the eighteenth century, with the first attempts to determine the orbits of visible planets. More recent work, early 1960s, was mostly developed targeting military applications, such as ballistic missile defense, battlefield situational awareness and orbital vehicle tracking. Nowadays, the target tracking problem has an ever-growing number of civilian applications, ranging from traditional applications such as air traffic control and building surveillance to emerging applications such as supply chain management and wildlife tracking. Certain applications of target tracking have become relatively more important in the last few years. Mainly air traffic control (ATC), which has and will experience a large grow due to the increase of civilian aviation (see figures 1.1 and 1.2) and highway vehicle surveillance, driven by current interest in intelligent transportation systems and by the proliferation of surveillance systems and their increased sophistication.

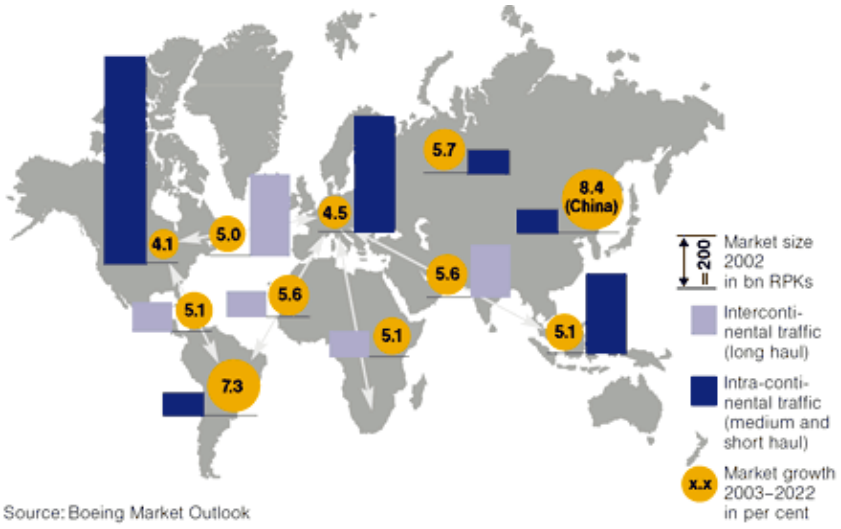


Figure 1.1: Market size 2002 in billions revenue passenger kilometers and yearly expected growth rates 2003-2022.

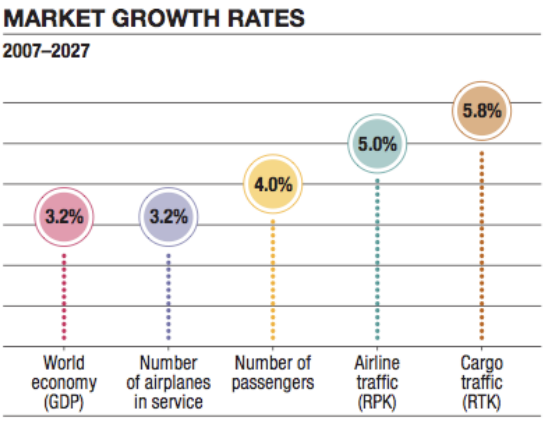


Figure 1.2: Expected market growth rates 2007-2027 (source: Boeing Market Outlook).

What these applications have in common, is that they intent to accurately and timely know the state of one or more moving targets based on a combination of the sensed data and the target history.

A major issue in target tracking design is thus the solving of two discrete-valued uncertainties, the target motion uncertainty and the measurement origin uncertainty.

The target motion uncertainty is present when a target undergoes a known or unknown maneuver during an unknown time period. In general, it is only possible to account target motion uncertainty by having different motion models. Note that the target motion uncertainty can also reveal itself by target nonlinearity, which will not be addressed in this thesis.

The measurement origin uncertainty is present when the measurements from the sensory system can have been originated by an external source, such as clutter, false alarms or neighboring targets, for example. In many situations the measurement origin uncertainty has far more impact in the tracking performance than the noise associated with the measurements or the target motion uncertainty.

Solving the measurement origin uncertainty has gain specially attention in current years, due to the rising of more complex and complete sensory systems such as RAdio Detection And Ranging (RADAR), infrared, laser RADAR and SOund Navigation and Ranging (SONAR), capable of observing diverse sources, including the targets of interested and background noise (see figure 1.3).

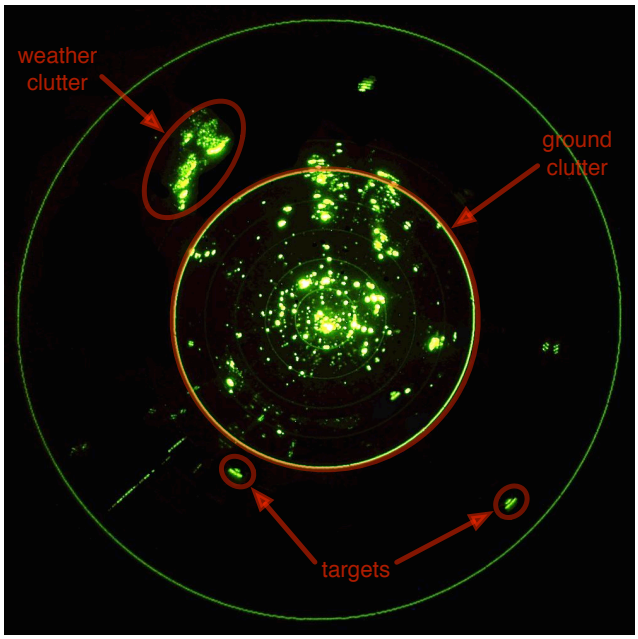


Figure 1.3: Plan position indicator screen of an ATC-radar with targets and clutter.

1.2 State-of-the-Art

The ubiquitous Kalman Filter [14] is the most widely used filtering technique and is the basis for most of the more complex target tracking estimators. The Kalman Filter is an optimal single target tracker if there is not target motion uncertainty nor measurement origin uncertainty.

The Multiple Model methods are the main stream approach in single target tracking under motion uncertainty and in the absence of measurement origin uncertainty. Basically, these methods resolve the target motion uncertainty by using multiple models at a time for a maneuvering target, being able to explain a larger number of target motions at each time. Numerous techniques have been developed to resolve the target motion uncertainty, amongst the more popular, there are the Minimum Mean-Square Error Autonomous Multiple Model [22] and the Interacting Multiple Model algorithm [11].

Furthermore, there are two main Multiple Model algorithm approaches, the Fixed Structure and the Variable Structure. The first has received a lot of attention from the scientific community, and little more there is to be improved, on the other hand, the latter is relatively new and has appeared has an alternative to upgrade the Fixed Structure Multiple Model algorithms. In the survey set [4], [7], [5], [6], [8] and [10], an interesting and almost complete study of all the main Variable Structure Multiple Model algorithms is made.

The Multiple Target Tracking problem extends the single target tracking scenario to a situation where the number of targets may not be known and varies with time, leading to the presence of measurement origin uncertainty in addition to the target motion uncertainty. Besides the need to estimate the targets current state, the identities of the targets may need to be known. The usual method to solve this problem is to assign a single target tracker to each target and use a data association technique to assign the correct measurement to each target. Numerous techniques have been developed to resolve the measurement origin uncertainty, such as the Nearest Neighbour Standard Filter [23], the Joint Probabilistic Data Association Filter [1], and the Multiple Hypothesis Tracking Filter [1].

1.3 Objectives

The aim of this thesis is to present a collection of tools that are capable of target tracking by solving one or both target motion and measurement origin uncertainties. This thesis represents an evolution from the more accepted and more documented tracking algorithms to the more recent ones, from the more simple to the more complex concepts in target tracking. Furthermore, during the thesis development, a new algorithm for Multiple Target Tracking arose naturally, by combining two of the studied approaches: the Likely Model-Set and the Joint Probabilistic Data Filter.

1.4 Contents

The outline of this thesis is as follows. The modeling of the target and sensory system is made in chapter 2. The Kalman Filter, viewed as the elemental filter or block of the Multiple Model estimators, is described in chapter 3. The three generations of Multiple Model methods are surveyed in chapter 4. This generations are: the autonomous, the cooperative and the variable structure multiple models. For all the three generations a representative algorithm is simulated. Furthermore, at the end of the chapter all three algorithms performances are discussed and compared. The tracking of multiple targets is addressed in chapter 5, where two alternative algorithms are presented. The first, is a well

documented algorithm whereas the second one was developed in the thesis context. Both this algorithms performances are compared. The final chapter, chapter 6, summarizes the work presented in this thesis and outlines future research based on the work developed.

Chapter 2

The Target and Sensor Models

This chapter addresses the importance of a correct modeling of the target and sensor in the problem of target tracking. In the first section (section 2.1), the system composed of target and sensor is described as a general state-space system.

In section 2.2, two possible models to explain the target motion are described. The target can have two types of motions: non-maneuvering motion (described in subsection 2.2.1) and maneuvering motion (described in subsection 2.2.2). In subsection 2.2.3 these two models are incorporated in a hybrid target model able to describe the multiple motions of the target.

In section 2.3, the sensor model is described along with the necessary coordinate transformation and linearization (subsection 2.3.1). In subsection 2.3.2, the sensor model is extended to the the multiple target tracking problem.

Finally in section 2.4 the difference between the terms Mode and Model are discussed.

2.1 The State Space Model

For the successful tracking of a moving target it is essential to extract the maximum useful information about the target state from the available observations. Both good models to describe the target dynamics and sensor will certainly help this information extraction. As the knowledge of information on the target's kinematics and sensor characteristics are generally known, most of the tracking algorithms base their performance on the a priori defined mathematical model of the target which are assumed to be sufficiently accurately. This section, addresses the problem of describing the target motion model and establishes a good compromise between accuracy and complexity.

The target is treated as a pontual mass and is described by a linear state-space model in the form

$$x_{k+1} = F_k x_k + G_k u_k + E_k w_k \tag{2.1}$$

and the measurement function has the form

$$z_k = H_k x_k + v_k \tag{2.2}$$

where $x_k \in \mathbb{R}^n$, $z_k \in \mathbb{R}^m$ and $u_k \in \mathbb{R}^p$ the target state,observation, and control input vectors, respectively, at the discrete time t_k; $w_k \in \mathbb{R}^q$ and $v_k \in \mathbb{R}^m$ are the process and measurement noise sequences with covariances $Q(t) \in \mathbb{R}^{q\times q}$ and $R(t) \in \mathbb{R}^{m\times m}$, respectively; and $F_k \in \mathbb{R}^{n\times n}$, $E_k \in \mathbb{R}^{n\times p}$, $G_K \in \mathbb{R}^{n\times q}$ and $H_k \in \mathbb{R}^{m\times n}$ are the possible time-varying discrete-time transition matrix, input gain, noise gain and measurement matrix, respectively.

The discrete-time model presented above is often obtained by discretizing the continuous-time model

$$\dot{x}(t) = A(t)x(t) + D(t)u(t) + B(t)w(t) \;\; x(t_0) = x_0 \tag{2.3}$$

$$z(t) = C(t)x(t) + v(t) \tag{2.4}$$

where $A(t)$, $D(t)$, $B(t)$ and $C(t)$ are the continuos-time equivalent to the F_k, E_K, G_k and H_k matrixes, respectively and $w(t)$ and $v(t)$ are the continuous-time process and measurement noise with covariances $Q(t)$ and $R(t)$, respectively.

Since the observations are usually only available at discrete time instants and the target motion is more accurately modeled by the continuous time equation, it is more appropriate to use the following continuous-discrete time form

$$\dot{x}(t) = A(t)x(t) + D(t)u(t) + B(t)w(t), \quad x(t_0) = x_0 \tag{2.5}$$

$$z_k = H_k x_k + v_k. \tag{2.6}$$

2.2 Mathematical Model for the Maneuvering Target

After addressing the importance of correct modeling the target and sensor, and introducing general equations for the continuous-time target system, in this section, the concrete model chosen to explain the target's motion is described.

Before presenting the model, the expressions for discretizing (sampling) of the continuous-time linear system (2.3), for a time-invariant continuous-time system with sampling rate Δt, using Euler discretization or Zero-Order Hold (ZOH) [12], [9], are

$$F_k \triangleq F(t_{k+1}, t_k) = F(t_{k+1} - t_k) = F(\Delta t) = e^{A\Delta t} \tag{2.7}$$

$$G_k \triangleq G(t_{k+1}, t_k) = \int_0^{\Delta t} e^{A(\Delta t - \tau)} D d\tau. \tag{2.8}$$

The process noise $w(t)$ is considered to be zero-mean and white Gaussian. The discretized process noise w_k retains the same characteristics

$$E[w_k] = 0, \quad E[w_k w_j'] = Q_k \delta_{kj} \tag{2.9}$$

with covariance Q_k given by

$$Q_k = \int_0^{\Delta t} e^{A(\Delta t - \tau)} B Q B' e^{A'(\Delta t - \tau)} d\tau. \tag{2.10}$$

The target is assumed to have a 2D horizontal motion model has a state described by

$$x = \begin{bmatrix} \mathrm{x} & \dot{\mathrm{x}} & \mathrm{y} & \dot{\mathrm{y}} \end{bmatrix}' \tag{2.11}$$

where (x, y) is the target position in Cartesian coordinates and $\dot{\mathrm{x}}$ and $\dot{\mathrm{y}}$ are the linear velocity of the target along the x-axis and the y-axis, respectively.

The target dynamics is described by

$$\begin{cases} \dot{\mathrm{x}}(t) = v(t) \cos \phi(t) \\ \dot{\mathrm{y}}(t) = v(t) \sin \phi(t) \\ \dot{v}(t) = a_t(t) \\ \dot{\phi}(t) = a_n(t)/v(t) \end{cases} \tag{2.12}$$

where v and ϕ are the linear velocity and (velocity) heading angle, respectively. The variable a_t and a_n are the target tangential and normal accelerations in the horizontal plane, respectively. See figure 2.1 for visual representation of this variables.

This model can explain several target motions including the special cases

- rectilinear, constant velocity motion $(a_n = 0, a_t = 0)$;

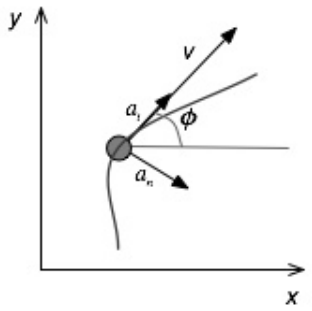


Figure 2.1: Geometry of 2D target motion.

- rectilinear, accelerated motion $(a_n = 0, a_t \neq 0)$;
- circular, constant speed motion $(a_n \neq 0, a_t = 0)$.

The two motions addressed in this thesis for modeling the target's dynamics are the first and last cases of the above list.

The first case represents the target in non-maneuver, its motion is (nearly) uniform with (nearly) constant linear velocity v. This model is commonly called Constant Velocity (CV) Model [20].

The last case above, with a constant and known a_n, is referred to as a Constant Turn (CT) Model with Known Turn Rate [20]. This model assume that the target moves with (nearly) constant speed v and (nearly) constant angular velocity (or turn rate) ω. The constant value is assumed to be known leading to a four-dimensional state vector.

2.2.1 Constant Velocity Model

From (2.12) for $a_n = 0$ and $a_t = 0$; and using the form of (2.3), the target state vector is given by

$$\dot{x}(t) = A_{CV}x(t) + B_{CV}w(t) \tag{2.13}$$

$$A_{CV} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_{CV} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.14}$$

where $w(t)$ is white gaussian noise with covariance $E[w(t)w(\tau)'] = Q(t)\delta(t-\tau)$

$$Q(t) = \text{diag}(\sigma_x^2,\ \sigma_y^2). \tag{2.15}$$

The discrete-time equivalent of the above model is described by

$$x_{k+1} = F_{CV}x_k + w_k \tag{2.16}$$

where, using the equation (2.7), the discrete-time transition matrix is given by

$$F_{CV} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.17}$$

The covariance matrix of the discrete-time process noise sequence w_k, calculated using the expression (2.10), is

$$Q_k = \begin{bmatrix} 1/3(\Delta t)^3\sigma_x^2 & 1/2(\Delta t)^2\sigma_x^2 & 0 & 0 \\ 1/2(\Delta t)^2\sigma_x^2 & \Delta t\sigma_x^2 & 0 & 0 \\ 0 & 0 & 1/3(\Delta t)^3\sigma_y^2 & 1/2(\Delta t)^2\sigma_y^2 \\ 0 & 0 & 1/2(\Delta t)^2\sigma_y^2 & \Delta t\sigma_y^2 \end{bmatrix}. \tag{2.18}$$

2.2.2 Constant Turn Model with Known Turn Rate

From (2.12) for $a_n \neq 0$ and $a_t = 0$; and using the form of (2.3), the target state vector is given by

$$\dot{x}(t) = A_{CT}(\omega)x(t) + B_{CT}w(t) \tag{2.19}$$

$$A_{CT}(\omega) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\omega \\ 0 & 0 & 0 & 1 \\ 0 & \omega & 0 & 0 \end{bmatrix}, \quad B_{CT} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.20}$$

where $w(t)$ is white gaussian noise with covariance given by (2.15). If ω is known this model is linear.

The discrete-time equivalent of the above model can be obtained by

$$x_{k+1} = F_{CT}(\omega)x_k + w_k \tag{2.21}$$

where, using once again the equation (2.7), the discrete-time transition matrix is given by

$$F_{CT}(\omega) = \begin{bmatrix} 1 & \frac{\sin\omega\Delta t}{\omega} & 0 & -\frac{1-\cos\omega\Delta t}{\omega} \\ 0 & \cos\omega\Delta t & 0 & -\sin\omega\Delta t \\ 0 & \frac{1-\cos\omega\Delta t}{\omega} & 1 & \frac{\sin\omega\Delta t}{\omega} \\ 0 & \sin\omega\Delta t & 0 & \cos\omega\Delta t \end{bmatrix}. \tag{2.22}$$

Note that the discretization of this continuos model has an issue, since it can be time-variant, regarding which of the discrete-time value of the angular velocity ω should be used. The most popular way, and the one here applied, is to use ω_k. Other ways include the more stable method propose in [16], which uses ω_{k+1}, or adopt the mean value $\frac{1}{2}(\omega_k + \omega_{k+1})$ instead.

The variable w_k remains a white gaussian noise sequence with covariance given by (2.10), resulting in

the matrix defined by $Q_k = \begin{bmatrix} Q_1 & Q_2 & Q_3 & Q_4 \end{bmatrix}$, where

$$Q_1 = \begin{bmatrix} \frac{(3\,\omega\Delta t - 4\,\sin(\omega\Delta t) + 1/2\sin(2\omega\Delta t))\sigma_y{}^2}{2\omega^3} + \frac{(\omega\Delta t - 1/2\sin(2\omega\Delta t))\sigma_x{}^2}{2\omega^3} \\ -\frac{\left(2\,\cos(\omega\Delta t) - 1 - \cos(\omega\Delta t)^2\right)\sigma_y{}^2}{2\omega^2} + \frac{\sin(\omega\Delta t)^2\sigma_x{}^2}{2\omega^2} \\ \frac{\left(2\,\cos(\omega\Delta t) - 1 - \cos(\omega\Delta t)^2\right)\sigma_y{}^2}{2\omega^3} + \frac{\left(\cos(\omega\Delta t)^2 - 2\,\cos(\omega\Delta t) + 1\right)\sigma_x{}^2}{2\omega^3} \\ \frac{(\omega\Delta t - 2\,\sin(\omega\Delta t) + 1/2\sin(2\omega\Delta t))\sigma_y{}^2}{2\omega^2} + \frac{(\omega\Delta t - 1/2\sin(2\omega\Delta t))\sigma_x{}^2}{2\omega^2} \end{bmatrix} \tag{2.23}$$

$$Q_2 = \begin{bmatrix} Q_1(2) \\ \frac{(\omega\Delta t - 1/2\sin(2\omega\Delta t))\sigma_y{}^2}{2\omega} + \frac{(1/2\sin(2\omega\Delta t) + \omega\Delta t)\sigma_x{}^2}{2\omega} \\ -\frac{(\omega\Delta t - 1/2\sin(2\omega\Delta t))\sigma_y{}^2}{2\omega^2} - \frac{(\omega\Delta t - 2\,\sin(\omega\Delta t) + 1/2\sin(2\omega\Delta t))\sigma_x{}^2}{2\omega^2} \\ \frac{\sin(\omega\Delta t)^2\sigma_x{}^2}{2\omega} - \frac{\sin(\omega\Delta t)^2\sigma_y{}^2}{2\omega} \end{bmatrix} \tag{2.24}$$

$$Q_3 = \begin{bmatrix} Q_1(3) \\ Q_2(3) \\ \frac{(\omega\Delta t - 1/2\sin(2\omega\Delta t))\sigma_y{}^2}{2\omega^3} + \frac{(3\,\omega\Delta t - 4\,\sin(\omega\Delta t) + 1/2\sin(2\omega\Delta t))\sigma_x{}^2}{2\omega^3} \\ \frac{\sin(\omega\Delta t)^2\sigma_y{}^2}{2\omega^2} + \frac{\left(\cos(\omega\Delta t)^2 - 2\,\cos(\omega\Delta t) + 1\right)\sigma_x{}^2}{2\omega^2} \end{bmatrix} \tag{2.25}$$

$$Q_4 = \begin{bmatrix} Q_1(4) \\ Q_2(4) \\ Q_3(4) \\ \frac{(1/2\sin(2\omega\Delta t) + \omega\Delta t)\sigma_y{}^2}{2\omega} + \frac{(\omega\Delta t - 1/2\sin(2\omega\Delta t))\sigma_x{}^2}{2\omega} \end{bmatrix}. \tag{2.26}$$

Unfortunately, this model requires the exact value of the angular velocity ω to be known a priori thus making it not usable in practical applications. A intuitive idea is to replace the above ω by its estimate however, this adds up to the already existing uncertainty of the target's state and may lead to unacceptably large errors into the system.

2.2.3 The Target as a Discrete-Time Hybrid System

In general, a target non-maneuvering and different maneuvering motions can only be explained by different motion models.

A natural solution to take advantage of the simpleness of the models described in sections 2.2.1 and 2.2.2, and still achieve good tracking results, is based on the use of multiple models with different fixed turn rates. The turn rate, or system behavior, at a time instant k is modeled by a discrete variable name mode state s_k.

The target system is hybrid, since it has both continuous and discrete components. Those that vary continuously are the base state $x(t)$ and those that may only jump are the mode state s_k.

In this approach, at each given time k the target can assume a mode state $s_k^{(i)}$, associated with an angular velocity ω_i. The mode s_k is considered to be in effect during the sampling period ending at k. The set of possible modes is the finite mode space $\mathbb{S}$. Note that the value of ω can be equal to zero, leading to the CV model.

The sequence of modes, and consequently the sequence of turn rates, is modeled as a Markov chain, which means that at each time step k its value may either jump or stay unchanged with transition probability, for a homogeneous Markov chain, defined by

$$P\{s_{k+1}^{(j)}|s_k^{(i)}\} = P\{\omega_{k+1} = \omega_j|\omega_k = \omega_i\} = \pi_{ij,k} = \pi_{ij} \quad \forall i, j, k. \tag{2.27}$$

The target model can be casted as a Markov Jump-Linear System (MJLS), i.e., a linear system whose parameters evolve with time according to a finite state Markov chain. The measurement equation will be omitted since the sensor has only discrete-time parameters and does not depend on the mode state. The discrete-time general representation of a hybrid system, where the discrete system components have a first order $\{s_k\}$-dependance, can be described by

$$x_{k+1} = F_k(s_k)x_k + w_k(s_k). \tag{2.28}$$

The above equations can be simplified in this context, where the system mode s_k is considered to be associated with an angular velocity ω_k, meaning that F_k is the only s_k-dependent factor of the system. Also the time-dependence of the transition matrix F_k can be dropped resulting in the following expressions

$$x_{k+1} = F(\omega_k)x_k + w_k \tag{2.29}$$

where

$$F(\omega_k) = \begin{cases} \lim\limits_{\omega_k \to 0} F_{CT}(\omega_k) = F_{CV} & \text{if } \omega_k = 0 \\ F_{CT}(\omega_k) & \text{otherwise.} \end{cases} \tag{2.30}$$

2.3 Mathematical Model for Sensory System

As was introduced in section 2.1 in the problem of target tracking it is essential to extract the most information about the target state from the observations, thus besides modeling the target it also is necessary to model the sensor.

The sensor described is a RADAR and is considered to be placed at the origin of the Cartesian axis, i.e., is placed in the coordinates $(x, y) = (0, 0)$, and the sensor coordinate system is polar providing in each measurement the range r and the bearing θ of the target (see figure 2.2).

The sensor measurements are modeled with the following additive noise

$$r = \bar{r} + v_r \tag{2.31}$$

$$\theta = \bar{\theta} + v_\theta \tag{2.32}$$

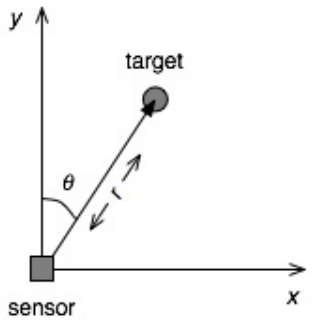


Figure 2.2: Sensor coordinate system.

where $x_k^p = (\bar{r}, \bar{\theta})$ are the nominal target position, in the sensor polar coordinates, and $v_k^p = (v_r, v_\theta)$ are the respective measurement errors, assumed to be zero-mean, Gaussian distributed and uncorrelated. Considering v_k^p the measurement error vector in polar coordinates at the time step k

$$v_k^p \sim \mathcal{N}(0, R_k^p) \tag{2.33}$$

with

$$R_k^p = \text{cov}(v_k^p) = \text{diag}(\sigma_r^2,\ \sigma_\theta^2) = R^p \ \ \forall k. \tag{2.34}$$

The sensor measurements in the sensor coordinate system is given by

$$z_k^p = \begin{bmatrix} r \\ \theta \end{bmatrix} = h(x_k) + v_k^p \tag{2.35}$$

where the error-free position of the target in polar coordinates is given by

$$x_k^c = h(x_k) = \begin{bmatrix} \bar{r}_k \\ \bar{\theta}_k \end{bmatrix} = \begin{bmatrix} \sqrt{\mathrm{x}_k^2 + \mathrm{y}_k^2} \\ f(\mathrm{y}_k, \mathrm{x}_k) \end{bmatrix} \tag{2.36}$$

where $f(\mathrm{y}_k, \mathrm{x}_k)$ is a four-quadrant inverse tangent.

At this point, there are two distinct coordinate systems, the Cartesian coordinate system of the target and the polar coordinate system of the sensor. Although there are several techniques to address the tracking problem in a multiple coordinate systems [19], the choice taken in this thesis is to track in Cartesian coordinates, therefore, there is the need to perform the conversion between the sensor coordinates to Cartesian coordinates.

2.3.1 Tracking in Cartesian Coordinates

The nominal position of the target in Cartesian coordinates corespondent to the pair $x_k^p = [r\ \ \theta]'$ is defined by $x_k^c = [\mathrm{x}\ \mathrm{y}]'$, which is a subvector of the actual state vector x. The correspondence polar-to-

Cartesian coordinates of the target position is exact and given by

$$x_k^c = \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \end{bmatrix} = \Phi(x_k^p) = \Phi(\bar{r}, \bar{\theta}) = \begin{bmatrix} \bar{r}\cos(\bar{\theta}) \\ \bar{r}\sin(\bar{\theta}) \end{bmatrix} \tag{2.37}$$

where $\Phi = h^{-1}$ is the polar-to-Cartesian transformation.

After the conversion above, the sensor model in Cartesian coordinates, can be modeled as

$$z_k = Hx_k + v_k^c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_k + v_k \tag{2.38}$$

where $x_k^c = Hx_k = [\mathrm{x\ y}]'$ is the nominal position of the target in Cartesian coordinates (a subvector of the actual state vector x) and v_k is the converted measurement noise into Cartesian coordinates. The measurement matrix is now time invariant, i.e., $H_k = H\ \forall k$.

The major advantage of the above expression is that the linear Kalman Filter (see chapter 3) can be applied for tracking, as the target dynamics are also linear.

The main challenge of the equation above is to obtain a linearized conversion of the measurement noise, since v_k^c in general is coupled across coordinates, is non-Gaussian and state dependent. Due to state dependency of v_k^c the model in equation (2.38) is actually non-linear which means that the Kalman Filter fails to deliver optimal results.

The alternative is also not perfect, as the techniques to track nonlinearity present only in the measurement noise and not in the measurement function are scarce and not deeply researched. The presence of this nonlinearity in v_k^c is actually much less significant then the presence of nonlinearity in the measurement function.

One of the approaches is to linearize the measurement noise is to perform a Taylor series expansion of $\Phi(x_k^p)$ around the noisy measurement z_k^p [19], omitting the time index k

$$x^c = \Phi(x^p) = \Phi(z^p - v^p) = \Phi(z^p) - J(z^p)v^p + \xi(v^p) \tag{2.39}$$

where $\xi(v^p)$ stands for higher order (≤ 2) terms and $J(z^p)$ is the Jacobian given by

$$J(z^p) = \frac{\partial \Phi}{\partial x^p}\Big|_{x^p = z^p} = \begin{bmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{bmatrix}. \tag{2.40}$$

Then the equation (2.38) can be written as

$$z_k = \Phi(z_k^p) = \Phi(x_k^p + v_k^p) = Hx_k + \underbrace{J(z_k^p)v_k^p + \xi(v_k^p)}_{v_k^c} \tag{2.41}$$

The most commonly applied approach is to treat v_k^c as a zero-mean sequence with covariance

$$R_k = J(z_k^p)R_k^p J(z_k^p)' \tag{2.42}$$

this is achieved by ignoring the higher order terms of the Taylor series expansion $\xi(v_k^p)$, i.e.,

$$z_k = \Phi(z_k^p) \approx Hx_k + \underbrace{J(z_k^p)v_k^p}_{v_k} \tag{2.43}$$

where v_k^c is approximated by the linearized version v_k. As stated in [17], the linearization described above is considered valid, i.e., the linearized v_k roughly retains the zero-mean and white sequence characteristic of v_k^c only if the following conditions are met

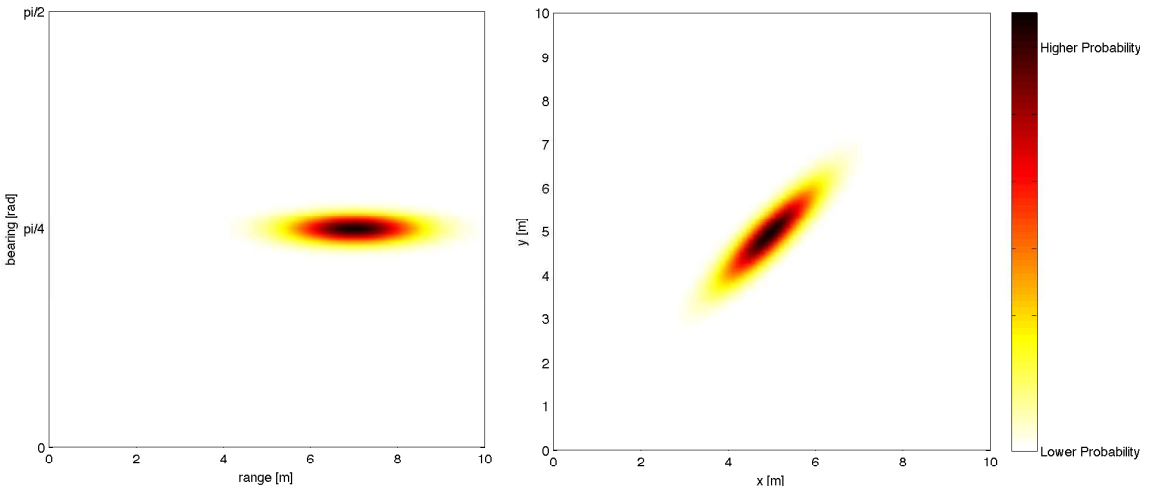
$$\frac{r\sigma_\theta^2}{\sigma_r} < 0.4, \quad \sigma_\theta < 0.4 \,\text{rad}. \tag{2.44}$$

Outside this conditions the noise sequence v_k reveals its state dependency, non zero-mean and non-white characteristics. Below in figure 2.3, is an example of coordinate conversion and what happens to the measurement probability density function from polar to Cartesian coordinates.

The distribution of the measurement data z_k (in polar or Cartesian coordinates) given the target position x_k (in polar or Cartesian coordinates) is calculated using

$$z_k \sim \mathcal{N}(x_k, R_k) \longrightarrow \mathcal{N}(z_k; x_k, R_k) = \frac{1}{2\pi\sqrt{\det(R_k)}} e^{-\frac{1}{2}(z_k - x_k)' R_k^{-1}(z_k - x_k)} \tag{2.45}$$

where R_k is measurement noise covariance in polar or Cartesian coordinates, depending on which coordinate system are the other variables.



(a) Polar coordinates.

(b) Cartesian coordinates.

Figure 2.3: Measurement probability density function for polar and Cartesian coordinates with range and bearing uncertainties of 1 m and 0.0349 rad, respectively; and for a target at a range of 7 m and a bearing of $\pi/4$.

To ensure that the linearization is valid the target should have its range r conditioned to

$$r < 0.4\frac{\sigma_r}{\sigma_\theta^2}. \tag{2.46}$$

For the same range and bearing uncertainties of figure 2.3, i.e., for

$$\sigma_r = 1 \text{ m}, \quad \sigma_\theta = 0.0349 \text{ rad} = 2^\circ \tag{2.47}$$

the area in which the sensor is considered valid is represented in figure 2.4. In the same figure is also represented a target trajectory when in CV mode and the correspondent measurements. It is clearly seen the dependence of the measurement noise covariance on the target's range r, the largest the range the more disperse are the measurements.

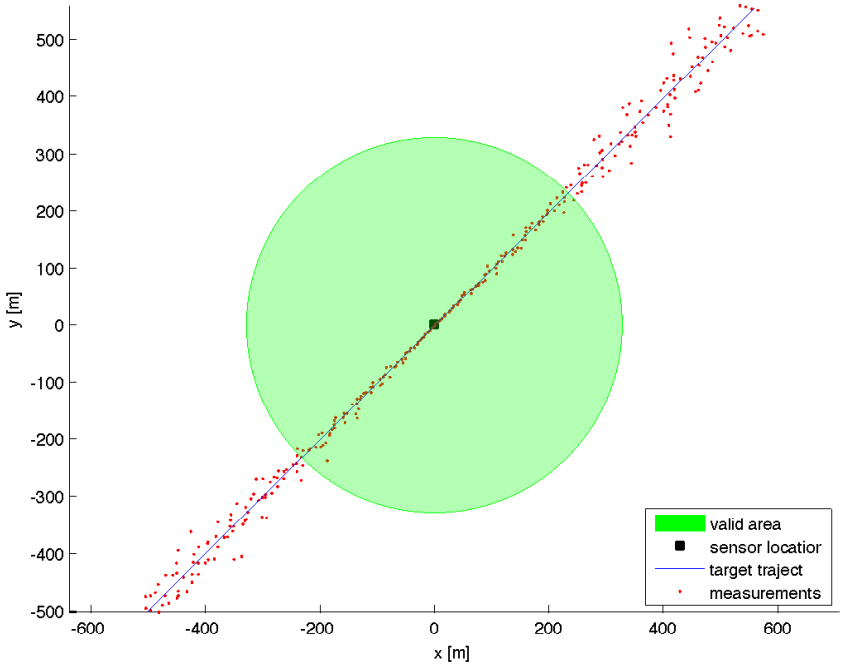


Figure 2.4: Validation area for the linearized sensor model for a range and bearing uncertainties of 1 m and 0.0349 rad, respectively.

2.3.2 Sensor Measurements for Multiple Targets

In the problem of multiple target tracking of two distinct targets, at each time step k, the sensor can have up to two measurements. It is assumed that at each time step k the tracking algorithm has access to a measurement vector z_k containing two measurements which are the Cartesian conversion of the polar measurements $z(1)_k^p$ and $z(2)_k^p$, associated with target 1 and target 2, respectively. The order in which this measurements are received is determined by their distance/range, the closest target being received first.

The multiple target sensor measurement equation is given by

$$z_k = \begin{bmatrix} z_k^{(1)} \\ z_k^{(2)} \end{bmatrix} = \begin{bmatrix} \Phi(\text{min}\{z(1)_k^p, z(2)_k^p\}) \\ \Phi(\text{max}\{z(1)_k^p, z(2)_k^p\}) \end{bmatrix} \tag{2.48}$$

where $z(r)_k^p$ indicates the noisy measurement originated from target r in polar coordinates, and $z(j)_k$ indicates the jth measurement of the sensor.

The same linearization discussed in subsection 2.3.1 can be performed for the above measurements to be able to use a linear model in the multiple target tracking algorithm analyzed in chapter 5.

2.4 Mode versus Model

Along the lines of [21], in this thesis the terms target mode and target model will be used to address two different realities. The target mode refers to the true target behavior or target motion. The target model is a mathematical usually simplified description of the of the target motion with a certain accuracy level. The tracking estimators are based on the target models, which are the known mathematical description of the target motion, and not on the true target modes.

Regarding the sensory system, it is assumed that a different characterization for its mode and a model can also be made. Its model is an approximation of the true mode since the model described in subsection 2.3.1 is a Cartesian transformation and linearized version of the sensor true mode described in section 2.3. On all the results presented in this thesis, the linear approximation of the sensor mode is always valid, i.e., conditions in (2.46) are met, so it is considered that the sensor mode and model are the same.

The system mode at a given time k is represented by

$$s_k \tag{2.49}$$

while the event that model j is in effect at time k is expressed by

$$m_k^{(j)} \triangleq \{s_k = m^{(j)}\}. \tag{2.50}$$

Here it is considered that the models in target tracking algorithm are a subset of the total mode set $\mathbb{S}_k$, i.e., the models have perfect accuracy but cannot explain all the target modes/motion behaviors.

Since the sensor mode and model are equal and constant during each simulation, for simplicity the system mode and model can be addressed as target mode and model, respectively.

Chapter 3

The Elemental Filter

In this chapter, the ubiquitous Kalman Filter algorithm is studied. Under the scope of this thesis, the Kalman Filter is presented as the most widely used building block, or elemental filter, of the Multiple Model tracking algorithms. Thus, this chapter begins by introducing the concept of the elemental filter and in section 3.2 the recursive Kalman Filter algorithm is described.

Even though the Kalman Filter, in this work, is used as a Multiple Model building block, it is also capable of tracking a single target based on a single model. Simulations regarding the Kalman Filter behavior as a single target tracker are discussed in section 3.3.

3.1 The Elemental Filter Concept

In the problem of single target tracking there are two main approaches: 1) Single Model (SM) based tracking and 2) Multiple Model (MM) based tracking.

The first approach is quite simplistic and it is only applicable if the target mode is time invariant and known, which for most of the true life applications is unfeasible due to motion uncertainty. The second approach addresses this issue by using more than one model to describe the target motion.

In the SM based tracking approach there is one single filter based on a unique model. The function of this elemental filter is to reduce the effect of the various disturbances (system and measurement noises) on the state estimate, based on the known target mode and measurements data.

On the other hand the MM algorithms use a set of models as possible candidates to describe the target mode at each time step k. This means that at each time the MM algorithms run a bank of elemental filters each based on a unique model in the set, and then compute the overall estimate based on all the estimates of each model. The MM algorithms will be discussed in depth in chapter 4.

In this chapter the algorithm of the elemental filter is described. The elemental filter is a discrete Kalman Filter (KF) which estimates the state in a way that minimizes the mean of the squared error (MSE).

3.2 The Discrete Kalman Filter

In 1960, R. E. Kalman published his famous paper [14] describing a recursive state estimator that minimized the mean of the squared error. This has become known as the Kalman Filter. The discrete KF consists of a recursive state estimation which, iteratively, computes the state vector from the prior knowledge of the discretized process dynamics, the discrete measurement model, and the measured data.

Let $x_{i|j}$, $i \geq j$, be the estimate of the state x_i using all the measurements information up to and including time j, $z^j = \{z_0, \ldots, z_k\}$. Thus, each cycle of the KF will output the current state estimate which is defined by

$$\hat{x}_{k|k} \triangleq E[x_k|z^k] \tag{3.1}$$

and the correspondent state covariance matrix defined by

$$P_{k|k} \triangleq E[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})'|z^k]. \tag{3.2}$$

For the algorithm validity the following statistical assumptions have to be met

- The initial state has known mean and covariance

$$E[x(0)|z^0] = \hat{x}_{0|0}, \;\; \text{cov}[x(0)|z^0] = P_{0|0}; \tag{3.3}$$

- The process and measurement noise are zero-mean and white with known covariance matrices

$$E[w_k] = 0, \;\; E[w_k w_j'] = Q_k \delta_{kj} \tag{3.4}$$

$$E[v_k] = 0, \;\; E[v_k v_j'] = R_k \delta_{kj}; \tag{3.5}$$

- All the above are mutually uncorrelated

$$E[x_0 w_k'] = 0, \;\; E[x_0 v_k'] = 0, \;\; E[w_k v_j'] = 0. \tag{3.6}$$

Most of the simplifications performed below relay on these assumptions.

It is possible to define the predicted state, which for the target model considered is ideally given by state equation (2.29), with $F(\omega_k) = F_k$

$$\hat{x}_{k|k-1} \triangleq E[F_{k-1} x_{k-1} + w_{k-1} | z^{k-1}]. \tag{3.7}$$

In the KF this variable is called a priori estimate of the state x_k and is computed from the estimate $x_{k-1|k-1}$ using a noise free version of (2.29)

$$\hat{x}_{k|k-1} = F_{k-1} \hat{x}_{k-1|k-1}. \tag{3.8}$$

Similarly, it is possible to compute the a priori estimate of the sensor measurement z_k using the a priori estimate and noise free version of the measurement equation (2.38)

$$\hat{z}_{k|k-1} = H_k \hat{x}_{k|k} \tag{3.9}$$

which is derived from

$$\hat{z}_{k|k-1} \triangleq E[H_k x_{k|k} + v_k]. \tag{3.10}$$

Since the value of the sensor measurement z_k is available, it is possible to compute the measurement prediction error, commonly called measurement residual

$$\tilde{z}_k \triangleq z_k - \hat{z}_{k|k-1}. \tag{3.11}$$

The measurement residual $\tilde{z}_k$ is a zero-mean white sequence which is used to correct the a posteriori state estimate by calculating

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{z}_k \tag{3.12}$$

where K_k is a gain matrix generally called Kalman Filter Gain.

The KF gain derivation takes into account the stochastic nature of the process and the measurement dynamics, in order to produce an optimal linear estimator that minimizes the mean squared error on the state estimate $x_{k|k}$. To compute the gain K it is necessary to define some more variables.

The state prediction error is given by

$$\tilde{x}_{k|k-1} = x_k - \hat{x}_{k|k-1} = F_{k-1}\tilde{x}_{k|k-1} + w_{k-1} \tag{3.13}$$

and the state prediction covariance by

$$\begin{aligned} P_{k|k-1} &\triangleq E[\tilde{x}_{k|k-1}(\tilde{x}_{k|k-1})'|z^k] \\ &= F_{k-1}E[\tilde{x}_{k-1|k-1}(\tilde{x}_{k-1|k-1})'|z^k](F_{k-1})' + E[w_{k-1}w_{k-1}'] \end{aligned} \tag{3.14}$$

which can be rewritten as

$$P_{k|k-1} = F_{k-1}P_{k-1|k-1}(F_{k-1})' + Q_{k-1}. \tag{3.15}$$

The residual in (3.11) can be rewritten to

$$\tilde{z}_k = H_k\tilde{x}_{k|k-1} + v_k. \tag{3.16}$$

Thus the predicted measurement covariance, or residual covariance, is given by

$$S_k \triangleq E[\tilde{z}_k\tilde{z}_k'] = H_kE[\tilde{x}_{k|k-1}(\tilde{x}_{k|k-1})'|z^k]H_k' + E[v_kv_k'] \tag{3.17}$$

which can be rewritten as

$$S_k = H_kP_{k|k-1}H_k' + R_k. \tag{3.18}$$

The a posterior state estimate error is defined by

$$\tilde{x}_{k|k} \triangleq x_k - \hat{x}_{k|k} = \tilde{x}_{k|k-1} - K_k\tilde{z}_k \tag{3.19}$$

so the state covariance matrix can be rewritten as

$$\begin{aligned} P_{k|k} &\triangleq E[\tilde{x}_{k|k}\tilde{x}_{k|k}'|z^k] \\ &= E[(\tilde{x}_{k|k-1} - K_k(H_k\hat{x}_{k|k-1}v_k))(\tilde{x}_{k|k-1} - K_k(H_k\hat{x}_{k|k-1}v_k))'] \\ &= P_{k|k-1} - K_kH_kP_{k|k-1} - P_{k|k-1}H_k'K_k' + K_kS_kK_k'. \end{aligned} \tag{3.20}$$

The KF gain is calculated by minimizing the trace of the state estimate covariance. Deriving the state estimate covariance trace in order of the KF gain results in

$$\frac{\partial \text{trace}(P_{k|k})}{\partial K_k} = -2(H_kP_{k|k-1})' + 2K_kS_k. \tag{3.21}$$

Equating the above equation to zero, leads to the optimal filter gain equation, defined by

$$K_k = P_{k|k-1}H_k(S_k)^{-1}. \tag{3.22}$$

Finally, by replacing the gain K on (3.20), the posteriori state estimate covariance is defined by

$$P_{k|k} = P_{k|k-1} - K_k S_k (K_k)'. \tag{3.23}$$

The KF state estimation cycle structure is divided in two main steps

1. Time update or Predict step: projects the current state estimate ahead in time computing the a priori variables;
2. Measurement update or Update step: adjusts the projected estimate by an actual measurement at that time computing the a posteriori variables.

A complete description of the KF algorithm is presented in table 3.1.

Table 3.1: One Cycle of Discrete KF Algorithm.

1.	Time update:	
	Predicted state:	$\hat{x}_{k\|k-1} = F_{k-1}\hat{x}_{k-1\|k-1}$
	Predicted covariance:	$P_{k\|k-1} = F_{k-1}P_{k-1\|k-1}(F_{k-1})' + Q_{k-1}$
2.	Measurement update:	
	Measurement residual:	$\tilde{z}_k = z_k - H_k\hat{x}_{k\|k-1}$
	Residual covariance:	$S_k = H_k P_{k\|k-1}(H_k)' + R_k$
	Filter gain:	$K_k = P_{k\|k-1}(H_k)'(S_k)^{-1}$
	Update state:	$\hat{x}_{k\|k} = \hat{x}_{k\|k-1} + K_k\tilde{z}_k$
	Update covariance:	$P_{k\|k} = P_{k\|k-1} - K_k S_k (K_k)'$

The KF design parameters necessary to run the KF algorithm are

- The model m;
- The system and measurement noises covariances (Q and R respectively);
- The initial state $\hat{x}_{0|0}$ and state covariance $P_{0|0}$.

The covariance calculations in the Minimum Mean Squared Error (MMSE) estimation problem are independent of the state and measurements and can, therefore, be performed offline. Combining together the predicted covariance and update covariance from 3.1 results in the Riccatti equation given by

$$P_{k+1|k} = F_k\{P_{k|k-1} - P_{k|k-1}H_k'[H_k P_{k|k-1}H_k' + R_k]^{-1}H_k P_{k|k-1}\}F_k' + Q_k. \tag{3.24}$$

For a time-invariant system, if it is observable, The prediction covariance matrix $P_{(k|k-1)}$ converges to a constant matrix $\bar{P}$ ($\lim_{k\to\infty} = \bar{P}$) and the Riccatti equation will converge to the steady state solution

$$\bar{P} = F\{\bar{P} - \bar{P}H'[H\bar{P}H' + R]^{-1}H\bar{P}\}F' + Q. \tag{3.25}$$

Note that this is not the case of the system studied since the measurement noise covariance R_k is time-dependent.

The optimal operating conditions of the KF rely on the following fundamental assumptions

A1. The true mode s is time invariant (i.e., $s_k = s,\ \forall k$);

A2. The true mode s at any time has a mode space $\mathbb{S}$ that is time invariant and identical to the time-invariant model m used (i.e., $\mathbb{S}_k = m,\ \forall k$).

3.3 Simulation Examples

In this section three examples are analyzed, beginning by showing the KF ability in target tracking under optimal conditions (subsections 3.3.1 and 3.3.2). The other example shows the performance of the KF algorithm under non-optimal conditions as mode and model mismatch occurs (subsection 3.3.3).

One important implementation detail is how to compute the converted measurement covariance R_k in (2.42) which is a function of the target range and bearing (in the term $J(z_k^p)$) as well as their uncertainties (in the term R_k^p). Therefore, the value of R_k should be computed using the best estimate available ate the time instant k. The most obvious way is to use the predicted state converted to polar coordinates using the transformation (2.36), except when in a first period after initialization when the uncertainty in the predicted state can be large. During this period the value of the measurement should be used to compute R_k. The threshold that determines which estimate should be used is given by [17]

$$\det[H_k P_{k|k-1} H_k'] \geq \det(R_k). \tag{3.26}$$

If the predicted measurement covariance based on the state predicted covariance $H_k P_{k|k-1} H_k$ is greater that the measurement error covariance R_k, then $H_k P_{k|k-1} H_k$ should be used.

3.3.1 Example 1 - Constant Velocity Model

In this example the target motion is described by the CV model described in subsection 2.2.1, with an initial state given by

$$x(0) = \begin{bmatrix} \mathrm{x}(0) & \dot{\mathrm{x}}(0) & \mathrm{y}(0) & \dot{\mathrm{y}}(0) \end{bmatrix}' = \begin{bmatrix} \mathrm{x}(0) & v\cos\phi(0) & \mathrm{y}(0) & v\sin\phi(0) \end{bmatrix}' \tag{3.27}$$

with

$$\mathrm{x}(0) = -170 \text{ m},\ \ \mathrm{y}(0) = -150 \text{ m},\ \ v = 5 \text{ m.s}^{-1} \text{ and } \phi(0) = 45^\circ. \tag{3.28}$$

The system noise covariance $Q(t)$ is given by equation (2.15) with

$$\sigma_x = 0.001 \text{ and } \sigma_y = 0.001. \tag{3.29}$$

The measurement noise covariance R_k^p is given by equation (2.34) with

$$\sigma_r = 2 \text{ m}, \quad \sigma_\theta = 1^\circ. \tag{3.30}$$

In this example, the KF based model is equal to the target and sensor mode with equal measurement and process noise characteristics.

In the actual implementation of the filter, the measurement noise covariance R_k (here, R_k denotes the time-invariant covariance in the sensor coordinates R_k^p) is usually easy to obtain by determining the covariance of the easy-to-obtain offline samples of the measurement noise. In regard to the process noise covariance Q determination, it is not as straightforward to obtain since the process usually is not directly observable. In most of the cases, the KF has acceptable results if the covariance Q accounts enough uncertainty.

In most cases, even if the values of noises covariances were known the dimensioning may still choose to tune these parameters to achieve better performance, as will be discussed in subsection 3.3.3. This tuning is frequently achieved offline and with the help of another distinct KF. For instances, if the Q and R are constant, both the state estimate covariance $P_{k|k}$ and the Kalman gain K_k will converge quickly to the constant value, see equation (3.25), thus these parameters can be calculated a priori by running the filter offline.

Note that the KF uses the discretized equations of the target mode, addressed in the correspondent target mode description section, in this case in subsection 2.2.1. The sampling time used in all simulation examples in this thesis is

$$\Delta t = 1 \text{ s}. \tag{3.31}$$

Plus, the initial position of the target and of the filter are different from each other, which creates a more realistic scenario. The initial state of the target is thus considered not to be known, only its distribution is known and given by

$$\hat{x}_{0|0} \sim \mathcal{N}(x(0), P_{0|0}) \tag{3.32}$$

where $x(0)$ is the initial true target state and $P_{0|0}$ is the considered variance which has the value given by

$$P_{0|0} = \begin{bmatrix} 10^2 & 0 & 0 & 0 \\ 0 & 0.2^2 & 0 & 0 \\ 0 & 0 & 10^2 & 0 \\ 0 & 0 & 0 & 0.2^2 \end{bmatrix}. \tag{3.33}$$

For the parameters defined above, the KF is in optimal conditions.

In this same figure it is also clear the initial filter linear velocity of the filter in contrast with the true target linear velocity.

In this situation, the state estimate has small error has can be seen in figure 3.1. In both variables, the error has its higher values at the first part of the simulation due to the initial estimate error.

Besides the influence of the initial state error, which is only present at the beginning of the simulation, there is also the influence of the system disturbances, the process and the measurement noises.

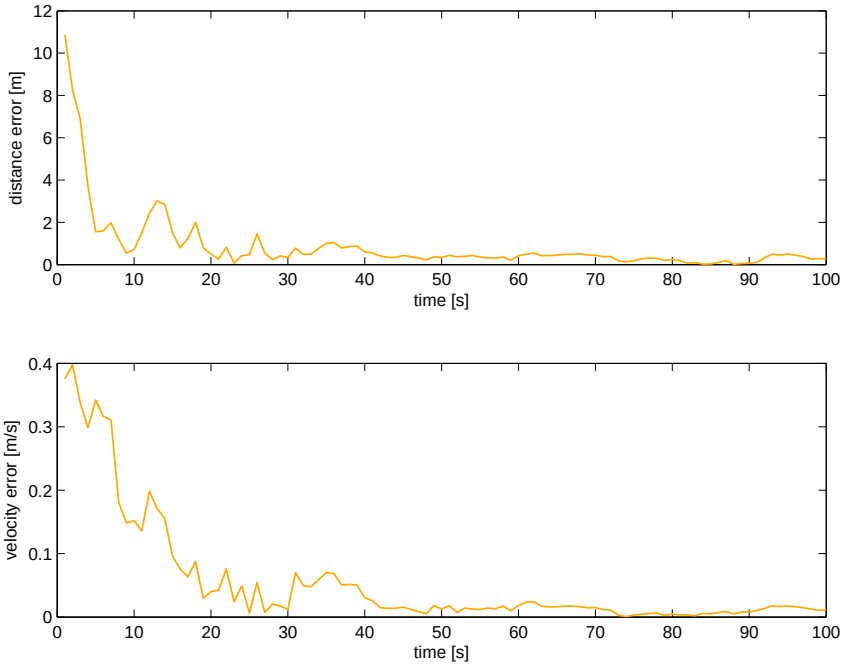


Figure 3.1: KF position and velocity error in CV model.

The influence of the process noise is constant throughout the whole simulation, more interesting is the influence of the measurement noise that, as shown in section 2.3.1, depends on the target state, specifically on the target range, thus it is expected an increase of the overall estimate error with the target range due to the increase of the measurement noise.

Even though the initial state estimate error, the filter is quick to adapt itself and converges to the true position and linear velocity value (see figure 3.1, once again). The quickness of the KF response depends mainly on the initial value of the state estimate, its covariance and the covariance of the disturbances (process and measurement noises).

Obviously, the closer the initial estimate is to the target state, the more quickly will the filter converge. Under optimal conditions, the KF will always converge for all values of the state estimate covariance, except the null value, but if its value is too small the filter will take longer to compensate the difference in the initial value. Regarding the disturbances covariance influence, they represent the belief of the filter in the data it receives, be the measurements data or the model description.

In figure 3.2 the KF position estimate is compared to the true target state and some state estimate error ellipsoids with a confidence level of 95 percent. The true target position, as expected, is always inside the ellipsoid centered at the state estimate and with shape defined by the state estimate covariance. As expected, with time the state estimate is more precise and the state estimate covariance decreases leading to the decrease of the ellipsoids contour.

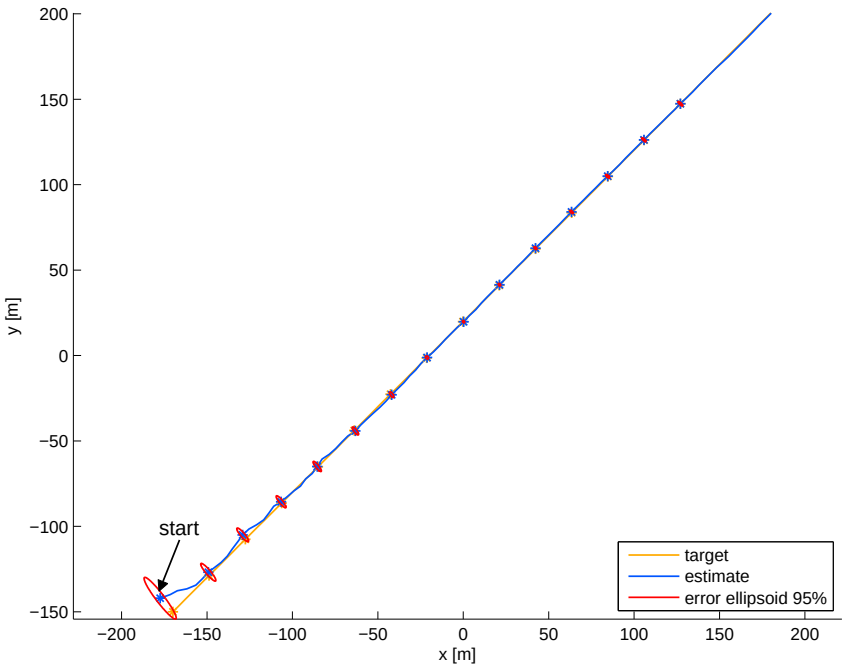


Figure 3.2: Target trajectory and KF estimate and error ellipsoid propagation in CV model.

3.3.2 Example 2 - Constant Turn Model

In this second example, the optimal conditions of operation for the KF are maintained but the target mode and KF based model are now the CT model described in subsection 2.2.2, with an angular velocity ω equal to 3 rads^{-1}.

The initial state of the target in this simulation is given by the equation (3.27) with

$$\mathrm{x}(0) = -150 \text{ m}, \quad \mathrm{y}(0) = 50 \text{ m}, \quad v = 5 \text{ m.s}^{-1} \text{ and } \phi(0) = -75^{\circ}. \tag{3.34}$$

The KF initial state estimate has the same distribution from the previous example (3.32) with covariance given by (3.33). The target and filter process and measurement noises characteristics are also maintained from the previous example.

The algorithm maintains its tracking capability, its state estimate error, regarding the position and the velocity, as in the previous example, is high at the beginning of the simulation due to the initial state estimate error but steadily is reduced, see figure 3.3.

The target trajectory in the CT mode, KF position estimate and some state estimate error ellipsoids with a confidence level of 95 percent are shown in figure 3.4. Once again, due to the initial state uncertainty, the ellipsoids contour is larger at beginning of the simulations but it steadily decreases with time.

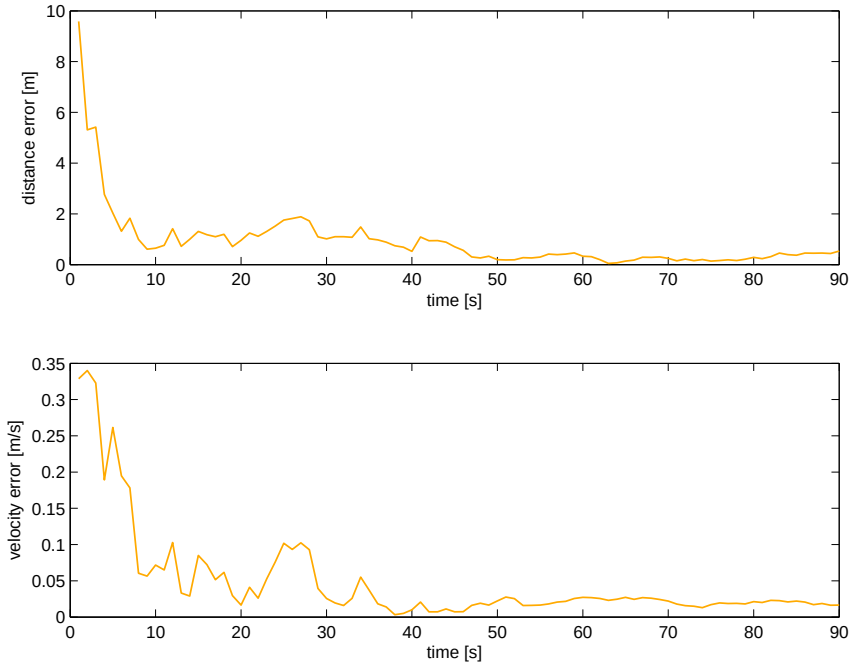


Figure 3.3: KF position and velocity error in CT model.

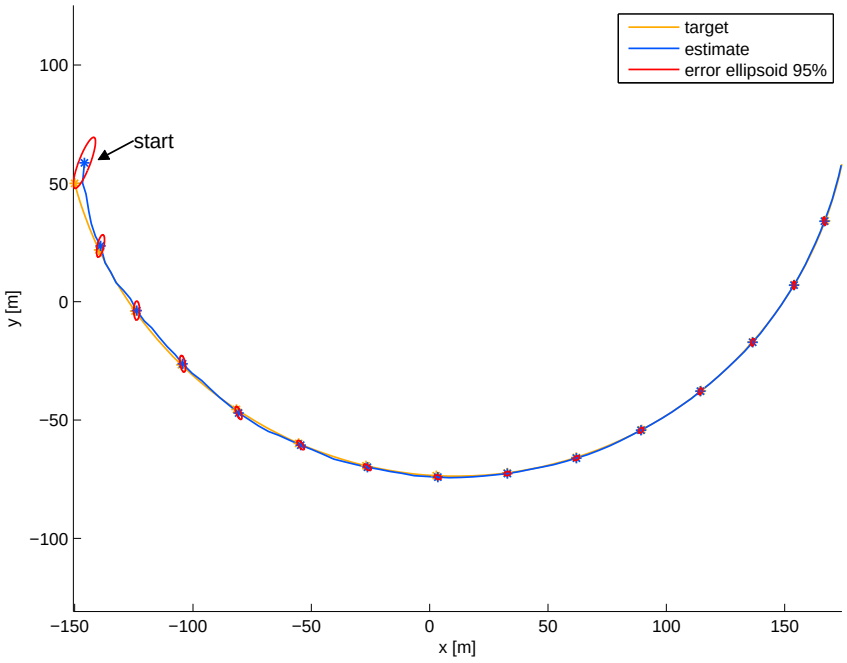


Figure 3.4: Target trajectory and KF estimate and error ellipsoid propagation in CT model.

3.3.3 Example 3 - Mode and Model mismatch

In this example, the model used to describe the target is different from the target's mode, thus a model mismatch occurs.

The target mode and KF based model are the CT model described in subsection 2.2.2, with an angular velocity equal to 3.1 rads^{-1} and 3 rads^{-1}, respectively. Every other target behavior and KF parameters are maintained from the previous simulation with the exception of the KF process noise covariance.

As mentioned before, the tune of the disturbances covariances can help the KF to deliver better estimates. The process noise covariance Q contributes to the overall uncertainty, it relates to the trust the KF has on the model used to describe the target motion. When Q is large, the Kalman Filter tracks large changes in the measured data more closely than for smaller Q. Regarding the measurement noise covariance R, it relates to the trust the KF has in measured data. If R is high, the Kalman Filter considers the measurements as not very accurate.

The covariance R is unchanged and defined by equation (2.34), whereas the process noise covariance Q is different from the true target process noise covariance to account the uncertainty of the KF model to describe the target mode. The covariance $Q(t)$ chosen is then higher and given by the equation (2.15) with

$$\sigma_x = 0.01 \text{ and } \sigma_y = 0.01. \tag{3.35}$$

For the parameters defined, the target position, correspondent estimate in the 2D space and some state estimate error ellipsoids with a confidence level of 95 percent are in figure 3.5. It can be seen clearly the initial position difference between the KF estimation and the target and the correspondent initial large contour of the error ellipsoid due to the initial state estimate uncertainty, directly related with the initial state estimate covariance $P_{0|0}$. Note also how, throughout the whole simulation, the error ellipsoids contour is overall larger in this experiment than in the previous where there was not a mode and model mismatch.

The estimate error (figure 3.6) is much higher then the previous simulations. Initially the high value of the error is clearly due to the uncertainty on the initial state, but the KF steadily reduces this error, indeed, after the 30th second, which seems to be the stabilization time in this example, the filter would be able to deliver as good estimates as in the previous examples if there was no mode and model mismatch. The error present after stabilization which is considerably higher than that of the KF estimation in optimal conditions is solely due to the mode and model mismatch.

As seen in this simulation, even in non-optimal conditions, the KF delivers a good estimate thus the KF operating conditions can be relaxed. Note that the track can indeed be lost if the KF is too far away from its optimal operating conditions, for example.

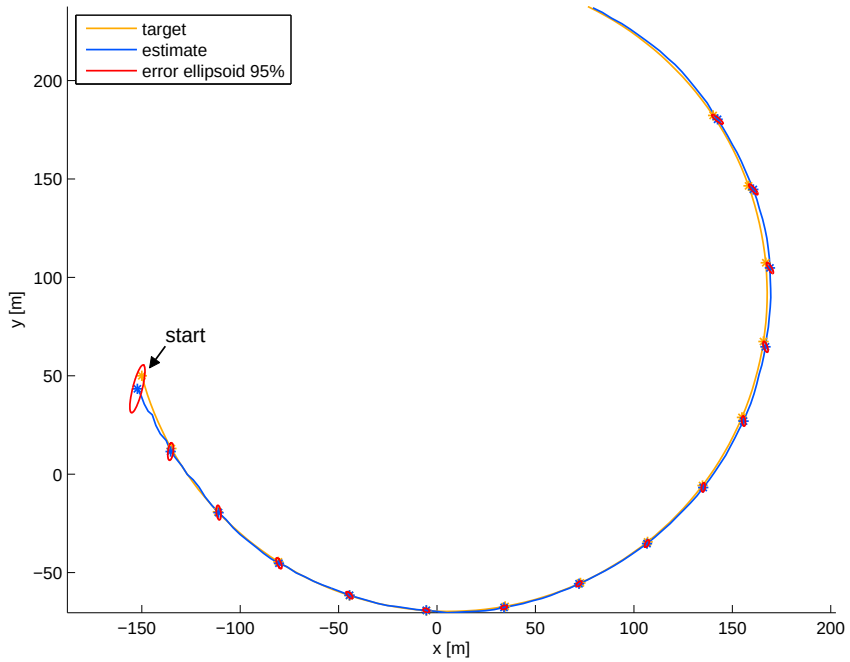


Figure 3.5: Target trajectory and KF estimate and error ellipsoid propagation with mode and model mismatch.

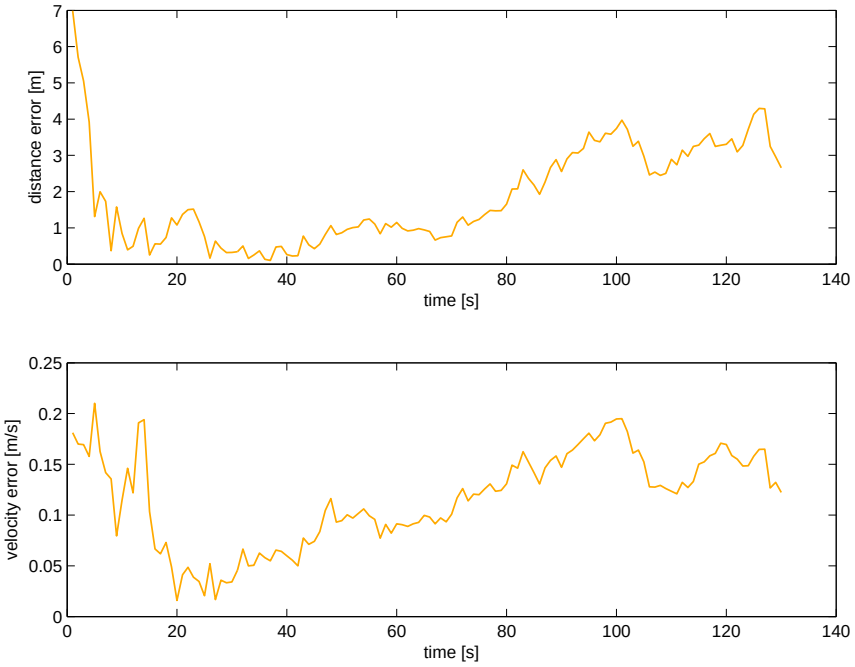


Figure 3.6: KF position and velocity error with mode and model mismatch.

Chapter 4

Single Target Tracking

This chapter addresses the problem of tracking a single target using a Multiple Model (MM) approach. In section 4.1, the MM tracking concept is presented.

This chapter introduces the three generations of MM algorithms, following the classification first introduced in [18]: the autonomous (section 4.2), the cooperating (section 4.3), and the variable structure (section 4.4). Each of the above sections is composed of an introduction to the peculiarity of each generation, a description of a representative algorithm of that generation and a simulation example of the algorithm. The algorithm chosen for the first, second and third generations are the Minimum Mean-Square Error Autonomous Multiple Model (subsection 4.2.1), the Interacting Multiple Model (subsection 4.3.1) and the Likely Model Set (subsection 4.4.1).

In the last section, section 4.5, the three algorithms studied earlier are simulated in the equal conditions and their performances are compared.

4.1 Introduction to Multiple Model algorithms

In chapter 3, the tracking of the single target is based on the choice of a (best) model to describe the maneuvering target. The chosen model is assumed as the true mode and a single filter runs based on it. This approach has several obvious flaws. First, the estimation does not take into account a possible mismatch between the target mode and the filter model. Second, the choice of the model is made offline and the estimation errors are not taken into account, even though they are the best indicator of the correctness of the chosen model.

The MM algorithms allow that more than one model can be used at a time to match the current target mode. Besides diminishing the probability of mode and model mismatch and using the estimation errors to output a better overall estimate, the MM algorithms greatest benefit comes when used to solve target motion uncertainty.

By using more than one model the MM algorithms are more capable to track targets with motion uncertainty. The basic idea is: i) to assume a set of models as possible candidates of the true mode in effect at the time; ii) run a bank of elemental filters, each based on a unique model in the set; and iii) generate the overall estimates by a process based on the results of these elemental filters. The model-set is designated by $\mathbb{M}$ and is composed of N models, i.e., $\mathbb{M} = \{m_1, \ldots, m_N\}$.

In general, four key components of MM estimation algorithms can be identified as follows (as stated in [21]):

1. Model-set determination: This includes both offline design and possibly online adaptation of the model set. A MM estimation algorithm distinguishes itself from non-MM estimators by the use of a set of models, instead of a single model. The performance of a MM estimator depends largely on the set of models used. The major task in the application of MM estimation is the design (and possibly adaptation) of the set of multiple models.

2. Cooperation strategy: This refers to all measures taken to deal with the discrete-valued uncertainties within the model set, particularly those for hypotheses about the model sequences. It includes not only pruning of unlikely model sequences, merging of 'similar' model sequences, and selection of (most) likely model sequences, but also iterative strategies, such as those based on the expectation-maximization algorithm.

3. Conditional filtering: This is the recursive (or batch) estimation of the continuous-valued components of the hybrid process conditioned on some assumed mode sequence. It is conceptually the same as state estimation of a conventional system with only continuous-valued state.

4. Output processing: This is the process that generates overall estimates using results of all filters as well as measurements. It includes fusing/combining estimates from all filters and selecting the best ones.

In figure 4.1 the above defined components of the MM algorithms is presented in a diagram.

This chapter introduces the three generations of MM algorithms: the autonomous (subsection 4.2), the cooperating (subsection 4.3), and the variable structure (subsection 4.4).

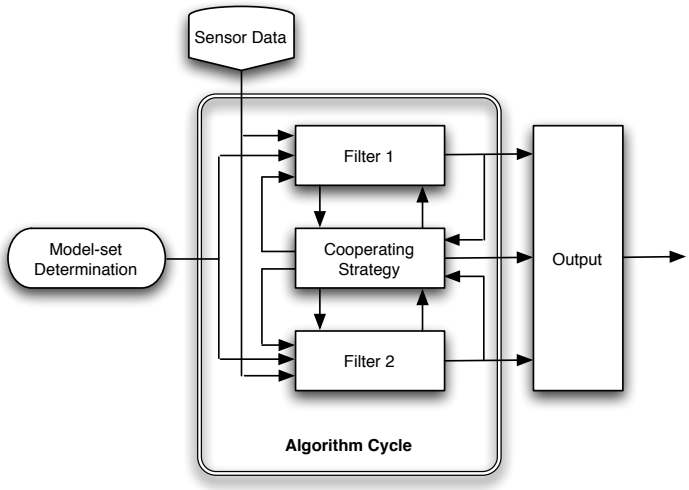


Figure 4.1: General structure of MM estimation algorithms with two model based filters.

4.2 Autonomous Multiple Model

The first generation is characterized by the fact that each of its elemental filters operates individually and independently of all the other elemental filters. Its advantage over many non-MM approaches is due to its superior output processing of results from elemental filters to generate the overall estimate.

The first generation of MM algorithms has two main assumptions

A1. The true mode s is time invariant (i.e., $s_k = s,\ \forall k$);

A2'. The true mode s at any time has a mode space $\mathbb{S}$ that is time invariant and identical to the time-invariant finite model set $\mathbb{M}$ used (i.e., $\mathbb{S}_k = \mathbb{M},\ \forall k$).

Comparing with the KF assumptions in 3.2, the second assumption A2 has been relaxed introducing the MM algorithms novelty, i.e., the simultaneous presence at each time step k of multiple different mode-matched filters.

Assumption A1 is a defining assumption of the Autonomous Multiple Model (AMM) algorithms. Under A1, each elemental filter does conditional filtering based on a constant model sequence. In other words, each filter works individually and independently of other filters. As such, each conditional filtering operation is autonomous, hence the name AMM.

In section 4.2.1 an example AMM algorithm is discussed.

4.2.1 The Minimum Mean-Square Error Autonomous Multiple Model algorithm

The AMM estimator considered in this section is the Minimum Mean-Square Error Autonomous Multiple Model (MMSE-AMM) algorithm, first introduced in [22].

Using Baye's formula, the probability of model i is correct given measurement data up to time step k, is given by the recursion

$$\begin{aligned}\mu_k^{(i)} &\triangleq P\{m^{(i)}|z^k\} = P\{m^{(i)}|z_k, z^{k-1}\} = \frac{p[z_k|z^{k-1}, m^{(i)}]P\{M_i|z^{k-1}\}}{p[z_k|z^{k-1}]} \\ &= \frac{p[z_k|z^{k-1}, m^{(i)}]\mu_{k-1}^{(i)}}{\sum_j p[z_k|z^{k-1}, m^{(j)}]\mu_{k-1}^{(j)}}\end{aligned} \tag{4.1}$$

where $p[z_k|z^{k-1}, m^{(i)}]$ is the likelihood function of model $m^{(i)}$, which under Gaussian assumption is given by

$$L_k^{(i)} \triangleq p[z_k|m^{(i)}, z^{k-1}] \overset{assume}{=} \mathcal{N}(\tilde{z}_k^{(i)}; 0, S_k^{(i)}) \tag{4.2}$$

where $\tilde{z}_k^{(i)}$ and $S_k^{(i)}$ are the residual and its covariance from the elemental filter matched to model $m^{(i)}$.

Thus each discrete KF based in one of the models $m^{(i)} \in \mathbb{M}$, processes the model-conditioned filtering, delivering the model-conditioned state estimates $\hat{x}_{k|k}^{(i)}$ and the model-conditioned covariances $P_{k|k}^{(i)}$. The probability of a model m_i being correct is given by 4.1. After initialization, at each time step k the filters run recursively on their own estimations from the previous cycle. Combining the model probability with the conditioned-model estimates the overall state estimate is given by

$$\hat{x}_{k|k} = \sum_i \hat{x}_{k|k}^{(i)} \mu_k^{(i)} \tag{4.3}$$

which under assumptions A1 and A2' is unbiased with an overall state covariance given by

$$P_{k|k} = \sum_i P_{k|k}^{(i)}[(\hat{x}_{k|k} - \hat{x}_{k|k}^{(i)})(\hat{x}_{k|k} - \hat{x}_{k|k}^{(i)})']\mu_k^{(i)}. \tag{4.4}$$

In table 4.1 the complete cycle of the MMSE-AMM algorithm is described, and a diagram of the algorithm steps is presented in figure 4.2.

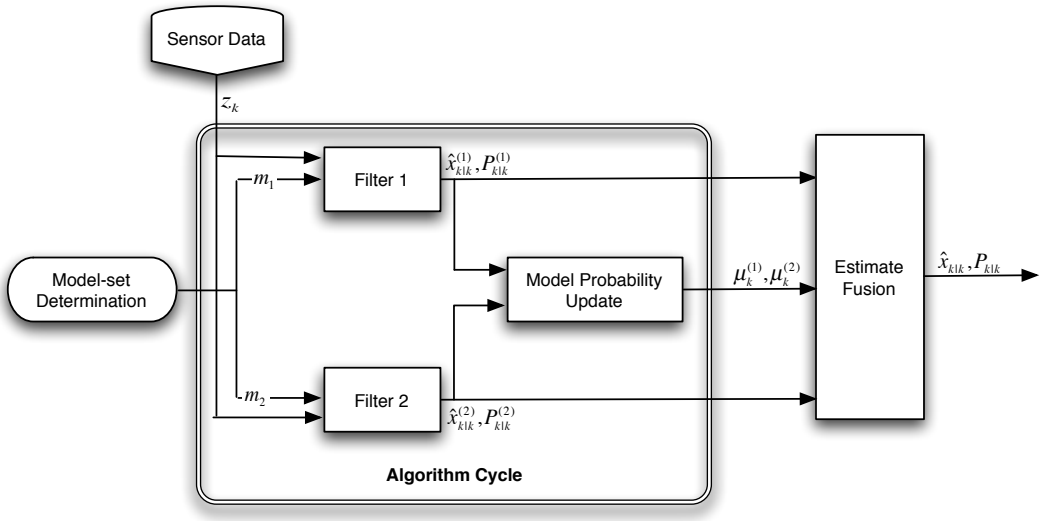


Figure 4.2: General structure of MMSE-AMM estimation algorithm with two model based filters.

Table 4.1: One Cycle of MMSE-AMM Algorithm.

1. Model-conditioned filtering ($\forall m_i \in \mathbb{M}$) :

Predicted state:	$\hat{x}^{(i)}_{k\|k-1} = F^{(i)}_{k-1}\hat{x}^{(i)}_{k-1\|k-1}$
Predicted covariance:	$P^{(i)}_{k\|k-1} = F^{(i)}_{k-1}P^{(i)}_{k-1\|k-1}(F^{(i)}_{k-1})' + Q^{(i)}_{k-1}$
Measurement residual:	$\tilde{z}^{(i)}_k = z_k - H^{(i)}_k\hat{x}^{(i)}_{k\|k-1}$
Residual covariance:	$S^{(i)}_k = H^{(i)}_k P^{(i)}_{k\|k-1}(H^{(i)}_k)' + R^{(i)}_k$
Filter gain:	$K^{(i)}_k = P^{(i)}_{k\|k-1}(H^{(i)}_k)'(S^{(i)}_k)^{-1}$
Update state:	$\hat{x}^{(i)}_{k\|k} = \hat{x}^{(i)}_{k\|k-1} + K^{(i)}_k\tilde{z}^{(i)}_k$
Update covariance:	$P^{(i)}_{k\|k} = P^{(i)}_{k\|k-1} - K^{(i)}_k S^{(i)}_k(K^{(i)}_k)'$

2. Model probability update ($\forall m_i \in \mathbb{M}$) :

Model likelihood:	$L^{(i)}_k \overset{assume}{=} \mathcal{N}(\tilde{z}^{(i)}_k; 0, S^{(i)}_k)$
Model probability:	$\mu^{(i)}_k = \dfrac{\mu^{(i)}_{k-1}L^{(i)}_k}{\sum_j \mu^{(j)}_{k-1}L^{(j)}_k}$

3. Estimate fusion:

Overall estimate:	$\hat{x}_{k\|k} = \sum_i \hat{x}^{(i)}_{k\|k}\mu^{(i)}_k$
Overall covariance:	$P_{k\|k} = \sum_i [P^{(i)}_{k\|k} + (\hat{x}_{k\|k} - \hat{x}^{(i)}_{k\|k})(\hat{x}_{k\|k} - \hat{x}^{(i)}_{k\|k})']\mu^{(i)}_k$

The design parameters of the MMSE-AMM and of the KF parameters are alike, except for the model set definition in contrast to a single model definition. The MMSE-AMM design parameters are:

- The model set $\mathbb{M}$ structure;
- The system and measurement noises covariances (Q and R respectively);
- The initial state $\hat{x}_{0|0}$ and state covariance $P_{0|0}$.

Under the assumptions A1 and A2', i.e., if the model set includes the target mode and no mode jump occurs, the true model probability will converge to unity and the MMSE-AMM will work as an isolated KF in sync with the true model.

If assumption A1 verifies but A2' doesn't, the probability of the model "nearest" to the target mode will be the highest (how close the probability is to one depends on how "near" is the model to the true mode) and the overall state estimate will be a weighted mean of all the KF (local) state estimates.

On the other hand, if assumption A2' doesn't verifies, the AMM fails to deliver good estimates. Its ability to handle mode jumps is very limited. This is due to the autonomous characteristics of the algorithm: the filters estimates have only access to their own previous estimates, there is not any cooperation strategy between the them. The KF, when based in models different from the target mode, takes a long time to correct its estimate, if mode jumps keep occurring there is a chance of a track loss.

In order to overcome these two limitations it is usual to apply a lower bound in the model probability, to facilitate the algorithm model switches. This way the MMSE-AMM can provide very reasonable position

estimates when the target is in slow maneuver. In order to improve this behavior and include it in a more structured way, the second generation MM algorithms was developed.

4.2.2 Simulation Examples

In this section two examples are analyzed, on the first example the MMSE-AMM target tracking is under optimal conditions (subsections 4.2.2) on the second example the MMSE-AMM is under non-optimal conditions and its performance with and without lower bounding is presented (subsection 4.2.2).

Example 1 - MMSE-AMM under optimal target tracking conditions

In this example the target motion is described by the CT model described in subsection 2.2.2, with an initial state given by (3.27) with

$$\text{x}(0) = -50 \text{ m}, \;\; \text{y}(0) = -100 \text{ m}, \;\; v = 3 \text{ m.s}^{-1} \;\; \text{and} \;\; \phi(0) = 0^\circ. \tag{4.5}$$

The process noise covariance $Q(t)$, given by the equation (2.15) is equal to the one defined in subsection 3.3.1, i.e., with uncertainties given by (3.35). The measurement noise covariance R_k^p is given by the equation (2.34) with uncertainties given by (3.30). To account the multiple model structure, both process and measurement covariances definitions should be rewritten, respectively, by

$$Q_k^{(i)} = Q_k \; \forall i, k \;\; \text{with } \sigma_x^2 = \sigma_y^2 = 0.001 \tag{4.6}$$

$$R_k^{p,(i)} = \text{diag}(\sigma_r^2, \; \sigma_\theta^2) \; \forall i, k \;\; \text{with } \sigma_r = 2 \text{ m}, \quad \sigma_\theta = 1^\circ \tag{4.7}$$

where, Q_k is the discretized continuous process covariance (2.15) using equation (2.10).

The model set $\mathbb{M}$ is composed of seven models ($N = 7$). The models differ from each other on the value of the angular velocity ω. Each model m_i corresponds to an angular velocity ω_i. The values of the angular velocity of the models in $\mathbb{M}$ are listed in the table 4.2, shown below

Table 4.2: Angular velocity of the seven models.

Angular velocity ω [rad.s^{-1}]
0.03
0.025
0.02
0.015
0.01
0.005
0

Each elemental filter has the initial state equal given by the distribution 3.32, i.e.,

$$\hat{x}_{0|0}^{(i)} \sim \mathcal{N}(x(0), P_{0|0}^{(i)}) \;\; \forall m_i \in \mathbb{M} \tag{4.8}$$

with initial state covariance estimate $P_{0|0}^{(i)}$ given by

$$P_{0|0}^{(i)} = \begin{bmatrix} 10^2 & 0 & 0 & 0 \\ 0 & 0.2^2 & 0 & 0 \\ 0 & 0 & 10^2 & 0 \\ 0 & 0 & 0 & 0.2^2 \end{bmatrix} \quad \forall m_i \in \mathbb{M}. \tag{4.9}$$

In this first example, the target's angular velocity is constant and equal to the angular velocity of one the models in the model-set. The angular velocity of the target chosen was $\omega = 0.03$ rad.s^{-1}.

In this example the algorithm does not have a lower bound in the probability. The initial probability of a model being correct is equal to all models, i.e.,

$$\mu_0^{(i)} = \frac{1}{N}, \quad \forall m_i \in \mathbb{M}. \tag{4.10}$$

The model probability of the model with angular velocity equal to the target's angular velocity clearly converges to unity as was expected. The convergence time is determined by the a priori models probability, the certainty in the a priori state estimate and the covariance Q of the models in the model set. In figure 4.3 is the graphic of the evolution of the models probability with time.

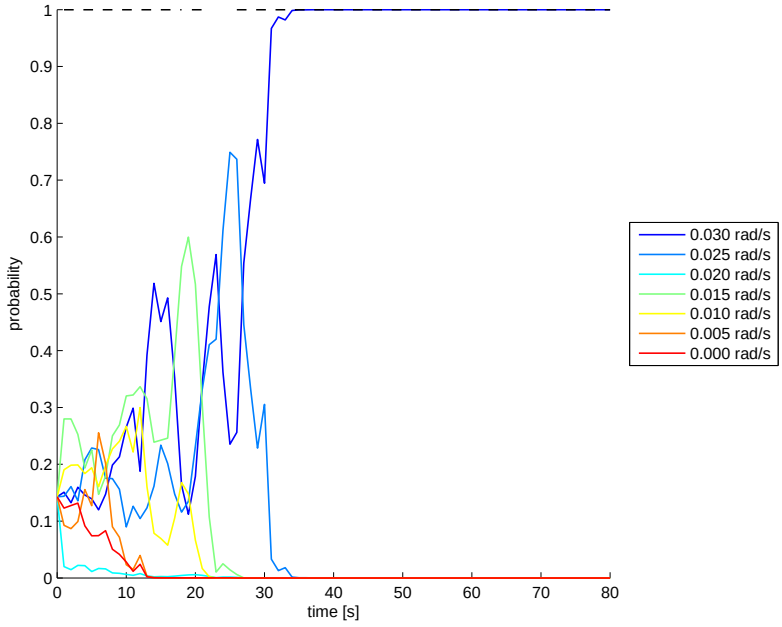


Figure 4.3: MMSE-AMM model probabilities under optimal operating conditions.

The error under these circumstances is quite low as presented in figure 4.4. The error is higher at the beginning when the algorithm is searching for the suitable model to match the target mode and compensate the initial state error. The error decreases steadily, as in the case of the isolated KF, as the probability of the correct model gets closer to unity, at that point the algorithm behaves almost like an isolated KF matched to the true target mode.

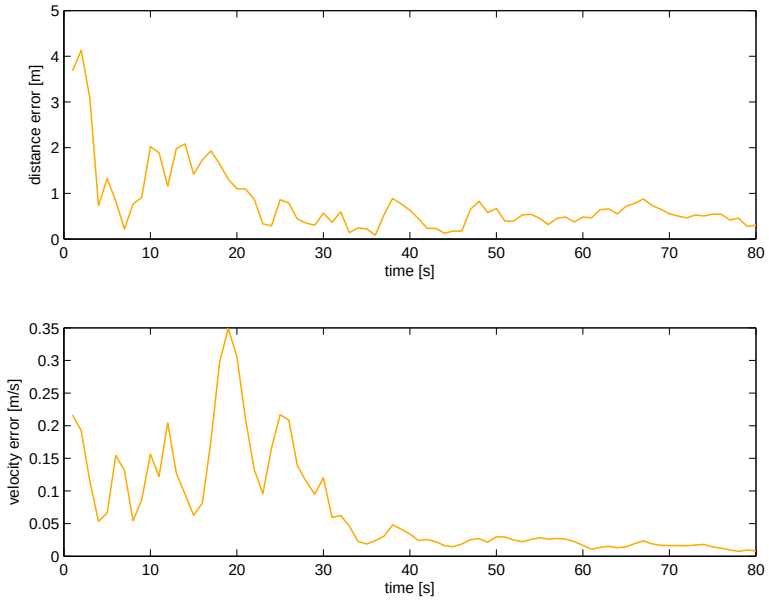


Figure 4.4: MMSE-AMM position and velocity error under optimal operating conditions.

The position estimate compared with the target's position state and the estimate error ellipsoids are represented in figure 4.5, as would be expected, due to the optimal nature of the MMSE-AMM algorithm, the contour of the error ellipsoids monotonously decrease with time.

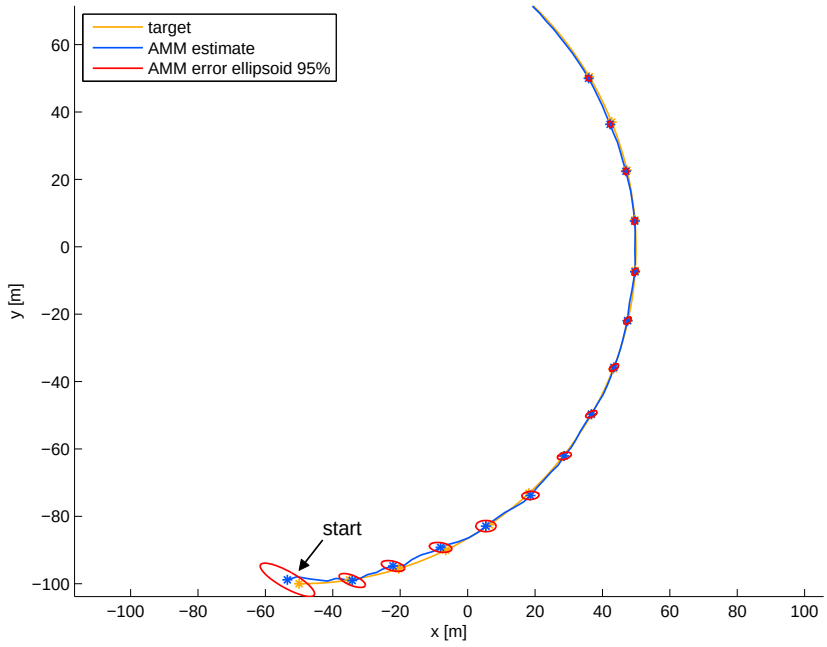


Figure 4.5: Target trajectory and MMSE-AMM with and without lower bound position estimate and error ellipsoid propagation under under optimal operating conditions.

Example 2 - Influence of the model probability lower bounding

In a second example, the probability lower bound is introduced and the target has a variable angular velocity. The target is in CT mode described in subsection 2.2.2 , with an initial state given by (3.27) with

$$\mathrm{x}(0) = 100 \text{ m}, \quad \mathrm{y}(0) = 100 \text{ m}, \quad v = 3 \text{ m.s}^{-1} \text{ and } \phi(0) = 100^\circ. \tag{4.11}$$

The angular velocity of the target at each time step is equal to the angular velocities of one the models in the model-set. The angular velocity ω of the target is given by the equation below and represented in figure 4.6,

$$\omega_k = \begin{cases} 0.03 & 1 \leq k \leq 40 \\ 0.025 & 41 \leq k \leq 80 \\ 0.02 & 81 \leq k \leq 120 \\ 0.015 & 121 \leq k \leq 160 \\ 0.01 & 161 \leq k \leq 200 \\ 0.005 & 201 \leq k \leq 240 \\ 0 & 241 \leq k \leq 280 \end{cases}, \quad \text{in rad.s}^{-1}. \tag{4.12}$$

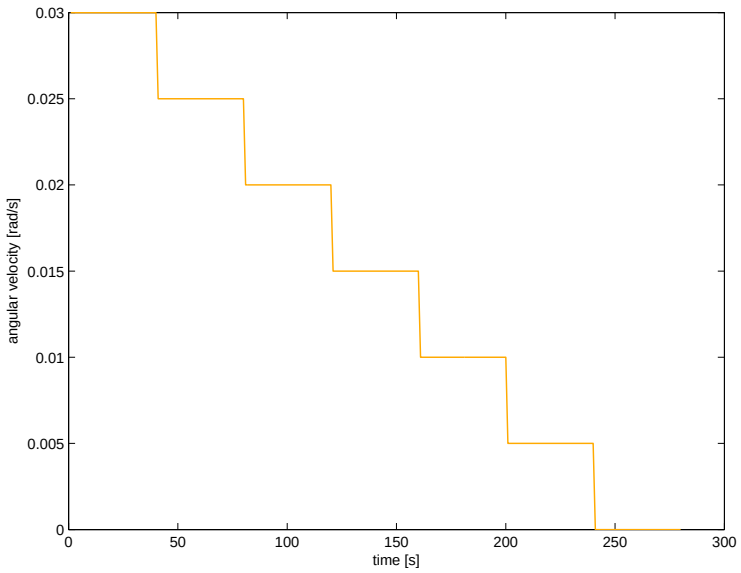


Figure 4.6: Non constant angular velocity of the target.

Also, to include the uncertainty in how the elemental filters explain the moving target the models m_i have a higher process noise covariance than the target given by

$$Q_k^{(i)} = Q_k \ \forall i, k \ \text{ with } \sigma_x^2 = \sigma_y^2 = 0.01. \tag{4.13}$$

where, Q_k is the discretized continuous process covariance (2.15) using equation (2.10). The same

principles discussed in subsection 3.3.3, regarding the influence of the noises covariances are maintained here, in the MM approach, since they consist of multiple KF.

The value chosen for the lower bound in the model probability was 10^{-3}. All the other parameters not specified here are identical to the ones used in the first example, including the filters initial state distribution and its covariance.

First, observe what happens to the model probability when the lower bounding is introduced in figure 4.7. The model probability takes the same time in both algorithms to rise to (near) unity (see the beginning of the algorithms), but the lower bounding approach is much more quickly to do the model switch. By the end of the simulation the classic MMSE-AMM algorithm realized three model switches against the almost six model switches of the lower bounded MMSE-AMM algorithm which are the same number of mode switches realized by the target.

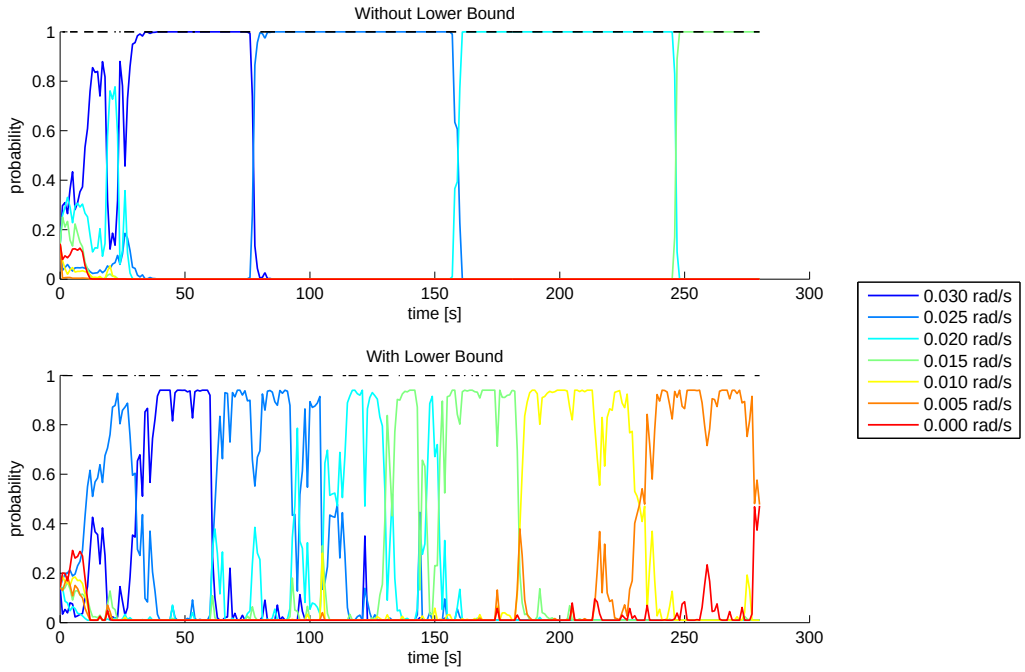


Figure 4.7: MMSE-AMM with and without lower bound model probabilities under non optimal operating conditions.

In figure 4.8 it is obvious the improvement of the MMSE-AMM algorithm tracking performance when the lower bound in model probability is introduced. The classic approach the MMSE-AMM is too slow to track the maneuvering target resulting in high errors (see figure 4.9). Regarding the error ellipsoids, the classic MMSE-AMM at some point looses track, since the real target position fails to be inside its estimate error ellipsoid. The lower bounded MMSE-AMM algorithm accounts for possible mode and model mismatch thus the error ellipsoids always contain the true target position, but they do not monotonously decrease with time revealing the non-optimality of this approach.

As the target keeps switching to different modes, without giving the classic MMSE-AMM the necessary (large) stabilization time, the error increases with time. The lower bounded model probability MMSE-AMM is a much faster tracker and more sensitive to the mode changes of the target.

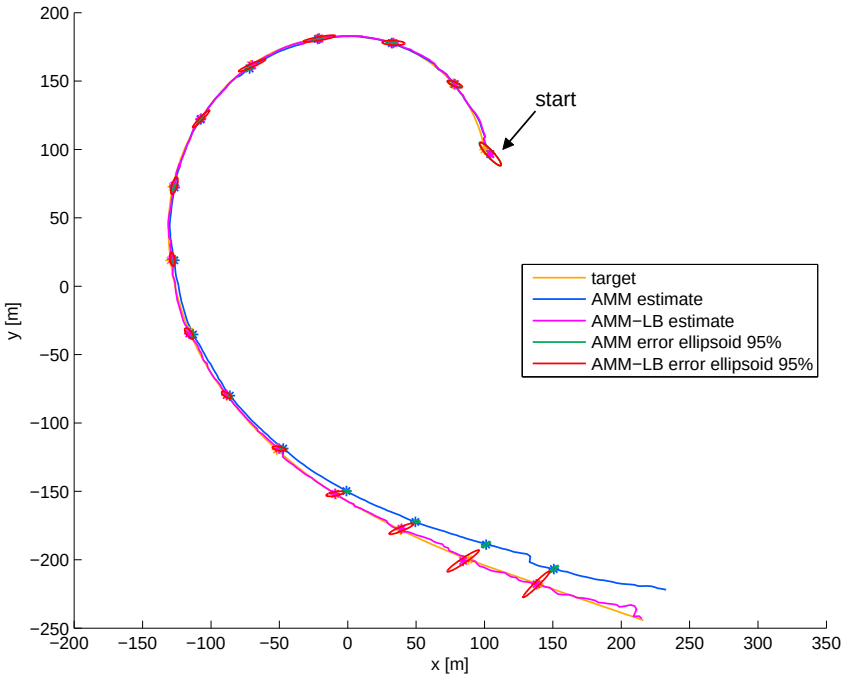


Figure 4.8: Target trajectory and MMSE-AMM with and without lower bound position estimate and error ellipsoid propagation under non optimal operating conditions.

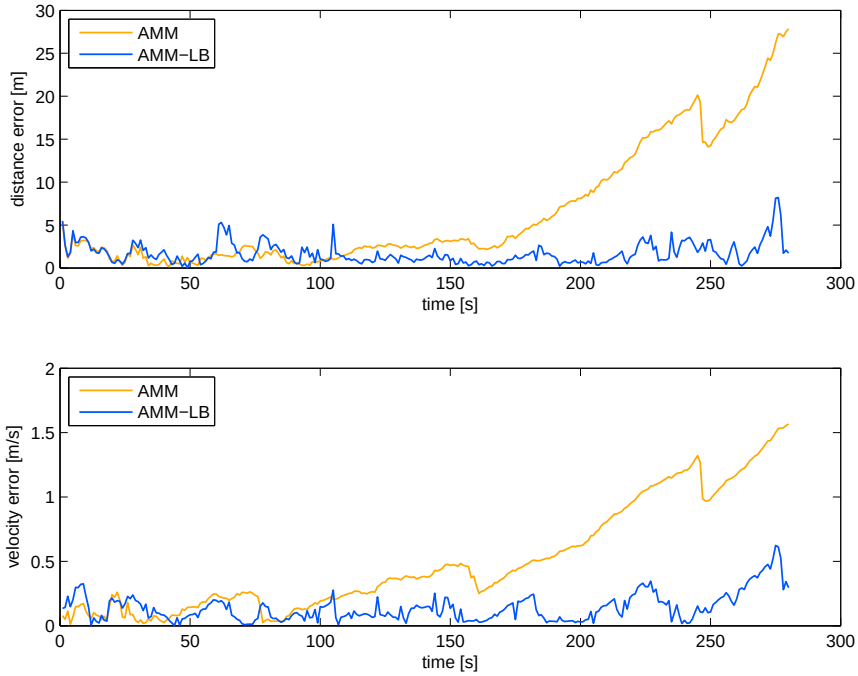


Figure 4.9: MMSE-AMM with and without lower bound position and velocity error under non optimal operating conditions.

4.3 Cooperating Multiple Model

The second generation of MM algorithms inherits the first generation superior output processing and introduces the internal cooperation between elemental filters. Rather then keeping the filters working independently, this generation brings in the filter reinitialization, where all the outputs of the filters from the previous cycle are used in the current cycle to achieve a better performance.

This generation relaxes the first generation assumptions (see section 4.2):

A1'. The true mode sequence $\{s_k\}$ is Markov, i.e., the mode evolves accordingly to the memoryless random process described by $P\{s_{k+1}^{(j)}|s_k^{(i)}\} = \pi_{ij,k} \ \ \forall i,j$;

A2'. The true mode s at any time has a mode space $\mathbb{S}$ that is time invariant and identical to the time-invariant finite model set $\mathbb{M}$ used (i.e., $\mathbb{S}_k = \mathbb{M}, \ \forall k$).

A new variable is introduced to account the model history or model sequence

$$m^{k,l} = \{m^{(i_{1,l})}, \ldots, m^{(i_{k,l})}\}, \ \ l = 1, \ldots, (N)^k \tag{4.14}$$

where $i_{k,l}$ is the model index at time k from history l and $1 \leq i_{k,l} \leq N \ \forall k$.

As the size of model sequence increases exponential with time ($(N)^k$), a exponentially increasing number of filters are needed to optimally estimate the state thus making the optimal solution not viable. Suboptimal approaches are commonly and successfully adopted, generally consisting in only keeping a limited number of model sequences associated with the largest probabilities, discard the rest, and normalize the model probability to ensure it sum is equal to unity.

In section 4.3.1, one of the most representative algorithms of the Cooperating Multiple Model (CMM), the Interacting Multiple Model (IMM) algorithm, is analyzed.

4.3.1 The Interacting Multiple Model algorithm

The IMM estimator was originally proposed by Bloom in [11]. It is one of the most cost-effective class of estimators for a single maneuvering target. The IMM has been receiving special attention in the last few years, due to its capability of being combined with other algorithms to resolve the multiple target tracking problem.

The sequence of events consisting of the true target mode sequence, $s^k = \{s_1, \ldots, s_k\}$, and the correspondent matching models sequence, $m^{(i^k)} = \{m^{(i_1)}, \ldots, m^{(i_k)}\}$, $m^{(i_n)} \in \mathbb{M}$, through time k, is denoted as

$$\{s^k = m^{(i^k)}\} = m_{(i^k)}^k = m_{i_1,\ldots,i_k}^k = \{m_1^{i_1}, \ldots, m_k^{i_k}\} = \{m_{(i^{k-1})}^{k-1}, m_k^{i_k}\} \tag{4.15}$$

where $m_{(i^{k-1})}^{k-1}$ is a parent sequence and $m_k^{i_k}$ is the last element. For simplicity the notation $m_{(i^k)}^k \in \mathbb{M}$ will be replaced by $i^k = 1, \ldots, N^k$.

Also of interest, is the definition of the mode transition probability, which given the Markov property, can be written as

$$P\{m_{(i)}^{k-1}, m_k^{(j)}\} = P\{m_{k-1}^i, m_k^j\} \stackrel{\triangle}{=} \pi_{ji} \tag{4.16}$$

under the assumptions A1' and A2' the above equation is equal to (2.27).

The base state estimator under the same assumptions at time k is given by

$$\hat{x}_{k|k} = E[x_k|z_k] = \sum_{i^k=1}^{(N)^k} E[x_k|m_{(i^k)}^k, z^k] P\{m_{(i^k)}^k|z^k\} = \sum_{j=1}^{(N)^k} \hat{x}_{k|k}^{(i^k)} \mu_{(i^k)}^k \tag{4.17}$$

which has an exponential increasing number of terms reveling the impracticability of the CMM optimal approach.

The IMM simplifies the optimal approach by keeping only N filters, i.e., only N hypotheses, resulting in

$$\hat{x}_{k|k} = \sum_{j=1}^{N} E[x_k|m_k^{(j)}, z^k] P\{m_k^{(j)}|z^k\} = \sum_{j=1}^{N} \hat{x}_{k|k}^{(j)} \mu_{(j)}^k \tag{4.18}$$

where the posterior mode-sequence probability is defined by

$$\mu_{(i^k)}^k = P\{m_{(i^k)}^k|z^k\} \tag{4.19}$$

and the posterior mode probability under assumptions A1' and A2' by

$$\mu_k^{(j)} = P\{m_k^{(j)}|z^k\}. \tag{4.20}$$

Like in the previous generation, under the assumptions A1' and A2', the base state estimate is unbiased with covariance given by

$$\begin{aligned} P_{k|k} &= E[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})'|z^k] \\ &\approx \sum_{j=1}^{N} [E[(x_k - \hat{x}_{k|k}^{(j)})(x_k - \hat{x}_{k|k}^{(j)})'|z^k] + (\hat{x}_{k|k} - \hat{x}_{k|k}^{(j)})(\hat{x}_{k|k} - \hat{x}_{k|k}^{(j)})'] P(m_k^{(j)}|z^k) \\ &= \sum_{j=1}^{N} [P_{k|k}^{(j)} + (\hat{x}_{k|k} - \hat{x}_{k|k}^{(j)})(\hat{x}_{k|k} - \hat{x}_{k|k}^{(j)})'] \mu_k^{(j)}. \end{aligned} \tag{4.21}$$

The simplification from equation (4.17) to equation (4.18) and from the first to the second line in the above equation, reflects the approximation that the past z^{k-1} can be explained entirely by a model-set of size N.

The input of each filter matched to model i consists in a mixture of the estimates $\hat{x}_{k-1|k-1}^{(i)}$ with the mixing probabilities $\mu_{k-1}^{j|i}$, i.e.,

$$\begin{aligned} \bar{x}_{k|k}^{(i)} &= E[x_{k-1}|z^{k-1}, m_k^{(i)}] = E\{E[x_{k-1}|m_{k-1}^{(i)}, m_k^{(j)} z^{k-1}]|z^{k-1}, m_k^{(i)}\} \\ &= \sum_{j=1}^{N} \hat{x}_{k-1|k-1}^{(j)} P\{m_{k-1}^{(j)}|z^{k-1}, m_k^{(i)}\} = \sum_{j=1}^{N} \hat{x}_{k-1|k-1}^{(j)} \mu_{k-1}^{j|i} \end{aligned} \tag{4.22}$$

where the estimate $\hat{x}^{(i)}_{k-1|k-1}$ is computed by the KF based on model m_i and mixing probabilities $\mu^{j|i}_{k-1}$ are given by

$$\mu^{j|i}_{k-1} = P\{m^{(j)}_{k-1}|z^{k-1}, m^{(i)}_k\} = \frac{\pi_{ji}\mu^{(j)}_{k-1}}{\sum_i \pi_{ji}\mu^{(j)}_{k-1}}. \tag{4.23}$$

Thus at time step $k-1$ there are only N estimates $\hat{x}^{(i)}_{k-1|k-1}$ ($\forall m_i \in \mathbb{M}$) and the associated covariances $P^{(i)}_{k|k}$ which approximately summarize the past z^{k-1}. This is one of the key features of the IMM algorithm resulting in a low computational complexity and still providing excellent state estimates.

The complete description of IMM algorithm cycle is in table 4.3. The model probability update step is inherited from the previous generation and the final combination step (step 4) is not part of the recursive algorithm but a final mixture for output purposes. A diagram of the IMM algorithm steps is presented in figure 4.10.

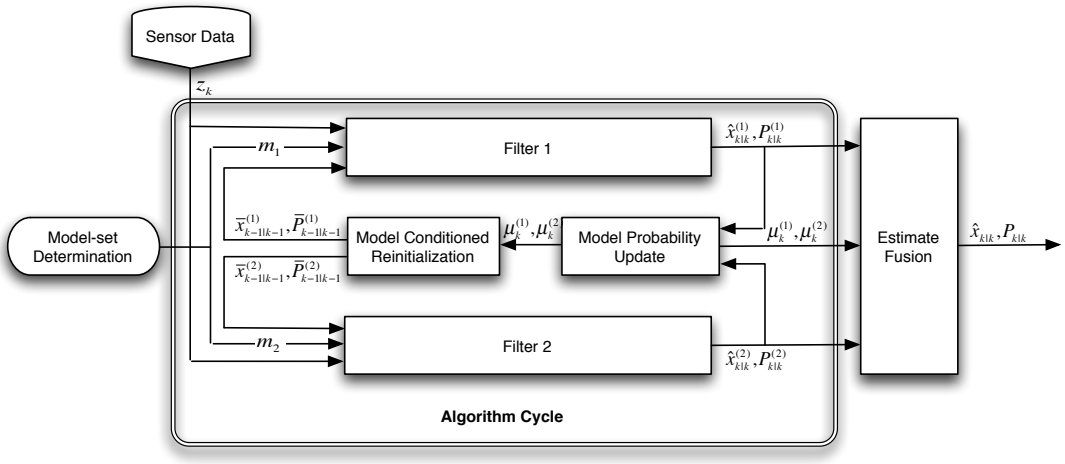


Figure 4.10: General structure of IMM estimation algorithm with two model based filters.

The IMM design parameters, which are once again the key to the algorithm performance, are

- The model set $\mathbb{M}$ structure;
- The system and measurement noises covariances (Q and R respectively);
- The initial state $\hat{x}_{0|0}$ and state covariance $P_{0|0}$;
- The jump structure (usually Markov) and the transition probability π_{ji} between the models from the selected set.

As mention in 4.2.1, one of the main drawbacks of the first generation MM algorithms is its inability to handle mode jumps because each elemental filters are not aware of the overall estimate. The IMM enables mode jumps in two ways: 1) by reinitialization of the filters, each elemental filter computes estimates based on the overall estimate making the residuals smaller and facilitating mode jumps; 2) by introducing a transition probability π_{ji}, which facilitates specific mode jumps, a priori considered to be more likely.

Table 4.3: One Cycle of IMM Algorithm.

1. Model-conditioned reinitialization ($\forall m_i \in \mathbb{M}$) :
   - Predicted model probability: $\mu^{(i)}_{k|k-1} = \sum_j \pi_{ji}\mu^{(j)}_{k-1}$
   - Mixing probabilities: $\mu^{j|i}_{k-1} = \pi_{ji}\mu^{(j)}_{k-1}/\mu^{(i)}_{k|k-1}$
   - Mixing estimate: $\bar{x}^{(i)}_{k-1|k-1} = \sum_j \hat{x}^{(j)}_{k-1|k-1}\mu^{j|i}_{k-1}$
   - Mixing covariance: $\bar{P}^{(i)}_{k-1|k-1} = \sum_j [P^{(j)}_{k-1|k-1} + (\bar{x}^{(i)}_{k-1|k-1} - \hat{x}^{(j)}_{k-1|k-1})(\bar{x}^{(i)}_{k-1|k-1} - \hat{x}^{(j)}_{k-1|k-1})']\mu^{j|i}_{k-1}$
2. Model-conditioned filtering ($\forall m_i \in \mathbb{M}$) :
   - Predicted state: $\hat{x}^{(i)}_{k|k-1} = F^{(i)}_{k-1}\bar{x}^{(i)}_{k-1|k-1}$
   - Predicted covariance: $P^{(i)}_{k|k-1} = F^{(i)}_{k-1}\bar{P}^{(i)}_{k-1|k-1}(F^{(i)}_{k-1})' + Q^{(i)}_{k-1}$
   - Measurement residual: $\tilde{z}^{(i)}_k = z_k - H^{(i)}_k\hat{x}^{(i)}_{k|k-1}$
   - Residual covariance: $S^{(i)}_k = H^{(i)}_k P^{(i)}_{k|k-1}(H^{(i)}_k)' + R^{(i)}_k$
   - Filter gain: $K^{(i)}_k = P^{(i)}_{k|k-1}(H^{(i)}_k)'(S^{(i)}_k)^{-1}$
   - Update state: $\hat{x}^{(i)}_{k|k} = \hat{x}^{(i)}_{k|k-1} + K^{(i)}_k\tilde{z}^{(i)}_k$
   - Update covariance: $P^{(i)}_{k|k} = P^{(i)}_{k|k-1} - K^{(i)}_k S^{(i)}_k(K^{(i)}_k)'$
3. Model probability update ($\forall m_i m_i \in \mathbb{M}$) :
   - Model likelihood: $L^{(i)}_k \overset{assume}{=} \mathcal{N}(\tilde{z}^{(i)}_k; 0, S^{(i)}_k)$
   - Model probability: $\mu^{(i)}_k = \frac{\mu^{(i)}_{k|k-1}L^{(i)}_k}{\sum_j \mu^{(j)}_{k|k-1}L^{(j)}_k}$
4. Estimate fusion:
   - Overall estimate: $\hat{x}_{k|k} = \sum_i \hat{x}^{(i)}_{k|k}\mu^{(i)}_k$
   - Overall covariance: $P_{k|k} = \sum_i [P^{(i)}_{k|k} + (\hat{x}_{k|k} - \hat{x}^{(i)}_{k|k})(\hat{x}_{k|k} - \hat{x}^{(i)}_{k|k})']\mu^{(i)}_k$

The IMM degenerates in the MMSE-AMM (without probability lower bounding) when the IMM's transition probability matrix π_{ij} is the identity matrix.

Other well studied CMM algorithms include the first order Generalized Pseudo-Bayesian (GPB1) and the second order Generalized Pseudo-Bayesian (GPB2) that, along with the IMM, are extensively discussed in [9].

4.3.2 Simulation Example

In this example the target and the model-set have exactly the same configuration as in the example in subsection 4.2.2. The only extra parameter not defined in the example mentioned and that is specific to the IMM algorithm is the transition probability matrix which has the value

$$\pi_{ji} = \begin{bmatrix} 0.99 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0.01 & 0.98 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0.98 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0.98 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0.98 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0.98 & 0.01 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0.99 \end{bmatrix}. \tag{4.24}$$

Note that the algorithm is only allowed to switch from a model to ”neighbor” model, i.e., models with closest angular velocity. This is taken from the target characteristics which is believed to have soft maneuvers. The target is also assumed to have a higher probability in keeping its current angular velocity than switching to another.

An interesting function to observe is the model probability $\mu_k^{(i)}$, see figure 4.11. Note that the maximum model probability saturates in a lower value than in the classic MMSE-AMM and that the probabilities of the models associated to the extreme angular velocities reach higher values, this is due to the format of the transition probability matrix (4.24).

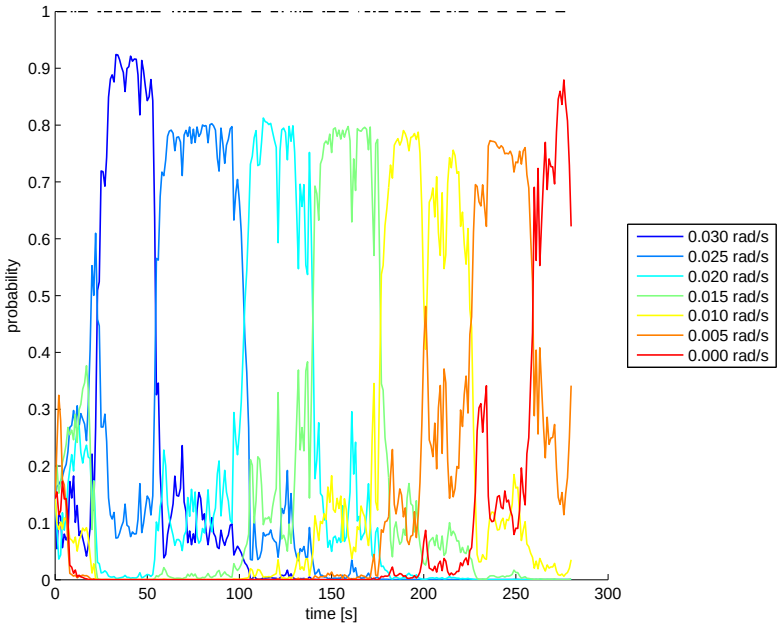


Figure 4.11: IMM model probabilities.

The error for the position and the velocity of the target is in figure 4.12, which is relatively small. The IMM tracker is fast to recognize mode changes, specially if the mode transition probability matrix is well defined, delivering a much smoother estimate and monotonously decreasing estimate error ellipsoids as can be observed in figure 4.13.

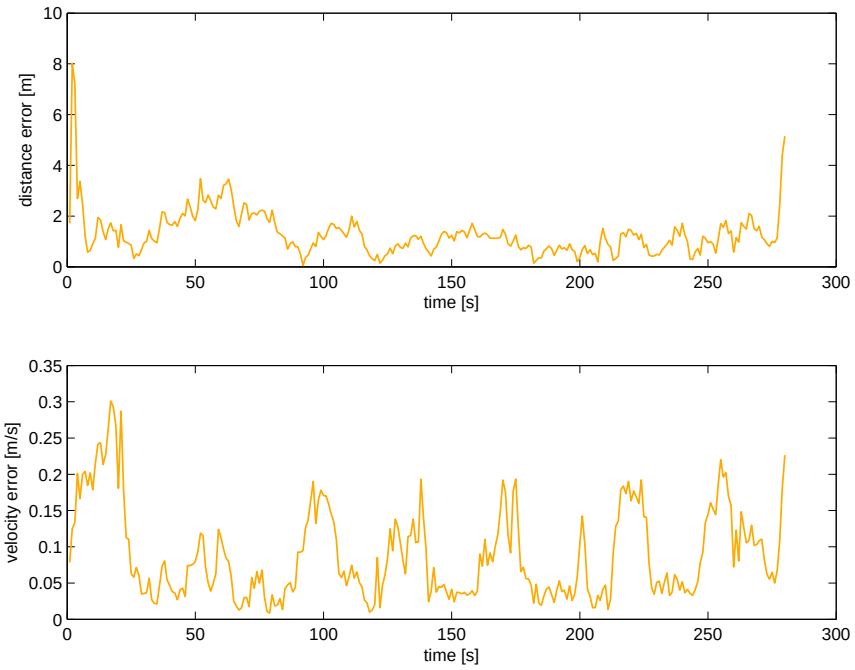


Figure 4.12: IMM position and velocity error.

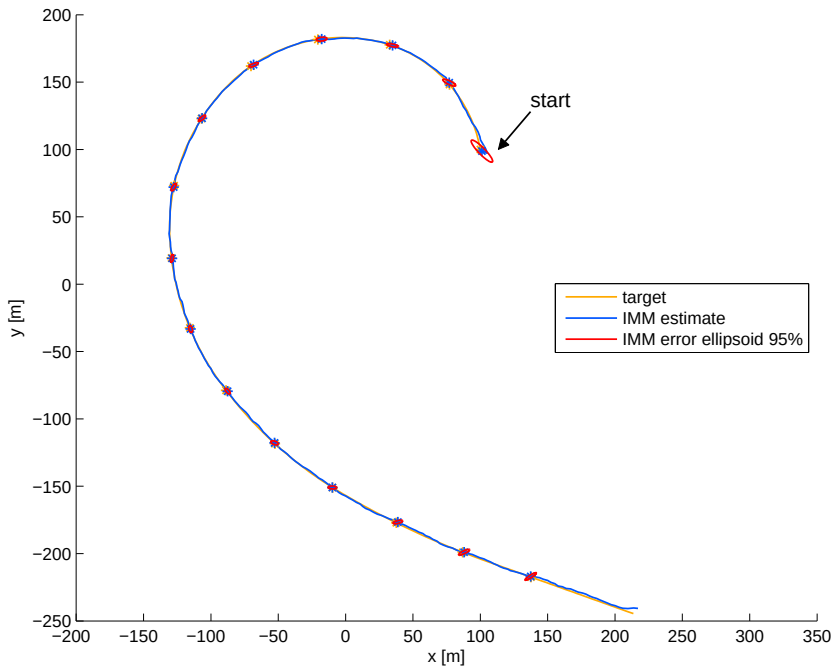


Figure 4.13: Target trajectory and IMM estimate and error ellipsoid propagation.

4.4 Variable Structure Multiple Model

Most existing MM algorithms have a fixed structure (FSMM), i.e., they have a fixed model set at all times. The first two generations of MM algorithms regard this type of model set structure and they have been shown to be quite capable of tracking targets in the presence of structural or parametric uncertainty. However these FSMM algorithms have little more room for improvement, since they have been exhaustively studied and improved for the last decades. In an attempt to break away from the FSMM a new force has been applied into the development of MM estimation with variable structure (VSMM), i. e, with a variable set of models. This approach intends to overcome the FSMM algorithms especially for problems involving many structural modes.

The VSMM algorithm aimed to tackle the estimation problems when the set of all true system models $\mathbb{S}$ is too large and when there is uncertainty in the models of the model set. To add more models to a FSMM may result in a huge computational burden and more, the excessive competition of "unnecessary" models can degrade the overall estimate, as was shown in [4]. The solution is to improve the model set and use a FSMM or, on the other hand, accept the fact that a perfect model set is hard to achieve and use an appropriate VSMM algorithm.

Still under active development, this generation is potentially much more advanced and has an open architecture. The VSMM algorithms' most interesting feature is its ability to adapt to the target maneuvering by creating new elemental filters if the existing ones are not good enough and by eliminating filters with poor performances. A filter with poor performances has its estimate so far away from the correct one that the overall estimate would be improved if that model was not present at all in the model set.

A VSMM should be applied to a tracking problem if the mode space is too large causing an unbearable computational burden for the available resources (possibly by being highly time invariant) or if the mode space is not known (possibly by scarce or highly complex a priori available information).

The third generation abandons the constant mode space, and consequently the fixed model-set ,assumption. Thus the second fundamental assumption of the second generation, which related to a fixed structure of the model set, is eliminated. The only fundamental assumption valid is:

A1". The true mode s is Markov.

The above assumption is maintained because most of the VSMM algorithms incorporate blocks of MM algorithms of the second generation.

A key innovative concept of this generation is the state dependency of a model set, i.e., given a current mode (and base state), the set of possible modes at the next time is a subset of the mode space, determined by the mode transition law.

To describe state dependency, it is usual to resort to graph theory, where each node is a mode and each directed edge a dependency determined by the mode transition law. The state dependency of the model set is the basis of the VSMM algorithms.

As in the previous generation, the optimal VSMM estimator, as described in [3] and [4], is computationally unfeasible. At each time k, there is a new possible mode set $\mathbb{S}_k$ which is the union of the state

dependent mode set at time k, i.e., $\mathbb{S}_k = \{\mathbb{S}_k^{(1)}, \ldots, \mathbb{S}_k^{(n)}\}$ that can be variable sized depending on the previous possible mode set $\mathbb{S}_{k-1}$ and the target's maneuvering restrictions. Thus the set of possible mode sequences through time k, defined as $\mathbb{S}^k = \{\mathbb{S}_1, \ldots, \mathbb{S}_k\}$, grows exponentially.

Most of the applicable VSMM algorithms replace the set of possible mode sequences at time k by one, and hopefully the best, model-set sequence $\mathbb{M}_k$. Note that the size of each state dependent model set may be variable trough time, when this happens a variable size model-set is clearly preferable to a fixed model-set.

This results in the two main tasks of the VSMM algorithm that are

1. Model-set adaptation;
2. MM estimation given a model-set.

To guarantee that no track loss occurs a perfect match between the modes and models at all the time should be assured by the model-set adaptation, i.e., $\{s_k \in \mathbb{M}_k\}\ \forall k$.

Perhaps, the most important task is the development of a good model-set adaptation. The model set adaptation consists of i) Model-set candidation and ii) Model-set decision given candidate sets. As for the model-set decision given candidate sets which mainly includes the termination of models, there exists general solutions with acceptable results based mainly on the models likelihood. The real ongoing not yet concluded study regards the model-set candidation, or model activation, where there are two main streams/classes of VSMM algorithms.

1. The active model-set family has a finite and defined a priori total model set. At each given time, a subset of the total model-set, called active or working model-set, is determined adaptively;
2. The model-set generation family generates the models in real time so it is not possible to define a total model-set in advance.

The MM estimation given the model-set addresses the question on how to initialize new activated models: i) assign initial probabilities to new models, and ii) obtain initial estimates and error covariances for the filters based on new models. The key to the optimal initialization is based on the state dependency of the model set concept, meaning that the new model probability is computed using only the probabilities of the models that may jump to this new model and the initial state estimate and covariance of a filter is computed using only the state estimate and covariance (and probabilities) of the filter based on those models that may jump to that model.

Defining $\mathbb{E}_n$ as a set of the models in $\mathbb{M}_{k-1}$ that are allowed to switch to the new model $m^{(n)}$, i.e.,

$$\mathbb{E}_n = \{m_l : m_l \in \mathbb{M}_{k-1}, \pi_{ln} \neq 0\}. \tag{4.25}$$

The initial state of the filter based on m_n can be obtained by

$$\begin{aligned} \bar{x}_{k-1|k-1}^{(n)} &= E[x_k | m_k^{(n)}, \mathbb{M}^{k-1}, z^{k-1}] \\ &= \textstyle\sum_{m_l \in \mathbb{E}_n} E[x_{k-1} | m_{k-1}^{(l)}, \mathbb{M}^{k-2}, z^{k-1}] P\{m_{k-1}^{(l)} | m_k^{(n)}, \mathbb{M}^{k-1}\} \\ &= \textstyle\sum_{m_l \in \mathbb{E}_n} \hat{x}_{k-1|k-1}^{(l)} \mu_{k-1}^{l|n} \end{aligned} \tag{4.26}$$

where the sequence at time k of sets of the total model-set is $\mathbb{M}^k = \{\mathbb{M}_1, \ldots, \mathbb{M}_k\}$ and

$$\mu_{k-1}^{l|n} = P\{m_{k-1}^{(l)} | m_k^{(n)}, \mathbb{M}^{k-1}\} = \frac{\pi_{ji}\mu_{k-1}^{(j)}}{\sum_{m_j \in \mathbb{E}_n} \pi_{ji}\mu_{k-1}^{(j)}}. \tag{4.27}$$

The assignment of initial model probabilities for new models follows the same logic

$$P\{m_{k-1}^{(n)} | \mathbb{M}^{k-1}, z^{k-1}\} = \sum_{m_j \in \mathbb{E}_n} P\{m_k^{(n)} | m_{k-1}^{(j)}\} P\{m_{k-1}^{(j)} | \mathbb{M}^{k-1}, z^{k-1}\}. \tag{4.28}$$

Analyzing the equations (4.26), (4.27) and (4.28) it is clear the similarity with the second generation IMM algorithm. This conclusion leads to the recursive variable structure version of the IMM algorithm called Variable Structure Interacting Multiple Model (VSIMM). The general cycle of the VSIMM algorithm based on time-varying model-set is presented in table 4.4.

Table 4.4: One Cycle of VSIMM $[\mathbb{M}_k, \mathbb{M}_{k-1}]$ Algorithm.

1. Model-conditioned (re)initialization $(\forall m_i \in \mathbb{M}_k)$:
   - Predicted model probability: $\mu_{k|k-1}^{(i)} = \sum_{m_j \in \mathbb{M}_{k-1}} \pi_{ji}\mu_{k-1}^{(j)}$
   - Mixing weight: $\mu_{k-1}^{j|i} = \pi_{ji}\mu_{k-1}^{(j)} / \mu_{k|k-1}^{(i)}$
   - Mixing estimate: $\bar{x}_{k-1|k-1}^{(i)} = \sum_{m_j \in \mathbb{M}_{k-1}} \hat{x}_{k-1|k-1}^{(j)} \mu_{k-1}^{j|i}$
   - Mixing covariance: $\bar{P}_{k-1|k-1}^{(i)} = \sum_{m_j \in \mathbb{M}_{k-1}} [P_{k-1|k-1}^{(j)} + (\bar{x}_{k-1|k-1}^{(i)} - \hat{x}_{k-1|k-1}^{(j)})(\bar{x}_{k-1|k-1}^{(i)} - \hat{x}_{k-1|k-1}^{(j)})']\mu_{k-1}^{j|i}$
2. Model-conditioned filtering $(\forall m_i \in \mathbb{M}_k)$:
   - Predicted state: $\hat{x}_{k|k-1}^{(i)} = F_{k-1}^{(i)} \bar{x}_{k-1|k-1}^{(i)}$
   - Predicted covariance: $P_{k|k-1}^{(i)} = F_{k-1}^{(i)} \bar{P}_{k-1|k-1}^{(i)} (F_{k-1}^{(i)})' + Q_{k-1}^{(i)}$
   - Measurement residual: $\tilde{z}_k^{(i)} = z_k - H_k^{(i)} \hat{x}_{k|k-1}^{(i)} - \bar{v}_k^{(i)}$
   - Residual covariance: $S_k^{(i)} = H_k^{(i)} P_{k|k-1}^{(i)} (H_k^{(i)})' + R_k^{(i)}$
   - Filter gain: $K_k^{(i)} = P_{k|k-1}^{(i)} (H_k^{(i)})' (S_k^{(i)})^{-1}$
   - Update state: $\hat{x}_{k|k}^{(i)} = x_{k|k-1}^{(i)} + K_k^{(i)} \tilde{z}_k^{(i)}$
   - Update covariance: $P_{k|k}^{(i)} = P_{k|k-1}^{(i)} - K_k^{(i)} S_k^{(i)} (K_k^{(i)})'$
3. Model probability update $(\forall m_i \in \mathbb{M}_k)$:
   - Model likelihood: $L_k^{(i)} \overset{assume}{=} \mathcal{N}(\tilde{z}_k^{(i)}; 0, S_k^{(i)})$
   - Model probability: $\mu_k^{(i)} = \frac{\mu_{k|k-1}^{(i)} L_k^{(i)}}{\sum_{m_j \in \mathbb{M}_k} \mu_{k|k-1}^{(j)} L_k^{(j)}}$
4. Estimate fusion:
   - Overall estimate: $\hat{x}_{k|k} = \sum_{m_i \in \mathbb{M}_k} \hat{x}_{k|k}^{(i)} \mu_k^{(i)}$
   - Overall covariance: $P_{k|k} = \sum_{m_i \in \mathbb{M}_k} [P_{k|k}^{(i)} + (\hat{x}_{k|k} - \hat{x}_{k|k}^{(i)})(\hat{x}_{k|k} - \hat{x}_{k|k}^{(i)})'] \mu_k^{(i)}$

Note that the model transition probability π_{ij} regards the total model-set and not only the current active model-set.

Also included in the VSMM task of MM estimation given a model-set is the fusion of two MM estimation.

This situation arises when after obtaining the MM estimation for a model-set $\mathbb{M}_1$ the model-set adaptation algorithm decides to add to it a new set of models $\mathbb{M}_2$. The efficient solution is to add the new set in the current cycle avoiding repeating the already performed computation, this is achieved by having two distinct MM estimators and a fusion algorithm. Both MM estimators have the same model-set history $\mathbb{M}^{k-1}$ but a different active model sets $\mathbb{M}_1$ and $\mathbb{M}_2$ characterized by

$$\{\hat{x}_{k|k}^{(i)}, P_{k|k}^{(i)}, L_k^{(i)}, \mu_k^{(i)}\}_{m_i \in \mathbb{M}_l} \quad l = \{1, 2\} \tag{4.29}$$

which can be computed, for instance, by a cycle of the VSIMM $[\mathbb{M}_l, \mathbb{M}_{k-1}] \quad l = \{1, 2\}$ algorithm (table 4.4).

The optimal MM estimator base on the model set $\mathbb{M}_k = \mathbb{M}_1 \cup \mathbb{M}_2$ and the common model-set history $\mathbb{M}^{k-1}$ is given by

$$\hat{x}_{k|k} = \sum_{m_i \in \mathbb{M}_k} \hat{x}_{k|k}^{(i)} \mu_k^{(i)} \tag{4.30}$$

$$P_{k|k} = \sum_{m_i \in \mathbb{M}_k} P_{k|k}^{(i)} [(\hat{x}_{k|k} - \hat{x}_{k|k}^{(i)})(\hat{x}_{k|k} - \hat{x}_{k|k}^{(i)})'] \mu_k^{(i)} \tag{4.31}$$

where

$$\mu_k^{(i)} \triangleq P\{m_k^{(i)} | \mathbb{M}^k = \mathbb{M}_1 \cup \mathbb{M}_2, \mathbb{M}^{k-1}, z^k\} = \frac{\mu_{k|k-1}^{(i)} L_k^{(i)}}{\sum_{m^{(j)} \in \mathbb{M}_k} \mu_{k|k-1}^{(j)} L_k^{(j)}}. \tag{4.32}$$

If there are common models in $\mathbb{M}_1$ and $\mathbb{M}_2$, i.e., $\mathbb{M}_1 \cap \mathbb{M}_2 \neq \emptyset$,extra computation is needed to calculate the value of the model probability $\mu_k^{(i)}$.

The section 4.4.1 describes the Likely Model Set (LMS) algorithm.

4.4.1 The Likely Model-Set algorithm

The LMS algorithm, described in [8], belongs to the family of active model set of the VSMM algorithms. When compared to other VSMM algorithms like Recursive Adaptive Model-Set (RAMS) [7] and the Model-Group Switching (MGS) [5], the LMS algorithm can be applied to a larger class of hybrid estimation problems, it is simpler with fewer design parameters, it is more cost effective. Its basis is to use a subset of the total model set as the active set for any given time. The total model set is finite and can be determined offline, prior to any measurements. The active or working model-set is determined adaptively. On the LMS algorithm this is achieved by implementing a simple rule of: 1) discarding the unlikely models; 2) keeping the significant models; and 3) activating the models adjacent to the principal models.

The adjacency of models is determined by wether one model is allowed to switch to another, if so, the first model is adjacent to the second one. By an appropriated setting of parameters, the number of principal models at any given time can be small and the number of unlikely models as large as needed.

This means that the number of active models at time step k can be much smaller than the total model set leading to a computational complexity saving and possibly to an improvement in estimation.

The LMS algorithm, regarding the VSMM tasks, can be decomposed in

1. Model-set adaptation
   (a) Model classification: Identify each model in $\mathbb{M}_{k-1}$ to be unlikely (if its probability is below the threshold t_1) , principal (if its probability exceeds the threshold t_2) or significant (if its probability is between t_1 and t_2);
   (b) Model-set adaptation: Obtain $\mathbb{M}_k$ by deleting all the unlikely models in $\mathbb{M}_{k-1}$ and by activating all the models adjacent from any principal models in $\mathbb{M}_{k-1}$.
2. MM estimation given a model-set
   (a) Model-set sequence conditioned estimator: Obtain at least one MM estimative using VSIMM algorithm based on the previous model-set $\mathbb{M}_{k-1}$ and a new model-set. The new and required model-set has all the models in $\mathbb{M}_{k-1}$ minus the unlikely ones. The optional new model-set contains all the models adjacent from the principal models if they did not belong to the new and required model-set;
   (b) Fusion of two MM estimators (if existing): Obtain the optimal estimation by fusing two MM estimators applying equations (4.30), (4.31) and (4.32).

The LMS algorithm has two variants regarding the logic of models deletion: 1) AND logic: A model is deleted only if its probability is both below the threshold t_1 and not among the K largest. This means that the model-set has at least K models; 2) OR logic: A model is deleted if its probability is below the threshold t_1 or it is not among the K largest. This means that the model-set has always K models.

The AND logic guarantees performance and relaxes computation while the OR logic guarantees computation but relaxes performance; the AND logic is more interesting. In table 4.5 is a description of the steps in a cycle of the LMS algorithm with AND logic. The flowchart diagram representing the LMS algorithm's process and decision steps is in figure 4.14.

As mentioned before, and confirmed in the algorithm description, one of the advantages of the LMS algorithm is the reduced number of design parameters. These are the same as for the IMM algorithm (since it previous uses a variation of the latter) with the addiction of the following parameters

- The unlikely and principal models probability thresholds, t_1 and t_2 respectively.

4.4.2 Simulation Example

In this example the target and the model-set have exactly the same configuration as in the example in subsection 4.3.2.

The total model-set is a priori defined, but at each time step only a subset of the total model-set is used. This subset has a minimum of $K = 3$ active models and can contain models with a value for the angular velocity from table 4.2.

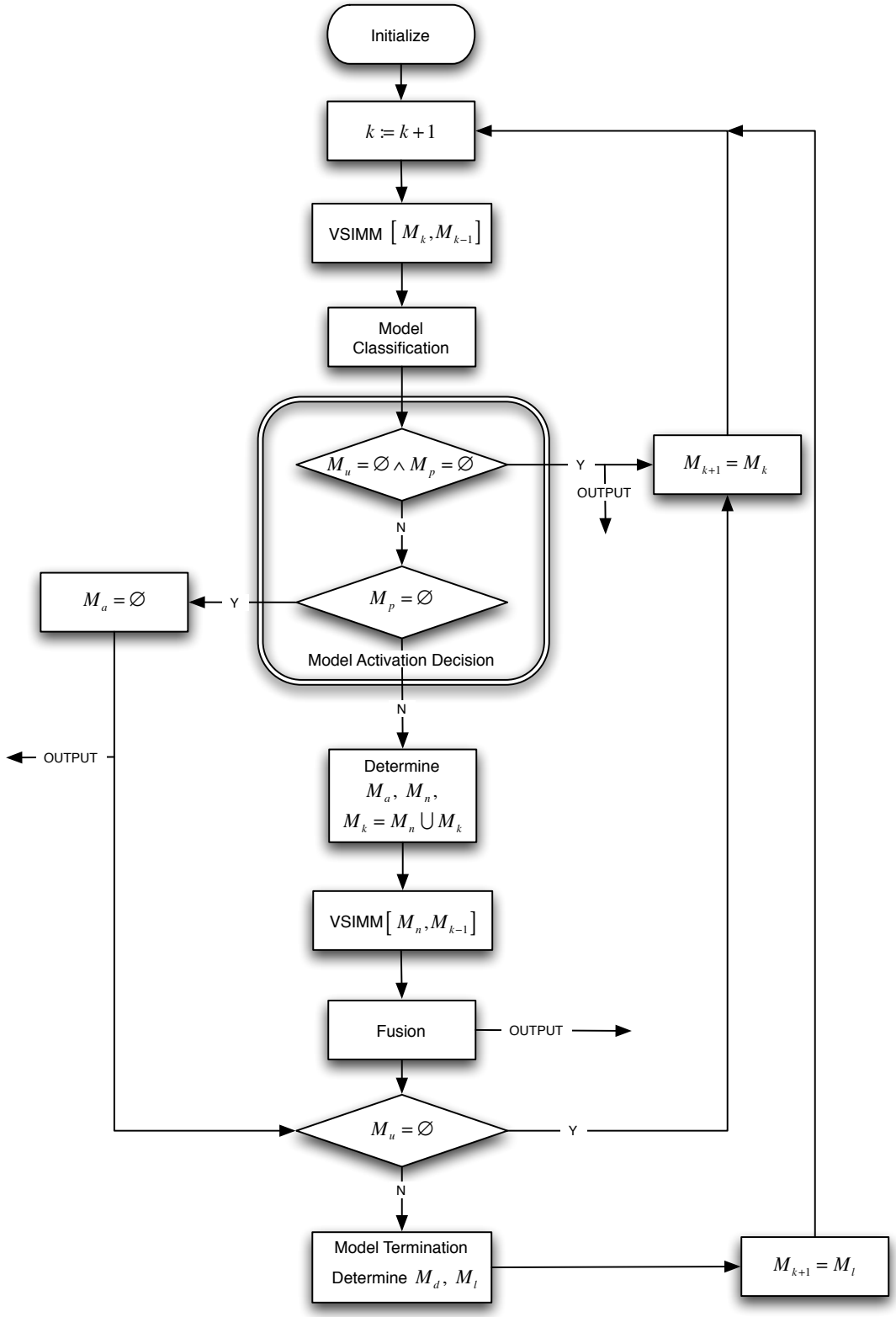


Figure 4.14: Flowchart of LMS estimation algorithm.

Table 4.5: One Cycle of LMS Algorithm.

1. Increase the time counter k by 1. Run the VSIMM $[\mathbb{M}_k, \mathbb{M}_{k-1}]$ cycle.
2. Classify all the models m_i's in $\mathbb{M}_k$ to be principal (i.e., $\mu_k^i > t_2$), unlikely (i.e., $\mu_k^i < t_1$) or significant (i.e., $t_1 \geq \mu_k^i \leq t_1$). Let the set of unlikely models be $\mathbb{M}_u$. If there is neither unlikely nor principal model, output $\hat{x}_{k|k}$, $P_{k|k}$ and $\{\mu_k^{(i)}\}_{m^{(i)} \in \mathbb{M}_k}$, let $\mathbb{M}_{k+1} = \mathbb{M}_k$ and go to step 1.
3. If there is no principal model, then let $\mathbb{M}_a = \emptyset$ and got to step 4. Otherwise, identify the set $\mathbb{M}_a$ of all models adjacent to any principal model. Find the set of new models $\mathbb{M}_n = \mathbb{M}_a \bigcap \bar{\mathbb{M}}_k$ (where $\bar{\mathbb{M}}_k$ is the complement of $\mathbb{M}_k$ and the union set $\mathbb{M}_k := \mathbb{M}_n \bigcup \mathbb{M}_k$.
   Then
   - Run VSIMM $[\mathbb{M}_n, \mathbb{M}_{k-1}]$ cycle, where $\mathbb{M}_n$ is the set of new and only new models.
   - Fusion: Calculate the estimates, error covariances, and mode probabilities for the union set $\mathbb{M}_k$:

$$\mu_k^{(i)} = \frac{\mu_{k|k-1}^{(i)} L_k^{(i)}}{\sum_{m_j \in \mathbb{M}_k} \mu_{k|k-1}^{(j)} L_k^{(j)}}, \forall m_i \in \mathbb{M}_k$$

$$\hat{x}_{k|k} = \sum_{m_i \in \mathbb{M}_k} \hat{x}_{k|k}^{(i)} \mu_k^{(i)}$$

$$P_{k|k} = \sum_{m_i \in \mathbb{M}_k} P_{k|k}^{(i)} [(\hat{x}_{k|k} - \hat{x}_{k|k}^{(i)})(\hat{x}_{k|k} - \hat{x}_{k|k}^{(i)})'] \mu_k^{(i)}$$

   where the estimates $\{\hat{x}_{k|k}^{(i)}\}$, error covariances $\{P_{k|k}^{(i)}\}$, likelihoods $\{L_k^{(i)}\}$, and predicted probabilities $\{\mu_{k|k-1}^{(j)}\}$ were obtained in the above VSIMM $[\mathbb{M}_k, \mathbb{M}_{k-1}]$ and VSIMM $[\mathbb{M}_n, \mathbb{M}_{k-1}]$ cycles.

4. Output $\hat{x}_{k|k}$, $P_{k|k}$ and $\{\mu_k^{(i)}\}_{m^{(i)} \in \mathbb{M}_k}$.
5. If there is no unlikely model, go to step 1; otherwise, identify the discardable set $\mathbb{M}_d = \mathbb{M}_u \bigcap \bar{\mathbb{M}}_a$, that is, the set of unlikely models that are not adjacent from any principal model.
6. Eliminate the models in $\mathbb{M}_d$ from $\mathbb{M}_k$ that have the smallest probability such that $\mathbb{M}_k$ has at least K models, that is, let the likely model-set be $\mathbb{M}_l = \mathbb{M}_k - \mathbb{M}_m$, where $\mathbb{M}_m$ is the set of models in $\mathbb{M}_d$ with smallest probabilities such that $\mathbb{M}_l$ has at least K models.
7. Let $\mathbb{M}_{k+1} = \mathbb{M}_l$ and go to step 1.

For more clarity, the initial state estimate is redefined, based on equations (4.8) and (4.9),

$$\hat{x}_{0|0}^{(i)} \sim \mathcal{N}(x(0), P_{0|0}^{(i)}) \ \ \forall m_i \in \mathbb{M}_{\mathbb{O}} \tag{4.33}$$

with initial state covariance estimate $P_{0|0}^{(i)}$ given by

$$P_{0|0}^{(i)} = \begin{bmatrix} 10^2 & 0 & 0 & 0 \\ 0 & 0.2^2 & 0 & 0 \\ 0 & 0 & 10^2 & 0 \\ 0 & 0 & 0 & 0.2^2 \end{bmatrix} \ \forall m_i \in \mathbb{M}_{\mathbb{O}}. \tag{4.34}$$

where $\mathbb{M}_{\mathbb{0}}$ is the initial active model-set, which for this simulation, is considered to be the total model-set.

The transition probability matrix is the same as the one defined by the equation (4.24) and all the models are considered to have equal a priori probability of being correct, i.e., the a priori model probability is given by (4.10), which can also be redefined by

$$\mu_0^{(i)} = \frac{1}{N_0}, \ \ \forall m_i \in \mathbb{M}_{\mathbb{0}} \tag{4.35}$$

where N_0 is the size of the initial active model set $\mathbb{M}_{\mathbb{0}}$, in this case, is equal to the total model-set size, i.e., $N_0 = N$.

The digraph associated with the total the model-set is quite linear and represented in figure 4.15, along with the correspondent transition probabilities.

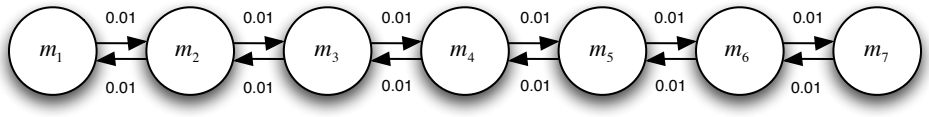


Figure 4.15: Digraph associated with the total the model-set.

By maintaing the size of the total model-set the main advantage will be the reduction of computer complexity. As mention before, the VSMM algorithms capability of providing a better estimates than the previous generations MM algorithms is most notorious when large models sets are needed.

In figure 4.16, it is possible to identify the models active at each time step. Note that there are times when the active model-set has more than three models. The number of active models is influence by the model probability thresholds. In this example, has the model-set is relatively small this thresholds can be relaxed, if they were more limitative the model-set would always be of size three.

The chosen thresholds, which are the only extra parameters needed to be define, were

$$t_1 = 0.1, \ \ t_2 = 0.4. \tag{4.36}$$

Although, these thresholds, guarantee a that the correct model is active during almost all the simulation, at the beginning, when the uncertainty is high and an incorrect model can have the highest probability, the correct model is discarded. A solution, to compensate this flaw, could be the definition of pre and after stabilization thresholds, specially if there is a considerable uncertainty in the target initial state. The pre stabilization thresholds should be more conservative, and not discard the models so soon, i.e., $t_1 << 0.1$.

The position and velocity error in this circumstances is represented in figure 4.17, which besides the initial state estimate uncertainty, is quite small. The representation of the target trajectory and LMS position estimate and error ellipsoid propagation in the state space with time will be omitted since it is similar to the IMM response in figure 4.13

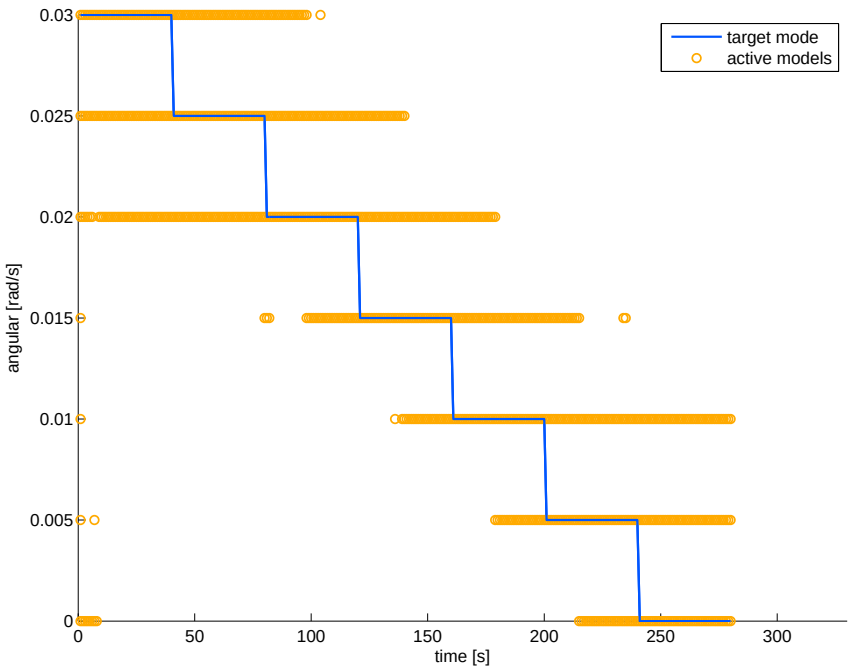


Figure 4.16: LMS active models.

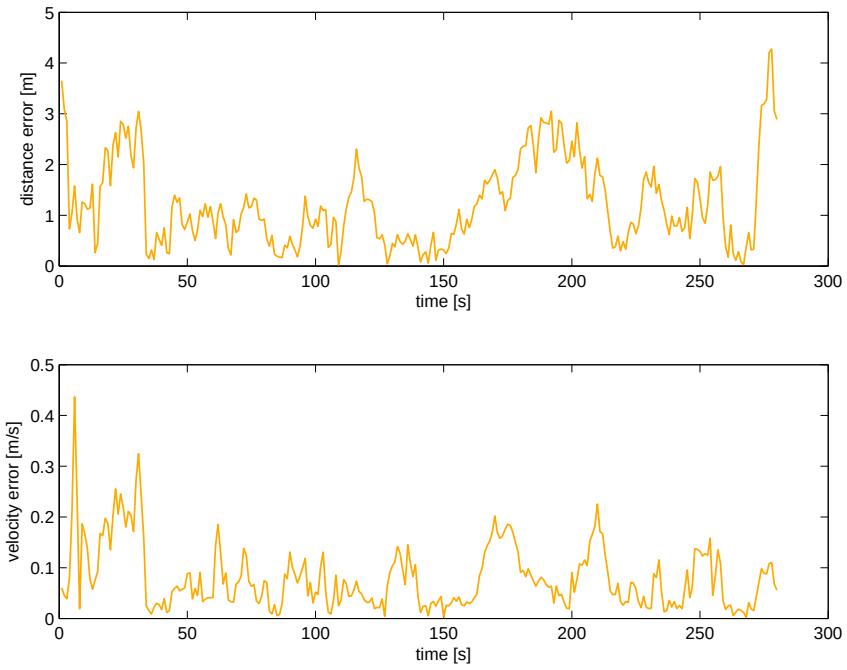


Figure 4.17: LMS position and velocity error.

4.5 Comparison of the MM algorithms

In this section a total of 30 simulations were made and the mean value of them was used to compute the results shown below.

Regarding the target trajectory the initial position of the target is given by (3.27) with

$$\mathrm{x}(0) = 100 \text{ m}, \;\; \mathrm{y}(0) = -100 \text{ m}, \;\; v = 1 \text{ m.s}^{-1} \;\; \text{and} \;\; \phi(0) = 45^{\circ}. \tag{4.37}$$

It is in CT mode described in subsection 2.2.2 with an angular velocity represented in figure 4.18 and given by

$$\omega_k = \begin{cases} 0.03 & 1 \leq k \leq 40 \\ -0.06\frac{k-40}{1000}k & 41 \leq k \leq 1040 \\ -0.03 & 1041 \leq k \leq 1080 \end{cases}, \quad \text{in rad.s}^{-1}. \tag{4.38}$$

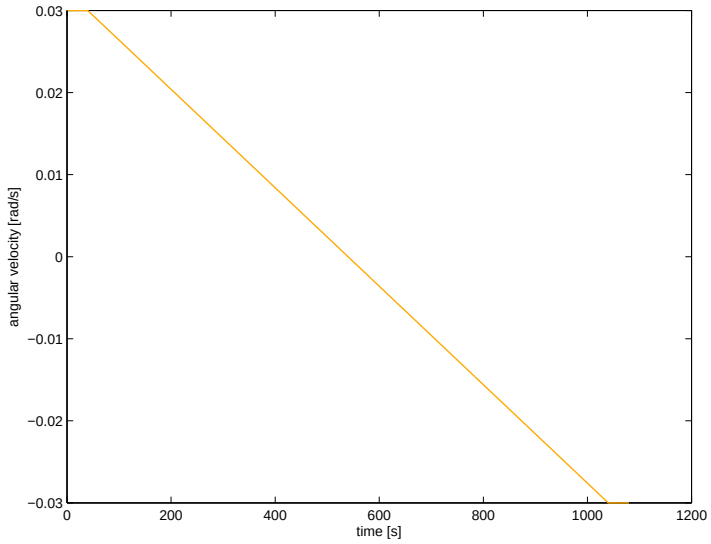


Figure 4.18: Non-constant and smoothed angular velocity of the target.

Note that the target's motion is smoothed and the current target mode may or may not be present in the mode-set. Under this motion non of the three generations of the MM algorithms are in optimal conditions. Tough, as will be observes later, the MM algorithms are still capable of tracking the target with satisfactory results.

Regarding the model-set, all the MM algorithms have the same model-set composed of thirteen models ($N = 13$) each associated with one of the angular velocities in table 4.6, where the positive sign means a anti-clockwise turn and the negative sign means clockwise turn.

The option to alter the values of the angular velocity was to allow a larger range of trajectories for target and better test the algorithms capability.

All the models are considered to have equal a priori probability of being correct, i.e., the a priori model

Table 4.6: Angular velocity of the thirteen models.

Angular velocity ω [rad.s^{-1}]
0.03
0.025
0.02
0.015
0.01
0.005
0
-0.005
-0.01
-0.015
- 0.02
-0.025
-0.03

probability for the FSMM and the VSMM algorithms is given by (4.10) and (4.35), respectively. The initial active model-set in the LMS algorithm is the total model-set.

The measurement noise covariance of the models is equal to the target's, while the process noise covariance is little higher and given by equation (3.35) like in previous simulations where there were target motion had mode switches.

Once again, for a realistic scenario, the filters do not know the initial state, only its distribution; thus each of the filters for each of the simulations has an initial state given by 4.8 and with variance given by 4.9. For the LMS algorithm, the equations are redefined to (4.33) and (4.34), respectively.

Regarding specific parameters of the MM algorithms, the values chosen are

- the MMSE-AMM model probability lower bound is of 10^{-3};
- the IMM and LMS transition probability matrix, of size (13×13), is given by

$$\pi_{ji} = \begin{bmatrix} 0.98 & 0.02 & 0 & \dots & 0 \\ 0.01 & 0.98 & 0.01 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0.01 & 0.98 & 0.01 \\ 0 & \dots & 0 & 0.02 & 0.98 \end{bmatrix}; \tag{4.39}$$

- the LMS thresholds were $t_1 = 0.01$ and $t_2 = 0.4$ and the minimum number of active models is $K = 3$.

The target trajectory and the mean of all simulations of the position estimate delivered from the three MM algorithms is represented in figure 4.19.

In this experiments, the mean of the MSE of the state vector in all simulations throughout the time of the estimate error was also calculated. Those values are represented in figure 4.20, and their mean values in table 4.7.

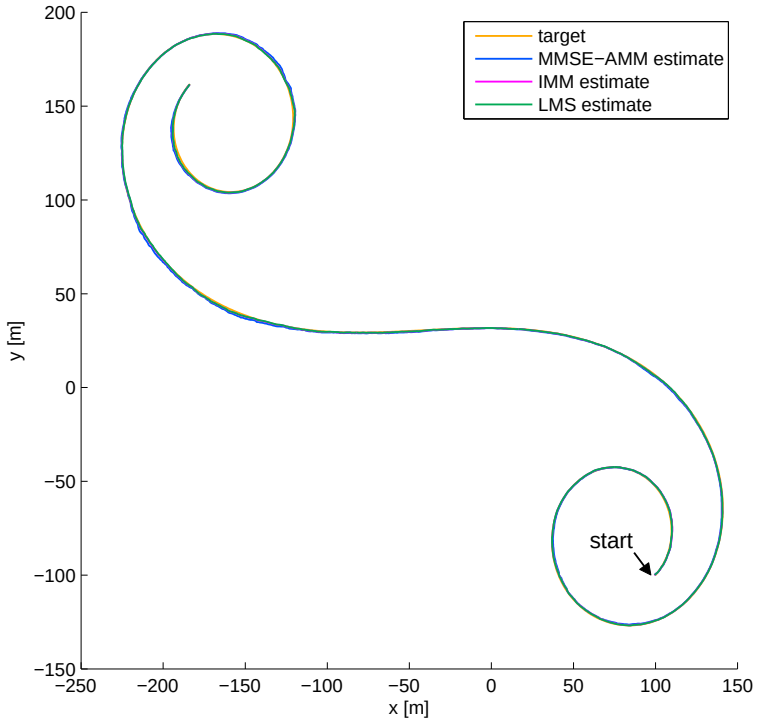


Figure 4.19: Target trajectory and the three generations of MM algorithms mean position estimate for 30 simulations.

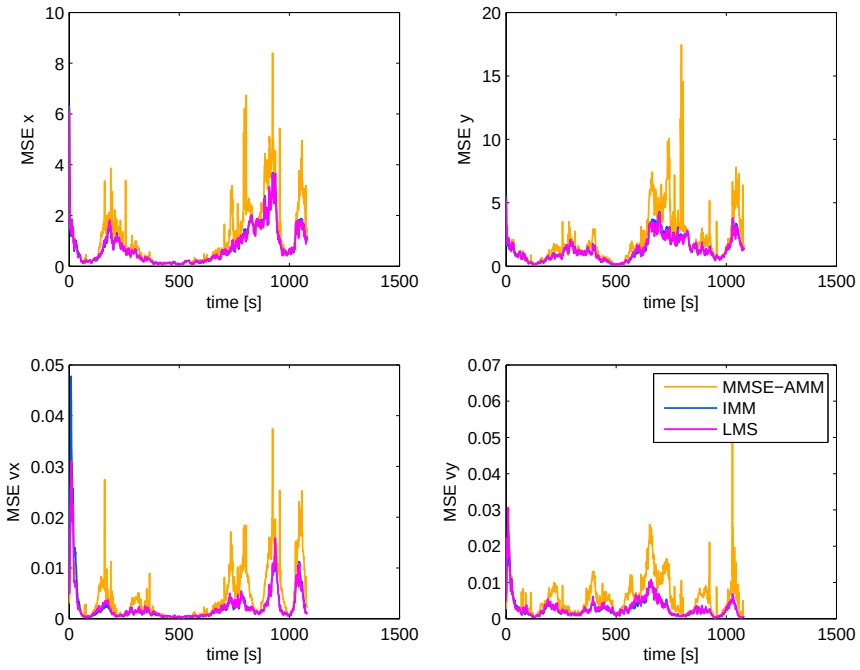


Figure 4.20: The three generations of MM algorithms' mean of the state MSE for 30 simulations.

Table 4.7: Mean state MSE for 30 simulations throughout the time.

Algorithm	MSE x [m]	MSE y [m]	MSE v_x [m/s]	MSE v_y [m/s]
MMSE-AMM	1.0729	1.9382	0.0044	0.0056
IMM	0.7424	1.2809	0.0025	0.0029
LMS	0.7425	1.2686	0.0023	0.0030

The superiority of the second and third generations over the first generation is obvious.

Regarding the second generation algorithm, the IMM, it is specially interesting since the implementing complexity doesn't suffer a tremendous increase and its specific parameters are of easy definition, delivering good estimates even when a fair dimensioning is made.

On the other had the third generation, the LMS, even when using a limited number of active models is able to deliver good estimates with a very low computer complexity. Its ability to overcome the second generation results isn't as obvious as the difference between the first two generations, and it highly depends on the choice of the simulation parameters. There is still a big need for improvement but this algorithm, but as it is now, it presents as a very good alternative to the IMM algorithm when dealing with large model-sets.

Chapter 5

Multiple Target Tracking

This chapter extends the previous studied target tracking problem to the scenario where multiple targets are detected by the sensory system. This chapter includes the main contribution of this thesis, since the new developed algorithm is here described.

In section 5.1, the problem of multiple target tracking and the main differences from the single target tracking problem are presented. Also some of the most common algorithms to solve the data association issue are mentioned.

In sections 5.2 and 5.3, two algorithms are presented and studied in depth. The first algorithm consists of a combination of the Interacting Multiple Model and the Joint Probabilistic Data Association Filter. The second algorithm is the new proposed algorithm that combines the Likely Model-Set with the Joint Probabilistic Data Association Filter. The tracking abilities of both algorithms are discussed and simulated separately in subsections 5.2.1 and 5.3.1. The section 5.4 contains a comparison of the two algorithms, regarding their simulated results under the same scenario.

5.1 Tracking Multiple Targets

The Multiple Target Tracking (MTT) problem extends the single target tracking to a situation where the number of targets may not be known and can also be variable variable with time. Moreover, the measurements obtained are also not known, since they can be originated from any of the targets. False alarms or measurements originated from clutter are an extra source of complexity in realistic applications. Thus, tracking multiple maneuvering targets is much more difficult than tracking a single maneuvering target since there is a challenge in correctly associating the measurements with the targets.

The MTT problem is composed of the following subproblems

- Single Target Tracking (STT): addresses the problem of tracking a single target. In the previous chapter, tracking a maneuvering target using MM has been shown to be highly effective. As studied earlier the STT estimates the positions of a target based on noisy observations of the target and with some prior regarding the target and sensor characteristics;
- Data association (DA): usually consists in ensuring that the correct measurement is given to each STT tracker so that the trajectories of each target can be accurately estimated.

The strategies to tackle the STT tracking problem resorting to MM were discussed in chapter 4. The DA problem has three classical approaches: the Nearest Neighbour Standard Filter [23], the Joint Probabilistic Data Association Filter [1], and the Multiple Hypothesis Tracking Filter [1].

In the following section 5.2.1, the STT algorithm IMM is combined with the DA filter Joint Probabilistic Data Association (JPDAF). In section 5.3 a new algorithm is described combining the LMS with the JPDAF algorithm.

5.2 Interacting Multiple Model Joint Probabilistic Data Association Filter

The algorithm presented in this section combines two well known approaches in target tracking: the IMM algorithm, discussed in 4.3.1 shown to be quite effective in estimating the state of highly maneuvering targets, and the JPDAF for resolving the MTT problem of data association. The IMM-JPDAF combination was introduced in literature in [2].

Assume that there are a total T targets r ($r = \{1, \ldots, T\}$). At each time step k there are total of n measurements $z_k^{(j)}$ ($z_k^{(j)} \in \mathbb{Z}_k = \{z_k^1, \ldots, z_k^{n_k}\}$), only $n_k \leq n$ of this measurements are considered valid.

A measurement $z_k^{(j)}$ is valid if at least one target r lies inside the validation gate $G_k(r)$. The validation gate of a target r is taken to be same for all models in $\mathbb{M}$ and chosen as the largest of them. The validation gate for each target r is given by

$$G_k(r) = \{y_k^{(j)} = z_k^{(l_j)} : [z_k^{(l_j)} - \hat{z}_{k|k-1}^{(i_r),(r)}]'(S_k^{(i_r),(r)})^{-1}[z_k^{(l_j)} - \hat{z}_{k|k-1}^{(i_r),(r)}] \leq \gamma\} \tag{5.1}$$

where i_r is the index of the model in the model set corresponding to the largest residual covariance

$$i_r := \text{argmax}_{i \in \mathbb{M}} \det(S_k^{(i)}). \tag{5.2}$$

Also, γ is an appropriated threshold, $S_k^{(i_r),(r)} = H_k^{(i_r)} P_{k|k-1}^{(i_r),(r)} (H_k^{(i_r)})' + R_k^{(i_r)}$ is the largest model-conditioned residual covariance for the target r and $\hat{z}_{k|k-1}^{(i_r),(r)} = H_k^{(i_r)} \hat{x}_{k|k-1}^{(i_r),(r)}$ is the correspondent predicted measurement.

Note that setting of the validation gate threshold value too low may make cause the algorithm to not associate any measurement with a target, and setting the value too high increases the computational burden as the algorithm is always trying to resolve target-measurement association. Ideally γ should allow the IMM-JPDAF to work as T isolated IMM algorithms when targets are far apart.

The set of validated measurements is denominated $\mathbb{Y}_k = \{y_k^1, \ldots, y_k^{n_k}\}$ with $n_k \leq n$.

The key to the JPDAF algorithm is the definition of the marginal events θ_{jr} and the evaluation of their conditional joint probabilities. A marginal association event θ_{jr} is said to be effective at time k when a validated measurement $y_k^{(j)}$ is associated with a target r, i.e., $y_k^{(j)} \in G_k(r)$. A joint association event Θ happens when a set of marginal events holds true simultaneously, i.e.,

$$\Theta = \bigcap_{j=1}^{n_k} \theta_{jr_j} \tag{5.3}$$

where r_j is the index of the target to which the measurement $y_k^{(j)}$ is associated with.

The validation matrix is given by

$$\Omega = [\omega_{jr}],\ j \in \{1 \ldots n_k\} \text{ and } r \in \{1 \ldots T\} \tag{5.4}$$

where ω_{jr} is a binary variable indicating whether measurement j lies in the validation gate of target r in event Θ.

Based on the validation matrix Ω, an association event Θ may be represented by the matrix

$$\hat{\Omega} = [\hat{\omega}_{jr}],\ j \in \{1 \ldots n_k\} \text{ and } r \in \{1 \ldots T\} \tag{5.5}$$

where

$$\hat{\omega}_{jr} = \begin{cases} 1 & \text{if } \theta_{jr} \subset \Theta \\ 0 & \text{otherwise} \end{cases}. \tag{5.6}$$

A feasible association event is one where each measurement has only one source, i.e., $\sum_{r=0}^{N} \hat{\omega}_{jr} = 1\ \forall j$ (where $r = 0$ indicates the measurement is a false alarm, originated from clutter for instances) and where at most one measurement is originate from each target $\delta_r := \sum_{j=0}^{n_k} \hat{\omega}_{jr} \leq 1\ (\forall r \in \{1, \ldots, T\})$ also called target indicator.

Other important indicators using the permutation matrix information are the binary measurement association indicator, given by $\tau_j := \sum_{r=1}^{N} \hat{\omega}_{jr}$, with $j = \{1, \ldots, n_k\}$ which indicates whether the validated

measurement $y_k^{(j)}$ is associated with a target in event Θ; and the number of false alarms (unassociated measurements) given by $\phi = \sum_{j=1}^{n_k}[1 - \tau_j]$.

The marginal association probability is the sum of the probabilities of the joint association events given that measurement j belongs to target r

$$\beta_{jr} = \sum_{\Theta} P\{\Theta|z^k\}\hat{\omega}_{jr}[\Theta], \quad j = 1, \ldots, n_k;\ r = 1, \ldots, T \tag{5.7}$$

with, assuming Gaussian distribution for the residual,

$$P\{\Theta|z^k\} = \frac{1}{c}\frac{\phi!}{V^\phi}\prod_{j=1}^{n_k}\{\mathcal{N}(y_k^{(j)}; \hat{z}^{r_j}, S^{r_j})\}^{\tau_j}\prod_{r=1}^{T}(P_D^r)^{\delta_r}(1 - P_D^r)^{1-\delta_r} \tag{5.8}$$

where c is a normalization factor, V is the volume limited by the validation gate, r_j is the index of the target that measurement j is associated with and P_D^r is the probability of detection of target r.

In table 5.1, the full description of the IMM-JPDAF algorithm is presented. The novelty regarding the classic IMM in table 4.3 appears in the filtering step (step 2), where the marginal association probability is introduced in the computation of the weighted measurement residual and on each local estimative off the state covariance. Regarding the state covariance update $P_{k|k}^{(i),(r)}$, its last term is a positive semidifinite matrix which increases the update covariance and the uncertainty of the state estimation due to incorrect measurements.

The IMM-JPDAF design parameters are

- The number of targets T;
- The model set $\mathbb{M}$ structure of each target $r = 1, \ldots, T$;
- The system and measurement noises covariances (Q and R respectively) of each target $r = 1, \ldots, T$;
- The initial state $\hat{x}_{0|0}$ and state covariance $P_{0|0}$ of each target $r = 1, \ldots, T$;
- The jump structure (usually Markov) and the transition probability π_{ji} between the models from the selected set of each target $r = 1, \ldots, T$;
- The probability of detection P_D^r of each target $r = 1, \ldots, T$;
- The validation gate threshold γ.

Table 5.1: One Cycle of IMM-JPDAF Algorithm.

For all targets r, $(r \in \{1 \dots T\})$:

1. Model-conditioned reinitialization ($\forall m_i \in \mathbb{M}$) :
   - Predicted model probability: $\mu_{k|k-1}^{(i),(r)} = \sum_j \pi_{ji}^{(r)} \mu_{k-1}^{(j),(r)}$
   - Mixing probabilities: $\mu_{k-1}^{j|i,(r)} = \pi_{ji}^{(r)} \mu_{k-1}^{(j),(r)} / \mu_{k|k-1}^{(i),(r)}$
   - Mixing estimate: $\bar{x}_{k-1|k-1}^{(i),(r)} = \sum_j \hat{x}_{k-1|k-1}^{(j),(r)} \mu_{k-1}^{j|i,(r)}$
   - Mixing covariance: $\bar{P}_{k-1|k-1}^{(i),(r)} = \sum_j [P_{k-1|k-1}^{(j),(r)} + (\bar{x}_{k-1|k-1}^{(i),(r)} - \hat{x}_{k-1|k-1}^{(j),(r)})(\bar{x}_{k-1|k-1}^{(i),(r)} - \hat{x}_{k-1|k-1}^{(j),(r)})'] \mu_{k-1}^{j|i,(r)}$

2. Model-conditioned joint probabilistic data association filtering ($\forall m_i \in \mathbb{M}$ and $j = \{1, \dots, n_k\}$) :
   - Predicted state: $\hat{x}_{k|k-1}^{(i),(r)} = F_{k-1}^{(i)} \bar{x}_{k-1|k-1}^{(i),(r)}$
   - Predicted covariance: $P_{k|k-1}^{(i),(r)} = F_{k-1}^{(i)} \bar{P}_{k-1|k-1}^{(i),(r)} (F_{k-1}^{(i)})' + Q_{k-1}^{(i)}$
   - Measurement validation: Find $y_k^{(j)} \in G_k(r)$
   - Association probability: $\beta_{jr} = \sum_\Theta P\{\Theta | z^k\} \hat{\omega}_{jr}[\Theta]$
   - Measurement residual: $\tilde{z}_k^{(ji),(r)} = y_k^{(j)} - H_k^{(i)} \hat{x}_{k|k-1}^{(i),(r)}$
   - Weighted measurement residual: $\tilde{z}_k^{(i),(r)} = \sum_{j=1}^{n_k} \beta_{jr} \tilde{z}_k^{(ji)}(r)$
   - Residual covariance: $S_k^{(i),(r)} = H_k^{(i)} P_{k|k-1}^{(i),(r)} (H_k^{(i)})' + R_k^{(i)}$
   - Filter gain: $K_k^{(i),(r)} = P_{k|k-1}^{(i)}(r) (H_k^{(i)})' (S_k^{(i),(r)})^{-1}$
   - Update state: $x_{k|k}^{(i),(r)} = x_{k|k-1}^{(i),(r)} + K_k^{(i),(r)} \tilde{z}_k^{(i),(r)}$
   - Update covariance: $P_{k|k}^{(i),(r)} = P_{k|k-1}^{(i),(r)} - (\sum_{j=1}^{n_k} \beta_{jr}) K_k^{(i),(r)} S_k^{(i),(r)} (K_k^{(i),(r)})' + K_k^{(i),(r)} [\sum_{j=1}^{n_k} \beta_{jr} \tilde{z}_k^{(ji),(r)} (\tilde{z}_k^{(ji),(r)})' - \tilde{z}_k^{(i),(r)} (\tilde{z}_k^{(i),(r)})'] (K_k^{(i),(r)})'$

3. Model probability update ($\forall m_i \in \mathbb{M}$) :
   - Model likelihood: $L_k^{(i),(r)} \overset{assume}{=} \mathcal{N}(\tilde{z}_k^{(i),(r)}; 0, S_k^{(i),(r)})$
   - Model probability: $\mu_k^{(i),(r)} = \frac{\mu_{k|k-1}^{(i),(r)} L_k^{(i),(r)}}{\sum_j \mu_{k|k-1}^{(j),(r)} L_k^{(j),(r)}}$

4. Estimate fusion:
   - Overall estimate: $\hat{x}_{k|k}^{(r)} = \sum_i \hat{x}_{k|k}^{(i),(r)} \mu_k^{(i),(r)}$
   - Overall covariance: $P_{k|k}^{(r)} = \sum_i [P_{k|k}^{(i),(r)} + (\hat{x}_{k|k}^{(r)} - \hat{x}_{k|k}^{(i),(r)})(\hat{x}_{k|k}^{(r)} - \hat{x}_{k|k}^{(i),(r)})'] \mu_k^{(i),(r)}$

5.2.1 Simulation Example

In this section, the simulation results of the IMM-JPDAF algorithm will be shown and discussed. One example will be exhibited with the objective to demonstrate the algorithm's capability of tracking two targets undergoing different trajectories. There will be some assumptions made regarding the data association problem. First, begin to assume that

- All events are feasible, thus $\delta_r = 1, \ \forall \Theta$;
- There is no clutter so a validated measurement is always associated with a target $\tau_j = 1$ ($j = \{1, \dots, n_k\}$ and $\forall \Theta$);

- There are no unassociated measurements thus $\phi = 0 \; \forall \Theta$;
- The target probability of detection is equal to all targets and equal to unity, i.e., $P_D^r = P_D = 1 \; r \in \{1, \ldots, T\}$.

This means that at each time step k there are T targets and T measurements ($n_k = n = T$). This results in a simplified combined version of 5.7 and 5.8

$$\beta_{jr} = \sum_{\Theta} \prod_{j=1}^{n_k} \{\mathcal{N}(z_k^{(j)}; \hat{z}^{r_j}, S^{r_j})\} \hat{\omega}_{jr}[\Theta] \tag{5.9}$$

The targets obey the models described in section 2.1 and they differ from each other in their initial state and angular velocity. The process and measurement noises covariances are equal to both targets and given by the equation (2.15) and (2.34) with uncertainties given by (3.35) and (3.30), respectively.

The target 1 angular velocity is constant equal to 0.01 rad.s^{-1}, while the angular velocity ω of the target 2 (with graphical interpretation in figure 5.1) is defined by

$$\omega_k = \begin{cases} -0.01 & 1 \leq k \leq 40 \\ 0 & 41 \leq k \leq 50 \\ 0.01 & 51 \leq k \leq 60 \\ 0.02 & 61 \leq k \leq 150 \end{cases}, \quad \text{in rad.s}^{-1}. \tag{5.10}$$

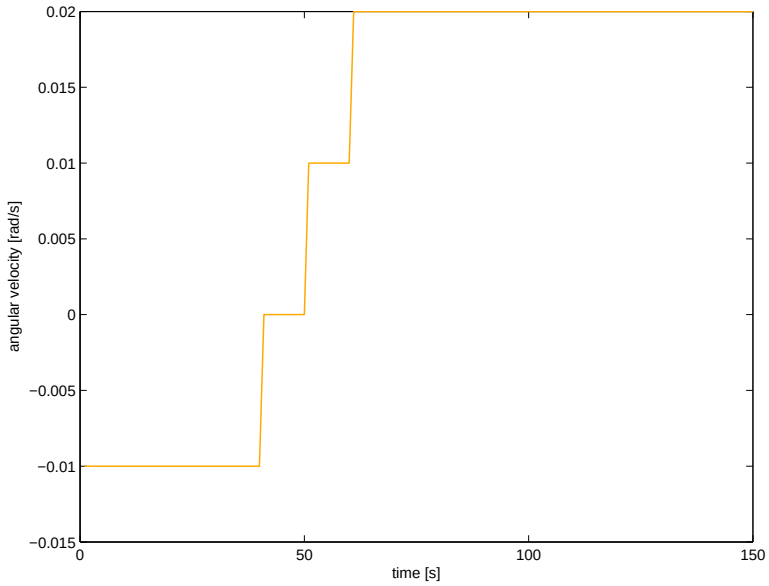


Figure 5.1: Angular velocity of target 2.

The initial state of target 1 is given by (3.27) with

$$\text{x}(0) = -200 \text{ m}, \;\; \text{y}(0) = -50 \text{ m}, \;\; v = 3 \text{ m.s}^{-1} \;\; \text{and} \;\; \phi(0) = 0^\circ \tag{5.11}$$

while initial state of target 2 is given by (3.27) with

$$\mathrm{x}(0) = -100 \text{ m}, \quad \mathrm{y}(0) = -150 \text{ m}, \quad v = 3 \text{ m.s}^{-1} \text{ and } \phi(0) = 90°. \tag{5.12}$$

Firstly, observe the representation of the targets trajectory in figure 5.2. In this circumstances the targets cross each others paths once, and for that single time a target-measurement association switch occurs.

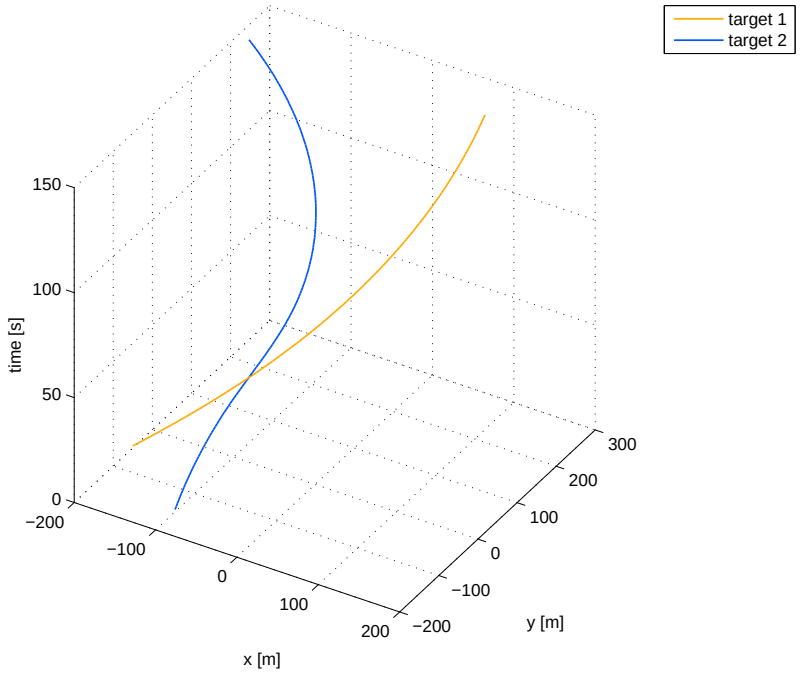


Figure 5.2: IMM-JPDAF targets trajectory three dimensional representation.

The two banks of filters in the IMM-JPDAF algorithm are equal to both targets and have a structure equivalent to the one studied in the previous simulations but with different values for the angular velocity ω_i which are listed in table 5.2, where the positive sign means an anti-clockwise turn and the negative sign means clockwise turn. The values chosen for ω guarantee that the multiple target tracker is able to identify a relatively wide range of maneuvers, similar to the simulation in subsection 4.5, while keeping a small model-set, similar to the other simulations in chapter 4.

The models have equal process and measurement noises covariances defined by the equation (4.13) and (4.7), respectively. To account the multiple target scenario, the process and measurement noises covariances should be rewritten to

$$Q_k^{(i),(r)} = Q_k \ \forall i,k,r \ \text{ with } \sigma_x^2 = \sigma_y^2 = 0.01 \tag{5.13}$$

$$R_k^{p,(i),(r)} = \text{diag}(\sigma_r^2,\ \sigma_\theta^2)\ \forall i,k,r \ \text{ with } \sigma_r = 2 \text{ m}, \quad \sigma_\theta = 1° \tag{5.14}$$

where, Q_k is the discretized continuous process covariance (2.15) using equation (2.10).

Table 5.2: Angular velocity of the seven models equal in both banks of filters.

Angular velocity ω [rad.s^{-1}]
0.03
0.02
0.01
0
-0.01
-0.02
-0.03

The initial state estimate of the bank of filters associated with target r is given by

$$\hat{x}_{0|0}^{(i),(r)} \sim \mathcal{N}(x(0), P_{0|0}^{(i),(r)}) \ \ \forall m_i \in \mathbb{M} \tag{5.15}$$

with initial state covariance estimate $P_{0|0}^{(i),(r)}$ given by

$$P_{0|0}^{(i),(r)} = \begin{bmatrix} 10^2 & 0 & 0 & 0 \\ 0 & 0.2^2 & 0 & 0 \\ 0 & 0 & 10^2 & 0 \\ 0 & 0 & 0 & 0.2^2 \end{bmatrix} \ \ \forall m_i \in \mathbb{M} \tag{5.16}$$

with $r = \{1, \ldots, T\}$.

The transition probability matrix is the same as the one defined by the equation (4.24) and all the models are considered to have equal a priori probability of being correct, i.e., the a priori model probability is given by (4.10). Finally the validation threshold γ is equal to 30 m.

In figure 5.3 it is clearly presented which measurement is associated with each target at the target position. In this example, at the same time, the targets cross each other and the measurement associated with each target changes. Note that at the beginning of the simulation, the measurement-target association is believed to be known, since the targets initial position is approximately known.

The innovation of the IMM-JPDAF algorithm is the introduction of the target-measurement association probability β, calculated using (5.9). The evolution of the probability β for each target with time for the example here studied, is shown figure 5.4. It is observable that the algorithm is very quick to switch the target-measurement association.

The root squared error for the position and the velocity of the target is represented in figure 5.5. As in the previous simulations, the error is higher at the beginning of the simulation, due to the initial state estimate error, which in this case is higher for target 1. During the course of the simulation, the error is overall larger for target 2, as excepted due to the nature of its switching motion. When the target's cross each other, right before the measurement associate with each target changes, there is a peak on the position and velocity error, which, in fact, is what helps the JPDAF detect that a measurement and target association has occur.

It is also interesting to observer the estimate error ellipsoid propagation of each target in figure 5.6 and how they do not suffer an increase on the targets cross, the same happens to the model probabilities of

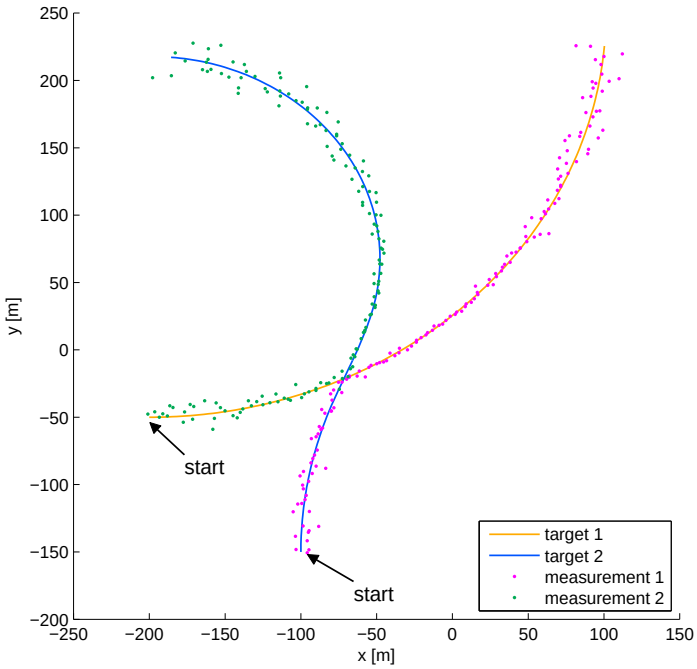


Figure 5.3: IMM-JPDAF targets trajectory and measurement association.

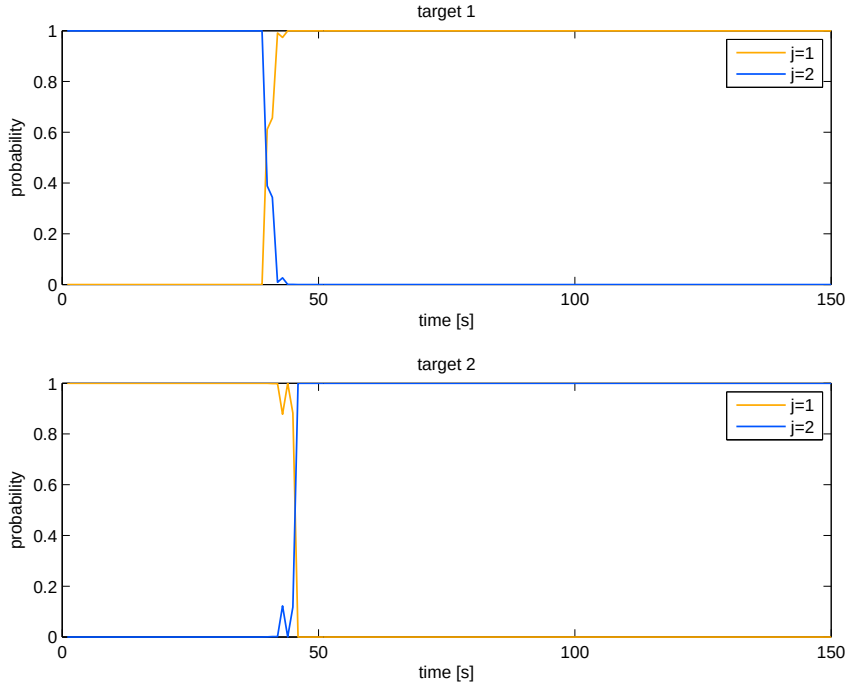
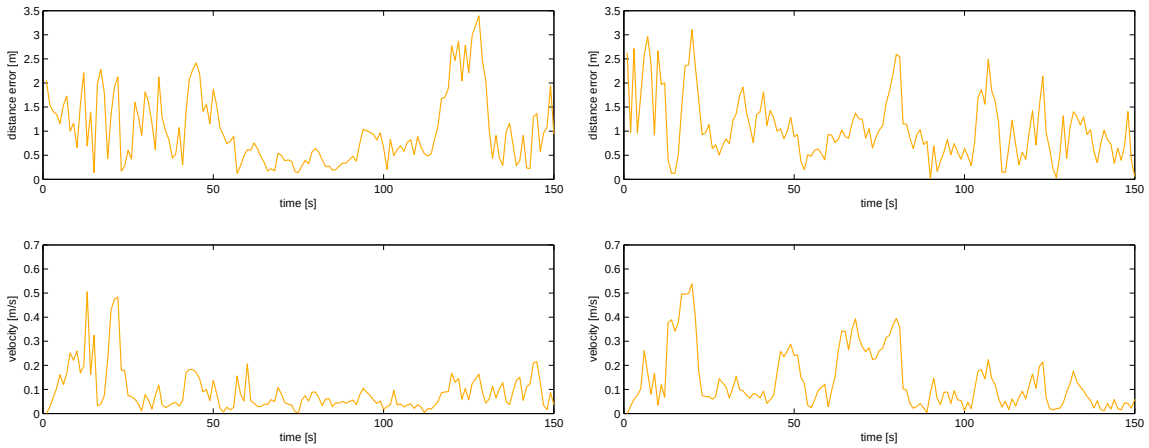


Figure 5.4: IMM-JPDAF measurement and target association regarding the value of β.



(a) target 1 (b) target 2

Figure 5.5: IMM-JPDAF root squared error for the position and velocity of each target

both filter banks (figure 5.7). Since the JPDAF is quick to resolve the DA problem, the two filter banks are almost ”blind” to the targets cross thus behave like two almost isolated IMM trackers.

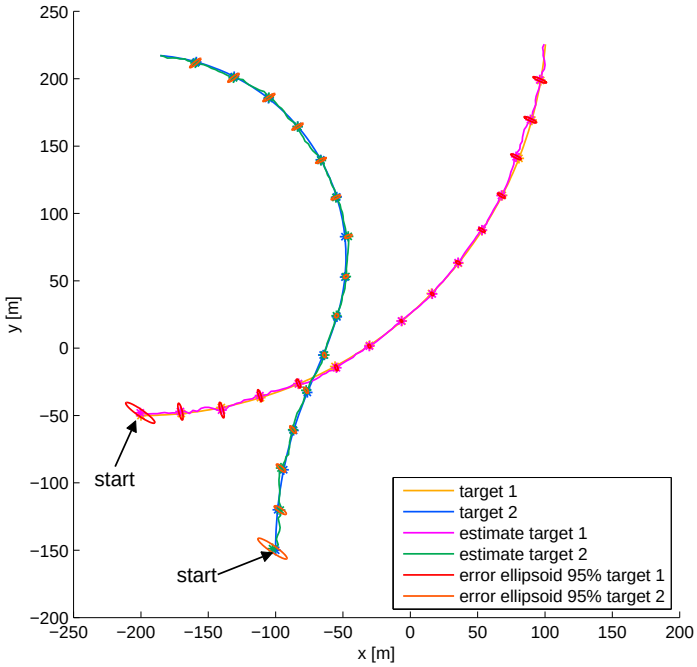


Figure 5.6: Trajectory of the two targets and correspondent IMM-JPDAF estimate and error ellipsoid propagation.

On another note, if the targets did not cross each other but there was a change in the measurement associated with each target, the error would have a similar form, since the novelty of the MTT is the DA problem, which has a much higher associated error than the caused by the STT problem.

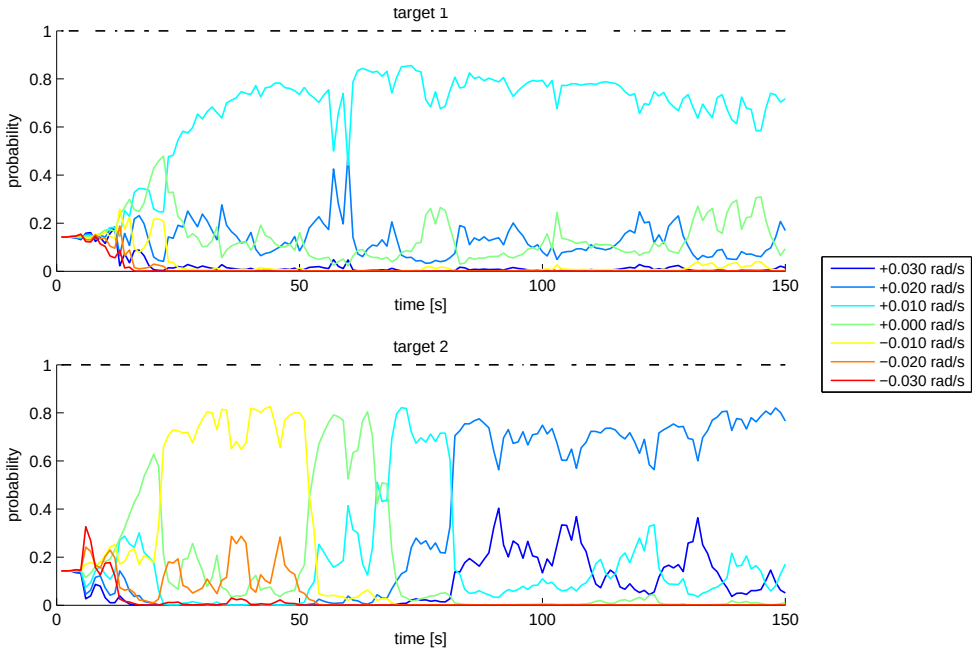


Figure 5.7: IMM-JPDAF model probabilities of the two filter bank associated with each target.

All the above results are quite astonishing, the algorithm performance in the presence of two targets is almost equivalent to the best performance of the MM algorithms studied earlier. Even when the targets cross each other the state estimate error increase is quite acceptable. This extraordinary performance is helped by the characteristics of the targets' motions which are very distinguishable, making the DA easier. Thus the algorithm is quick to find which measurement belongs to whom when they are moving close to each other or even when crossing each other.

5.3 Likely Model-Set Joint Probabilistic Data Association Filter

The algorithm presented in this section was elaborated with the belief that the evolution of the MTT algorithms will necessary pass by the VSMM. The improvement that these algorithms can bring to the MTT algorithms performance are the a direct consequence of the improvement on they have already brought to STT problem. Furthermore, by having different model-sets associated with each target, the data association performance can be improved.

The new algorithm that combines two previous studied approaches in this thesis: the LMS algorithm, the STT discussed in 4.4.2, and JPDAF algorithm, the DA solver discussed in the previous section. Analogously, to the IMM-JPDAF, this algorithm will be named LMS-JPDAF. This algorithm appears as a natural evolution in this thesis, after the IMM-JPDAF it seems logic to try the combination LMS-JPDAF.

The first step in the description of this new algorithm is the redefinition of the VSIMM algorithm, described in section 4.4, into a VSIMM-JPDAF, i.e., an algorithm that combines the JPDAF with the variable structure IMM algorithm. This algorithm appears as an empirical variable structure adaptation of the IMM-JPDAF, and is described in table 5.3.

In relation to the DA problem, in this algorithm at each time step, it is solved regarding only the active model-set, reducing the computational burden and making it easier to determine which measurement is associated with each target. For each cycle of the LMS-JPDAF, the DA association problem should only be solved once, thus there is the need to define an adapted VSIMM-JPDAF, which will be named VSIMM-JPDAF*. The VSIMM-JPDAF* will be run whenever there are newly activated models and uses the association probability β previously calculated. This algorithm is also described in table 5.3.

Table 5.3: One Cycle of VSIMM-JPDAF and VSIMM-JPDAF* Algorithm.

For all targets r, $(r \in \{1 \dots T\})$:

1. Model-conditioned reinitialization ($\forall m_i \in \mathbb{M}_k$) :
   - Predicted model probability: $\mu_{k|k-1}^{(i),(r)} = \sum_{m_j \in \mathbb{M}_{k-1}} \pi_{ji}^{(r)} \mu_{k-1}^{(j),(r)}$
   - Mixing probabilities: $\mu_{k-1}^{j|i,(r)} = \pi_{ji}^{(r)} \mu_{k-1}^{(j),(r)} / \mu_{k|k-1}^{(i),(r)}$
   - Mixing estimate: $\bar{x}_{k-1|k-1}^{(i),(r)} = \sum_j \hat{x}_{k-1|k-1}^{(j),(r)} \mu_{k-1}^{j|i,(r)}$
   - Mixing covariance: $\bar{P}_{k-1|k-1}^{(i),(r)} = m_j \in \mathbb{M}_{k-1}[P_{k-1|k-1}^{(j),(r)} + (\bar{x}_{k-1|k-1}^{(i),(r)} - \hat{x}_{k-1|k-1}^{(j),(r)})(\bar{x}_{k-1|k-1}^{(i),(r)} - \hat{x}_{k-1|k-1}^{(j),(r)})']\mu_{k-1}^{j|i,(r)}$
2. Model-conditioned joint probabilistic data association filtering ($\forall m_i \in \mathbb{M}_k$ and $j = \{1, \dots, n_k\}$) :
   - Predicted state: $\hat{x}_{k|k-1}^{(i),(r)} = F_{k-1}^{(i)} \bar{x}_{k-1|k-1}^{(i),(r)}$
   - Predicted covariance: $P_{k|k-1}^{(i),(r)} = F_{k-1}^{(i)} \bar{P}_{k-1|k-1}^{(i),(r)} (F_{k-1}^{(i)})' + Q_{k-1}^{(i)}$
   - * Measurement validation: Find $y_k^{(j)} \in G_k(r)$
   - * Association probability: $\beta_{jr} = \sum_{\Theta} P\{\Theta | z^k\} w_{jr}[\Theta]$
   - Measurement residual: $\tilde{z}_k^{(ji),(r)} = y_k^{(j)} - H_k^{(i)} \hat{x}_{k|k-1}^{(i),(r)}$
   - Weighted measurement residual: $\tilde{z}_k^{(i),(r)} = \sum_{j=1}^{n_k} \beta_{jr} \tilde{z}_k^{(ji)}(r)$
   - Residual covariance: $S_k^{(i),(r)} = H_k^{(i)} P_{k|k-1}^{(i),(r)} (H_k^{(i)})' + R_k^{(i)}$
   - Filter gain: $K_k^{(i),(r)} = P_{k|k-1}^{(i)}(r) (H_k^{(i)})' (S_k^{(i),(r)})^{-1}$
   - Update state: $x_{k|k}^{(i),(r)} = x_{k|k-1}^{(i),(r)} + K_k^{(i),(r)} \tilde{z}_k^{(i),(r)}$
   - Update covariance: $P_{k|k}^{(i),(r)} = P_{k|k-1}^{(i),(r)} - (\sum_{j=1}^{n_k} \beta_{jr}) K_k^{(i),(r)} S_k^{(i),(r)} (K_k^{(i),(r)})' + K_k^{(i),(r)} [\sum_{j=1}^{n_k} \beta_{jr} \tilde{z}_k^{(ji),(r)} (\tilde{z}_k^{(ji),(r)})' - \tilde{z}_k^{(i),(r)} (\tilde{z}_k^{(i),(r)})'] (K_k^{(i),(r)})'$
3. Model probability update ($\forall m_i \in \mathbb{M}_k$) :
   - Model likelihood: $L_k^{(i),(r)} \overset{assume}{=} \mathcal{N}(\tilde{z}_k^{(i),(r)}; 0, S_k^{(i),(r)})$
   - Model probability: $\mu_k^{(i),(r)} = \frac{\mu_{k|k-1}^{(i),(r)} L_k^{(i),(r)}}{\sum_j \mu_{k|k-1}^{(j),(r)} L_k^{(j),(r)}}$
4. Estimate fusion:
   - Overall estimate: $\hat{x}_{k|k}^{(r)} = \sum_{m_i \in \mathbb{M}_k} \hat{x}_{k|k}^{(i),(r)} \mu_k^{(i),(r)}$
   - Overall covariance: $P_{k|k}^{(r)} = \sum_{m_i \in \mathbb{M}_k} [P_{k|k}^{(i),(r)} + (\hat{x}_{k|k}^{(r)} - \hat{x}_{k|k}^{(i),(r)})(\hat{x}_{k|k}^{(r)} - \hat{x}_{k|k}^{(i),(r)})'] \mu_k^{(i),(r)}$

* These two steps are omitted on the VSIMM-JPDAF* algorithm.

In table 5.4, the full description of the LMS-JPDAF algorithm is presented. It has exactly the same

structure as the LMS algorithm described in table 4.5, only the steps regarding the VSIMM algorithm are replace by the VSIMM-JPDAF (and VSIMM-JPDAF*) cycle.

Table 5.4: One Cycle of LMS-JPDAF Algorithm.

---

1. Increase the time counter k by 1. Run the VSIMM-JPDAF $[\mathbb{M}_k, \mathbb{M}_{k-1}]$ cycle.
2. Classify all the models m_i's in $\mathbb{M}_k$ to be principal (i.e., $\mu_k^i > t_2$), unlikely (i.e., $\mu_k^i < t_1$) or significant (i.e., $t_1 \geq \mu_k^i \leq t_1$). Let the set of unlikely models be $\mathbb{M}_u$. If there is neither unlikely nor principal model, output $\hat{x}_{k|k}$, $P_{k|k}$ and $\{\mu_k^{(i)}\}_{m^{(i)} \in \mathbb{M}_k}$, let $\mathbb{M}_{k+1} = \mathbb{M}_k$ and go to step 1.
3. If there is no principal model, then let $\mathbb{M}_a = \emptyset$ and got to step 4. Otherwise, identify the set $\mathbb{M}_a$ of all models adjacent to any principal model. Find the set of new models $\mathbb{M}_n = \mathbb{M}_a \bigcap \bar{\mathbb{M}}_k$ (where $\bar{\mathbb{M}}_k$ is the complement of $\mathbb{M}_k$ and the union set $\mathbb{M}_k := \mathbb{M}_n \bigcup \mathbb{M}_k$.
   Then

   - Run VSIMM-JPDAF* $[\mathbb{M}_n, \mathbb{M}_{k-1}]$ cycle, where $\mathbb{M}_n$ is the set of new and only new models.
   - Fusion: Calculate the estimates, error covariances, and mode probabilities for the union set $\mathbb{M}_k$:

$$\mu_k^{(i)} = \frac{\mu_{k|k-1}^{(i)} L_k^{(i)}}{\sum_{m_j \in \mathbb{M}_k} \mu_{k|k-1}^{(j)} L_k^{(j)}}, \forall m_i \in \mathbb{M}_k$$

$$\hat{x}_{k|k} = \sum_{m_i \in \mathbb{M}_k} \hat{x}_{k|k}^{(i)} \mu_k^{(i)}$$

$$P_{k|k} = \sum_{m_i \in \mathbb{M}_k} P_{k|k}^{(i)} [(\hat{x}_{k|k} - \hat{x}_{k|k}^{(i)})(\hat{x}_{k|k} - \hat{x}_{k|k}^{(i)})'] \mu_k^{(i)}$$

     where the estimates $\{\hat{x}_{k|k}^{(i)}\}$, error covariances $\{P_{k|k}^{(i)}\}$, likelihoods $\{L_k^{(i)}\}$, and predicted probabilities $\{\mu_{k|k-1}^{(j)}\}$ were obtained in the above VSIMM-JPDAF $[\mathbb{M}_k, \mathbb{M}_{k-1}]$ and VSIMM-JPDAF* $[\mathbb{M}_n, \mathbb{M}_{k-1}]$ cycles.

4. Output $\hat{x}_{k|k}$, $P_{k|k}$ and $\{\mu_k^{(i)}\}_{m^{(i)} \in \mathbb{M}_k}$.
5. If there is no unlikely model, go to step 1; otherwise, identify the discardable set $\mathbb{M}_d = \mathbb{M}_u \bigcap \bar{\mathbb{M}}_a$, that is, the set of unlikely models that are not adjacent from any principal model.
6. Eliminate the models in $\mathbb{M}_d$ from $\mathbb{M}_k$ that have the smallest probability such that $\mathbb{M}_k$ has at least K models, that is, let the likely model-set be $\mathbb{M}_l = \mathbb{M}_k - \mathbb{M}_m$, where $\mathbb{M}_m$ is the set of models in $\mathbb{M}_d$ with smallest probabilities such that $\mathbb{M}_l$ has at least K models.
7. Let $\mathbb{M}_{k+1} = \mathbb{M}_l$ and go to step 1.

The LMS-JPDAF design parameters are

- The number of targets T;
- The model set $\mathbb{M}$ structure of each target $r = 1, \ldots, T$;
- The system and measurement noises covariances (Q and R respectively) of each target $r = 1, \ldots, T$;
- The initial state $\hat{x}_{0|0}$ and state covariance $P_{0|0}$ of each target $r = 1, \ldots, T$;

- The jump structure (usually Markov) and the transition probability π_{ji} between the models from the selected set of each target $r = 1, \ldots, T$;

- The unlikely and principal models probability thresholds, t_1 and t_2, respectively, of each target $r = 1, \ldots, T$, as where defined in the LMS section 4.4.1;

- The probability of detection P_D^r of each target $r = 1, \ldots, T$;

- The validation gate threshold γ.

5.3.1 Simulation Example

In this section, simulation results of the LMS-JPDAF algorithm will be shown and discussed. The example from the previous section will be maintained with all its assumptions, the only difference is in the MTT algorithm used. The only new parameters needed to be defined are the unlikely and principal models probability thresholds, t_1 and t_2. They are chosen differently for each target since they have different maneuver motions, which are considered to be known superficially, i.e., there is the knowledge if a target is in high or slow maneuver. These thresholds are given by

$$t_1 = 0.01, \;\; t_2 = 0.4 \;\; \text{and} \;\; t_1 = 0.01, \;\; t_2 = 0.2 \tag{5.17}$$

for target 1 and target 2, respectively.

Using these thresholds the active models at each time step, by each of the LMS algorithms associated with each target, are represented in figure 5.8. By maintaing different active models for each target, the algorithm eases the DA problem, besides reducing the computer complexity. Also this figure illustrates the novelty of this algorithm by revealing its variable structure.

The evolution of the probability β for each target with time for the example here studied is quite similar to the IMM-JPDAF response in figure 5.4, this algorithms maintains the same speed of solving the DA problem present in the IMM-JPDAF algorithm.

The root squared error for the position and the velocity of the target is represented in figure 5.9, which are of the same range of values than the root squared error of the IMM-JPDAF, represented in figure 5.5. The LMS-JPDAF state estimate and error ellipsoid propagation is similar to the IMM-JPDAF response in figure 5.6.

Since the performance of a MTT algorithm is closely related to the DA algorithm performance, the LMS-JPDAF can achieve better results than the IMM-JPDAF by differentiating the targets active model-sets. Note that the IMM-JPDAF can also have different model-sets for each target, but this would only happen if the targets did not share the same motion. The LMS-JPDAF allows the targets to have the same total model-set, but can have different active models for each target at each time.

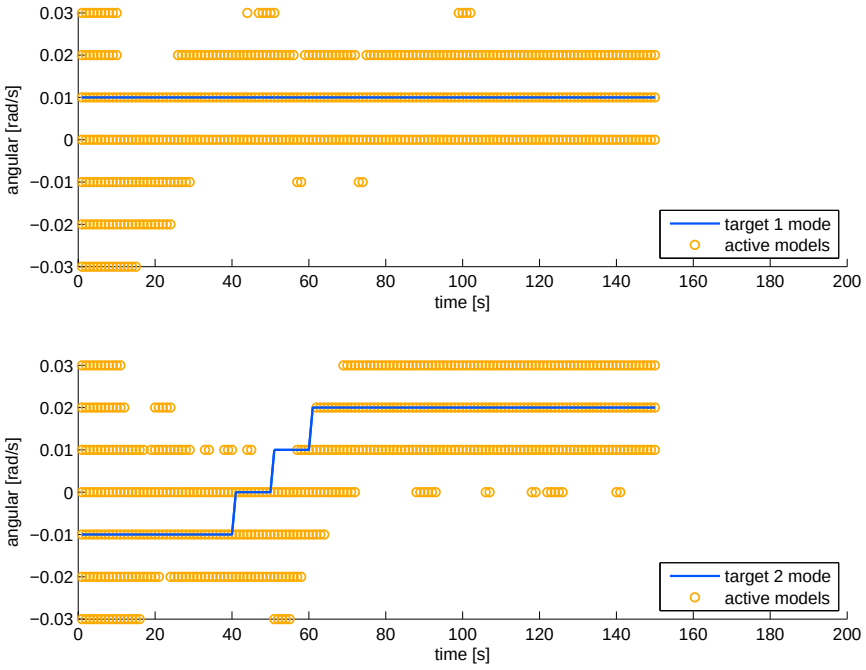
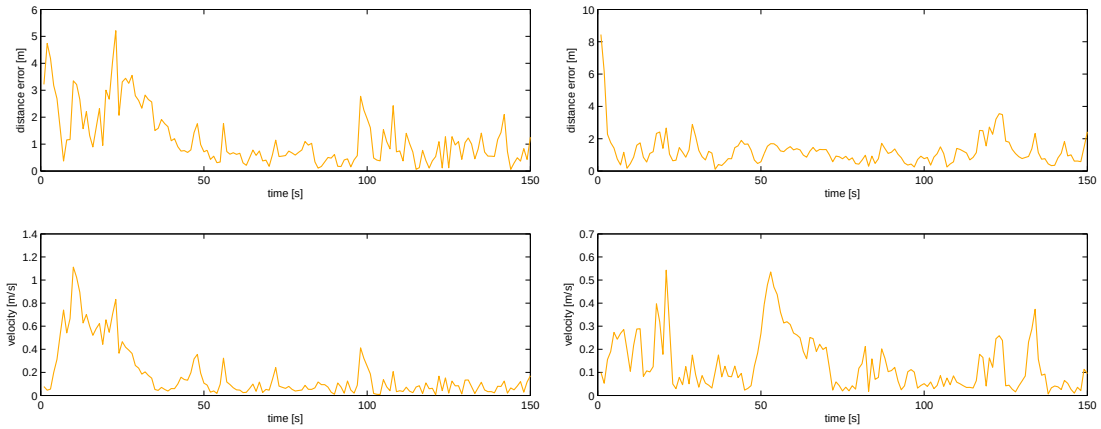


Figure 5.8: LMS-JPDAF active models for each target.



(a) target 1 (b) target 2

Figure 5.9: LMS-JPDAF root squared error for the position and velocity of each target.

5.4 Comparison of the MTT algorithms

In this section, a resume comparison of the two studied algorithms to solve the MTT problem is presented. Only an analysis of the root squared error of the position and velocity will be made. Similarly to section 4.5, a total of 30 simulations were made and the mean value of them was used to compute the results shown below.

Once again, the simulation parameters are the same used in sections 5.2 and 5.3. The only difference

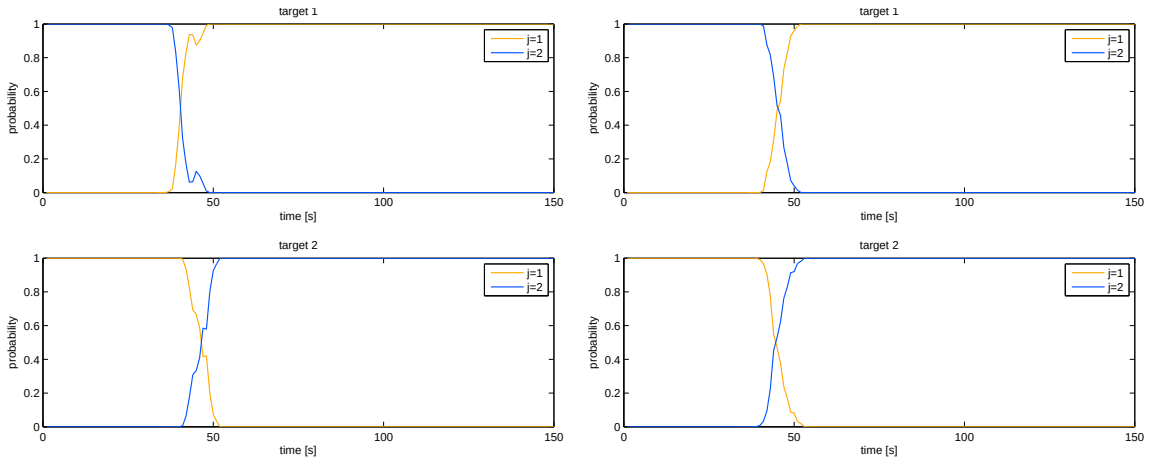
is on the choice of the following parameters

- The validation threshold given by $\gamma = 70$;
- The initial state covariance estimate $P_{0|0}^{(i),(r)}$ given by equation

$$P_{0|0}^{(i),(r)} = \begin{bmatrix} 10^2 & 0 & 0 & 0 \\ 0 & 0.2^2 & 0 & 0 \\ 0 & 0 & 10^2 & 0 \\ 0 & 0 & 0 & 0.2^2 \end{bmatrix} \tag{5.18}$$

with $r = \{1, \ldots, T\}$, $\forall m_i \in \mathbb{M}$ and $\forall m_i \in \mathbb{M}_O$, for the IMM-JPDAF and LMS-JPDAF, respectively.

The mean value of β for each of the algorithms is shown in figure 5.10. From this example, its possible to observe that the LMS-JPDAF provides a softer, though longer, measurement and target association switch. This provides a softer error as will be seen below, but also increases this algorithm's inertia to change.

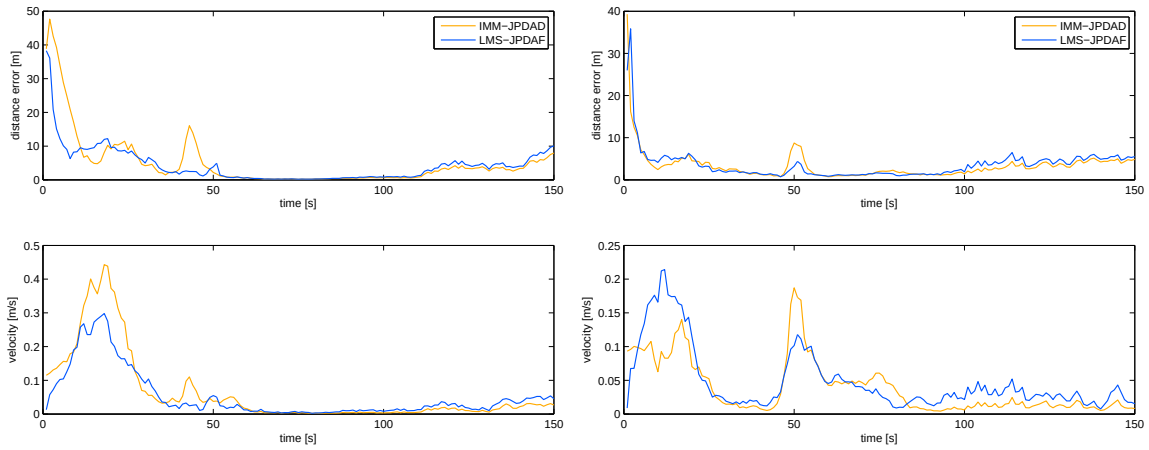


(a) IMM-JPDAF algorithm.

(b) LMS-JPDAF algorithm.

Figure 5.10: IMM-JPDAF and LMS-JPDAF measurement and target association regarding the value of β.

The mean of the root squared error for the position and velocity of each target for the 30 simulations is shown in figure 5.11. Although the overall error is larger for the LMS-JPDAF, specially at the beginning of the simulation, the peak during the targets crossover and data association switch is much smaller, since this is the highest value of the error, the LMS-JPDAF can be useful for some applications where the error maximum needs to be smaller.



(a) target 1 (b) target 2

Figure 5.11: MTT algorithms mean of the root squared error for the position and velocity of each target for 30 simulations.

Chapter 6

Conclusion

In this chapter the main conclusions of this thesis are presented. It does not intent to prove which of the algorithms studied is the best nor does it compare the algorithms performances, since they were analyzed based on the results of specific simulated scenarios and not based on theoretical deductions. This conclusion intents to resume the work on this thesis and presents the overall behavior of the studied algorithms behavior has it can be drawn from the simulated results.

To finalize, in section 6.1 some additional research regarding the scope of this thesis is purposed in order to make this overview on target tracking using multiple models even more complete.

The aim of this thesis was to present an overview of a collection of tools used in the problem of target tracking. Under this thesis scope, a new algorithm for target tracking was developed. This chapter will thus emphasize the main results drawn from the algorithms studied, bearing in mind that the validity of these conclusions may be limited to the scenarios simulated.

The MM algorithms all revealed capable trackers under target motion uncertainty. The first generation of MM algorithms is the most limitative, since it assumes the target is in constant maneuver, i. e, the target has one (unknown) constant mode. The other two generations have more complex concepts, even though they are not computationally more complex. Their tracking capability is much better, since they assume the target can have multiple maneuver motions. The CMM generation has little more room for improvement, leading to the general belief in the tracking community, that the VSMM algorithms will be the future of target tracking, since they allow lower computational complexity with larger model-sets. The VSMM algorithms are relatively new and need more research to be well established as a robust alternative algorithm to STT.

Regarding the MM comparison realized in section 4.5, this results represent only an example since the algorithms may have different responses for different target motions. In the simulations performed in this thesis, the MMSE-AMM clearly stands out negatively, since it is the poorer tracker. The IMM and LMS responses are similar, but the LMS is quite sensitive to its design parameters, specifically to the probability thresholds which lack a mathematical dimensioning and are not easy to choose intuitively. With further development, the VSMM algorithms have it all to surpass their previous generations performance.

The MTT algorithms, besides solving the target motion uncertainty also deal with measurement origin uncertainty. The JPDAF algorithm revealed to be a successful DA algorithm, but the assumptions made, regarding the false alarms, presence of clutter and the number of detected targets, reduced drastically the measurement origin uncertainty. Nevertheless, the results were quite satisfactory.

The IMM-JPDAF combines the superior tracking capabilities of the CMM algorithm with a classic DA algorithm. The LMS-JPDAF arises as a natural evolution from the CMM multiple target trackers to the VSMM multiple target trackers. There is still the need and the capability for improvement under the group of CMM multiple target trackers, but the study of the VSMM multiple target trackers seems pertinent.

For the example analyzed the LMS-JPDAF surpassed the IMM-JPDAF performance under measurement origin uncertainty, although provided a worse STT when the targets were far apart. The better solving of the target measurement association can be explained by the accentuating of the targets different motions by the LMS-JPDAF, i.e., by making the active model-sets for each target different it becomes easier to identify which measurement belongs to each target. Also for the example analyzed, the LMS-JPDAF provided a softer, though longer, measurement and target association switch. Tough this leads to a softer error it also can cause an increases on this algorithm's inertia to change.

6.1 Additional Research Proposed

The additional research that can be realized in this area is vast. There are several additional topics to extend this thesis work, they are: i) Test further and study even more in depth the algorithms in

simulation environment, ii) Analyze the algorithms performance in a real life implementation and iii) Study alternative algorithms.

Regarding the test of the already studied algorithms, more simulations could be realized, specially for the IMM-JPDAF algorithm, where some simplifying assumptions were made regarding the number of targets detected by the sensor which was time-invariant. Also further algorithm development could be done in order to study its performance in the presence of clutter, a subject which has also a great coverage in literature.

The LMS-JPDAF algorithm needs also further study and performance improvement to make it a competitor of the available MTT algorithms. Regarding this performance improvement, it it would be interesting to analyze possible mathematical designs of the model probability thresholds, characteristic of the LMS algorithm, and study whether they should be static or adaptive, for instances, if there is a high uncertainty on the state estimate, the thresholds can be relaxed.

More theoretical work could be done to fundament the algorithms results, namely the study of the Cramer Rao lower bounds.

Since all the discussed algorithms, are studied in environments that try simulate a real life situation, it would be valuable to analyze their performances in a real life application. It would be interesting to implement the algorithms using real sensor measurements and real targets, and test their capability and robustness. Of course, the more interesting algorithms would be the second and third generations of MM and LMS, since they represent more recent algorithms with real implementations not study profoundly.

Regarding the study of alternative algorithms, there are a lot to choose from, more or less documented and more or less studied. With special interest, there are the VSMM and MTT algorithms, which are relatively new areas with constant development, in contrast with the first two MM generations, which have already been deeply studied.

An interesting subject with plenty room for exploration is the MTT problem, with the study of more alternative algorithms, such as the Monte Carlo techniques known as Particle Filtering [15], which have drawn the attention of the tracking community.

Bibliography

[1] Y. Bar-Shalom and T.E. Fortmann, Tracking and Data Association, 1988, Academic Press

[2] Y. Bar-Shalom, K. C. Chang and H. A. P. Blom, Tracking splitting targets in clutter by using an Interacting Multiple Model Joint Probabilistic Data Association filter, in *Multitarget-Multisensor Tracking: Applications and Advances* (Y. Bar-Shalom and W. D. Blair, eds.), vol.II, pp.93-110, MA: Artech House.

[3] Y. Bar-Shalom and X. R. Li, Mode-set adaptation in multiple-model estimators for hybrid systems, in *Proceedings of the 1992 American Control Conference*, Chicago, IL, June 1992, 1794-1799

[4] Y. Bar-Shalom and X. R. Li, Multiple-model estimation with variable structure, in *IEEE Transactions on Automatic Control*, 41(4):478-493, April 1996

[5] Y. Bar-Shalom and X. R. Li, Multiple-model estimation with variable structure-part III: model-group switching algorithm, in *IEEE Transactions on Aerospace and Electronic Systems*, 35(1):225-241, January 1999 pdf pdf

[6] Y. Bar-Shalom and X. R. Li, Multiple-model estimation with variable structure-part IV: design and evaluation of model-group switching algorithm, in *IEEE Transactions on Aerospace and Electronic Systems*, 35(1):242-254, January 1999

[7] Y. Bar-Shalom and X. R. Li, Multiple-model estimation with variable structure-part II: model-set adaptation, in *IEEE Transactions on Automatic Control*, 45(11):2047-2060, November 2000

[8] Y. Bar-Shalom and X. R. Li, Multiple-model estimation with variable structure-part V: likely-model set algorithm, in *IEEE Transactions on Aerospace and Electronic Systems*, 36(2):448-466, April 2000

[9] Y. Bar-Shalom and X. R. Li, Estimation with Applications to Tracking and Navigation, 2001, John Wiley & Sons, Inc., New York, NY, USA,.

[10] Y. Bar-Shalom and X. R. Li, Multiple-model estimation with variable structureÑpart VI: expected-mode augmentation, in *IEEE Transactions on Aerospace and Electronic Systems*, 41(3): 853-867, July 2005

[11] H. A. P. Bloom, An efficient filter for abruptly changing systems, in *Proceedings of the 23rd IEEE Conference on Decision and Control* Las Vegas, NV, Dec. 1984, 656-658.

[12] R. G. Brown and P. Y. C. Hwang, Introduction to Random Signals and Applied Kalman Filtering, 3rd ed. , New York: Wiley, 1997.

[13] T. E. Fortmann, Y. Bar-Shalom and M. Scheffe, Multitarget tracking using joint probabilistic data association in *Proceeding 1th IEEE Conference on Decision and Control*, 807-8 12.1980.

[14] R. E. Kalman, A New Approach to Linear Filtering and Prediction Problems, in *Transaction of the ASME-Journal of Basic Engineering*, pp. 35-45 (March 1960).

[15] R. Karlsson and F. Gustafsson, Monte Carlo Data Association for Multiple Target Tracking, in *Proceedings of The IEE Workshop on Target Tracking : Algorithms and Applications*, The Netherlands, 2001.

[16] K. Kastella and M. Biscuso Tracking algorithms for air traffic control applications. in *Air Traffic Control Quarterly*, 3, 1 (Jan. 1996), 19-43.

[17] D. Lerro and Y. Bar-Shalom, Tracking with debiased consistent converted measurements versus EKF, in *IEEE Trans. on Aerospace and Electronic Systems*, vol. 29, July 1993, pp. 1015-1022.

[18] X. R. Li, Engineer's Guide to Variable-Structure Multiple-Model Estimation for Tracking, in *Multitarget-Multisensor Tracking: Applications and Advances* (Y. Bar-Shalom and W. D. Blair, eds.), vol. III, ch. 10, pp. 499-567, Boston, MA: Artech House.

[19] X. R. Li and V.P. Jilkov, A Survey of Maneuvering Target Tracking. Part III: Measurement Models. in *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, pages 423-446, San Diego, USA, Jul.-Aug. 2001

[20] X. R. Li and V.P. Jilkov, Survey of maneuvering target tracking. Part I. Dynamic models, in *Aerospace and Electronic Systems, IEEE Transactions on*, Oct. 2003

[21] X. R. Li and V.P. Jilkov, A Survey of Maneuvering Target Tracking. Part V: Multiple-Model Methods in *IEEE Transactions on Aerospace and Electronic Systems* VOL. 41, NO. 4, Oct. 2005, 1256-1320

[22] D. T. Magill, Optimal adaptive estimation of sampled stochastic processes in *IEEE Transactions on Automatic Control*, AC-10 (1965), 434-439.

[23] D. Reid, An algorithm for tracking multiple targets in *IIEEE Transactions on Automatic Control*, 24 no. 6, 1979.