



**Fuzzy Mathematics and Logic for Multisource
Data Association and Target Tracking
(Phase II)**

September 1998

Center for Multisource Information Fusion
The State University of New York at Buffalo
Buffalo, New York 14260

Principle Investigators: Dr. Tarunraj Singh
Email: tsingh@eng.buffalo.edu

Dr. James Llinas
Email: llinas@acsu.buffalo.edu

Research Assistants: Dirk Tenne
Email: tenne@eng.buffalo.edu

Li-Chieh Chen
Email: lcchen2@eng.buffalo.edu

Abstract

Target tracking is the prediction of the future location of a dynamic system (the target) based on developing an optimal estimator which employs both imperfect sensor observations of the target motion, and its past states. The accuracy of the prediction, therefore, depends in part on how accurately a target's past and present positions were measured. Unfortunately, due to environmental and man-made clutter and other limitations of the sensors and the signal processing algorithms, the positions of a target cannot be measured with perfect accuracy. A wide variety of estimator forms have been developed to deal with the target tracking problem. One of the early forms, a so-called fixed-coefficient estimator (or "filter") called the " α - β " filter, has been (and continues to be) employed on many operational systems. Thus, in spite of its simplicity and limitations, it continues to be of interest. This paper explores many aspects of developing a fuzzy logic gain modified α - β filter. The fixed coefficient α - β - γ filter is analyzed in detail to characterize the performance of the α - β filter which is a subset of the α - β - γ filter¹. The α - β - γ filter is analyzed with the intention of eventually developing a fuzzy α - β - γ filter.

This is a phase 2 of a continuing project sponsored by the Office of Naval Research/Spawar System Center (ONR/SSC) on the synthesis and analysis of fuzzy-logic scheduled α - β filter. The focus of this phase of the study was the control-theoretic based analysis of the fuzzy α - β filter and a development of a testbed for the realistic simulation of the target tracking process.

The α - β - γ filter is a sampled data target tracker which can asymptotically track a constant accelerating target. The α - β - γ parameters are studied to characterize the stability of the filter, its performance viz its transient behavior and the influence of measurement noise. A closed form equation for the mean square response of the system to white noise is derived. In addition metrics to gauge the transient response and the steady state tracking error are derived. These equations are shown to reduce to some results presented in the literature for the α - β filter by equating γ to zero. The equations for the noise ratio and the transient and steady state error are exploited to optimally select the α - β - γ parameters, so as to minimize the noise transmission capability of the filter.

The proposed fuzzy logic based α - β filter changes the smoothing parameter, α and β , as a function of the maneuver error and error rate with the tracking performance which is comparable to a Kalman filter, but without modeling a target system and a sensor noise model. Furthermore, the computational cost is less than those of the traditionally Kalman filter. The fuzzy logic theory overcomes the difficulty of

¹The coefficients α , β and γ signify position, velocity and acceleration smoothing gains respectively

defining an explicit relationship between the current system states and the actions required to achieve a certain system performance by using linguistic rules. The proposed fuzzy filter exploits an analogy of control of a second order system a rest-to-rest maneuver, to define appropriate rules.

Since the evaluation of candidate tracking algorithms and the identification of an optimal one, is not feasible from a strictly analytically point of view, such problems are frequently evaluated using Monte Carlo simulations. Accordingly, we have designed a Testbed within which such problems can be efficiently studied. The Testbed provides a graphical user interface controlling the simulation environment to conduct comparisons.

Acknowledgments

This research was supported by the Office of Naval Research/Spawar System Center under Contract N00014-97-1-0584. The authors gratefully acknowledge helpful discussions and comments provided by Mr. Ed. Jahn and Dr. Dave Johnson.

Contents

1	Introduction	1
2	Filter Analysis	5
2.1	Introduction	5
2.2	Stability Analysis	6
2.2.1	α - β Tracker	6
2.2.2	Jury's Stability Test	9
2.2.3	α - β - γ Tracker	15
2.3	Performance of the Target Tracker	21
2.3.1	Noise Ratio	21
2.3.2	Maneuver Error	27
2.4	Design of Optimal Filters	34
2.4.1	Circular Trajectory	35
2.4.2	Straight Line Maneuver	38
2.5	Remarks	45
3	Fuzzy Logic Approach of Target Tracking	47
3.1	Principles of the Fuzzy Algorithm	48

3.1.1	Fuzzy Logic on an Example	48
3.1.2	A General Fuzzy Logic Approach	52
3.2	The Fuzzy Algorithm in Target Tracking	56
3.2.1	The Fuzzy Logic Inputs	56
3.2.2	The Fuzzy Logic Rule Base	60
3.3	Summary	65
4	Tracker Testbed	68
4.1	Parameters for the Testbed Setup and Simulation Execution	68
4.1.1	Filter initialization for Simulation	70
4.1.2	Noise Measurement Generation in Simulation	70
4.1.3	Pairwise Sensor Fusion	72
4.1.4	Position Estimation of Filters in Simulation	73
4.1.5	Performance Measures for Simulation Output Analysis	73
4.2	The Functional Structure and the GUI	73
5	Results	81
5.1	The Proposed Fuzzy Gain α - β Tracker	81
5.2	Benchmark Tests	84
5.3	Remarks	90
6	Conclusion and Recommendation	92
A	Kalman Filter Process Noise	97

Chapter 1

Introduction

The Center of Excellence in Multisource Information Fusion (CMIF) at the State University of New York at Buffalo ("UB"), has been involved in the development of a tracking Testbed under the sponsorship of ONR/SSC. This project was in part stimulated by the earlier work carried out at Orincon¹, sponsored by NRaD. This Testbed was developed as a tool within which to study alternative methods for tracking submarines which are detected by systems of sensor arrays deployed in littoral environments. There are various characteristics of this Anti-submarine Warfare (ASW) problem that create difficulties in developing technically optimal yet cost-effective solutions to the target tracking problem, e.g.:

- crossing targets
- sparse data
- varying target localization capabilities (environmentally dependent)
- false alarms

among other factors. Additionally, the trend toward low-cost, compact sensor nodes provides a requirement that the tracking algorithm be computationally efficient. A

¹Orincon Corporation

solution to such problems, i.e., the identification and evaluation of candidate tracking algorithms for such problems, is not feasible from a strictly analytical point of view. Thus, such problems are frequently examined empirically and parametrically. Accordingly, we have designed a Testbed within which such problems can be efficiently studied. The Testbed provides a graphical user interface to control the simulation environment and to conduct comparisons.

This project also focuses on the viability of using Fuzzy Logic (FL) and Mathematics in such ASW tracking applications. The motivation was based on the potential simplification and compatibility that FL offers for implementing adaptive yet simple tracking methods, especially when compared to traditional Kalman Filter-based methods. In particular, this work proposes an assessment of a Fuzzy Logic-controlled α - β filter, which are nominally simple but fixed-coefficient filters. As the result, the proposed target tracker balances computational simplicity with adaptation. The research objectives of this project can be summarized by the following items:

- methods for higher-fidelity simulation of sensor-measurement processes, including modeling of environmental effects on target localization capability
- developing formal methods of control for α - β filters especially as related to stability management and measurement noise
- extending these methods to α - β - γ filter
- researching FL-based methods for adaptive α - β filter

The performance of the target tracker is first developed for the α - β - γ filter and later reduced to the α - β filter. The α - β - γ parameters are studied in detail to characterize the stability of the filter, its performance viz its transient behavior and the influence of measurement noise. The equations for the noise ratio and the transient and steady state error are exploited to optimally select the α - β - γ parameters so as to minimize the noise transmission capability of the filter. Chapter 3 explains the basic fuzzy logic

algorithm before introducing the fuzzy logic based α - β filter. The proposed fuzzy filter exploits an analogy from the system control theory, the rest-to-rest maneuver, to define the fuzzy relational matrix, which is discussed later in this report.

Furthermore, this project addresses a variety of enhancements to the Testbed capacity developed during the first phase. These enhancements are briefly listed below:

Task 1 Realistic Simulation of the Sensor Field

This task focuses on studying the effect of sparseness of data, and represent the correct effects of sensor system localization capabilities which is related to the environmental effects on the sensor resolution.

Task 2 Generating realistic target trajectories

The objective of this task is to provide realistic submarine maneuvers such as (i) straight line maneuvers, (ii) single gradual turn maneuvers, (iii) straight line speedup maneuvers and (iv) turn with speedup maneuvers .

Task 3 Integration of Standard Trackers into the Testbed

The objective of this task is to integrate standard tracking filters such as the Kalman filter, α - β - γ filter into the Testbed.

Task 4 Development and Integration of Adaptive Tracking Filters

This task integrates the proposed fuzzy logic filter and Chan's adaptive tracking filter [5] into the Testbed.

Task 5 Evaluation of Various Metrics for Comparison of the Different Filters

This task involves conceiving of different metrics to evaluate the performance of different filters using the constant coefficient α - β filter as a baseline. One of the metric, which is studied is the evolution of the mean error and error-rate from Monte Carlo simulations for different tracking filters when the target is performing a straight line maneuver. Following

this, more complicated maneuvers are studied. Another measure of interest, is the sum of the square of the tracking errors over a finite time horizon simulation. This is evaluated for each of the Monte Carlo runs and plotted to estimate a error distribution profile. Another metric used is the computational cost of each target tracker.

Task 6 Development of a Graphical User Interface (GUI)

This task focuses on the development of a mouse-and-menu-driven interface which permits the user to select from various options such as: (i) filter for tracking, (ii) target maneuvers, (iii) simulation parameter, etc. The GUI is developed on MATLAB ².

The focus of this report is on theoretical aspects of target tracking filter such as α - β - γ filter, developing a control theoretic based adaptive α - β filter and conducting a performance comparison within a graphical simulation environment.

²MATLAB is a registered trademark of The MathWorks, Inc., Natick, Mass.

Chapter 2

Filter Analysis

2.1 Introduction

Numerous applications such as air-traffic handling, missile interception, anti-submarine warfare require the use of discrete-time data to predict the kinematics of a dynamic object. The use of passive sonobuoys which have limited power capacity constrain us to implement target-trackers which are computationally inexpensive. With these considerations in mind, we analyze an α - β - γ filter to study its ability to predict the object kinematics in the presence of noisy discrete-time data.

There exists a significant body of literature which addresses the problem of *track-while-scan systems* [16], [9], [2] and [1]. Sklansky [16] in his seminal paper analyzed the behavior of an α - β filter. His analysis of the range of values of the smoothing parameters α and β which resulted in a stable filter constrained the parameters to lie within a stability triangle. He also derived closed form equations to relate the smoothing parameters for critically damped transient response and the ability of the filter to smooth white noise using a figure of demerit which was referred to as the *noise ratio*. Finally he proposed, via a numerical example, a procedure to optimally select the α - β parameters to minimize a performance index which is a function of

the noise-ratio and the tracking error for a specific maneuver. Following his work, Benedict and Bordner [2] used calculus of variations to solve for an optimal filter which minimizes a cost function which is a weighted function of the noise smoothing and the transient (maneuver following) response. They show that the optimal filter is coincident with an α - β filter with the constraint that $\beta = \alpha^2/(2 - \alpha)$.

Numerous researchers using assumptions of the noise characteristics develop optimal filters [12], [14] and [7] which are commonly called Kalman Filters. Those filters were first introduced in the 60's by Kalman [10] and [11].

Kalata [8] proposed a new parameter which he referred to as the *tracking index* to characterize the behavior of α - β and α - β - γ filters. The tracking index was defined as the ratio of the position maneuverability uncertainty to the position measurement uncertainty. He also presented a technique to vary the α - β - γ parameters as a function of the tracking index.

In this chapter, a detailed analysis of the α - β - γ filter is carried out. Section 2 discusses the bounds on the smoothing parameters for a stable filter. This is followed by a closed form derivation of the noise ratio for the α - β - γ filter in Section 3. In Section 4, a closed form expression for the steady state errors and a metric to gauge the transient response of the filter are derived followed by the optimization of the smoothing parameters for various cost functions in Section 5. The chapter concludes with some remarks in Section 6.

2.2 Stability Analysis

2.2.1 α - β Tracker

The α - β tracker is an one-step ahead position predictor that uses the current error called the innovation to predict the next position. The innovation is weighted by the smoothing parameter α and β . These parameters influence the behavior of the system

in terms of stability and ability to track the target. Therefore, it is important to analyze the system using control theoretic aspects to gauge stability and performance.

The form of the equations for the α - β tracker can be derived by considering the motion of a point mass with constant acceleration which is described by integrating *Newtons First Law* yielding $x(t) = x_0 + v_0t + \frac{1}{2}at^2$, where v is velocity and a is acceleration. If the acceleration is negligible and the equation of motion is written in discrete time where the initial condition x_0 and v_0 are substituted by the smoothed condition, the one-step ahead prediction equation for an α - β tracker is obtained :

$$x_p(k+1) = x_s(k) + Tv_s(k) \quad (2.1)$$

The use of the equation of motion without neglecting the acceleration would lead to the α - β - γ tracker discussed later on, in this work. Equation (2.1) states that between each time step, a linear motion is assumed and the smoothed conditions x_s and v_s are the initial conditions for each time step. The smoothed conditions are derived using the innovation and the previous states according to Equations (2.2) and (2.3).

$$x_s(k) = x_p(k) + \alpha(x_o(k) - x_p(k)) \quad (2.2)$$

$$v_s(k) = v_s(k-1) + \frac{\beta}{T}(x_o(k) - x_p(k)) \quad (2.3)$$

The innovation in Equation (2.2) and (2.3) defined as $(x_o(k) - x_p(k))$ represents the error between the observed and predicted position. As can be seen, α and β , the smoothing parameter, change the dynamics of the system. The input-output relationship between the observed and predicted position, $G = \frac{x_p}{x_o}$, is referred to as the system in this work.

Since the prediction equation, Equation (2.1), is in recursive form, it needs to be initialized. The initialization procedure requires two observed target positions to calculate the smoothed initial velocity. Therefore, the target position prediction begins with the third time step. The initial predicted target position is defined to be equal to the observed position at the second time step

$$x_p(2) = x_o(2) , \quad (2.4)$$

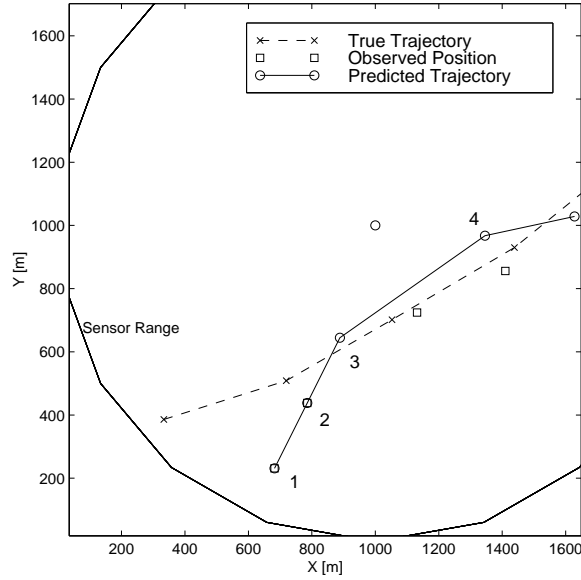


Figure 2.1: Target Track Initialization

leading to a zero initial innovation, which means that the smoothing parameter have no influence of the initial prediction. The smoothed initial velocity is calculated by the finite difference of the two observed target positions divided by the appropriate time,

$$v_s(2) = \frac{x_o(2) - x_o(1)}{T} \quad (2.5)$$

According to Equation (2.2) the smoothed position at the second time step equals the predicted target position at the same time. The first predicted position can therefore be calculated by the following equation:

$$x_p(3) = x_o(2) + Tv_s(2) , \quad (2.6)$$

which is an extrapolation of the first two observed target positions. Figure (2.1) illustrates the track initialization. It can be seen that the third predicted position is on the straight line formed by the first two observations and the first and the third position are equidistant from the second.

The behavior of the system can be expressed in terms of the smoothing parameters α and β ; furthermore, regions of stability and different transient response characteristics can be specified in the α - β space. Writing Equation (2.1), (2.2) and (2.3) in the z -domain and substituting x_s and v_s into the prediction Equation (2.1) yields the transfer function of the system in the z -domain $G(z)$ as follows,

$$G(z) = \frac{x_p}{x_o} = \frac{\alpha(z-1) + \beta z}{z^2 + (\alpha + \beta - 2)z + (1 - \alpha)} \quad (2.7)$$

which can now be used to determine the region of stability of the α - β filter. Stability requires that the roots of the characteristic polynomial lie within the unit circle in the z -domain. The characteristic polynomial is given by the denominator of Equation (2.7). To prove that the roots lie within the unit circle, one can transform Equation (2.7) into the w -domain, mapping the unit circle of the z -domain to the left half plane of the w -domain and applying one of the known stability criteria in continuous domain. Another approach is to check the stability directly in the z -domain using *Jury's Stability Test*.

2.2.2 Jury's Stability Test

The Jury's Stability Test can be used to analyze the stability of the system without explicitly solving for the poles of the system. Therefore, it is used to determine the bounds on the parameters which result in a stable transfer function in the z -domain.

For a system with a characteristic equation $P(z) = 0$, where

$$P(z) = a_0 z^n + a_1 z^{n-1} + \dots + a_{n-1} z + a_n \quad (2.8)$$

and $a_0 > 0$, we construct the table where the first row consists of the elements of the polynomial $P(z)$ in ascending order and the second row consists of the parameters in descending order [13]. The table is shown in Table 2.1. where

$$b_k = \begin{vmatrix} a_n & a_{n-1-k} \\ a_0 & a_{k+1} \end{vmatrix}, \quad k = 0, 1, 2, \dots, n-1 \quad (2.9)$$

Row	z^0	z^1	\dots	z^{n-1}	z^n
1	a_n	a_{n-1}	\dots	a_1	a_0
2	a_0	a_1	\dots	a_{n-1}	a_n
3	b_{n-1}	b_{n-2}	\dots	b_0	
4	b_0	b_1	\dots	b_{n-1}	
\vdots	\vdots	\vdots	\vdots	\vdots	
2n-5	p_3	p_2	\dots		
2n-4	p_0	p_1	\dots		
2n-3	q_2	q_1	q_0		

Table 2.1: General Form of Jury's Stability Table

$$c_k = \begin{vmatrix} b_{n-1} & b_{n-2-k} \\ b_0 & b_{k+1} \end{vmatrix}, \quad k = 0, 1, 2, \dots, n-2 \quad (2.10)$$

$$q_k = \begin{vmatrix} p_3 & p_{2-k} \\ p_0 & p_{k+1} \end{vmatrix}, \quad k = 0, 1, 2 \quad (2.11)$$

Note, that the last row of the table contains only three elements. The Jury's test states that a system is stable if all of the following conditions are satisfied:

$$|a_n| < a_0 \quad (2.12)$$

$$P(z)|_{z=1} > 0 \quad (2.13)$$

$$P(z)|_{z=-1} \begin{cases} > 0 \text{ for even } n \\ < 0 \text{ for odd } n \end{cases} \quad (2.14)$$

$$\begin{aligned} |b_{n-1}| &> |b_0| \\ |c_{n-2}| &> |c_0| \\ &\dots \end{aligned} \quad (2.15)$$

$$|q_2| > |q_0|$$

Exploiting this scheme for the characteristic polynomial of the α - β filter leads to the following *Jury's* table shown in Table 2.2. The condition that $a_0 > 0$ is satisfied since $a_0 = 1$. To satisfy the constraint

$$|a_n| < a_0 \quad (2.16)$$

Row	z^0	z^1	z^2
1	$(1 - \alpha)$	$(\alpha + \beta - 2)$	1

Table 2.2: Jury's Stability Table of the α - β Filter

we require

$$|1 - \alpha| < 1, \quad (2.17)$$

which is equivalent to

$$0 < \alpha < 2. \quad (2.18)$$

To satisfy the constraint

$$P(z)|_{z=1} > 0, \quad (2.19)$$

we require

$$1 + (\alpha + \beta - 2) + (1 - \alpha) > 0 \quad (2.20)$$

which can be rewritten as

$$\beta > 0. \quad (2.21)$$

To satisfy the constraint

$$P(z)|_{z=-1} > 0, \text{ for even } n, \quad (2.22)$$

we require

$$1 - (\alpha + \beta - 2) + (1 - \alpha) > 0 \quad (2.23)$$

which can be rewritten as

$$2\alpha + \beta < 4. \quad (2.24)$$

Equations 2.18, 2.21 and 2.24 defines the region where α and β may lie for the tracker to be stable. Plotting the boundaries of these constraints, one arrives at the stability triangle shown in Figure (2.2).

The stability area can be divided by the critical damped curve into an over-, and underdamped area as well as into areas with certain eigenfrequencies of the system. The system is said to be critically damped if the poles are coincident. Therefore, critical damping is obtained by solving the following equation.

$$z_1 = z_2, \quad \text{or} \quad (\alpha + \beta - 2)^2 = 4(1 - \alpha) \quad (2.25)$$

since,

$$z_{1,2} = \frac{-(\alpha + \beta - 2) \pm \sqrt{(\alpha + \beta - 2)^2 - 4(1 - \alpha)}}{2} \quad (2.26)$$

Solving Equation (2.25) leads to the following relationship

$$\beta_{1,2}^d = 2 - \alpha \pm 2\sqrt{1 - \alpha} \quad (2.27)$$

for critical damped response of the filter. The dashed line in Figure 2.2 corresponds to the solution $\beta_1^d = 2 - \alpha - 2\sqrt{1 - \alpha}$ and the dash-dot line to the solution $\beta_2^d = 2 - \alpha + 2\sqrt{1 - \alpha}$. Equation (2.27) is valid for all $\alpha \leq 1$, and the system is oscillating if the poles in Equation (2.26) contain a non-zero imaginary part. This can be seen by using the transformation between z - and s -domain.

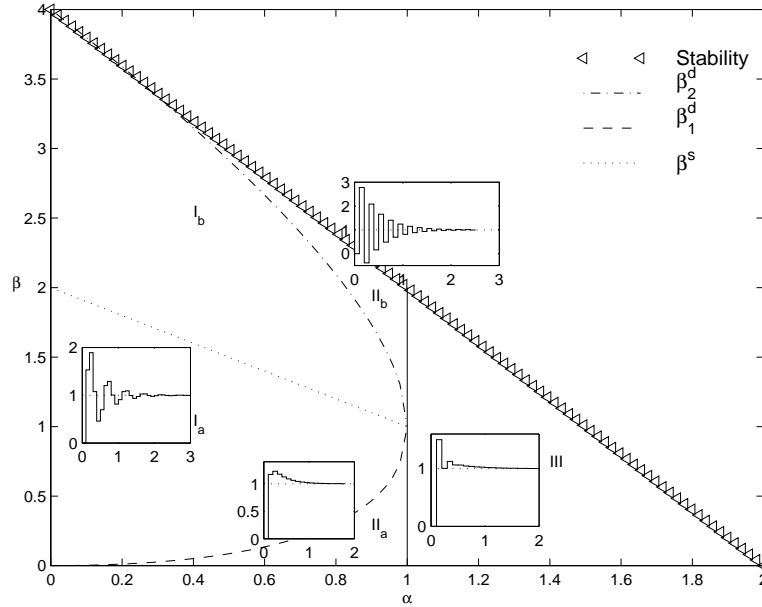
$$z = e^{sT} = e^{(\sigma + j\omega)T} = e^{\sigma T} e^{j\omega T} \quad (2.28)$$

We now show that, even though the smoothing parameters are chosen to be in the overdamped area, the system can oscillate under certain circumstances. These circumstances need to be investigated to achieve a specific transient response. Furthermore, expressions for the eigenfrequencies of the system will be derived.

The first part involves analyzing the space where α is less than one followed by the analysis for the region where we consider α greater than one. All areas discussed are also shown in Figure 2.2. Equation (2.26) shows that if the system is underdamped, the z -poles become a complex conjugate pair. In this case, Equation (2.26) can be rewritten as follows:

$$z_{1,2} = Ae^{j\psi}, \quad \text{where} \quad A = \sqrt{1 - \alpha} \quad (2.29)$$

$$\psi = \arctan \frac{\pm \sqrt{4\beta - (\alpha + \beta)^2}}{2 - \alpha - \beta}$$

Figure 2.2: Regions of the α - β Tracker

Comparing Equation (2.28) with Equation (2.29) yields the following equation for the eigenfrequency ω .

$$\omega = \frac{1}{T} \arctan \frac{\pm \sqrt{4\beta - (\alpha + \beta)^2}}{2 - \alpha - \beta} \quad (2.30)$$

Equating ω to zero, which corresponds to the critically damped case, simplifies Equation 2.30, resulting in Equation 2.27. The effect of α and β on the eigenfrequency can be easily interpreted using Equation 2.30 unlike Equation 2.26. Representing the poles of Equation (2.28) as a vector in the complex domain, as shown in Figure (2.3), various regions of the α - β space can be easily analyzed. The highest frequency is obtained if the vector in Figure 2.3 lies on the real axis and points to negative infinity, so that $\omega T = \pi$ implying $\omega = \frac{\pi}{T} [\frac{rad}{s}] = \frac{1}{2T}$ [Hz] which in turn is on the critical damped curve corresponding to β_2^d . Equation (2.30) can be used to determine when the real part of the poles changes sign, which corresponds the vector in Figure 2.3 subtending an angle $\pi/2$ to the real axis which corresponds to a frequency of $\omega = \frac{1}{4T}$ [Hz]. If the denominator of the argument in Equation (2.30) becomes zero, the real part is also zero, and leads to the line $\beta^s = 2 - \alpha$ illustrated by the dotted line in Figure 2.2.

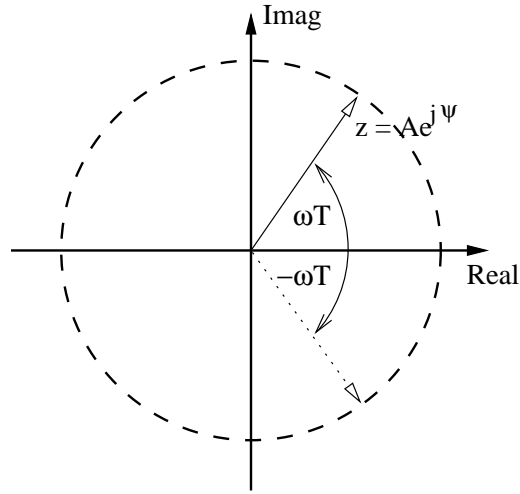


Figure 2.3: The Poles as a Vector in the Complex Domain

This divides the region for α less than one into an area where the poles have positive and negative real parts as follows:

$$\begin{aligned} \text{if } \beta < \beta^s &\rightarrow \text{positive real part } (I_a) \\ \text{if } \beta > \beta^s &\rightarrow \text{negative real part } (I_b) \end{aligned}$$

Although the regions II_a and II_b correspond to the overdamped area as shown in Figure (2.2), the region II_b corresponds to oscillatory dynamics with a constant frequency of $\omega = \frac{1}{2T} [\frac{1}{s}]$. Remembering that the vector in Figure 2.3 corresponding to the critical damped curve β_2^d lies on the real axes with a phase of π , any set of α and β in the region II_b does not change the phase since it only changes the magnitude of the vector. Within the region II_b the frequency of Equation (2.30) becomes complex which if substituted in Equation (2.28), leads to a negative real pole. For example, for the stability boundary we have,

$$\beta = 4 - 2\alpha, \quad 0 < \alpha \leq 1 \quad (2.31)$$

which results in

$$\psi = \arctan\left(\frac{\pm j\alpha}{\alpha - 2}\right) = j \operatorname{arctanh}\left(\frac{\pm \alpha}{\alpha - 2}\right). \quad (2.32)$$

The roots of the filter can now be represented as

$$z_{1,2} = \sqrt{(1 - \alpha)} \exp \left[-\operatorname{arctanh} \left(\frac{\pm \alpha}{\alpha - 2} \right) \right] e^{j\pi} \quad (2.33)$$

which corresponds to an oscillatory response with a frequency of $\omega = \frac{\pi}{T}$ [$\frac{rad}{s}$] which cannot be derived from Equation 2.26.

If α becomes greater than 1, the roots of Equation (2.26) are never negative, so the above approach cannot be applied. When $\alpha > 1$, Equation 2.26 leads to the following two poles.

$$z_1 = \tilde{A}_1 e^{j0} \quad \text{and} \quad z_2 = \tilde{A}_2 e^{j\pi}, \quad (2.34)$$

where

$$\begin{aligned} \tilde{A}_1 &= -\frac{\alpha + \beta - 2}{2} + \frac{1}{2} \sqrt{(\alpha + \beta - 2)^2 + 4(\alpha - 1)} \\ \tilde{A}_2 &= \frac{\alpha + \beta - 2}{2} + \frac{1}{2} \sqrt{(\alpha + \beta - 2)^2 + 4(\alpha - 1)} \end{aligned}$$

As can be seen in Equation (2.34), in the region $\alpha > 1$, one pole is oscillating with $\omega = \frac{1}{2T}$ and the other corresponds to an overdamped mode with zero frequency.

2.2.3 α - β - γ Tracker

The α - β tracker is obtained by neglecting the acceleration term in the equation of motion of a point mass. Deriving a tracker which includes the acceleration, is a better representation of the equation of motion, leading to the α - β - γ tracker.

The equation of the higher order one-step ahead prediction is the same as for the α - β tracker with an additional term representing the influence of the acceleration.

$$x_p(k+1) = x_s(k) + T v_s(k) + \frac{1}{2} T^2 a_s(k), \quad (2.35)$$

The additional information about the acceleration allows us to predict the velocity of the target as well. Equation (2.36).

$$v_p(k+1) = v_s(k) + T a_s(k), \quad (2.36)$$

where the smoothed kinematic variables are again calculated by weighting the innovation as follows:

$$x_s(k) = x_p(k) + \alpha(x_o(k) - x_p(k)) \quad (2.37)$$

$$v_s(k) = v_p(k) + \frac{\beta}{T}(x_o(k) - x_p(k)) \quad (2.38)$$

$$a_s(k) = a_s(k-1) + \frac{\gamma}{2T^2}(x_o(k) - x_p(k)) \quad (2.39)$$

Similar to the α - β filter are the assumptions about the initial conditions of the α - β - γ filter. Since the velocity is also predicted, Equation (2.36), the initialization requires three observed target positions. Equations 2.4 and 2.5 are now used one time step ahead:

$$x_p(3) = x_o(3) \quad (2.40)$$

$$v_s(3) = \frac{x_o(3) - x_o(2)}{T}, \quad (2.41)$$

and the smoothed initial acceleration is calculated by the finite difference of the two initial velocities as follows:

$$a_s(3) = \frac{v_p(3) - v_p(2)}{T} = \frac{x_o(3) - x_o(2)}{T^2}. \quad (2.42)$$

The first target position prediction is now available at the fourth time step:

$$x_p(4) = x_p(3) + Tv_s(3) + \frac{1}{2}T^2a_s(3) \quad (2.43)$$

$$v_p(4) = v_s(3) + Ta_s(3). \quad (2.44)$$

Depending on the target, the initial acceleration might be neglected and set to be zero. Thus, the amount of required initial observed points reduces to the requirements of an α - β filter.

Applying the Laplace Transform to Equation (2.35) to (2.39) and solving for the ratio $\frac{x_p}{x_o}$ leads to the transfer function in z -domain which is

$$G(z) = \frac{\alpha + (-2\alpha - \beta + \frac{1}{4}\gamma)z + (\alpha + \beta + \frac{1}{4}\gamma)z^2}{z^3 + (\alpha + \beta + \frac{1}{4}\gamma - 3)z^2 + (-2\alpha - \beta + \frac{1}{4}\gamma + 3)z + \alpha - 1} \quad (2.45)$$

Equation 2.45 can now be used to determine the bounds of α , β and γ for stability. For this complex system, the *Jury's Stability Test* is used as described in Section 2.2.2, to determine the region of stability.

Writing the coefficients of the characteristic polynomial in Jury's Table, and calculating the determinants b_2 , b_1 and b_0 (Equation 2.9) yield the Table 2.3. The

Row	z^0	z^1	z^2	z^3
1	$\alpha - 1$	$-2\alpha - \beta + \frac{1}{4}\gamma + 3$	$\alpha + \beta + \frac{1}{4}\gamma - 3$	1
2	1	$\alpha + \beta + \frac{1}{4}\gamma - 3$	$-2\alpha - \beta + \frac{1}{4}\gamma + 3$	$\alpha - 1$
3	$\alpha(\alpha - 2)$	$\alpha(4 - 2\alpha - \beta + \frac{1}{4}\gamma) - \frac{1}{2}$	$\alpha(\alpha + \beta - 2 + \frac{1}{4}\gamma) - \frac{1}{2}$	

Table 2.3: Jury's Stability Table of the α - β - γ Filter

condition $a_0 > 0$ is satisfied since $a_0 = 1$. To satisfy the constraint $|a_n| < a_0$, the coefficients require $|\alpha - 1| < 1$, which is equivalent to

$$0 < \alpha < 2 . \quad (2.46)$$

Substituting $z = 1$ and applying the constraint $P(z)|_{z=1} > 0$, requires satisfaction of the inequality

$$1 + (\alpha + \beta + \frac{1}{4}\gamma - 3) + (-2\alpha - \beta + \frac{1}{4}\gamma + 3) + \alpha - 1 > 0, \quad (2.47)$$

which can be rewritten as

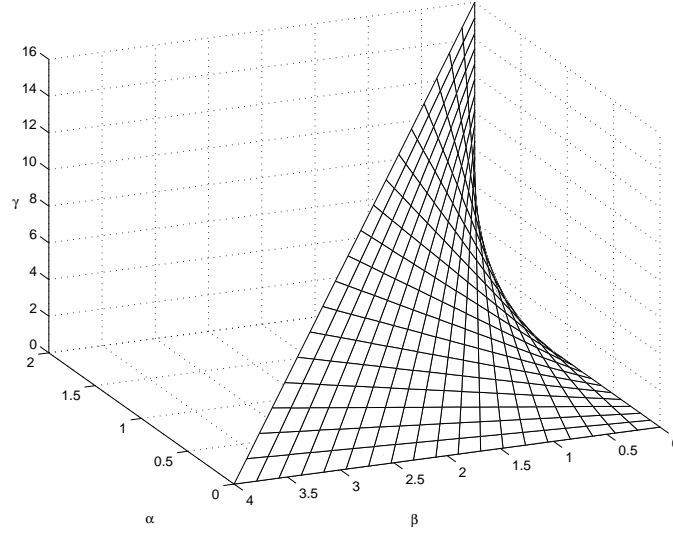
$$\gamma > 0 . \quad (2.48)$$

Satisfying the constraint $P(z)|_{z=-1} < 0$, for odd n , yields

$$2\alpha + \beta < 4 , \quad (2.49)$$

which is the same constraint for α and β as for the α - β tracker. The final condition $|b_2| > |b_0|$ requires

$$|\alpha(\alpha - 2)| > |\alpha(\alpha - 2) + \alpha(\beta + \frac{1}{4}\gamma) - \frac{1}{2}\gamma| \quad (2.50)$$

Figure 2.4: Stability Area of the α - β - γ Tracker

Observing Equation (2.50) and knowing the fact that $\alpha(\alpha - 2)$ is always negative within the stability area, we have:

$$\alpha\left(\beta + \frac{1}{4}\gamma\right) - \frac{1}{2}\gamma > 0. \quad (2.51)$$

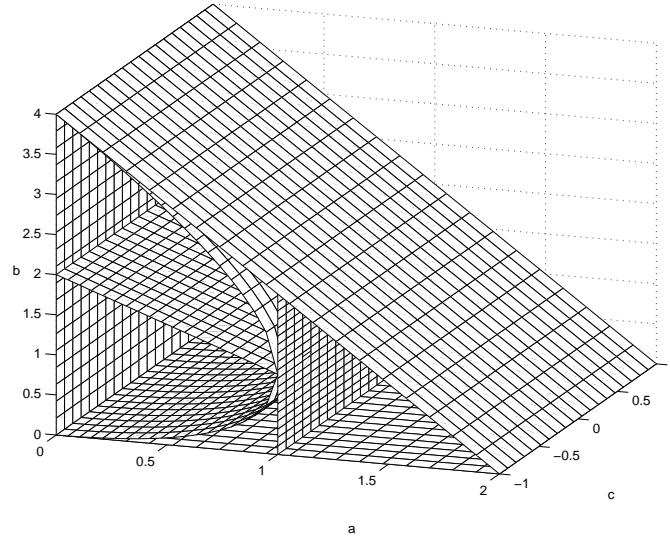
This statement leads to the constraint on γ for which the α - β - γ tracker is stable, which is

$$\boxed{\gamma < \frac{4\alpha\beta}{2 - \alpha}} \quad (2.52)$$

Figure (2.4) illustrates the bounding surfaces which include the stable volume in the α - β - γ space based on Equation (2.46), (2.48), (2.49) and (2.52).

It is desirable to divide the stability volume into regions which are characterized by specific class of transient responses such as, underdamped, overdamped, and critically damped. However, the difficulty of factorizing the characteristic polynomial of the transfer function in the α - β - γ space prompt us to conceive of a new space which we refer to as the a - b - c space. In this space, the characteristic polynomial is represented as

$$(z + c)(z^2 + (a + b - 2)z + 1 - a), \quad (2.53)$$

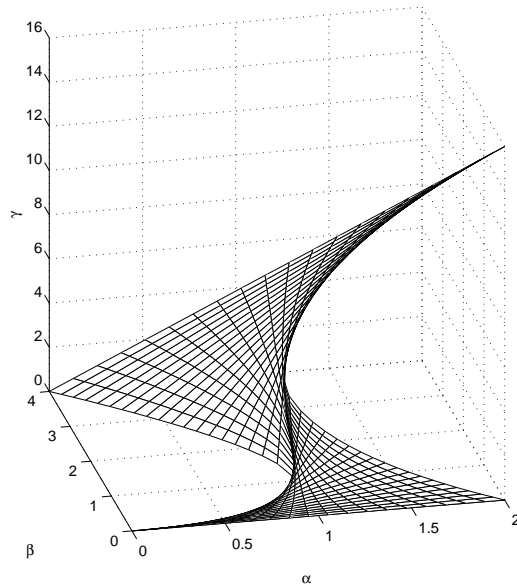
Figure 2.5: Stability Prism in the a - b - c Space

where the second order factor has a form which is identical to the characteristic equation of the α - β filter and the third pole is real and is located at $z = -c$. Comparing the denominator of Equation (2.45) with Equation (2.53) the following transformation is derived:

$$\begin{aligned}\alpha &= 1 + c(1 - a) \\ \beta &= a(1 + c) + \frac{1}{2}b(1 - c) \\ \gamma &= 2b(1 + c).\end{aligned}\tag{2.54}$$

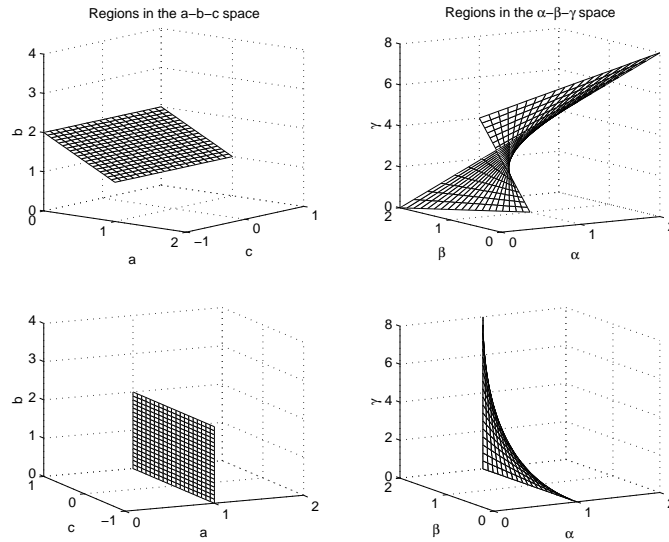
The usefulness of this transformation, becomes evident when one derives the stability volume of the α - β - γ filter. Since, c is constrained to lie within -1 and 1 , and the a - b space resembles the α - β space, the stability volume in the a - b - c space is a prism (Figure (2.5)) with a triangular cross-section which is derived from the α - β filter. Mapping the stability prism in the a - b - c space to the α - β - γ space using Equation (2.54), we rederive the stability volume illustrated in Figure (2.4).

Since, the pair of poles of Equation (2.53) which are functions of a and b are responsible for oscillation of the system, the a - b - c space is divided by extruding the lines which divide the stability triangle of the α - β filter (Figure 2.2), in the c dimen-

Figure 2.6: Critical Damped Surfaces of the α - β - γ Space

sion. These surfaces, shown in Figure (2.5), are transformed using Equation (2.54) to the α - β - γ space. Figure (2.6) shows the surfaces in the α - β - γ space corresponding to each critically damped surface of the a - b - c space. Figure (2.7) shows the transformation of the two surfaces dividing the stability area at $a = 1$ and $b = 2 - a$.

Observing Figures (2.6) and (2.7), illustrates the fact that for $\gamma = 0$, the third order tracker reduces to the α - β tracker. Substituting $\gamma = 0$ in the transfer function (Equation (2.7)), results in a pole zero cancellation at $z = 1$, resulting in a second order tracker. From Equation (2.54), we can infer that c equals -1 when $\gamma = 0$, and furthermore a and b degenerate to α and β . The cross-section at $c = -1$ therefore corresponds to the α - β tracker. Note that $c = 0$ does not result in degenerating the $\alpha - \beta - \gamma$ to the α - β filter.

Figure 2.7: Mapping between a - b - c Space and α - β - γ Space

2.3 Performance of the Target Tracker

2.3.1 Noise Ratio

Measurement noise significantly effects the performance of target trackers. It is therefore, of interest to characterize the noise filtering strength of the trackers.

The process of tracking includes in both α - β tracker and α - β - γ trackers, measurement noise and system noise. Therefore, expressions are derived to gauge the influence of noise on the output of the filter.

In this section, we derive a closed form expression for the noise ratio parameterized in terms of the a - b - c parameters whose relationship to α , β and γ is uniquely known.

Studying the effect of noisy signals requires a metric which measures the influence of the noise on the system. Since, the response of the system to a noisy input, can reflect this influence, the *noise ratio* is defined as the ratio of the root mean square value (RMS) of the system response to the RMS value of the noisy input. The

noise ratio is defined as follows.

$$\rho \equiv \sqrt{\frac{\overline{x_p^2(t)}}{\overline{x_o^2(t)}}} \quad (2.55)$$

Since, we require the tracker to reject measurement noise, a small value of ρ implies an excellent filtering of noise. The mean-square of $x_p(t)$ can be derived in the time domain using the standard integral over the time from $-\infty$ to $+\infty$.

$$\overline{x_p^2(t)} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T x_p^2(t) dt \quad (2.56)$$

Assuming the input is known, the response $x_p(t)$ of the system can be evaluated as follows by using the transfer function, $G(s)$.

$$x_p^2(t) = [\mathcal{L}^{-1}\{G(s)x_o(s)\}]^2, \quad (2.57)$$

where the \mathcal{L}^{-1} represents the inverse Laplace Transformation and $x_o(s)$ is the Laplace transformation of the input. The input noise is assumed to be white noise, so that the value of the noise input at any time is independent of previous values. Therefore, the sampled noise can be evaluated as a train of independent impulses, where the * indicates the sampling.

$$x_o^*(t) = \sum_{n=0}^N x_o(nT)\delta(t - nT) \quad (2.58)$$

Using Equation (2.57) yields the response for the impulse train.

$$x_p^2(t) = \left[\sum_{n=0}^N x_o(nT)\delta(t - nT) \right]^2 [\mathcal{L}^{-1}\{G(s)\}]^2, \quad (2.59)$$

since, the impulse responses are uncorrelated. Thus the the mean square value of the system response to each impulse equals to the mean square value of the system response according to the complete impulse train, the RMS value of $x_p(t)$ can be calculated by first deriving the response to each impulse and than determining the ensemble average.

$$\overline{x_p^2(t)} = \sum_{n=0}^N \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T [x_o\delta(t - nT)(nT)\mathcal{L}^{-1}\{G(s)\}]^2 dt. \quad (2.60)$$

The averaged impulse value is taken over an arbitrary time interval $nT < t < (n+1)T$ to latter deriving the ensemble average. Since this interval is one sampling time unit long the Equation (2.55) can now be derived as follows:

$$\rho^2 = \frac{1}{T} \lim_{T \rightarrow \infty} \int_0^{+\infty} [\mathcal{L}^{-1}\{G(s)\}]^2 dt \quad (2.61)$$

Fortunately, the definition of the noise ratio reduced to find the integral of the inverse Laplace transform of the transfer function $G(s)$, of the tracker.

Equation (2.61) could be solved in the time domain by finding the Laplace inverse of $G(s)$ [16], or by integrating Equation (2.61) in the discrete domain (z -domain) by rewriting the continuous time integral (Equation (2.61)) in the discrete domain as :

$$\rho^2 = \frac{1}{T} \sum_{n=0}^{+\infty} T g^2(n) = \sum_{n=0}^{+\infty} g^2(n) \quad (2.62)$$

Applying Parseval's Theorem [13] and the Residue Theorem to the sum of Equation (2.62) yield the final expression in the discrete domain.

$$\sum_{n=0}^{+\infty} g^2(n) = \frac{1}{2\pi j} \oint_C G(z)G(z^{-1})z^{-1}dz = \sum_{\nu=1}^p \text{Res}_{z_\nu} [G(z)G(z^{-1})z^{-1}], \quad (2.63)$$

where p is the number of poles on or inside the unit circle. The discrete transfer functions for the α - β tracker (Equation (2.7)) and for the α - β - γ tracker (Equation (2.45)) can be used to solve for the noise ratio. Note, that these transfer functions differ from those used by Benedict and Bordner [2] and by Simpson [15] respectively. Since the transfer function of the α - β - γ tracker can always be reduced to the α - β filter by setting γ to zero, only the noise-ratio of the three parameter filter is derived in the following section.

The derivations of the residues of the α - β - γ filter become difficult since the poles within the unit circle contain third order roots. Therefore, it is convenient to solve the residues in the a - b - c space introduced in Equation (2.54). The transfer function, Equation (2.45), is rewritten in the a - b - c space as:

$$G(z) = \frac{(1 + c + a + b)z^2 + (-2 + bc - 2c + ca - a)z + 1 + c - ca}{(z + c)(z^2 + (a + b - 2)z + 1 - a)}, \quad (2.64)$$

where the poles are decomposed into a set of second order poles and one first order pole such that:

$$z_1 = -c \quad (2.65)$$

$$z_{2,3} = -\frac{1}{2}(a+b-2) \pm \sqrt{(a+b-2)^2 - 4(1-a)}. \quad (2.66)$$

The line integral in Equation (2.63) is carried out along the unit circle guaranteeing that all stable poles of $G(z)$ lie within and the poles of $G(z^{-1})$ lie outside the unit circle. The three residues lying within the unit circle can be derived as follows:

$$\text{Res } f(z) = \lim_{z \rightarrow z_\nu} (z - z_\nu) \cdot f(z). \quad (2.67)$$

From Equation (2.63), the noise-ratio can now be calculated by using the Equations (2.65) to (2.67) which leads to the following result:

$$\begin{aligned} \rho^2 &= \frac{-(4ba^2 + 2ab^2 + 4b^2) - 4b(1+c)k_1}{2a(b+2a-4)b + a(1+c)k_2} \quad (2.68) \\ k_1 &= k_{11}c^2 + k_{12}c + k_{13} \\ k_{11} &= 4a + 2a^3 - ab + ba^2 - 6a^2 \\ k_{12} &= ba^2 + ab^2 - 6ab - 2a^3 + 8a \\ k_{13} &= 6a^2 - 2ab^2 - 4ba^2 + 4a - 2b^2 + 7ab \\ k_2 &= (b+2a-4)(c^2a - ca - c^2 - 2b + bc + 1) \end{aligned}$$

Constant noise-ratio surfaces are obtained by solving Equation (2.68) for either a , b or c . A simple solution for b as a function of $f(a, c, \rho)$ exists, which consists of two solutions of b , where one is always outside the stability prism. A typical constant noise-ratio ($\rho^2 = 10$) surface is shown in Figure (2.8). Applying the transformation of Equation (2.54) to each point of the constant noise-ratio surface in the a - b - c space yields the constant noise-ratio surface in the α - β - γ space shown in Figure (2.9).

As mentioned in the section of stability, the α - β - γ filter reduces to a two parameter tracker if γ becomes zero inferring that $c = -1$ and a and b degenerate to α and β . Thus, the noise-ratio of the α - β - γ filter reduces to the α - β filter by applying

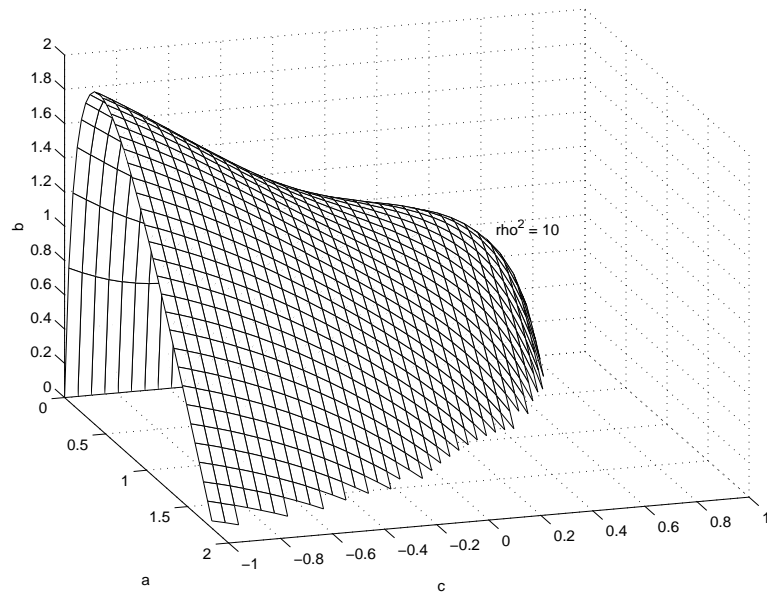


Figure 2.8: Constant ρ -Surface in the a - b - c Space

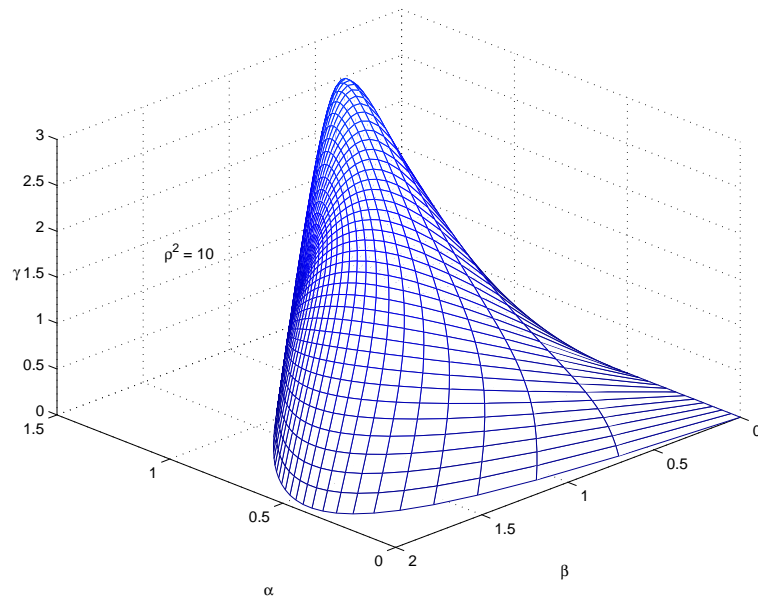
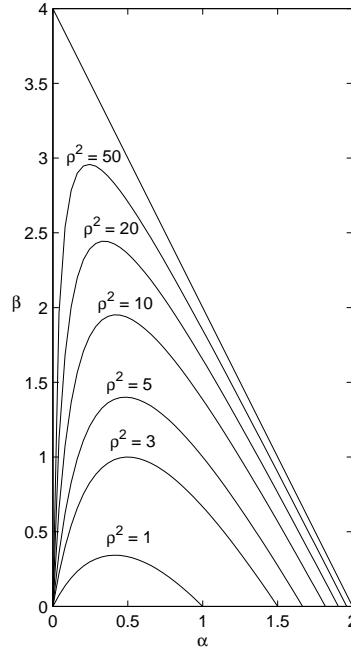


Figure 2.9: Constant ρ Surface in the α - β - γ Space

Figure 2.10: Constant ρ curves in the α - β Space

these conditions to Equation (2.68).

$$\rho^2 = \frac{2\alpha^2 + \alpha\beta + 2\beta}{\alpha(4 - \beta - 2\alpha)} \quad (2.69)$$

The line of constant noise-ratio for the α - β filter is also included in the Figures 2.8 and 2.9 by setting $c = -1$ and respectively $\gamma = 0$. However for clarity, the constant noise ratio curves are plotted in the α - β space for various noise ratios as shown in Figure 2.10. Equation (2.69) is different from those derived by Sklansky [16] and Benedict and Bordner [2]. For some reasons Benedict and Bordner use a different transfer function $G(z)$, to represent the impulse response of the filter. To prove the veracity of Equation (2.69), numerical simulations are carried out. Results of simulating an α - β filter for normally distributed white noise are used to calculate the noise ratio by calculating the ratio of the root-mean-square value of the output and input. Table 2.4 displays the simulation result for the parameter set $\alpha = 0.5$ and $\beta = 0.7$. As is clear from the table, the solution of Sklansky and Benedict and Bordner do not match the results of the simulation while Equation 2.69 matches the simulated results.

Simulation	Sklansky	Benedict & Bordner	Proposed (Eq. 2.69)
1.9540	1.2058	0.7391	1.9565

Table 2.4: Noise-ratio of different Approaches

Finally, various constant noise-ratio surfaces are shown in Figure (2.11), where ρ^2 is chosen to $\rho^2 = 1, 5, 50$. Different viewpoints in the α - β - γ space and a - b - c space are selected to better reveal the shapes of these surfaces.

2.3.2 Maneuver Error

Numerous metrics can be used to gauge the performance of a tracker, such as: the steady state tracking error, the transient response of the tracker, the noise ratio, besides others. In this work, we endeavour to derive closed form expressions for the steady state errors and the sum of the square of the errors which captures the transient behavior of the filter for the α - β and the α - β - γ filters for two classes of trajectories. The first is a circular trajectory with the target moving at constant speed and the second is a straight line trajectory where the target moves with constant acceleration. These metrics can then be exploited to determine optimal sets of parameters for any cost function which are weighted combinations of these metrics.

The Tracking Error

The tracking error of the two or three parameter filter is defined as the difference between the observed and the predicted position. Since the performance of the filter due to the maneuver error is developed in the discrete complex domain, a error-input transfer function needs to be derived.

The innovation in the smoothing Equations (2.2) and (2.3) represents the maneuver error of the filter. Substituting the innovation with the error $e(k)$, and transforming the prediction Equation (2.1) and the smoothing Equations (2.2) and

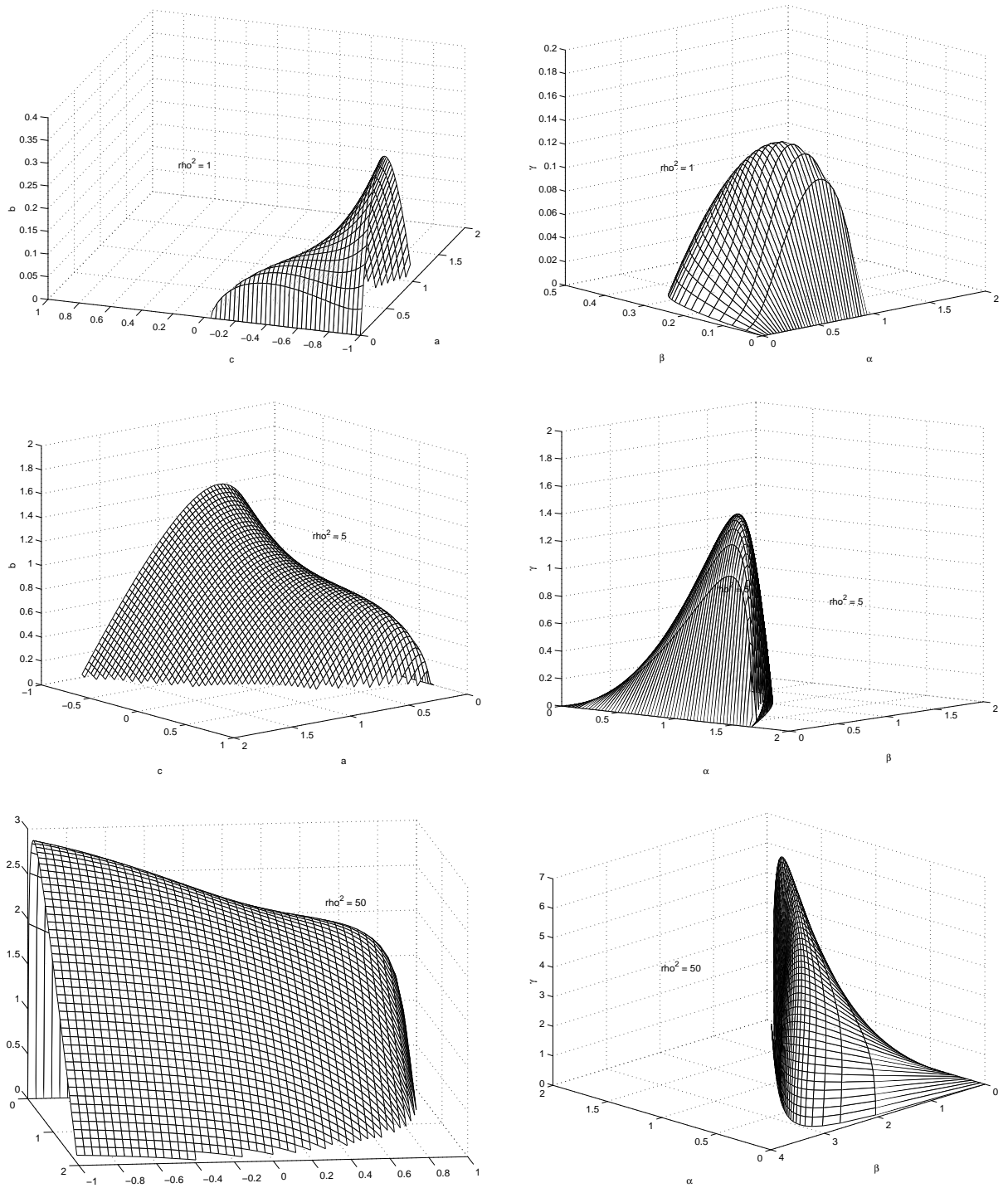


Figure 2.11: Constant noise-ratio Surfaces for different values of ρ

(2.3) into the z -domain, yields the relationship between the maneuver error and the predicted position, which is:

$$\frac{e}{x_p} = \frac{(z-1)^2}{\alpha(z-1) + \beta z} . \quad (2.70)$$

Furthermore, the transfer function of the maneuver error and the observed position can be obtained by substituting the predicted position with the transfer function of the α - β filter given in Equation (2.7). Thus, the desired relationship becomes:

$$e_{\alpha\beta} = \frac{(z-1)^2}{z^2 + (\alpha + \beta - 2)z + (1 - \alpha)} x_o \quad (2.71)$$

Extending this approach to the α - β - γ filter, described by Equations (2.35) to (2.45), leads to the following maneuver error transfer function:

$$e_{\alpha\beta\gamma} = \frac{(z-1)^3}{z^3 + (\alpha + \beta + \frac{1}{4}\gamma - 3)z^2 + (-2\alpha - \beta + \frac{1}{4}\gamma + 3)z + \alpha - 1} x_o . \quad (2.72)$$

Likewise in the transfer function $G(z)$ of Equation (2.45), the maneuver error transfer function of the α - β - γ filter reduces to the maneuver error transfer function by setting γ to zero, which leads to a pole-zero cancellation at $z = 1$.

Circular Trajectory

The tracking error of a filter following a circular trajectory can be resolved into components e_x and e_y in the x and y directions respectively. The net error magnitude is given by:

$$e(t) = \sqrt{e_x^2(t) + e_y^2(t)} . \quad (2.73)$$

Assuming that the target traces a circular path while moving at constant speed resulting in a constant angular velocity, the target's position can be described in the time domain using the projection of a rotating phasor, via the cosine and sine function. The position can also be written in the complex domain as:

$$\begin{aligned} x_o(t) &= Re^{j\omega t} \\ y_o(t) &= jRe^{j\omega t} \end{aligned} \quad (2.74)$$

Substituting Equation (2.74) into the error transfer function (Equation (2.72)), and substituting $z = e^{j\omega T}$, the steady state values of the errors can be represented as,

$$\begin{aligned} e_x(nT) &= Re_{\alpha\beta\gamma}(z = e^{j\omega T})e^{j\omega nT} = \hat{R}e^{j(\omega nT + \psi)} \\ e_y(nT) &= jRe_{\alpha\beta\gamma}(z = e^{j\omega T})e^{j\omega nT} = j\hat{R}e^{j(\omega nT + \psi)}, \end{aligned} \quad (2.75)$$

$$\text{where } \hat{R} = |Re_{\alpha\beta\gamma}(e^{j\omega T})|, \quad \psi = \arg \{Re_{\alpha\beta\gamma}(e^{j\omega T})\}.$$

Equation (2.75) states that the magnitude of the maneuver error for a circular path is constant at all sampling instants, and only the phase is changing with each time step. To derive an expression for the tracking error for an α - β - γ filter, we substitute the corresponding transfer function $F(e^{j\omega T})$ into equation 2.75. The maneuver error, Equation (2.73), of a circular target path of an α - β - γ tracker can now be simplified to the following equation:

$$e = \frac{4R \sin^3(\frac{\omega T}{2})}{\sqrt{\sin^2(\frac{\omega T}{2})[(2 - \alpha) \cos \omega T + \alpha + \beta - 2]^2 + \alpha^2 \sin^2 \omega T} + \gamma(\frac{\gamma}{32}(\cos \omega T + 1) - \frac{\alpha}{4} \sin^2 \omega T)}. \quad (2.76)$$

To determine the steady state error for an α - β filter, we can substitute the corresponding transfer function $F(e^{j\omega T})$, or set γ to zero in Equation (2.76). This leads to the maneuver error of the α - β filter.

$$e = \frac{4R \sin^2(\frac{\omega T}{2})}{\sqrt{[(2 - \alpha) \cos \omega T + \alpha + \beta - 2]^2 + \alpha^2 \sin^2 \omega T}}, \quad (2.77)$$

where the maneuver error of the α - β filter is the same as the closed form expression derived by Sklansky [16]. Sklansky [16] proposed simplifying the expression for $\omega T \ll \pi$ by neglecting the term under the radical, reducing Equation (2.77) to:

$$e \approx \frac{a_r T^2}{\beta}, \quad (2.78)$$

where a_r is the centripetal acceleration given by $R\omega^2$.

Straight Line Maneuvers

Besides the circular maneuver, the straight line maneuver is one which needs to be studied carefully. This section presents derivation of closed form expression for steady state errors and the sum of the square of the errors for α - β and α - β - γ filters.

Unlike the derivation of the error for the circular path, the error for a straight line trajectory can be simplified by assuming that the target path is coincident with one of the reference axis reducing the problem to a single dimension. Assuming constant acceleration along the straight line and zero initial position, the targets position is given by:

$$x_o(t) = \frac{1}{2}a_s t^2 + vt \Leftrightarrow x_o(z) = \frac{a_s T^2 z(z+1)}{2(z-1)^3} + \frac{vTz}{(z-1)^2} \quad (2.79)$$

where a_s is the acceleration and v is the initial velocity. The choice of constant acceleration is made to illustrate the ability of a higher order tracker to track the target without a steady state error. Simulating the α - β tracker for this maneuver, results in a steady state error, in contrast to the α - β - γ tracker which exhibits no steady state error. Similar to the derivation of Equation (2.75), the maneuver error for a straight line is obtained by substituting the z-transform of x_o into Equation (2.72). The maneuver error for the α - β tracker and the α - β - γ trackers are:

$$e_{\alpha\beta} = \frac{\frac{1}{2}a_s T^2 z(z+1) + vTz(z-1)}{(z^2 + (\alpha + \beta - 2)z + 1 - \alpha)(z-1)} \quad (2.80)$$

$$e_{\alpha\beta\gamma} = \frac{\frac{1}{2}a_s T^2 z(z+1) + vTz(z-1)}{z^3 + (\alpha + \beta + \frac{1}{4}\gamma - 3)z^2 + (-2\alpha - \beta + \frac{1}{4}\gamma + 3)z + \alpha - 1}. \quad (2.81)$$

The steady state error is obtained by applying the final value theorem [13] to Equation (2.80) and (2.81), which for the α - β tracker leads to the interesting result,

$$\begin{aligned} e_{\alpha\beta}(t \rightarrow \infty) &= \lim_{z \rightarrow 1} (z-1)e_{\alpha\beta}(z) \\ &= \frac{a_s T^2}{\beta} \end{aligned} \quad (2.82)$$

which is coincident with the maneuver error for a circular path with high sampling rate as derived by Sklansky [16] (Equation (2.78)). As can be seen from Equation (2.81),

the steady state error of the α - β - γ tracker

$$e_{\alpha\beta\gamma}(t \rightarrow \infty) = 0 \quad (2.83)$$

vanishes for the case of constant acceleration.

The transient response can be characterized by a metric which is defined as the sum of the square of the tracking errors as time tends to infinity. This metric can be defined in the continuous and discrete time as

$$J_c = \int_0^{\infty} e^2(t) dt, \quad (2.84)$$

$$J = \sum_{k=0}^{\infty} T e^2(k). \quad (2.85)$$

This metric can be calculated by simulating the response of the transfer function given in Equation (2.80) and (2.81), and summing the resulting error square $= (u - y)^2$, where y is the system response and u is the true target position. Instead using simulations, a closed form expression can be derived using *Parseval's Theorem* [13]. Rewriting Equation (2.85) and using Parseval's Theorem and the *Residue Theorem* we have:

$$J = \frac{1}{2\pi j} \oint_C e(z)e(z^{-1})z^{-1} dz = \sum_{\nu=1}^p \operatorname{Res}_{z_\nu} [e(z)e(z^{-1})z^{-1}], \quad (2.86)$$

where the line integral is carried out along a closed curve C , which is also given by the sum of the residues at the singular points within the closed curve C . Similar to the derivations of the noise-ratio, the closed curve C is the unit circle which includes all the stable poles of $e(z)$.

As shown in Equation (2.82), the α - β tracker exhibits a non-zero final value, so that the metric defined in Equation (2.85) is not finite. However, subtracting the steady state error from the transient error yields a finite solution for the metric. To derive the finite metric which characterizes the transient behavior of the α - β filter tracking a constant acceleration profile, a new error transfer function is defined as

$$e_{\alpha\beta}(z) = \frac{\frac{1}{2}a_s T^2 z(z+1) + vTz(z-1)}{(z^2 + (\alpha + \beta - 2)z + 1 - \alpha)(z-1)} - \frac{a_s T^2}{\beta} \frac{z}{z-1}, \quad (2.87)$$

where the z-transform of the steady state error is subtracted from the transfer function of the filter. Solving Equation (2.86) by solving for the residues of the three poles of $e_{\alpha\beta}(z)$ yields a closed form equation for the metric for the α - β tracker.

$$J_{\alpha\beta} = \frac{v^2 T^3 (\alpha - 2)}{\alpha\beta(2\alpha + \beta - 4)} + \frac{\frac{1}{4} a_s^2 T^5 (2\beta + \alpha\beta + 2\alpha^2)}{\alpha\beta^3} + \frac{v a_s T^4}{\beta^2} \quad (2.88)$$

The first expression in Equation (2.88) is the transient cost function, corresponding to a constant velocity target motion. The second expression corresponds to the metric for a target path which is parameterized by constant acceleration and zero initial velocity. These two expressions are labeled as follows:

$$J_{\alpha\beta,v} = \frac{v^2 T^3 (\alpha - 2)}{\alpha\beta(2\alpha + \beta - 4)}, \quad J_{\alpha\beta,a} = \frac{\frac{1}{4} a_s^2 T^5 (2\beta + \alpha\beta + 2\alpha^2)}{\alpha\beta^3}. \quad (2.89)$$

Since the α - β - γ filter tracks the target without steady state error, the metric defined in Equation (2.85) is finite. Thus, the cost function of an α - β - γ tracker for an accelerating target along the straight line given in Equation (2.86) is obtained by using the error transfer function (Equation (2.81)). Since, the derivations of the cost function in the α - β - γ space is more complex than in the a - b - c space, Equation (2.81) is transformed to the a - b - c space, where the error transfer function becomes:

$$e(z) = \frac{\frac{1}{2} a_s T^2 z(z+1) + v T z(z-1)}{(z+c)(z^2 + (a+b-2)z + 1-a)}. \quad (2.90)$$

A closed form expression of the metric is now obtained in the a - b - c space which is

$$J_{abc} = \frac{2(c-ca-1)T^3 v^2}{a(2a+b-4)(c-1)[c(a+b+ca) - (1-c)^2]} + \frac{\frac{1}{2} a_s^2 T^5 (c(a-1) - 1)}{ab(1+c)[c(a+b+ca) - (1-c)^2]}. \quad (2.91)$$

Like the α - β filter, the first term is the contribution to the metric of the tracker performance for a constant velocity input and the second term represents the cost for a constant acceleration target. Equation (2.91) cannot be reduced to an α - β tracker by substituting c with -1 , since this results in the second term being divided by zero, resulting in an infinite value for $J_{\alpha\beta\gamma}$ which is consistent with our intuition since the integral of a constant value is infinity.

The different metric definitions given in Equation (2.76), (2.88) or (2.91) can now be used in an optimization process to find an optimal set of smoothing parameters α , β and γ for a specific maneuver.

2.4 Design of Optimal Filters

We have derived regions of stability for the α - β and the α - β - γ filters. Selection of the smoothing parameters within the region of stability is a function of the trajectory of the target, the noise in the measurement, steady state error, and the transient response of the filter. To arrive at the optimal set of parameters, a constrained parameter optimization problem is formulated. This is feasible since closed form expressions for various metrics have been derived in Sections 2.3.1 and 2.3.2. Furthermore, closed form solutions for the optimal parameter are derived for certain target trajectories.

Define a figure of demerit which consists of two terms: the first, is a function of the tracker error and the second is a function of the noise ratio. This provides us with the flexibility to include steady state error, transient error etc. and permits us to weight them based on their importance. Thus, the cost function becomes:

$$f \equiv f(e, \rho, \kappa) \tag{2.92}$$

where κ is a weighting factor to penalize the contribution of one term relative to the other. The constraints for the optimization are defined by the stability region of the tracker. As was shown in the Section 2.3.2, there are different metrics to capture the tracking error for specific maneuvers such as straight line or circular target path. The maneuver error can be characterized by transient response and/or the steady state error. The choice of the appropriate cost function depends on the optimization purpose.

2.4.1 Circular Trajectory

The first task attempted in this work was an extension of the optimization of a cost defined by Sklansky [16] for the design of a filters for tracking submarines. To study the variation of the smoothing parameters as a function of the relative importance of the steady state error for a circular maneuver and the noise ratio, a series of optimizations were carried out. The maneuver error e in Equation (2.92) is obtained for a circular target path in steady state conditions. Substituting Equation (2.77) and (2.69) in the cost function f , Equation (2.92) leads to the cost function

$$J = e^2 + \kappa \rho^2 \overline{x_o^2}. \quad (2.93)$$

Assuming that the smallest turn radius of a submarine with $v = 3$ knots speed is 45 m, leads to the angular velocity of $\omega = v/R = 0.034$ rad/sec. Assume that the variance of the measurement noise is 200 m , and weight the noise ratio by κ in Equation (2.92). The cost function (Equation 2.93) is illustrated in Figure (2.12), where $\kappa = \frac{1}{100}$ and the sampling rate is assumed to $T = 3s$. The contour plot (Figure (2.13)) illustrates the location of the minima. To illustrate the effect of changing the weighting parameter, a series of optimizations are carried out and the resulting set of α - β parameters are plotted. Figure (2.14) illustrates that the optimal set of parameters monotonically decrease with an increase of κ . It can be seen that for large values of the weighting factor, the smoothing parameters asymptotically tend to $\alpha = 0.0032$ and $\beta = 0$, which can be shown by setting the maneuver error in the cost function to zero and solving for the optimal α - β . The shape of the curve does not change for circular maneuvers of different radii, since the increase in the radius of the trajectory is equivalent to increasing the weighting factor. The optimal smoothing parameters are displayed in the α - β space in Figure (2.15) where the cost function is a weighted combination of the noise-ratio and the steady-state maneuver error when the target follows a circular path. Figure (2.15) shows various values of the weighting factor κ along the line of optimal solutions. It can be seen that the optimal solutions lie in the underdamped region only.

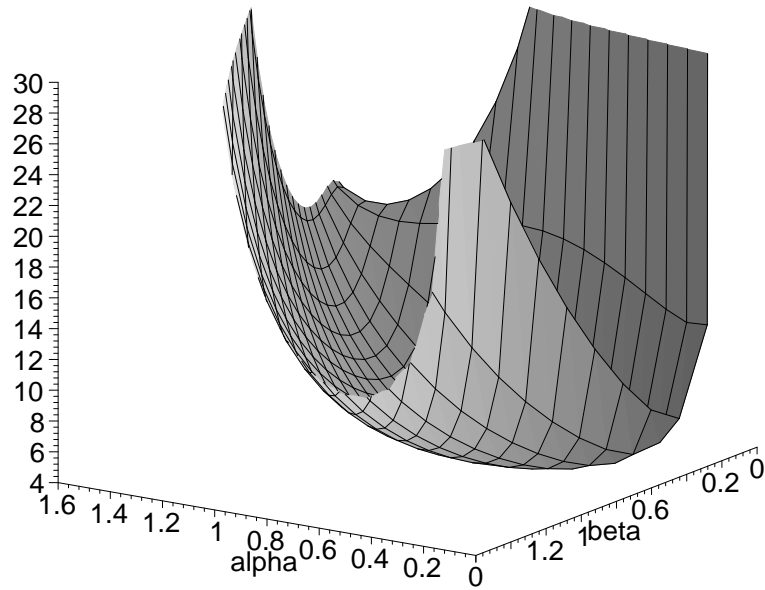


Figure 2.12: Cost Function of a specific submarine motion

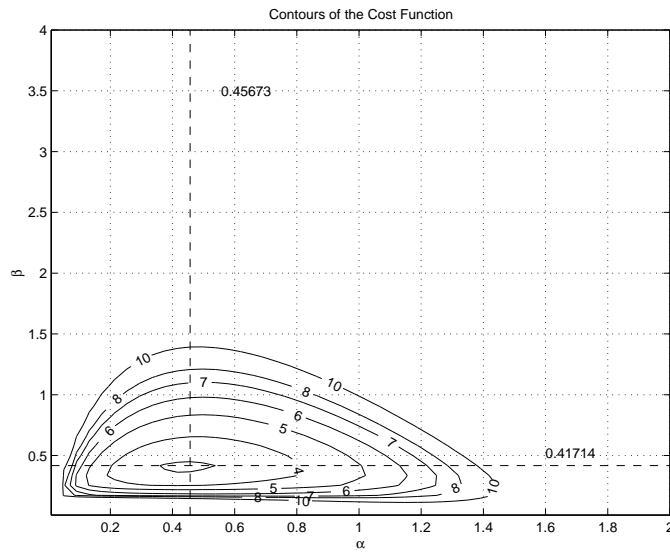


Figure 2.13: Contours of the Cost Function including the Minimum Value

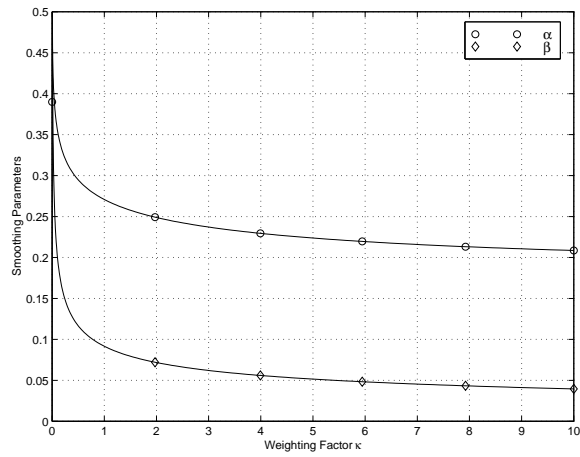


Figure 2.14: Optimal Solution of α and β for tracking a circular path

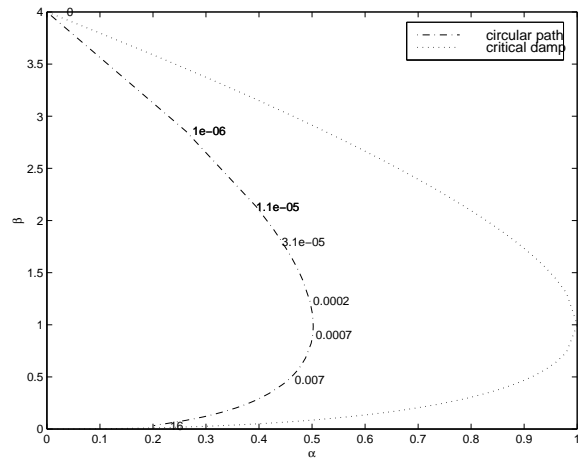


Figure 2.15: Optimal Solution of α and β for tracking a circular path

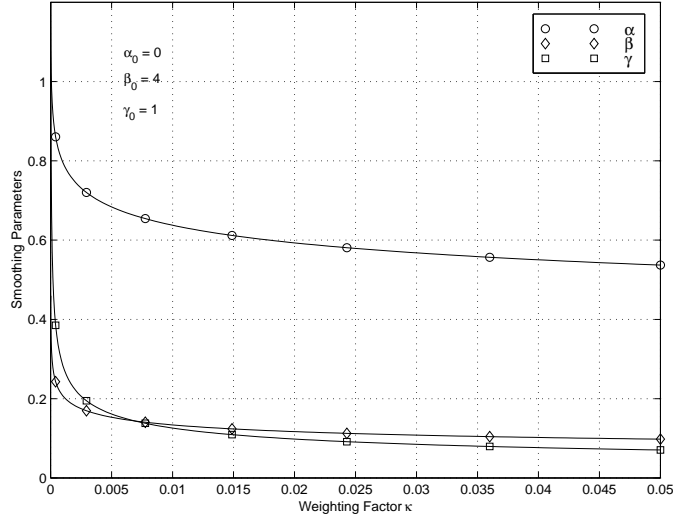


Figure 2.16: Optimal Solution of α , β and γ for tracking a circular path

Extending the same procedure to an α - β - γ tracker, leads to similar results for the optimal smoothing parameter. The change of the parameters versus the weighting factor is shown in Figure (2.16).

2.4.2 Straight Line Maneuver

In practice the tracker rarely reaches steady state because the target path is continuously changing. Including the transient response of the tracker to the cost function leads to a more realistic metric. Section (2.3.2) contains the development of the transient error cost function for tracking a straight path. A closed form solution is given for an α - β tracker (Equation (2.88)), and an α - β - γ tracker (Equation (2.91)). Depending on the target path the tracker can exhibit a steady state error. We can now modify Equation (2.93) by including the metric which measures the transient performance, as follows:

$$f = J_{\alpha\beta(\gamma)} + \kappa_{SS} e_{\alpha\beta(\gamma)}^2(t \rightarrow \infty) + \kappa \rho^2 \overline{x_o^2}, \quad (2.94)$$

where $J_{\alpha\beta(\gamma)}$ is the transient error metric and $e_{\alpha\beta(\gamma)}^2$ the steady state error of the α - β tracker or the α - β - γ tracker. The weighting factor κ_{SS} adjusts the influence

for the steady state error. The transient error cost function in Equation (2.88) and (2.91) are sorted as a function of velocity and acceleration. The effect of the velocity and acceleration on the optimal set of the smoothing parameter of an α - β tracker is shown in Figure (2.17). Three different sets of cost functions are used in this figure. The three curves in the graph correspond to objective functions which weighs the noise ratio to the tracking errors for targets moving with constant velocity, constant acceleration and the steady state tracking error for a constant acceleration input.

The solid line in Figure (2.17) exhibits variation of the optimal parameters as a function of the weighting parameter for targets with constant velocity. A closed form expression of this optimal curve can be derived as the following algorithm shows below. The objective function (Equation (2.94)) for the proposed case reduces to:

$$f = J_{\alpha\beta,v} + \overline{\kappa x_o^2} \rho^2 . \quad (2.95)$$

The optimal solution is obtained by searching for the parameter where the gradient of f vanishes. In this two parameter case the following equations need to be satisfied:

$$\frac{df}{d\alpha} = \frac{dJ_{\alpha\beta,v}}{d\alpha} + \overline{\kappa x_o^2} \frac{d(\rho^2)}{d\alpha} = 0 \quad (2.96)$$

$$\frac{df}{d\beta} = \frac{dJ_{\alpha\beta,v}}{d\beta} + \overline{\kappa x_o^2} \frac{d(\rho^2)}{d\beta} = 0$$

Which can obviously be rewritten in the matrix form as:

$$\begin{bmatrix} \frac{dJ_{\alpha\beta,v}}{d\alpha} & \frac{d(\rho^2)}{d\alpha} \\ \frac{dJ_{\alpha\beta,v}}{d\beta} & \frac{d(\rho^2)}{d\beta} \end{bmatrix} \begin{bmatrix} 1 \\ \overline{\kappa x_o^2} \end{bmatrix} = 0 \quad (2.97)$$

Equation 2.97 can be satisfied only if the determinant of the Jacobian matrix is zero, since the vector $[1 \quad \overline{\kappa x_o^2}]'$ never vanishes. Equating the determinant to zero, we can solve for β resulting in the equation:

$$\beta = \frac{\alpha^2}{2 - \alpha} , \quad (2.98)$$

which matches the optimal solution proposed by Benedict and Bordner [2]. In addition to Equation (2.98), the optimal set of the parameters α and β requires satisfaction of

one of Equation (2.96), which for instance, is

$$0 = [2\beta(4\beta\alpha + 4\alpha^2 - 4\beta + \beta^2)]\kappa + 2v^2T^3(-\alpha^2 + 4\alpha - 4 + \beta) . \quad (2.99)$$

This equation determines the location of the optimal parameters on the curve given by Equation (2.98) as a function of the weighting factor κ . In Figure (2.17) some values of the weighting factor are shown. Very small penalty on the noise ratio optimizes the settling time of the tracker, which is, of course, the shortest if the two poles lie at $z = 0$. This leads to an infinitesimally quick tracker response and is given at the smoothing parameter $\alpha = \beta = 1$. Increasing the penalty of the noise ratio, moves the parameter α and β towards the critically damped curve.

The optimal curve for constant acceleration is obtained by setting the initial velocity and the steady state weight to zero. Optimizing with respect to the transient response of accelerating targets results in higher values for β as shown by the dash-dot line in Figure (2.17). The equation describing this curve is obtained by applying the same algorithm as for the case of constant velocity maneuver. It can be shown that the dash-dot line in Figure (2.17) is described by:

$$\beta_{1,2} = 3 - \frac{5}{2}\alpha \pm \frac{1}{2}\sqrt{(36 - 60\alpha + \alpha^2)} \quad (2.100)$$

and the second equation that needs to be satisfied is:

$$0 = [4\beta^3(4\alpha^2 - 4\beta + \beta^2 + 4\beta\alpha)]\kappa + a_s^2(\alpha^2 - \beta)(2\alpha - 4 + \beta)^2 . \quad (2.101)$$

The dashed line in Figure (2.17) which corresponds to a cost function which includes the steady state error reveals that minimizing the steady state error of the tracker by increasing the weight κ_{SS} , forces β to maximize. The two equations describing the optimal solutions for the case where the objective function has a strong penalty on the steady state error are derived as follows:

$$\beta_{1,2} = 2 - 2\alpha \pm 2\sqrt{(1 - 2\alpha)} \quad (2.102)$$

$$0 = [(4\alpha^2 - 4\beta + \beta^2 + 4\beta\alpha)]2\kappa \quad (2.103)$$

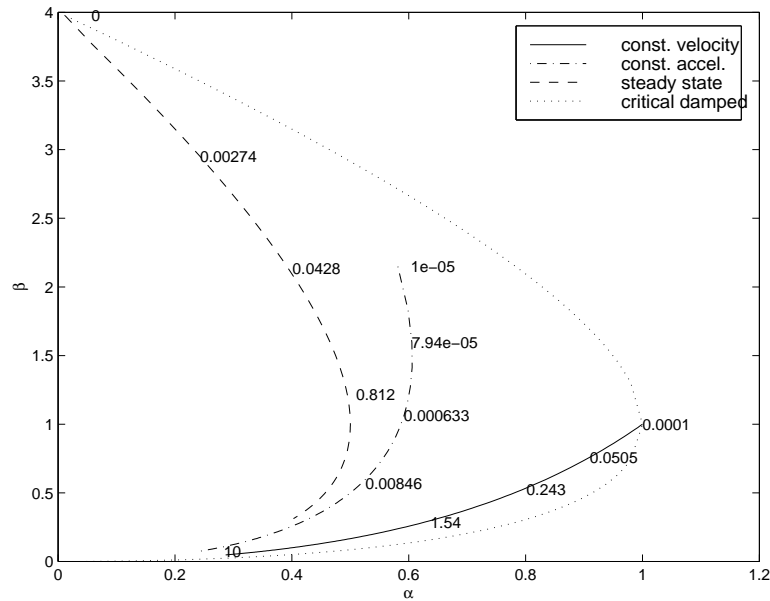


Figure 2.17: Optimal Solution of α and β for a straight line maneuver

Observing Figure (2.17) reveals that a better noise smoothing is obtained with smaller values of β and α , conversely, faster response is obtained with higher values of β . The special case of constant velocity has its fastest response at $\alpha = \beta = 1$. The three discussed objective functions containing constant velocity, constant acceleration and steady state are illustrated in the subplots of Figure (2.18). A change in the variance can easily translated into a change of the weighting factor since those parameters appear as constants in the same metric.

The equation of the transient error cost functions of an α - β - γ tracker is also divided in two fractions depending only on the velocity and the acceleration. Two sets of the optimal smoothing parameter are shown in Figure (2.19), which have been derived by the constrained parameter optimization algorithm. The upper plot uses constant acceleration input without initial velocity, whereas the lower plot shows the optimal solution for a combined cost function with an acceleration of $0.03m/s^2$ and a velocity of 3 knots. As also observed in Figure (2.17), higher parameter cause a faster system response. Conversely, for better noise smoothing, the optimal parameters

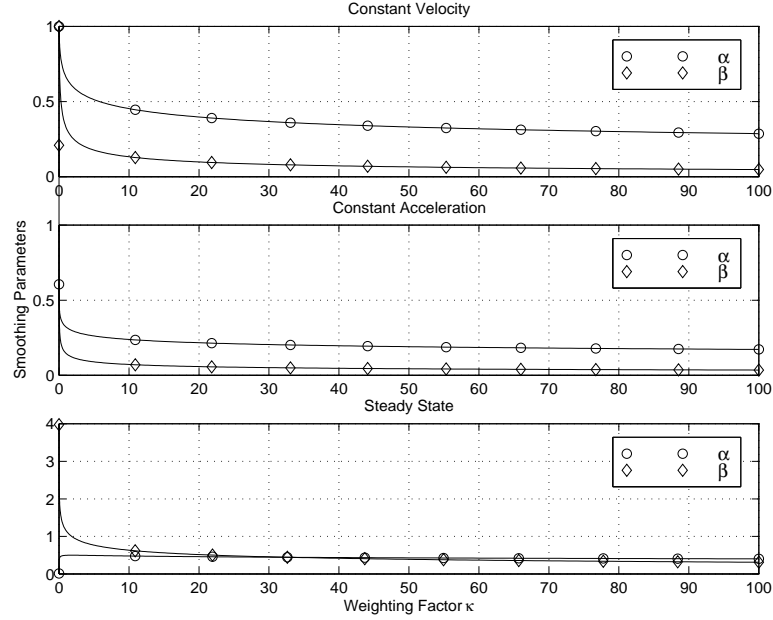


Figure 2.18: Optimal Solution of α and β for a straight line maneuver with respect to different objectives

are decreasing. A closed form expression of the optimal set of parameter can be derived for a straight line target trajectory where the target is accelerating. Define a matrix of the gradients of the terms of the cost function with respect to a , b and c , we have

$$\begin{vmatrix} \frac{dJ_{abc,v}}{da} & \frac{dJ_{abc,a}}{da} & \frac{d(\rho^2)}{da} \\ \frac{dJ_{abc,v}}{db} & \frac{dJ_{abc,a}}{db} & \frac{d(\rho^2)}{db} \\ \frac{dJ_{abc,v}}{dc} & \frac{dJ_{abc,a}}{dc} & \frac{d(\rho^2)}{dc} \end{vmatrix}, \quad (2.104)$$

where the previously introduced transformation into the a - b - c space is carried out (Equation (2.54)). Equating the determinant of Equation 2.104 to zero and solving for the parameter b , leads to the optimal solution in the a - b - c space:

$$b_1 = \frac{(1+c)^2(a-1) + a^2c}{c(2-a)} \quad (2.105)$$

$$b_2 = \frac{(1+c)(c+1-1)}{c} \quad (2.106)$$

It can be seen that b_1 reduces to the optimal solution of the α - β filter (Equation (2.98))

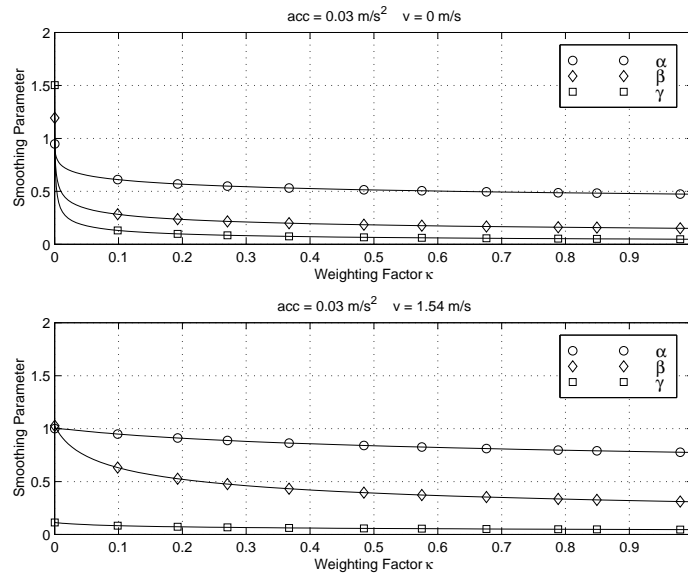


Figure 2.19: Optimal Solution of α , β and γ for a straight line maneuver with two different Cost Functions

if $c = -1$, whereas b_2 vanishes for this case. The two additional conditions to be satisfied cannot easily be simplified.

Since the sensor noise of a sonobuoy is fixed to some number, the optimal solutions of the α , β and γ parameters are derived for a constant initial velocity and changing acceleration of a target moving along a straight line. The objective function for this case contains the transient maneuver error cost (Equations (2.88) or (2.91)) and the noise-ratio (Equations (2.68) or (2.69)). We arrive at the optimal solutions shown in Figure (2.20) by using a numerical constrained optimization. For clarity the α - β filter is first explored. The dash-dot line in this figure represents the optimal parameters for zero acceleration given in Equation (2.98). Results are shown for three different initial velocities, 1, 3 and 20 knots, where the circle emphasizes zero acceleration, which lies on the dash-dot line. Increasing the initial velocity results in higher values for α and β . Since the target is capable to accelerate and decelerate, optimal solutions are derived for positive (solid line) and negative acceleration (dashed line). Whereas the optimal parameters for a target with positive acceleration result in a

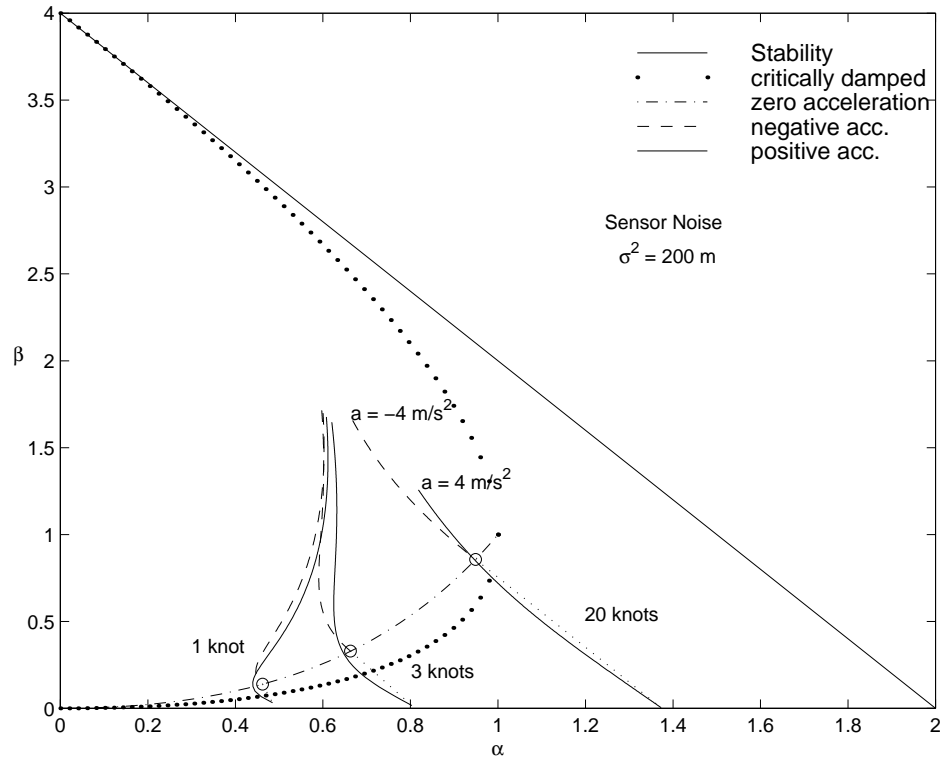


Figure 2.20: Optimal α and β parameters at constant sensor noise and changing acceleration

jump to the overdamped region, the decelerating target starts without a discontinuity. Each case in Figure (2.20) is derived for a figure of acceleration between $-4 \frac{m}{s^2}$ and $4 \frac{m}{s^2}$. Negative initial velocity results in the same optimal solutions, whereas the positive and negative acceleration curves are interchanged.

Similar results are obtained for the α - β - γ filter. Figure (2.21) shows the change of the smoothing parameters versus the acceleration. It can be observed that for zero acceleration the filter reduces to the α - β filter since the α - β filter is able to track a non-accelerating target. Increasing the acceleration results in increasing γ and since the rate of changing the velocity increases, β also increases.

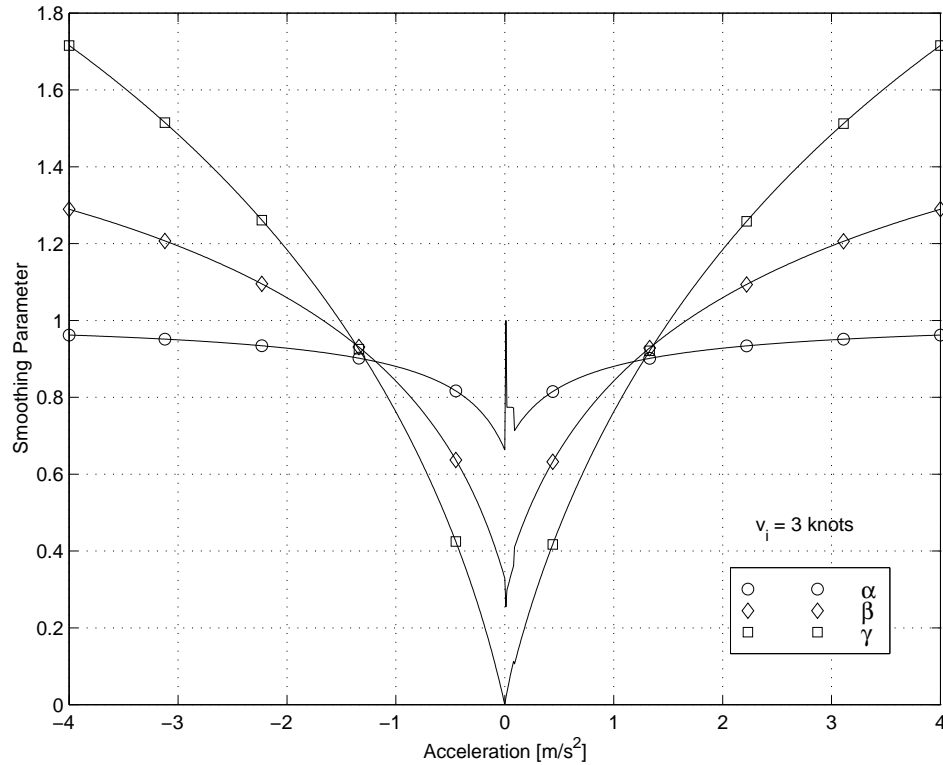


Figure 2.21: Optimal α , β and γ parameters at constant sensor noise and changing acceleration

2.5 Remarks

This chapter has focused on the design of α - β - γ filters. The issue of determination of stability volume is first addressed. A simple technique to simplify the procedure to determine the stability bounds on the α , β and γ filters is proposed. This includes parameterizing the characteristic equation of the α - β - γ filter via a nonlinear transformation to what is referred to as the a , b , c space. In this space, the characteristic equation appears to be the product of the characteristic equation of an α - β filter and a first order pole which is only a function of the parameter c . One can now easily determine the bounds on the parameters with knowledge of the bounds on the parameters of an α - β filter. Therefore the stability volume in the a - b - c is a prism which can be transformed into the α - β - γ space. To quantify the performance

of α - β - γ filters, various metrics are defined such as the noise ratio which is a figure of demerit to represent the noise filtering capability of the tracker. A closed form solution to the noise ratio is arrived at in the a - b - c space which reduces to the noise ratio for the α - β filter when γ is equated to zero. The resulting solution is shown to be different from that derived in the literature. Numerical simulations are carried out to evaluate the veracity of the derived solution. Closed form equations to characterize the transient performance of the α - β - γ tracker for straight line and circular maneuvers are also derived. These are subsequently used in conjunction with the noise ratio to determine the optimum set of the tracker parameters based on a cost function which is a weighted combination of the noise ratio, the transient response metric and the steady state error. Variation of the tracker parameters for different weights of the cost function are studied to provide the designer with information for the optimal selection of the α , β and γ parameters.

Chapter 3

Fuzzy Logic Approach of Target Tracking

The fuzzy logic theory overcomes the difficulty of defining an explicit relationship between the current system states and the actions required to achieve a certain system performance by using linguistic rules. These rules could be determined with an experts knowledge or simply by stating the required cause effect relation in common language, which has the form of **if-then** statements. Furthermore, the rules could be learned during some kind of training algorithm [17].

This chapter uses a simple fuzzy controller to introduce the fuzzy elements interactions before exploring a more sophisticated representation of the fuzzy algorithm. The rule base of the proposed fuzzy filter is determined by an analogy to the optimal solution of the rest-to-rest maneuver of a second order system [6], where we propose a relationship between the damping and stiffness determined by the rest-to-rest maneuver to the smoothing parameter α and β .

3.1 Principles of the Fuzzy Algorithm

3.1.1 Fuzzy Logic on an Example

The interaction between the fuzzy elements is first shown on an single-input-single-output example, followed by description of a general fuzzy logic algorithm. Suppose one is interested in keeping a constant flow out of a valve, one would open the valve if the flow slows down and vice versa. The appropriate rules may formulated as:

- r_{14} : IF flow is *zero* THEN valve opening is *large*
- r_{23} : IF flow is *small* THEN valve opening is *medium*
- r_{32} : IF flow is *medium* THEN valve opening is *small*
- r_{41} : IF flow is *large* THEN valve opening is *zero*

A human operator might use these rules since he is capable of distinguishing between the linguistic terms *small*, *medium* etc. Since the flow rate and the valve position are quantified using crisp values, these rules cannot be applied directly to the system. However, what one human operator determines the flow to be *small*, in contrast, others state that the flow exhibits *medium* behavior. The resulting valve opening depends on characterizing the current flow. The above two operators might negotiate between the valve positions *medium* and *small*, which again are uncertainties of the operator. Certain human operators also have different belief about the linguistic terms *small*, *medium*, etc. These vague informations are modelled in the fuzzy theory by assigning the input variables (here the flow) to memberships of linguistic terms as opposed to declaring the input to a fixed value in the input space on which a mathematical model calculates the required control. The membership functions, μ_A , are defined over the entire universe of discourse such that every input has a specific membership of a fuzzy set like *large*, *medium*, etc. The membership functions of the above example are defined as triangular shapes over the universe of discourse

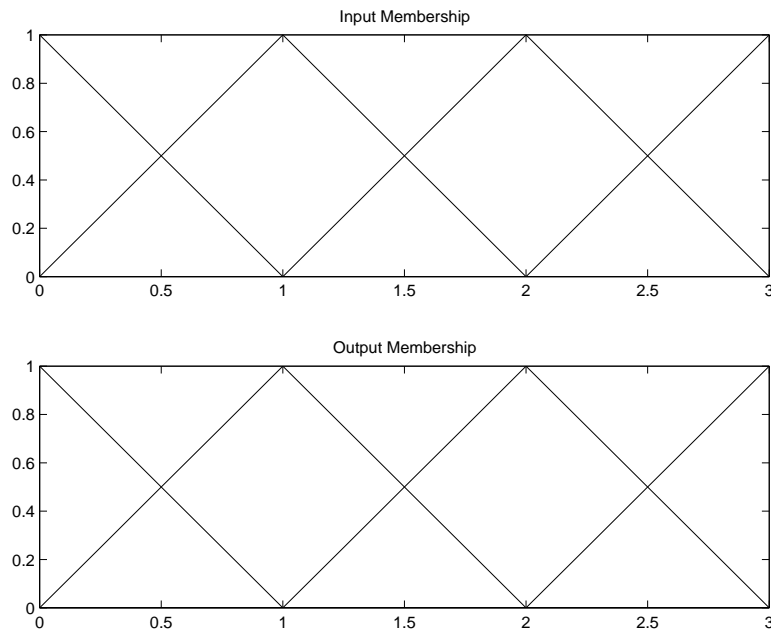


Figure 3.1: Input and Output Membership function of the Example

$[0, 3]$ as shown in Figure (3.1). The shape of the membership functions is arbitrary and changes are reflected in different behavior of the fuzzy system. The membership functions and the applied fuzzy algorithm may be represented in the continuous or discrete space depending on the computational cost. Throughout this work only the discrete representation is considered, whereas the continuous representation could be applied in similar formulations.

For reasons of generality and brevity it is convenient to introduce a label for each rule. Since each input could possibly be related to every output, a matrix of rules with the size of [number of input membership functions \times number of output membership functions] is constructed. Thus, the four given example rules are labeled as shown above. The first rule, r_{14} , for instance, shows that the first input membership function is combined with the fourth output membership function. This labeling scheme is also convenient with the use of confidences of each rule. A confidence of a rule expresses the belief in this rule, and is a gain between zero and one. If more than one rule is applied to one input membership function, the confidences should be

normalized such that the sum equals one. This is not necessarily required, but its use permits scaling the output. The general form of the fuzzy rule can now be written as:

$$r_{ij} : \text{IF } \mathbf{x} \text{ is } \mathbf{A}^i \text{ THEN } \mathbf{y} \text{ is } \mathbf{B}^j (c_{ij}), \quad (3.1)$$

where \mathbf{x} and \mathbf{y} are the input and output variables of dimension m and n . Each input and output set contains p_m and q_n membership functions. For a single input–single output fuzzy system the rule and confidence matrixes are of size $[p_1 \times q_1]$, whereas for multi–input–output fuzzy systems the rule and confidence matrixes are of size $[p_1 \cdot p_2 \dots p_m \times q_1 \cdot q_2 \dots q_n]$ thus avoiding higher order matrixes.

Since the input and often the output are crisp values, processing the crisp values using linguistic rule bases requires the input to be fuzzyfied and the output to be defuzzyfied. This procedure assigns a certain numerical value referred to as degrees of membership functions (μ_{A^i}) and vice versa for degrees on output memberships (μ_{B^i}) a crisp output value. Since both schemes differ from each other, they are discussed separately. The input fuzzyfication assigns degrees of active membership functions to a crisp value. As shown as in Figure (3.2), Part A the example input, x , has non-zero membership in the input set *zero* and *small* with the degree of $\mu_{A^1}(\hat{x})$ and $\mu_{A^2}(\hat{x})$. To explain the output defuzzyfication we need to construct first the fuzzy output according to the fuzzy inference. Since this example input activates the membership functions of *zero* ($i = 1$) and *small* ($i = 2$), which means that all rules r_{1j} and r_{2j} , where $j = 1 \dots q$, need to be evaluated. According to the example rules there are two rules r_{14} and r_{23} . The first rule suggests: If the input is *zero* then the output should be *large*, which means that the output membership function *large* is activated. Since the input has a fractional membership of the input set *zero*, the corresponding membership of the output set *large* needs also be adjusted. For purpose of demonstration the simple truncation algorithm is applied, where the output set *large* is truncated to the membership of the input set $\mu_{A^1}(\hat{x})$. Similarly the second fuzzy influence can be constructed. Part C in Figure (3.2) visualizes this

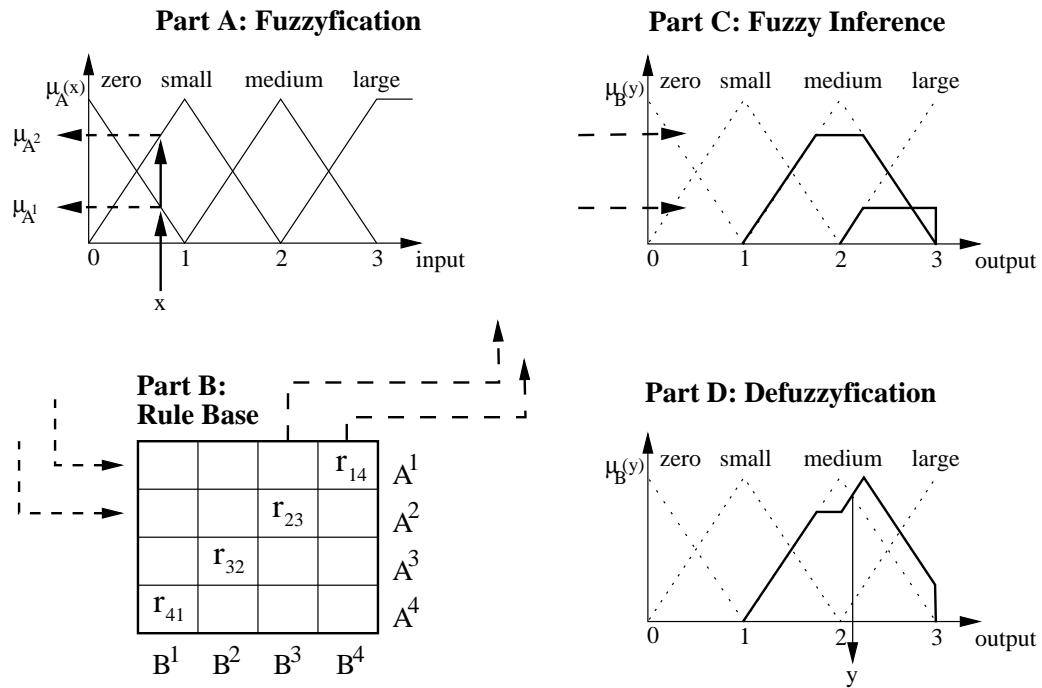


Figure 3.2: Fuzzy Example Algorithm

algorithm. Note, that a multivariable input space is mapped in a hyper membership set, where the inputs are combined with respect to each rule. Furthermore, since rule r_{14} and rule r_{23} are firing, a compromise between these two suggested outputs needs to be achieved. During this step, the *and* operation is carried out which is either the algebraic (Addition) or logic (Maximum) operator. Combining the two fuzzy influences with the addition operator yields the fuzzy output shown in Figure (3.2), Part D. This linguistic (fuzzy) output is quite certain that the output should be somewhat *medium* and also somewhat *large*, let us say somewhat between *medium* and *large*. The human operator might be able to interpret this linguistic terms as opposed to most technical applications requiring a crisp value. This final step of defuzzification loses all the important information about the fuzzy output by reducing it to one crisp value, the defuzzified output.

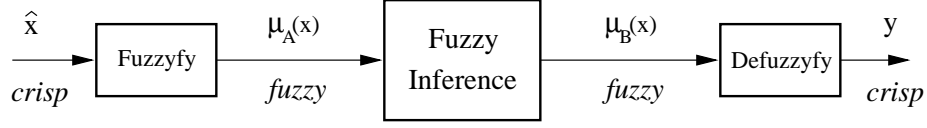


Figure 3.3: Fuzzy Flow Chart

3.1.2 A General Fuzzy Logic Approach

The above example of a fuzzy algorithm contains a lot of motivation for use of fuzzy systems instead of equations describing the same process. However, implementing the fuzzy algorithm in a lot more complex requiring a more sophisticated process. Figure (3.3) recalls the steps taken by the fuzzy decision process.

The fuzzyfication maps the crisp value $\hat{\mathbf{x}}$ into the fuzzy set $\mu_{\mathbf{A}}(\mathbf{x})$. A common approach is the singleton fuzzyfication where a binary membership is used. Equation (3.2) represents the singleton method.

$$\mu_{\mathbf{A}}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} = \hat{x} \text{ (continuous) or nearest } \mathbf{x}^{\mathbf{d}} \text{ (discrete)} \\ 0 & \text{otherwise} \end{cases} \quad (3.2)$$

The fuzzy influence combines the input sets with the output sets based on the fuzzy rules, which results in the fuzzy set with the membership $\mu_{r_{ij}}(\mathbf{x}, \mathbf{y})$ for each rule and the fuzzy rule base $\mu_{\mathbf{R}}(\mathbf{x}, \mathbf{y})$ combining the separate rules. The individual membership functions are calculated as:

$$\mu_{r_{ij}}(\mathbf{x}, y) = \mu_{\mathbf{A}^i}(\mathbf{x}) \cap \mu_{\mathbf{B}^j}(\mathbf{y}) \cap c_{ij} , \quad (3.3)$$

where \cap is the *AND*-operator. If \mathbf{x} is multivariate, each input set needs to be related to the other according to:

$$\mu_{\mathbf{A}^i} = \bigcap_{k=1}^m \mu_{\mathbf{A}_k^i}(\mathbf{x}_k) \quad (3.4)$$

Combining the individual sets with the *OR*-operator (\cup) yields the input-output-relation:

$$\mu_{\mathbf{R}}(\mathbf{x}, y) = \bigcup_{i,j} \mu_{r_{ij}}(\mathbf{x}, y) , \quad (3.5)$$

	<i>Algebraic</i>	<i>Mamdani</i>
AND	Product	Min
OR	Summation	Max

Table 3.1: Common Fuzzy Intersections

where the indices i and j are within the range 1 to $p_1 \cdot p_2 \dots p_m$ and 1 to $q_1 \cdot q_2 \dots q_n$. The fuzzy intersection *AND* may be a algebraic product operator or a the logic minimum operator. Whereas, the *OR* intersection can be derived using the addition or maximum operator. These are the most popular fuzzy intersection, where the logic operators in the literature referred to as Mamdani intersection. Table 3.1 summarizes the most popular fuzzy intersections.

The fuzzy rule base Equation (3.5) can now be used to generate the fuzzy output distribution for a given input set $\mu_{\mathbf{A}}(\hat{x})$ obtained by the fuzzyfication process. This procedure is known as the compositional rule of inference, which may be written as:

$$\mu_{\mathbf{B}}(y) = \mu_{\mathbf{A}}(\hat{\mathbf{x}}) \circ \mu_{\mathbf{R}}(\mathbf{x}, \mathbf{y}) , \quad (3.6)$$

where “ \circ ” is the composition operator and $\mu_{\mathbf{B}}(y)$ is the degree of output membership functions. The composition procedure is similar to the example in Figure (3.2) where the suggested output by each rule is truncated and than added to get the final output distribution. Therefore, the composition operator performs a comparison of the two membership functions and summarizes over all possible values of $\hat{\mathbf{x}}$. The composition operator for fuzzy sets defined on a continuous domain in a more general expression may be written as:

$$\mu_{\mathbf{B}}(y) = \int_{\mathbf{x}} \mu_{\mathbf{A}}(\hat{\mathbf{x}}) \mu_{\mathbf{R}}(\mathbf{x}, \mathbf{y}) d\mathbf{x} \quad (3.7)$$

and similarly in the discrete domain as:

$$\mu_{\mathbf{B}}(y^d) = \sum_{\mathbf{x}^d} \mu_{\mathbf{A}}(\hat{\mathbf{x}}^d) \mu_{\mathbf{R}}(\mathbf{x}^d, \mathbf{y}^d) \quad (3.8)$$

For a singleton fuzzyfication Equation (3.8) reduces to:

$$\mu_{\mathbf{B}}(y^d) = \mu_{\mathbf{R}}(\mathbf{x}^{dc}, y^d) , \quad (3.9)$$

where \mathbf{x}^{dc} is the discrete value closest to the input $\hat{\mathbf{x}}$.

Figure (3.4) illustrates the rule base given in Equation (3.5) for the fuzzy example introduced above. The universe is discrete, and the *AND* and *OR* operators are represented with the algebraic operations. The relational matrix represents the

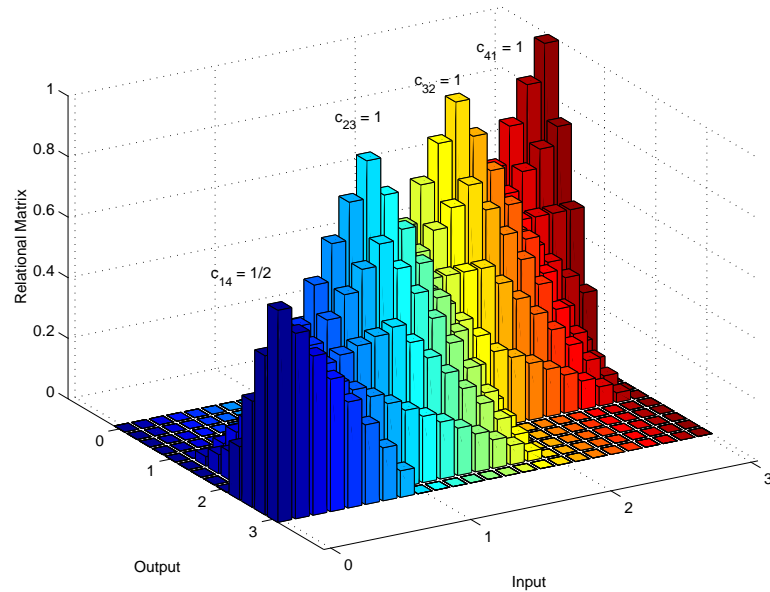


Figure 3.4: Relational Matrix of the above Example

shape of the fuzzy input and output sets and shows the underlying fuzzy algorithm. Observing the relational matrix in Figure (3.4), illustrates the mapping of the input space to the output space with respect to the four example rules (four spikes). The influence of the rule confidences on the relational matrix is also shown in Figure (3.4), where the confidence of the rule r_{14} is reduced to $c_{14} = 0.5$. Thus, the weight of the output according to this rule reduces and the spike of the relational matrix is therefore scaled down. If the input is singleton fuzzyfied, Equation (3.9) can be applied, which means that the fuzzy output is given by the appropriate column of the discrete relational matrix. Figure (3.5) shows such a slice through the relational matrix at the crisp input value of 1.25.

As mentioned before, the fuzzy, imprecise output, $\mu_{\mathbf{B}}(y^d)$ needs to be trans-

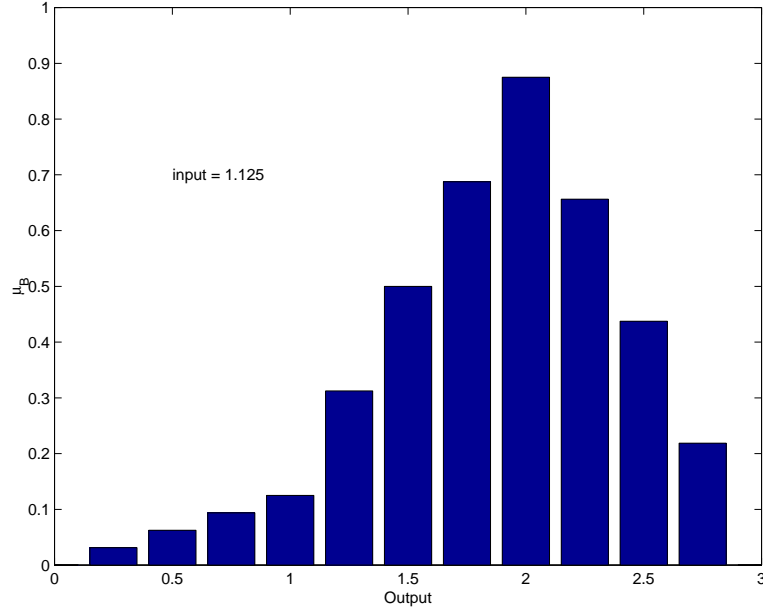


Figure 3.5: Discrete Fuzzy Output Membership

formed to a crisp value using a defuzzification scheme. The most popular defuzzification is the center of gravity defuzzification, which is defined for the continuous and discrete domain as follows:

$$y(\mathbf{x}) = \frac{\int_Y \mu_B(y) y dy}{\int_Y \mu_B(y) dy} \quad y(\mathbf{x}^d) = \frac{\sum_{Y^d} \mu_B(y^d) y^d}{\sum_{Y^d} \mu_B(y^d)}. \quad (3.10)$$

Exploiting the Equation (3.10) and applying algebraic operators, reduces the calculation of the output to [4]:

$$y(\mathbf{x}^d) = \frac{\sum_{\mathbf{X}^d} \mu_A(\mathbf{x}^d) \sum_i \mu_{A^i}(\mathbf{x}^d) w_i}{\sum_{\mathbf{X}^d} \mu_A(\mathbf{x}^d)}, \quad (3.11)$$

where w_i are the weights similar to the joints of an Associative Memory Network (AMN). The relationship between the confidences and the weights can be shown to be:

$$w_i = \sum_j^q c_{ij} y_j^c, \quad (3.12)$$

where y_j^c is the center of the j^{th} output membership function. For a singleton input

fuzzyfication Equation (3.11) reduces to:

$$y^s(\mathbf{x}) = \sum_i^p \mu_{A^i}(\hat{\mathbf{x}})w_i = \sum_i^p \sum_j^q \mu_{A^i}(\hat{\mathbf{x}})c_{ij}y_j^c \quad (3.13)$$

This representation is less computational expensive as opposed to storing the relational matrix and carrying out the matrix operations.

3.2 The Fuzzy Algorithm in Target Tracking

Unlike the fixed gain α - β filter the fuzzy logic based α - β filter changes the smoothing parameter, α and β , as a function of the maneuver error and error rate with the tracking performance comparable to a Kalman filter, but without modeling a target system and a sensor noise model. Furthermore, the computational cost is less than that of the Kalman filter. Moreover, the fuzzy algorithm reaches a decision by considering several different situations at the same time resulting in a more human like decision.

3.2.1 The Fuzzy Logic Inputs

The maneuver error is now defined as the difference between the observed position and the predicted position of the target. Since, the filter estimates the targets future position, the maneuver error at the current time influences the choice of the appropriate smoothing parameter α and β . The maneuver error at the current time may be written as:

$$e(k) = x_o(k) - x_p(k) . \quad (3.14)$$

Besides the information ‘‘How close the prediction is to the true target position’’, the change of the maneuver error is also an important fact since different action must be taken whether the maneuver error is about to increase or decrease. The error rate is defined as follows:

$$\Delta e(k) = e(k) - e(k - 1) . \quad (3.15)$$

For purpose of generality of the target tracker, normalization of the maneuver error and error rate are desired. Chan [5] introduced an effective normalization scheme, where the normalization denominator is variable and depends on the magnitude of the error. The following three situations may occur:

1. The magnitude of the error is less than the difference between the current and previous observed position.

$$E(k) = \frac{x_o(k) - x_p(k)}{x_o(k) - x_o(k-1)} \quad (3.16)$$

2. The magnitude of the error is greater than the difference between the current and previous observed position.

$$E(k) = \frac{x_o(k) - x_p(k)}{|x_o(k) - x_p(k)|} \quad (3.17)$$

3. The predicted position equals the current observed position and the current and previous observations are the same.

$$E(k) = 0 \quad (3.18)$$

Plotting the Equations (3.16) to (3.18), yields the normalized error shown in Figure (3.6), which is a saturation function with changing slope and symmetric about the origin. Since the normalized maneuver error lies within the interval $[-1, 1]$, the maximum absolute maneuver error rate is 2. Similar to the above normalization scheme the error rate is also transformed into the interval $[-1, 1]$ according to the following equations:

$$\Delta E(k) = \begin{cases} \frac{E(k) - E(k-1)}{E(k-1)} & \text{if } |E(k) - E(k-1)| < |E(k-1)| \\ \frac{E(k) - E(k-1)}{|E(k) - E(k-1)|} & \text{if } |E(k) - E(k-1)| > |E(k-1)| \\ 0 & \text{if } E(k) - E(k-1) = E(k-1) = 0 \end{cases} \quad (3.19)$$

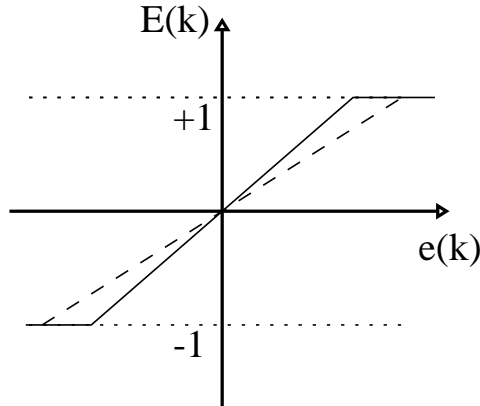


Figure 3.6: Error Normalization Scheme

This normalization scheme might be applied for each dimension separately, where every normalized error and error rate contains the information about the relative position of prediction and observation (sign of the error). In a multivariable dimensional space one could calculate the magnitude of the error and the error rate, and use those to determine the appropriate α and β . This is proposed by Chan [5] and the corresponding error magnitude are defined as:

$$\tilde{E}(k)^2 = \frac{\sum_i^n E_i(k)^2}{n} \quad (3.20)$$

$$\Delta\tilde{E}(k)^2 = \frac{\sum_i^n \Delta E_i(k)^2}{n}, \quad (3.21)$$

where the summations are carried out over the number of dimensions. These error and error rate magnitude lie within the interval $[0, 1]$. On the contrary, the relative position of the prediction to the observation and the direction of changing the error are important to choose an appropriate control. Figure (3.7) illustrates the four situations. The predicted position “A” in Figure (3.7) exhibits a positive maneuver error and a positive error rate increases the error of the next time step, whereas a negative error rate decreases the next error. On the other hand, a positive error rate for the predicted position “B” in Figure (3.7) decreases the maneuver error of the next time step. Likewise, a negative error rate induces the opposite effect.

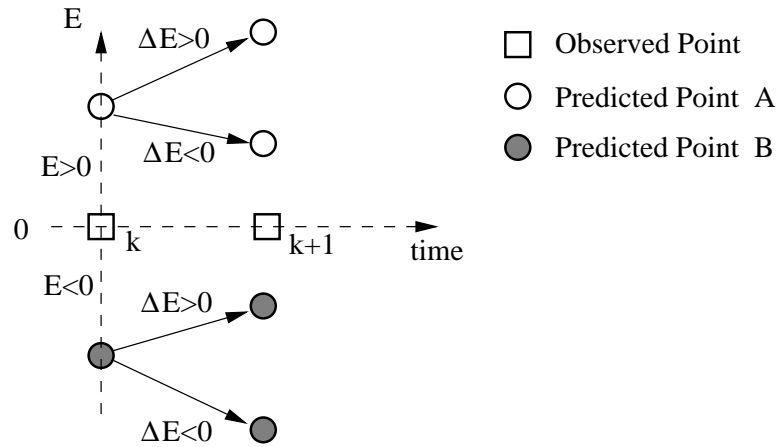


Figure 3.7: Development of Maneuver Error

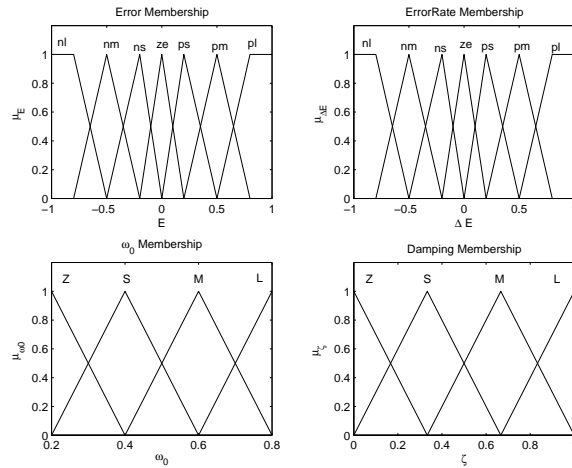


Figure 3.8: Membership Functions

Hence, a normalized representation of the error and error rate is given in Equations (3.16)-(3.18) and Equation (3.19), a general set of input membership functions can be defined. With regard to the normalization scheme, the input membership functions are generated as shown as in Figure (3.8). The fact that large errors are mirrored into the normalized bounds -1 and $+1$ leads to a larger region of support (ROS) of the negative-large (nl) and positive-large (pl) membership functions. Furthermore, the other membership functions shrink around zero together.

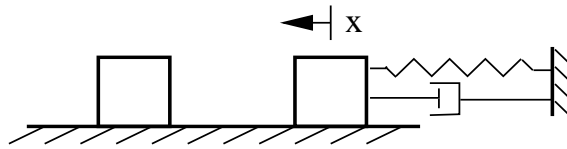


Figure 3.9: Rest-to-Rest Maneuver

3.2.2 The Fuzzy Logic Rule Base

The fuzzy logic rule base is built by combining the input membership sets via the linguistic rules with the output membership sets. The linguistic rules are of the form of Equation (3.1), where every possible input combination should be covered. In case of the seven error and seven error rate input sets shown in Figure (3.8), a minimum of 49 rules is required. To increase the modelling capability a input combination of the maneuver error and error rate may call different output membership functions at the same time with the use of continuous confidences [4]. Determining the rules is difficult and might be done with the help of experts knowledge about the system [5]. In Chans work this expert's knowledge is used to determine the specific amount of the smoothing parameter α and β . The proposed fuzzy filter exploits an analogy from system control theory, the rest-to-rest maneuver of a second order system [6], to define appropriate rules.

The rest-to-rest maneuver requires a mass at rest to move as quickly as possible to another desired position of rest. The control law is modelled in the form of a non-linear spring-damper system shown in Figure (3.9). The coordinate system is fixed at the desired final position of rest, so that the first rest is expressed as an initial condition, and the controller of the rest-to-rest maneuver is designed to exhibit Lyapunov stability. The results of this work [6] are used to determine the 49 rules as explained in the following section.

The transfer function of a spring-mass-dashpot system shown in Figure (3.9)

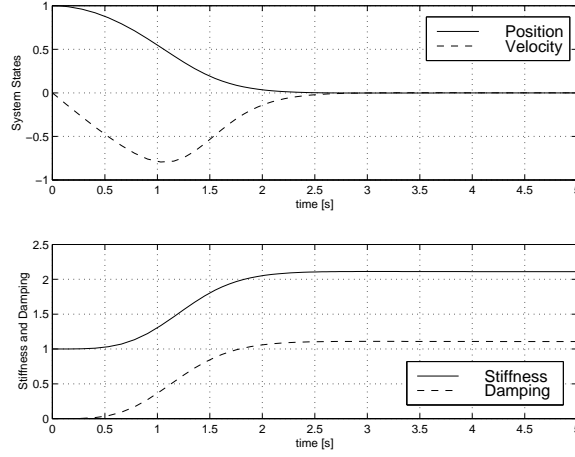


Figure 3.10: States and Control of the Rest-to-Rest Maneuver

can be represented in continuous time domain as:

$$G(s) = \frac{x(s)}{u(s)} = \frac{m}{s^2 + 2\zeta\omega_0 s + \omega_0^2}, \quad (3.22)$$

where ω_0 and ζ are the undamped eigenfrequency and the damping ratio. The transfer function of Equation (3.22) is later used to define a relation between the smoothing parameter α and β , and the undamped eigenfrequency and the damping ratio. The non-linear control law is defined in [6] as follows:

$$u = -(k_1 + k_2(x - x(0))^2)x - (c_1 + c_2(x - x(0))^2)\dot{x}, \quad (3.23)$$

where the constants k_1 , k_2 , c_1 and c_2 are determined to satisfy the above mentioned performance. The solution of the original problem, the rest-to-rest maneuver, with an initial position error and zero initial velocity is shown in Figure (3.10). We can infer from Figure (3.10) that the stiffness increases from a non-zero value and the damping starts at zero. The closer the mass to the final position, the higher the stiffness and damping. This knowledge may now be used to determine the rules according to which the control action is taken. A convenient structure of writing these rules is shown in Table (3.2). Within this table all possible input combinations are captured, since the normalized error, Equations (3.16)–(3.18), is entered in the first row and the normalized error rate, Equation (3.19), is entered in the first column. The original

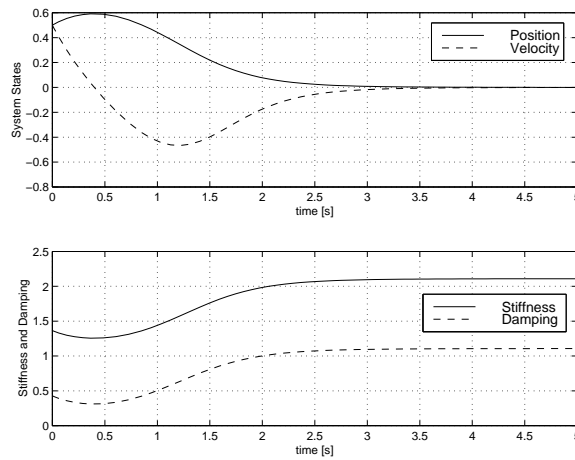


Figure 3.11: States and Control of the Rest-to-Rest Maneuver for non-zero initial position and velocity

rest-to-rest maneuver, explained above, is represented in Table 3.2 at the row ΔE equals *zero*, where *large* errors require *small* stiffness and damping and *zero* error requires *large* stiffness and damping. For similar reasons, the column with *zero* error rate requires *large* stiffness and damping, trying to keep the error and error rate small. As shown as in Figure (3.7) positive error and negative error rate exhibits the same situation as negative error and positive error rate. Thus, the Table 3.2 is mirrored symmetrical about the two main-diagonals, which is emphasized by the light and dark gray table entries. The gray table entries are determined by the use of the modified rest-to-rest maneuver, where the initial rest position is changed to different values of initial position and velocity. For instance, the rule of *positive medium* normalized error and *positive medium* normalized error rate are obtained by setting the initial velocity and the initial position to 0.5 as shown in Figure (3.11). The rule table entry in the upper right and lower left rectangulars are obtained with the same method. Underlying to this method is the following rule observed by Figure (3.7):

- If the error decreases by itself (positive error and negative error rate and vice versa), do not apply a large control (small stiffness and damping), only if the error is small and the error rate is large increase the damping of the system to

Entries $\omega_0-\zeta$		E						
		nl	nm	ns	ze	ps	pm	pl
ΔE	nl	S-L	M-L	L-L	L-L	S-L	Z-M	Z-S
	nm	S-M	M-M	L-L	L-L	M-M	Z-S	S-Z
	ns	Z-S	S-M	M-L	L-L	M-M	S-Z	M-Z
	ze	Z-Z	S-S	M-M	L-L	M-M	S-S	Z-Z
	ps	M-Z	S-Z	M-M	L-L	M-L	S-M	Z-S
	pm	S-Z	Z-S	M-M	L-L	L-L	M-M	S-M
	pl	Z-S	Z-M	S-L	L-L	L-L	M-L	S-L

Table 3.2: Fuzzy Rule Base (Compare the linguistic Terms with Figure (3.8))

prevent overshoot!

- If the error increases by itself, increase the damping moderate to counteract this situation

With the knowledge of the fuzzy rule base, Table 3.2, and the membership sets defined in Figure (3.8) the fuzzy relational matrix can now be derived according to Equation (3.5). Using the singleton input-fuzzyfication and the Center of Gravity Method to defuzzyfy the controllers output, the control surfaces shown in Figure (3.12) can be calculated.

Since the fuzzy controller is defined in terms of *stiffness* and *damping*, it is required to express those terms as smoothing parameter α and β . For this purpose we recall the transfer function of the α - β filter derived in Chapter 2. The discrete transfer function, Equation (2.7), shall be mapped into another discrete domain where the terms stiffness and damping are defined. Thus, the area of the unit circle of the z -domain is transformed into the left half plane, often referred to as the w -domain. This mapping is described by the following equation:

$$z = \frac{1+w}{1-w} . \quad (3.24)$$

Applying this transformation to the transfer function of Equation (2.7), yields the

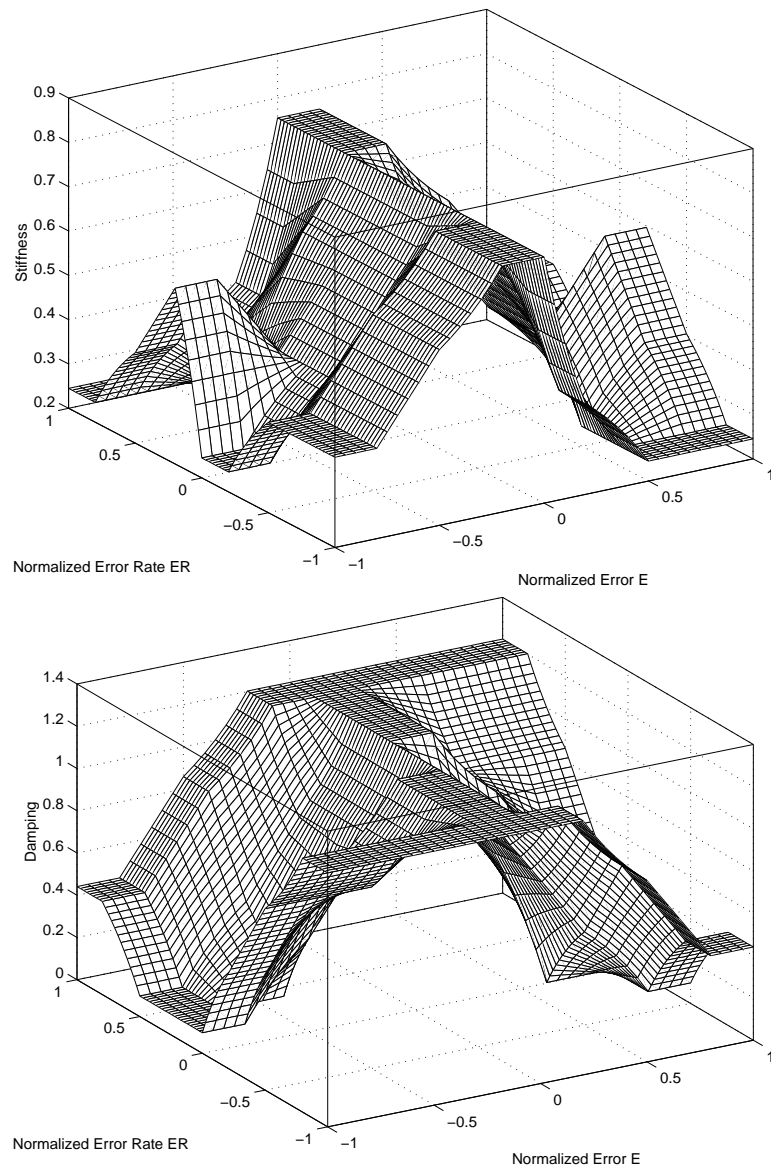


Figure 3.12: Stiffness and Damping Control Surface

transfer function of α - β filter with respect to the w -domain.

$$G(w) = \frac{-(2\alpha + \beta)w^2 + 2\alpha w - \alpha}{(4 - 2\alpha - \beta)w^2 + 2\alpha w + \beta} \quad (3.25)$$

The w -domain exhibits similar behavior to the continuous s -domain since their stability regions are both defined in the left half plane. This similarity is used to relate the stiffness and damping to the smoothing parameter α and β . Comparing the characteristic polynomial of Equation (3.22) with this of Equation (3.25) leads to the following relationship:

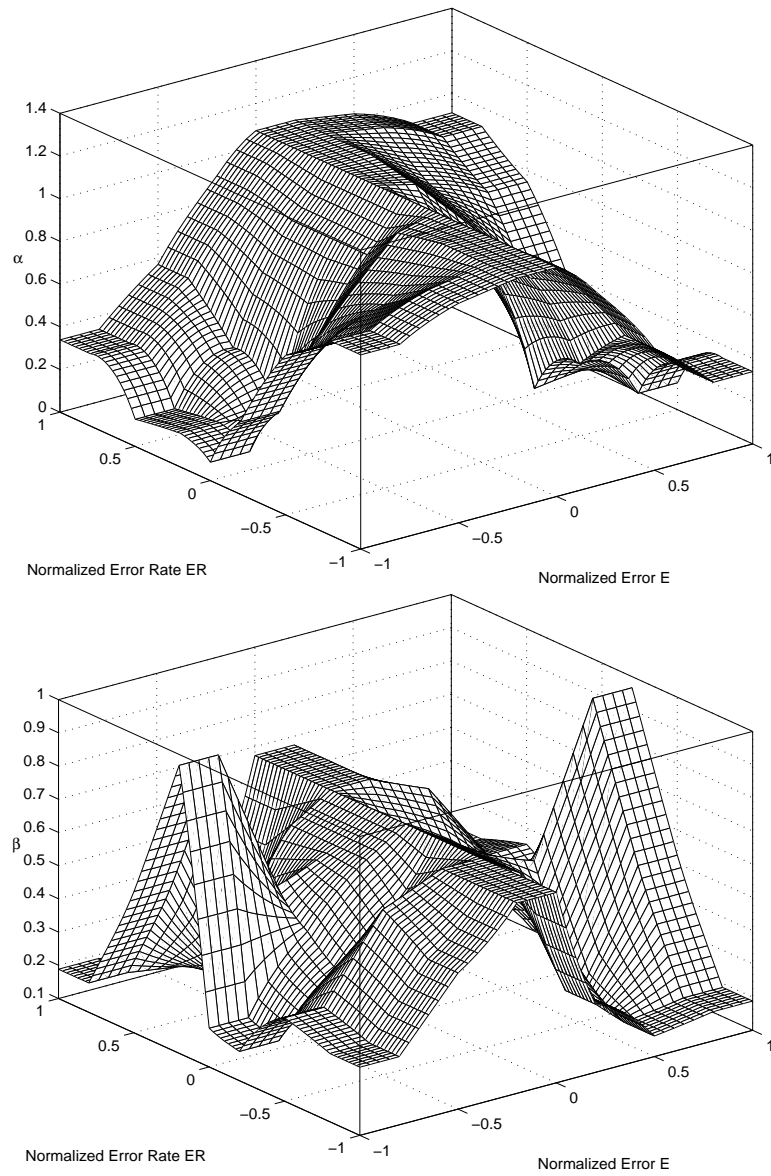
$$\alpha = \frac{4\zeta\omega_0}{2\zeta\omega_0 + \omega_0^2 + 1} \quad \beta = \frac{4\omega_0^2}{2\zeta\omega_0 + \omega_0^2 + 1} \quad (3.26)$$

Applying the transformation Equation (3.26) to the control surface shown in Figure (3.12) yields finally to the input-output-relationship between maneuver error and error rate and the smoothing parameter α and β , which is shown in Figure (3.13). During this mapping we assumed the direct relationship between stiffness and eigenfrequency.

3.3 Summary

This chapter proposed a *Fuzzy Logic Algorithm* applied to the standard α - β filter developed in Chapter 2. Unlike the fixed gain α - β filter, the fuzzy logic approach adapts the smoothing parameter α and β as a function of the maneuver error and error rate. Thus, the adaptive α - β filter is capable of tracking various types of maneuvers whereas the fixed α - β filter can only be optimized for one trajectory and certain sensor noise.

The fuzzy logic algorithm is first illustrated via single-input-single-output example and is discussed in detail in section 3.1.2 following the scheme illustrated in Figure (3.3). It is shown how the fuzzy relational matrix can be created using the input and output membership sets and the fuzzy rules. Finally, the relational matrix is used to calculate the output fuzzy set (μ_B) which is defuzzified for further usage.

Figure 3.13: Smoothing Parameter α and β Control Surface

The final section discusses the implementation of the fuzzy logic algorithm into the *target tracking* process. The linguistic rule base is determined by an analogy to the rest-to-rest maneuver, where the system variables, *stiffness* and *damping*, are derived. A relationship between those system variables and the smoothing parameters α and β is proposed to conclude with the control surfaces shown in Figure (3.13).

Chapter 4

Tracker Testbed

4.1 Parameters for the Testbed Setup and Simulation Execution

There are three types of parameters for our testbed. They are sensor field parameters, trajectory generation parameters, and simulation parameters.

The first type relates to the characteristic of sensors field. The testbed sensor field is a two-dimensional sensor array with three parameters, the dimension of the sensor array, the distance between two sensors, and the detection range of each sensor. The dimension of the sensor array determines the number of sensors in each direction. The testbed sensor array has same number of sensors in both X and Y directions. If a value 'n' is assigned to the dimension of sensor array, the total number of sensors would be 'n x n'. The X and Y distances are measured one center point to the other. Each sensor has a circular detection field with its boundary determined by a user-specified radius. Thus, detection range and distance control the density of the sensor field. Using these three parameters, the sensor field could be adjusted to approximate real situations having different sensor deployment patterns. In our testbed the initial detection range is 1 kilometer. The distances between two sensors are set to be 2,

5, and 10 kilometers for dense, medium, and sparse deployment cases, respectively. The sensor measurement noise variances in position (range) and direction (angle) are assumed to be provided by the manufacturer. These noises will be added to the true trajectory sampling points during the execution of simulation.

The second type includes target initial position, initial velocity, initial acceleration, and its radius of turn. Based on the magnitudes of the initial velocity in axes X and Y, the initial direction is also derived. With these parameters specified, four types of trajectory could be generated.

1. Straight line with constant velocity: Target travels with the initial velocity and remain the same direction throughout the simulation time.
2. Straight line with constant acceleration: Target travels with the initial acceleration, and remain the same direction throughout the simulation time.
3. Single gradual turn with constant angular velocity: With the radius of turn specified, the initial velocity is converted into constant angular velocity. Target travels in circle with this constant angular velocity throughout the simulation time.
4. Single gradual turn with constant angular acceleration: With the radius of turn specified, the initial velocity is converted into the initial angular velocity. The magnitude of acceleration is converted into angular acceleration. Target travels in circle with constant angular acceleration throughout the simulation time.
5. Interactive: Other than these options, the fifth type of trajectory generation is also provided. In order to simulate different cases of target maneuvers, the testbed allows the user to define trajectories interactively by using mouse click points. These points are way-points along the simulated true trajectory which is formed as a spline fit to these points. Within the sensor field specified in the previous section, the user provides a series of mouse click points to define the

way-points and thus the trajectory for a target. Each point, with coordinates X and Y , stands for the location of a target at certain simulation time. The time between two consecutive points is set to one unit. The total simulation time is equal to the total number of points specified in the sensor field with the first point excluded. Therefore, with different spacing and direction between connective points, the user could provide different variations of acceleration in X and Y directions, which facilitate the simulation of highly maneuvering target.

The third type includes the parameters relevant to simulation, such as simulation time, sampling time, and number of Monte Carlo runs. Simulation time determines the total time of each simulation run. Sampling time is the time duration between two consecutive sensor scans.

4.1.1 Filter initialization for Simulation

In order to initialize the filters, at least two measurements are needed. That is, the filters will not be activated until the first two measurements are obtained. The initialization steps are carried out under the following conditions. First, if the first few points of target trajectory locate outside the sensor detection range, these points will be trimmed and not considered for filter simulation. Second, if the first measurement is in the sensor detection range, while the next measurement is not obtained at the second sampling time, the second measurement should be derived by interpolation.

4.1.2 Noise Measurement Generation in Simulation

In order to simulate a realistic measurement scenario, zero mean and user specified variances for each sensor are used to generate noise. Adding noise to each true location point generates the noisy measurements of each target point. For example, if at certain simulation time the target point is at (X, Y) , the measurement of sensor

n would be $(X + \text{Randn}(0, \sigma_n^2), Y + \text{Randn}(0, \sigma_n^2))$, where $\text{Randn}(0, \sigma_n^2)$ stands for Gaussian distributed random number with zero mean and variance σ_n^2 for the sensor n. If a target point is located within the overlapping range of different sensors, the measurements from different sensors need to be fused. Since the sensor measurement noise variances in position and direction are given in Polar coordinate, the transformation of noise to the Cartesian coordinate is necessary.

Consider a two-dimensional radar system that measures range (R_o) and azimuth angle (η_o), and tracks in the x, y Cartesian coordinate system. The measurements are formed through the equations [3]:

$$x_o = R_o \cos \eta_o, y_o = R_o \sin \eta_o, \quad (4.1)$$

Then, the measurement covariance matrix is

$$R_c = \begin{vmatrix} \sigma_{x_o}^2 & \sigma_{x_o y_o}^2 \\ \sigma_{x_o y_o}^2 & \sigma_{y_o}^2 \end{vmatrix}, \quad (4.2)$$

where, using a first order expansion,

$$\sigma_{x_o}^2 = \sigma_{R_o}^2 \cos^2 \eta_o + R_o^2 \sin^2 \eta_o \sigma_{\eta_o}^2$$

$$\sigma_{y_o}^2 = \sigma_{R_o}^2 \sin^2 \eta_o + R_o^2 \cos^2 \eta_o \sigma_{\eta_o}^2$$

$$\sigma_{x_o y_o}^2 = \frac{1}{2} \sin 2\eta_o [\sigma_{R_o}^2 - R_o^2 \sigma_{\eta_o}^2]$$

$\sigma_{R_o}^2, \sigma_{\eta_o}^2$ = range and azimuth angle measurement variances, respectively.

4.1.3 Pairwise Sensor Fusion

In the proposed testbed, two α - β trackers, one for the X dimension the other for the Y dimension, are used for the overall sensor field. This is a decoupled-filter approach being used as an initial prototype for simplicity. Decoupled filters offer improved computational simplicity although, for Cartesian coordinates, they ignore the X-Y coupled errors. After studying the optimal and sub-optimal fusion techniques for α - β tracker, [18] provide optimal forms for an α - β fusion tracker. One of these forms employs a "data compression" or measurement fusion approach, which we use here. Equations below demonstrate the procedures and calculations needed for pairwise measurement fusion, an approach which yields a variance-weighted fused measurement.

- Z_1 = measurement from sensor 1
- Z_2 = measurement from sensor 2
- Z_F = fused measurement
- σ_1^2 = noise variance of measurement from sensor 1
- σ_2^2 = noise variance of measurement from sensor 2
- σ_F^2 = noise variance of fused measurement

$$Z_F = Z_1 + W(Z_2 - Z_1), \text{ where, } W = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}, \quad (4.3)$$

$$\sigma_F^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}, \quad (4.4)$$

4.1.4 Position Estimation of Filters in Simulation

Within the five types of filters, only the Chan's α - β filter combines the errors from both X and Y directions to update the α and β values. The other filters, including the proposed α - β filter, calculate errors in x and y direction separately and therefore update the filter gains separately in each direction. In the proposed α - β filter, error in each direction serves as the "crisp" input of the fuzzy logic controllers. Two fuzzy logic controllers, one for generating the stiffness value, the other for the damping value, are used. The new α and β values are calculated based on the stiffness and damping values and used to predict the target position for the next simulation time.

4.1.5 Performance Measures for Simulation Output Analysis

In order to compare the performance of different filters, three performance measures are adopted. To compare the accuracy of position prediction, the mean and variance of prediction errors are calculated. The prediction error is obtained by taking the difference of the predicted position and target true position. To compare the complexity of computation, the "floating point operation count" is calculated. It indicates the cumulative number of floating point operations at each time step for a specific filter.

4.2 The Functional Structure and the GUI

The simulation environment of this testbed consists of three major components, which are the parameter setup, the trajectory generation and the simulation execution. The simulation execution component is further decomposed into noise generation, filter execution, and result analysis. These components are constructed using MATLAB M-files and supported with graphical user interface (GUI). This main structure of the Testbed is shown in Figure (4.1).

The steps for using the testbed are illustrated in the flow chart of Figure (4.2)

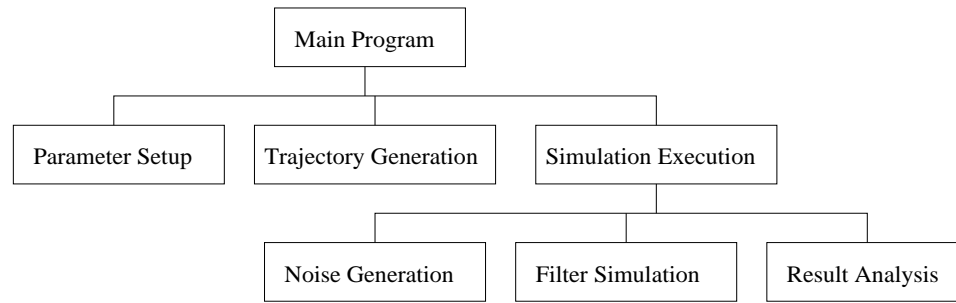


Figure 4.1: System Structure of Simulation Testbed

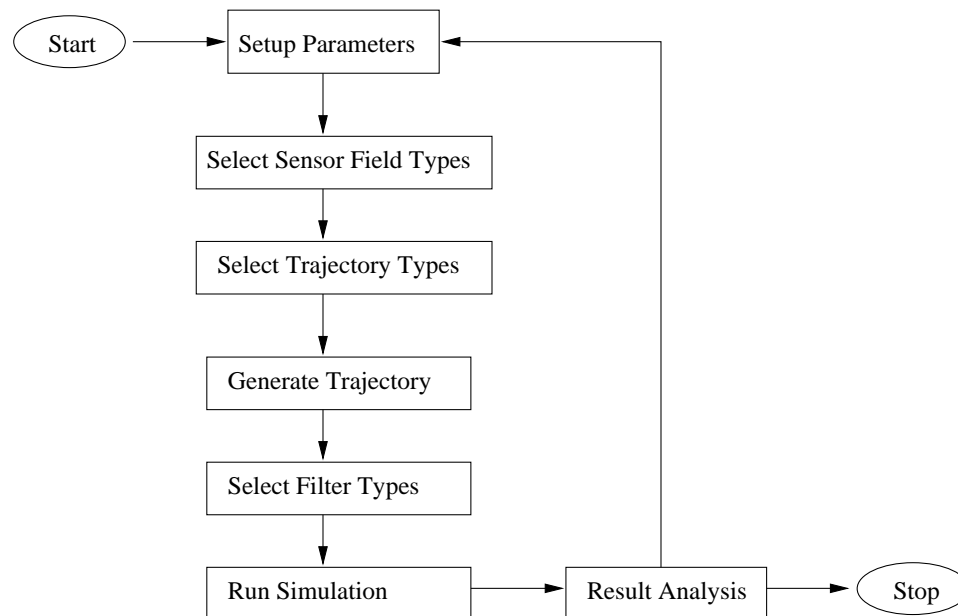


Figure 4.2: Flowchart of Simulation Testbed

and each step is explained below.

1. The Testbed can be started with executing the file `start.m` at the Matlab prompt and the initial menu shown in Figure (4.3) pops up. From there the user has three options: (i) reading the manual, (ii) entering the simulation setup or (iii) informing about the Testbed designers.
2. Entering the simulation setup, the users obtains the trajectory dialogue box shown in Figure (4.4). This interface gives various possibilities explain as ex-

plained in the following steps.

3. In the “Model Parameter” dialogue box (Figure (4.5)), the users specifies the values of target initial velocity, acceleration, radius of turn, sensor range and direction noises, simulation time, sampling time, and number of Monte Carlo runs.
4. Select sensor field type: In the “Sensor Spacing” list box, users select sensor deployment spacing from three options: dense, medium, or sparse.
5. Select trajectory type: In the “Maneuver” list box, user selects trajectory generation type from five options: straight line, single gradual turn, straight line with speed up, single gradual turn with speed up, or interactive.
6. Generate trajectory: After the “Generate Case” button is clicked, a cross-hairs presents the user to specify the trajectory starting point (for cases 1, 2, 3, 4 discussed in section 4.1) or traveling points (case 5 in section 4.1).
7. Select filter type: In the “Tracker” list box, users select either one of the filters or all filters for simulation and benchmark. They could be the α - β filter, the α - β - γ filter, the Chan’s α - β filter, the Proposed α - β filter, or the Kalman filter.
8. Run simulation: A mouse click on the “Simulation” button triggers the execution of simulation.
9. Result analysis: The duration of the simulation is emphasized by the “waitbar” shown in Figure (4.6). If the simulation is complete, the “Result Analysis” dialogue box (Figure (4.7)) displays the performance measures, error mean, error variance, and floating point operation count, for each filter. Users could further click on the buttons “Error Plot” and “Trajectory Plot” to view graphical representation of the tracking performance.

Modules	Major Supporting Files	Descriptions
Main Program	<code>start.m</code> <code>generate_field.m</code>	testbed start up program major user interface with: <ul style="list-style-type: none"> • sensor field • sensor field type, trajectory type, and filter type selection list boxes • parameter setup, trajectory generation, simulation execution buttons
Parameter Setup	<code>model_parameters.m</code> <code>set_parameter.m</code>	user interface for setting up the parameters update the parameter values when changed
Trajectory Generation	<code>generate_case.m</code>	generate trajectory
Noise Generation	<code>noise_measurement.m</code>	generate noise measurement and add to the true trajectory
Filter Simulation	<code>run_simulation.m</code> <code>alphabetatkr.m</code> <code>alphabetagamma_tkr.m</code> <code>chan_alphabetatkr.m</code> <code>proposed_alphabetatkr.m</code> <code>kf_tkr.m</code> <code>fuzzyengine.m</code> <code>alphatkr.fis</code> <code>betatkr.fis</code>	main file for simulation execution α - β filter α - β - γ filter Chan's α - β filter Proposed α - β filter Kalman filter fuzzy controller of proposed filter fuzzy controller of α for Chan's filter fuzzy controller of β for Chan's filter
Result Analysis	<code>result_analysis.m</code> <code>calculate_error_meanvar.m</code>	display simulation results for filters calculate estimation error mean and variance

Table 4.1: File Structure



Figure 4.3: The Startup Menu

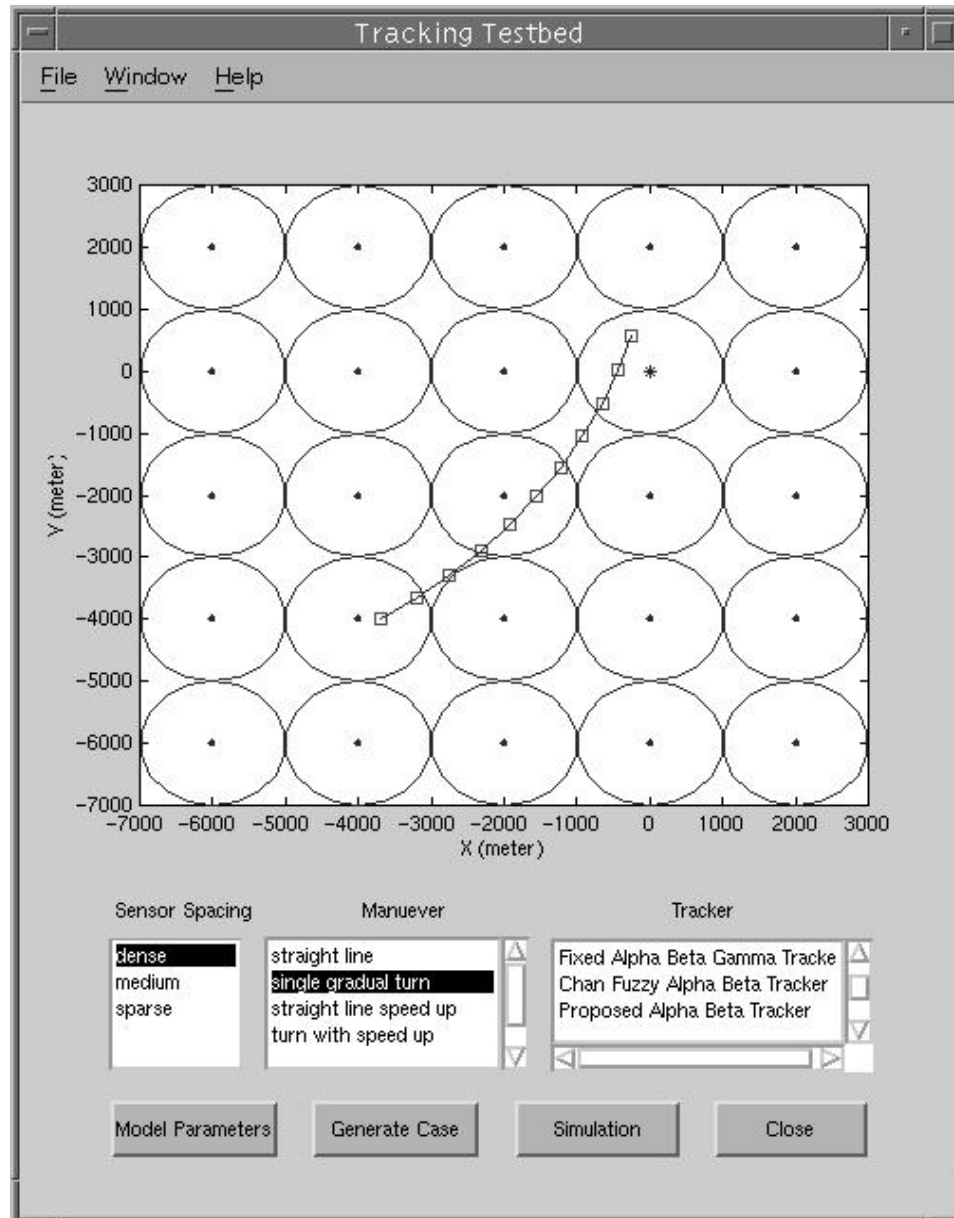


Figure 4.4: Layout of the Simulation Setup

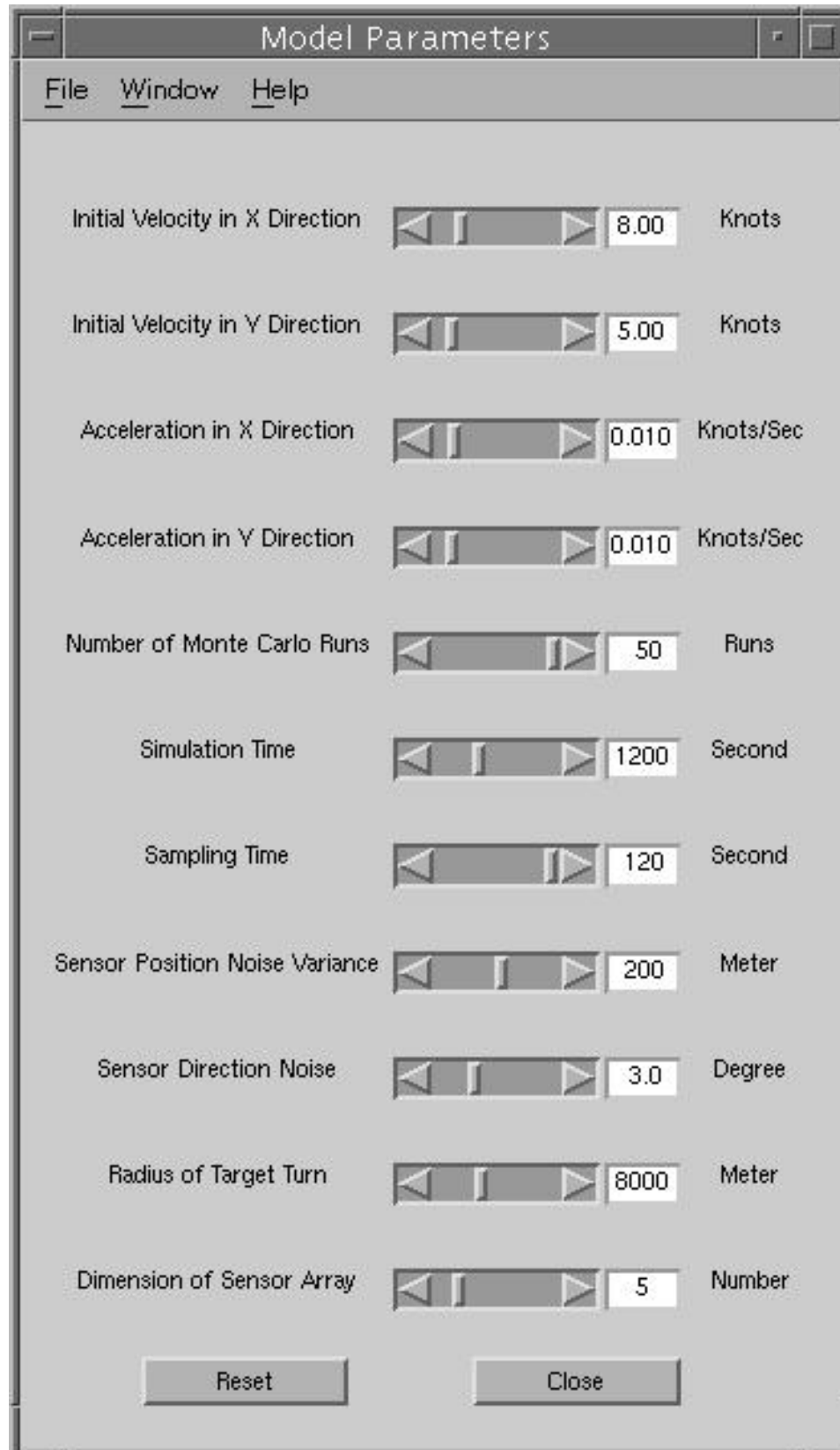


Figure 4.5: Layout of the Model Parameter Dialogue

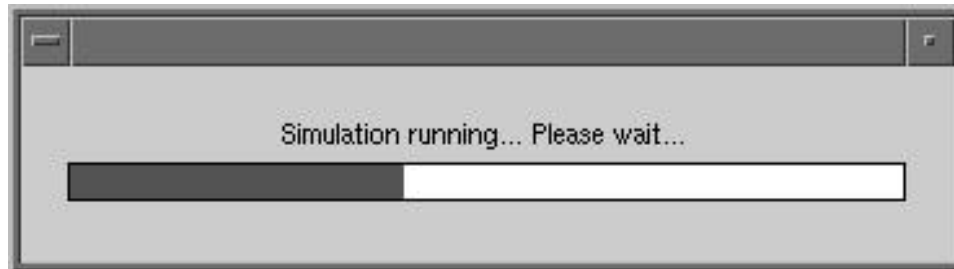
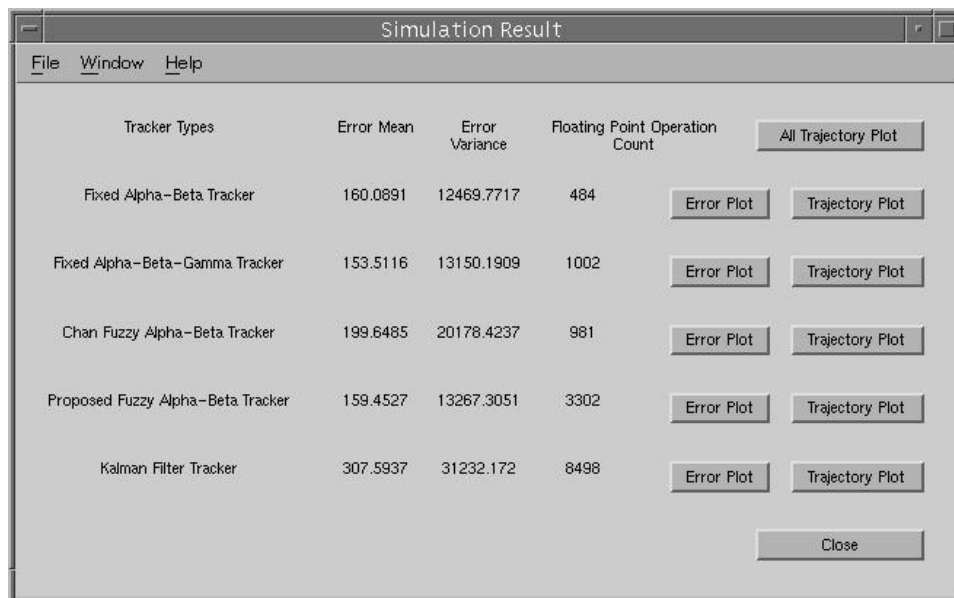


Figure 4.6: Waitbar, indicates a running simulation



Tracker Types	Error Mean	Error Variance	Floating Point Operation Count		
Fixed Alpha-Beta Tracker	160.0891	12469.7717	484	Error Plot	Trajectory Plot
Fixed Alpha-Beta-Gamma Tracker	153.5116	13150.1909	1002	Error Plot	Trajectory Plot
Chan Fuzzy Alpha-Beta Tracker	199.6485	20178.4237	981	Error Plot	Trajectory Plot
Proposed Fuzzy Alpha-Beta Tracker	159.4527	13267.3051	3302	Error Plot	Trajectory Plot
Kalman Filter Tracker	307.5937	31232.172	8498	Error Plot	Trajectory Plot

Buttons: All Trajectory Plot, Error Plot, Trajectory Plot, Close

Figure 4.7: Layout of the Result Analysis

Chapter 5

Results

Following the analysis of the standard α - β -(γ) filter, the development of a fuzzy logic modified α - β filter and the GUI Testbed, we may now discuss the filter performance with respect to realistic target trajectories of a submarine.

The first section concentrates on the proposed fuzzy logic filter, where basic filter results are shown, for instance, the selected smoothing parameter. The following section describes certain benchmark test, comparing the five tracking filter, (i) fixed α - β filter, (ii) fixed α - β - γ filter, (iii) Chan's FL α - β filter, (iv) Kalman filter and (v) proposed FL α - β filter. The tests are carried out as Monte-Carlo simulation for different target trajectories.

5.1 The Proposed Fuzzy Gain α - β Tracker

In this section, we shall discuss the proposed fuzzy logic target tracker, in particular, the choice of the smoothing parameter α and β , and the influence of limits of the discrete universe of discourse for the fuzzy output sets shown in Figure (3.8).

Consider the single gradual turn of the submarine shown in Figure (5.1). The limits of the discrete universe of discourse are selected as shown in Table 5.1 after

numerous iterations.

Stiffness		Damping	
lower	upper	lower	upper
0.25	0.35	0.54	0.89

Table 5.1: First Set of the Limits of the Fuzzy Output Membership Sets

As will be shown later, these limits change the region where the α and β parameter are active. We will first analyze the tracker for a benchmark maneuver shown in Figure (5.1), where the RMS error is 253 [m]. This figure illustrates the true trajectory by “x”s, the noisy measurements by squares and the estimated trajectory by circles. The effect of initialization of the filter can be seen in addition to the effects of missing observations. The missing observation occurs when the submarine is outside the sensor range and is indicated by missing “squares” corresponding to the “x”s. The target tracker predicts future location of the target along the direction prior to the missing observations for every sampling instant. This can be seen in Figure (5.1) from the 7th to the 12th sampling instant. When a target is detected later, the tracker accomodates the error in the prediction and uses the measurement to change the prediction of the future trajectory as shown at the 13th time instant.

The choice of the dynamic α and β parameters depends on the maneuver error and error rate. Figure (5.2) illustrate the change in the α - β parameters within the stability triangle for the trajectory illustrated in Figure (5.1) based on the proposed fuzzy rules. It can be seen that the α - β parameters lie far from the point $\alpha=\beta=1$, which corresponds to the fastest tracker response. This is to accomodate for the noise in the measurement. At the start of the tracking process, the tracker is characterized by the parameter set $\alpha=\beta=1$. It is interesting to note that the fuzzy logic α - β scheduler places the parametes in the vicinity of the optimal α - β paramters ($\alpha = 0.5377$ and $\beta = 0.2018$) as determined for a straight line maneuver with position noise variance of 200 [m] and direction noise variance of 3 [degree]. It should be noted that the “optimal” solution is optimal only for a target moving along a straight line

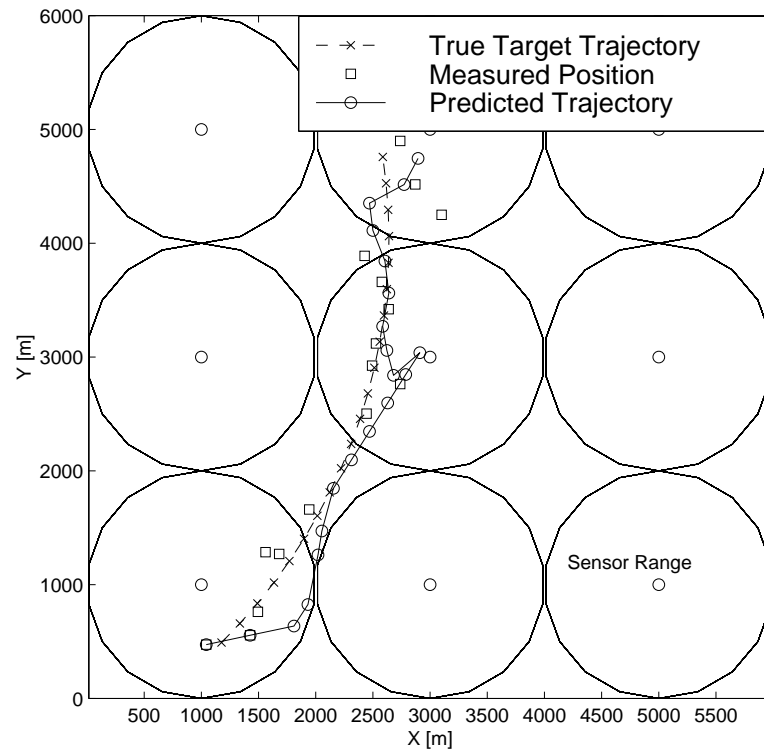


Figure 5.1: Single Gradual Target Turn and its Prediction using the Fuzzy Logic Gain Modified α - β Tracker

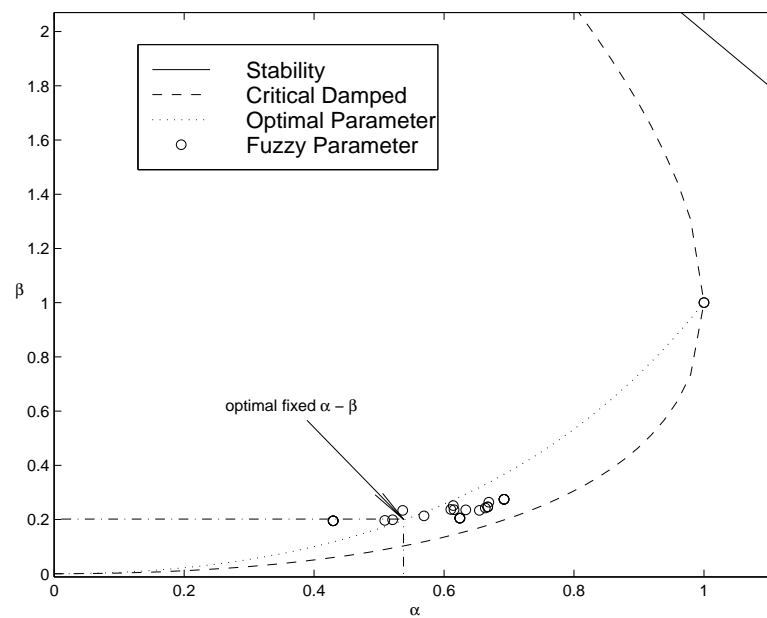


Figure 5.2: Fuzzy Logic Modified Smoothing Parameter α and β

with constant velocity and since realistic targets do not satisfy those constraint, the fuzzy logic perturbs the filter parameters to accomodate maneuvering targets and the measurement noise.

Finally, the effect of the changing the limits of the discrete universe of discourse is shown on the next example. The limits are changed to the sets shown in Table 5.2.

Stiffness		Damping	
lower	upper	lower	upper
0.2	0.8	0	1.3

Table 5.2: Second Set of the Limits of the Fuzzy Output Membership Sets

Figure (5.3) shows the region where the filter parameters reside for the trajectory illustrated in Figure (5.1). Figure (5.4) illustrates the performance of the filter for the current universe of discourse of the output sets. It is clear that the predicted track is not as smooth as in the previous case since the system dynamics are characterized by very fast response which results is a more aggressive filter which can potentially amplify measurement noise. The limits of the discrete universe of discourse of the output membership sets can be used to adapt the tracker to different targets. By increasing the stiffness one could probably track highly maneuvering targets more easily.

5.2 Benchmark Tests

Benchmark tests are carried out to perform comparison between different target tracker where the Testbed (Chapter 4) can be used. Within the Testbed five target tracker have been installed which are listed below.

Tracker 1: Fixed parameter α - β filter

Tracker 2: Fixed parameter α - β - γ filter

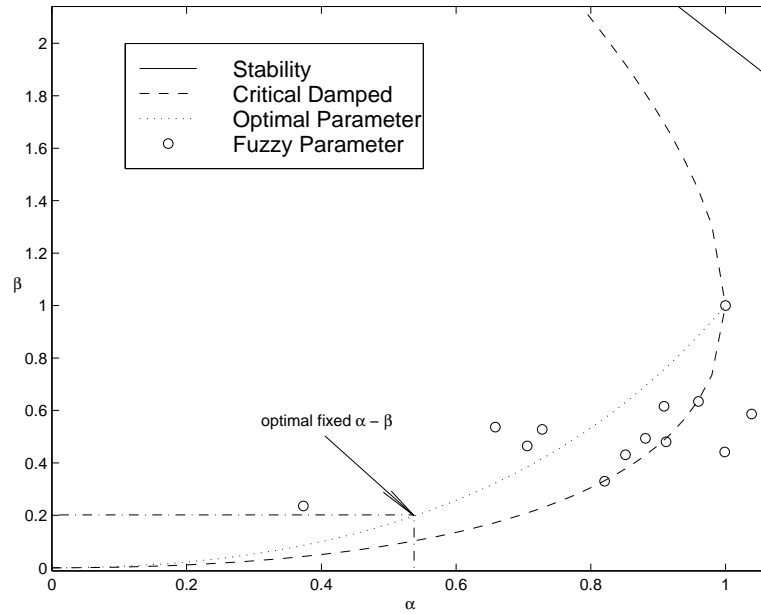


Figure 5.3: Fuzzy Logic Modified Smoothing Parameter α and β with modified limits

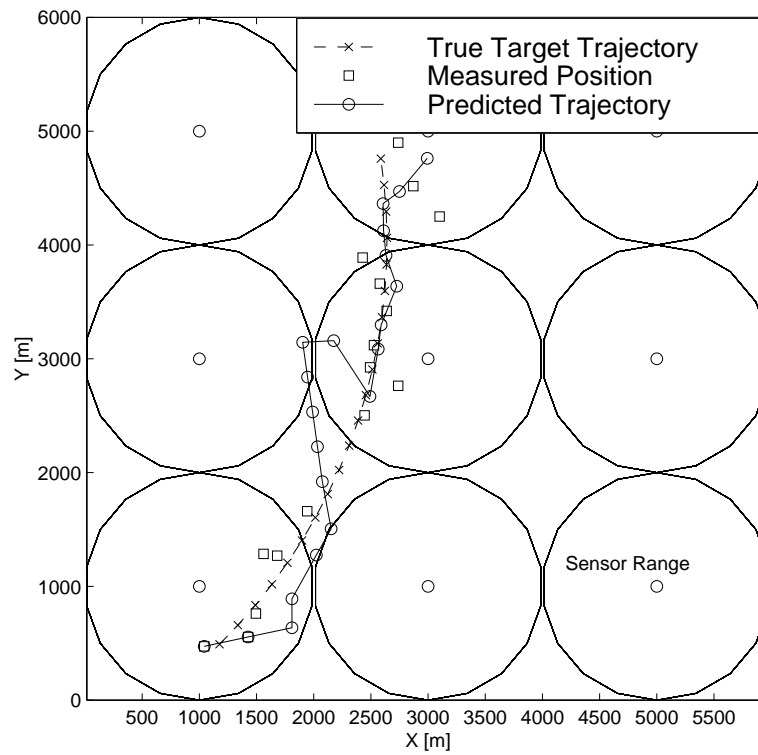


Figure 5.4: Single Gradual Target Turn and its Prediction using the Fuzzy Logic Gain Modified α - β Tracker with modified limits

Tracker 3: Chan's fuzzy logic modified α - β filter [5]

Tracker 4: Proposed fuzzy logic modified α - β filter

Tracker 5: Kalman filter

These tracker labels are used in the following tables for brevity. Each tracker has been introduced in Chapter 4. One requirement on the Testbed was to simulate realistic target trajectories, which now may be used to perform the benchmark test. These tests evaluate the performance of the target trackers for a fixed trajectory, where the observed data vary with different measurement noise with every Monte Carlo run. The target tracker performance is basically evaluated on four realistic target maneuvers of a submarine, which are:

Maneuver 1: Targets moving with constant speed on a straight line

Maneuver 2: Targets moving with constant acceleration on a straight line

Maneuver 3: Targets moving with constant speed on a single gradual turn

Maneuver 4: Targets moving with constant acceleration on a single gradual turn

The four benchmark target maneuvers are shown in Figure (5.5). In addition, three sensor distributions are provided for each maneuver which are (i) dense, (ii) medium and (iii) sparse as illustrated in Chapter 4. The following discusses the target tracker performance for those three sensor distributions and the four benchmark trajectories listed above. The simulations have been performed over 50 Monte Carlo runs with a sensor noise characteristic of 200 [m] position and 3 [deg] of direction variance. The target trajectories are generated with a fixed sampling time of 60 seconds over a time window of 1200 seconds, which means 20 successive scans.

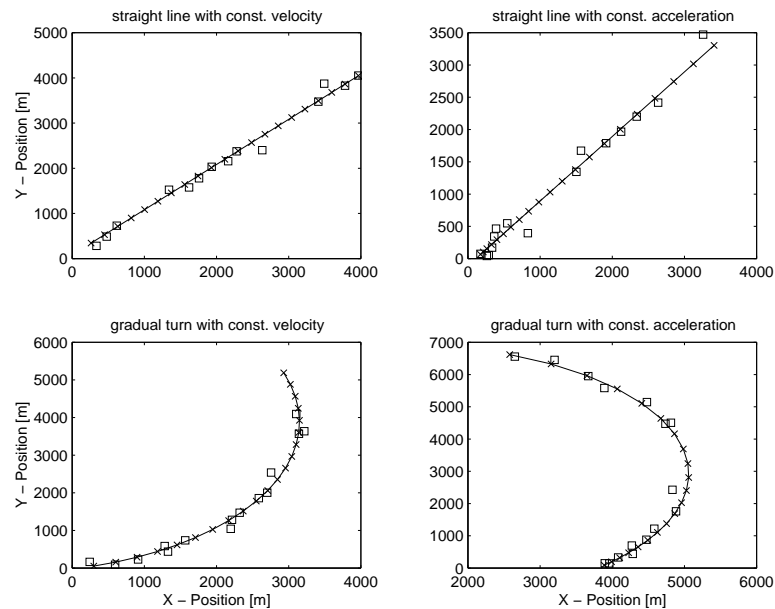


Figure 5.5: The Four Benchmark Target Maneuvers including Example Noise

Dense Sensor Field

The sensor distribution of a dense sensor field is shown in the Chapter 4, where the number of missed observations is possibly low. The simulation results are listed in Table 5.3.

Since the smoothing parameter α and β are optimal parameters calculated for a target moving with constant speed on a straight line and the corresponding sensor noise characteristic, the fixed gain α - β filter exhibits the lowest maneuver error. Both, the Kalman and the fixed gain α - β - γ filter are designed to track accelerating targets. Simulations of the Kalman filter without modeling the acceleration as process noise has shown excellent performance for this maneuver, whereas the maneuver error increases in case of accelerating targets. As shown in Section 5.1, the fuzzy logic modified smoothing parameters are in the region of the optimal α - β parameters, which leads to a similar maneuver error as the fixed gain filter. The fixed gain α - β - γ tracker is as expected, the filter with the lowest maneuver error since its parameter are chosen to perform optimal. The Kalman filter also exhibits a low maneuver

Filter	Maneuv. 1		Maneuv. 2		Maneuv. 3		Maneuv. 4	
	Mean	Var.	Mean	Var.	Mean	Var.	Mean	Var.
1	141	12792	211	25386	251	22259	436	120538
2	168	20286	189	24348	225	19276	397	94391
3	185	21779	227	52676	275	44510	333	57686
4	149	13563	204	23780	253	24382	439	113478
5	198	36173	190	27188	246	25604	376	95828

Table 5.3: Simulation Results for a dense Sensor Field at 60 [s] sampling time

Error	Tracker 1	Tracker 2	Tracker 3	Tracker 4	Tracker 5
Mean	224	222	271	223	263
Var.	2853	33360	67674	30171	35603

Table 5.4: Simulation Results for a dense Sensor Field at 30 [s] Sampling Time

error since the Kalman filter is designed for accelerating targets. Even for the single gradual turn, the fixed gain α - β - γ filter exhibits a reasonable performance. In the case of targets traveling with constant speed all five target tracker exhibit similar performance. Whereas tracking targets with constant acceleration yields different performances of the five target trackers. In general, the maneuver error is higher than for the other maneuvers for *Maneuver 4*, and the fixed gain α - β and the proposed fuzzy logic modified gain α - β filter show a higher maneuver error, whereas the Kalman filter and Chan's fuzzy logic gain modified α - β filter show relatively low maneuver errors. However, decreasing the sampling time from 60 seconds to 30 seconds, yields an improvement of the performance. Table 5.4 shows the maneuver errors for a sampling time of 30 seconds. Decreasing the sampling time increases the number of successive scans, which in turn yields a better performance.

Filter	Maneuv. 1		Maneuv. 2		Maneuv. 3		Maneuv. 4	
	Mean	Var.	Mean	Var.	Mean	Var.	Mean	Var.
1	381	108535	700	411456	749	541677	1523	1635908
2	1786	9770645	1135	1159524	1050	1159934	1612	2200747
3	583	307365	1172	1386552	753	562431	1249	1443696
4	378	111253	666	402310	687	522226	1548	1845937
5	405	123283	1173	990780	809	488069	1629	2054990

Table 5.5: Simulation Results for a medium Sensor Field at 60 [s] sampling time

Filter	Maneuv. 1		Maneuv. 2		Maneuv. 3		Maneuv. 4	
	Mean	Var.	Mean	Var.	Mean	Var.	Mean	Var.
1	1682	1227601	1841	2060016	3392	7071981	7892	64618001
2	3031	12065839	3743	22692198	5159	31505394	10126	175835941
3	2412	4627991	3058	13744394	6152	55393143	12519	231254121
4	1619	1205443	1613	1948927	3572	7245144	7799	65240824
5	4335	25675807	3473	11642534	8251	64635098	12914	255547298

Table 5.6: Simulation Results for a sparse Sensor Field at 60 [s] sampling time

Medium and Sparse Sensor Field

The simulations with a medium or sparse sensor field are similar to those of the dense sensor field. The differences are in generating the target trajectory. Since the distances between the sensors becomes larger, a longer target trajectory is required to obtain successive scans. The simulation results are listed in Tables 5.5 and 5.6.

The proposed target tracker exhibits an excellent performance for a medium or sparse sensor field. However, increasing the sampling time leads to a similar behavior mentioned for the dense sensor field. The question arises how the sampling time may be included in the fuzzy logic α - β scheduler since certain sampling times exhibit

	<i>Tracker 1</i>	<i>Tracker 2</i>	<i>Tracker 3</i>	<i>Tracker 4</i>	<i>Tracker 5</i>
FLOPS	470	974	(941)	2287	8282

Table 5.7: FLOPS used by the Target Tracker for 20 scans

different performances. Future work proposes a sampling time depending mapping between the control system parameters stiffness and damping and the smoothing parameter α and β .

Floating Point Operations

Measuring the computational cost accurately is not a feasible procedure in MATLAB, which provides a counter for floating point operations (flops) where the most important operations are counted (for example: additions and subtractions are counted as one flop). The used flops of every target tracker is listed in Table 5.7, where the target trajectory consists of 20 points. As expected, the fixed α - β filter requires less computational effort than the other, and the Kalman filter requires the most resources because of its complexity. The value of Chan's fuzzy logic filter differs from the proposed because Chan's filter is coded using the MATLAB Fuzzy Toolbox, where the fuzzy evaluation at every sampling time is not counted as flops. For reasons of flexibility, the proposed fuzzy logic filter does not require this toolbox, and each fuzzy evaluation is counted as 54 flops. Thus, one fuzzy evaluation might be counted as 54, and for 20 evaluations Chan's flops need to be increased by approximately $54\text{flops} \times 20 = 1080\text{flops}$.

5.3 Remarks

This chapter has illustrated the target tracking process for the Anti-Submarine Warfare. Different tracking filter have been used to conduct performance comparison for

some benchmark target trajectories and three sensor field distributions. The first section has shown that the proposed fuzzy logic scheduler selects α and β within an optimal region. Furthermore, the active region of the fuzzy logic may be adopted to achieve certain performances by changing the limits of the discrete universe of discourse of the fuzzy output functions stiffness and damping.

The similarities in choosing the smoothing parameters of the “optimal” fixed gain and fuzzy logic modified gain α - β filter are reflected in the simulation results, where the proposed fuzzy tracker mostly improves the performance of the fixed α - β filter. Finally, it has been shown that tracking within a medium or sparse sensor field becomes more difficult since we lack observed data. Whereas, the proposed fuzzy logic gain modified α - β filter has shown acceptable tracking performance where the maneuver error is small compared to the other filter.

Chapter 6

Conclusion and Recommendation

The examination of the applicability of Fuzzy Mathematics and Logic for association and tracking applications has shown, first of all, that such methods are fundamentally viable, at least for target tracking. The literature review of Phase I has shown that such methods have frequently if not consistently demonstrated comparable or better performance when compared to either fixed coefficient or Kalman type filters, for a variety of problem types. This is not to say that such methods are categorically better for association and tracking problems. Typical tradeoffs on target tracker are expected to depend on a variety of parameters such as the number of targets, target maneuverability, environment and sensor characteristics such as resolution and sampling rate among possibly other parameters. In this work five target tracker have been tested on several target trajectories by performing Monte Carlo simulations. It has been shown that the proposed fuzzy logic gain modifying α - β filter exhibits acceptable performance.

The focus on this work has centered on the modern-day Anti-Submarine Warfare problem and a particular type of sensing (sonobuoys) and target environment. In that particular environment, this research has examined the use of fuzzy logic methods for adaptive parameter control for widely known α - β filter. In essence, this approach generates a pseudo-adaptive filtering scheme in which the filter coefficients

are set by the fuzzy logic controller at each simulation time interval according to problem condition and estimated error and error rate. A distinctive aspect of this work is on a detailed research of the fixed α - β - γ filter to characterize its performance. The α - β - γ filter has analyzed with the intention of eventually developing a fuzzy α - β - γ filter and to characterize the α - β filter which is a subset of the α - β - γ filter. This aspects and the employment of such an approach, to our knowledge, has not been previously researched. The results are briefly listed below:

- proposing a simple technique to simplify the performance derivations of the α - β - γ filter via a non-linear transformation
- introducing the stability prism in the a - b - c space
- arriving at a closed form solution of the noise-ratio in the a - b - c space
- evaluating the veracity of the closed form solution of the noise-ratio for the α - β filter which is different from those derived in the literature
- deriving closed form solutions to characterize the transient performance of the α - β - γ filter
- performing an optimization to provide the designer with information for the optimal selection of the α , β and γ parameter
- determining a fuzzy rule base by an analogy to the rest-to-rest maneuver in terms of stiffness and damping
- proposing a control-theoretic relationship between the smoothing parameter α and β and the rule base parameter stiffness and damping

The Testbed has been developed as a graphical user interface controlling various kinds of simulation parameter. The Testbed performance entails:

- realistic simulation of the sensor field

- generating realistic target trajectories
- user selectable target tracker
- build-in metrics for comparison

Bibliography

- [1] Yaakov Bar-Shalom and Thomas E. Fortmann. *Tracking Data and Association*. Academic Press, Inc., 1988.
- [2] T. R. Benedict and G. W. Bordner. Synthesis of an optimal set of radar track-while-scan smoothing equations. In *IRE Transaction on Automatic Control*, volume AC-1, July 1962.
- [3] S. S. Blackman. *Multiple-Target Tracking with Radar Applications*. Artech House, Inc., 1986.
- [4] Martin Brown and Chris Harris. *Neurofuzzy Adaptive Modelling and Control*. Prentice Hall International (UK), 1994.
- [5] Keith C.C. Chan, Vika Lee, and Henry Leung. Robust target tracking using a fuzzy filter. In *International Conference on Systems, Man and Cybernetics*, volume 5. IEEE, October 1995.
- [6] L. Ducourau, T. Singh, and R. W. Main. Automated parameter optimization for structural and controller design. *Proceedings of the CSME Mechanics in Design Conference*, May 6-9 1996.
- [7] Bernard Friedland. Optimal steady-state position and velocity estimation using noisy sampled data. *IEEE Transaction on Aerospace and Electronic Systems*, AES-9(6), November 1973.
- [8] Paul R. Kalata. The tracking index: A generalized parameter for $\alpha - \beta$ and $\alpha - \beta - \gamma$ target trackers. In *IEEE Transactions on Aerospace and Electronic Systems*, volume AES-20, No.2. ECE Department, Drexel Univeristy Philadelphia, Pennsylvania, March 1984.
- [9] Paul R. Kalata. $\alpha - \beta$ target tracking systems: A survey. In *American Control Conference/WM12*. ECE Department, Drexel Univeristy Philadelphia, Pennsylvania, 1992.
- [10] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82, March 1960.

- [11] R. E. Kalman and R. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83D, March 1961.
- [12] Allen J Kanyuck. Transient response of tracking filters with randomly interrupted data. *IEEE Transaction on Aerospace and Electronic Systems*, AES-6(3), May 1970.
- [13] Katsuhiko Ogata. *Discrete-Time Control Systems*. Prentice-Hall, Engelwood Cliffs, New Jersey, University of Minnesota, 1987.
- [14] C.C. Schooler. Optiaml $\alpha - \beta$ filters for systems with modeling inaccuracies. In *IEEE Transactions on Aerospace and Electronic System*, volume AES-11, November 1975.
- [15] H. R. Simpson. Performance measures and optimization condition for a third order sampled-data tracker. In *IEEE Transaction on Automatic Control*, volume AC-12, June 1962.
- [16] Jack Sklansky. Optimizing the dynamic parameter of a track-while-scan system. *RCA Laboratories, Princeton, N.J.*, June 1957.
- [17] Li-Xin Wang. *Adaptive Fuzzy Systems and Control : Design and Stability analysis*. Prentice Hall, 1994.
- [18] John Yosko and Paul R. Kalata. Optimal and sub-optimal fusion of $\alpha - \beta$ target tracks. In *American Control Conference/WM12*. JJM Systems Inc., One Ivybrook Blvd., Suite 190, Ivyland, Pennsylvania, 1992.

Appendix A

Kalman Filter Process Noise

The performance of the Kalman filter is highly depended on the model approximating the real system. In this work a second order decoupled Kalman filter has been used. Which means that the Kalman filter is designed to track targets with constant velocity, and the acceleration is modeled as process noise. The uncertainty of the submarine motion due to the unknown acceleration makes it difficult to determine a specific value for the process noise. The influence of the process noise assumption on the mean tracking error is shown in Figure (A.1), where a trajectory without missing observations has been simulated over 50 Monte Carlo runs. As expected, the maneuver error is high if the process noise is set to be zero, since the model does not contain any assumptions about an accelerating target. As is clear from Figure A.1, the assumed variance of the process noise has a significant effect on the resulting mean tracking error. It can be seen that a specific value of the process noise results in a minimum tracking error for a target maneuvering with constant acceleration. The selection of the process noise is an open research issue which the authors hope to address in the future.

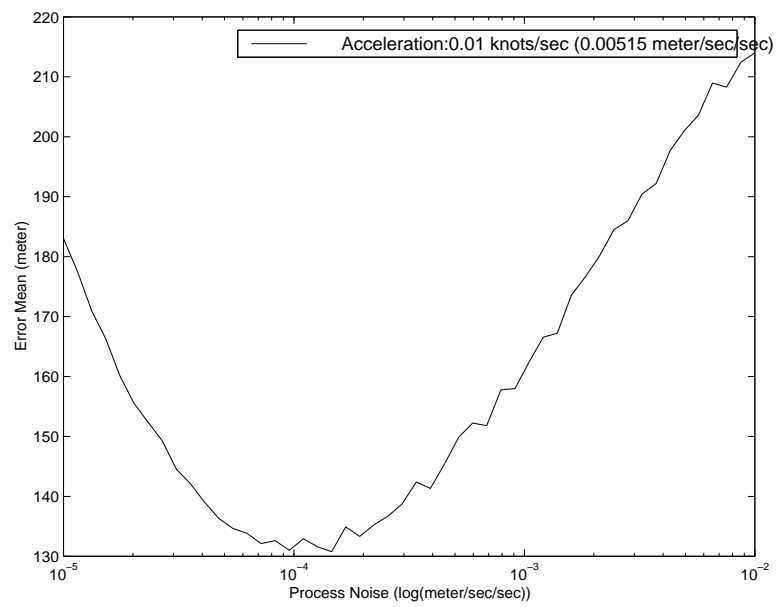


Figure A.1: The Influence of the Process Noise on the Maneuver Error Mean