Algorithms for Tracking Single Maneuvering and Multiple Closely-Spaced Targets

Mohamed Nabil Abdelghaffar Eltoukhy

A Thesis

In the Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy (Electrical and Computer Engineering) at

Concordia University

Montréal, Québec, Canada

December, 2020

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Mohamed Nabil Abdelghaffar Eltoukhy

Entitled: Algorithms for Tracking Single Maneuvering and Multiple Closely-Spaced Targets

and submitted in partial fulfillment of the requirements for the degree of

Doctor Of Philosophy (Electrical and Computer Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

	Chair
Dr. Yuhong Yan	
	External Examiner
Dr. Magdy A. Bayoumi	
	External to Program
Dr. Chun-Yi Su	
	Examiner
Dr. Chunyan Wang	
	Examiner
Dr. Wei-Ping Zhu	
	Thesis Co-Supervisor
Dr. M. Omair Ahmad	
	Thesis Co-Supervisor
Dr. M.N.S. Swamy	

Approved by:

Dr. Wei-Ping Zhu, Graduate Program Director

December 21, 2020

Dr. Mourad Debbabi, Dean,
Gina Cody School of Engineering and Computer Science

Abstract

Algorithms for Tracking Single Maneuvering and Multiple Closely-Spaced Targets

Mohamed Nabil Abdelghaffar Eltoukhy, Ph.D.

Concordia University

Target tracking is crucial in monitoring and controlling air traffic in civilian and military applications. Target tracking is a process of estimating the current position and predict the future position of one or more targets using the measurements received by a radar system.

One of the major challenges in tracking a single target is when it performs a maneuver and the angle of maneuver is not known. The interacting multiple model (IMM) algorithm is the most commonly-used algorithm for tracking a maneuvering target with an *a priori* knowledge of the target turn rate, since it provides a very good tracking performance with moderate complexity. However, the tracking performance of such an algorithm deteriorates or may even fail when the target performs a maneuver with a turn rate larger than that assumed in the design of the algorithm. A few methods have been reported to overcome this limitation of an assumed turn rate by actually estimating it adaptively. Two of such algorithms use nonlinear filters that leads to a large complexity, and one of them uses linear filters and models providing good tracking performance, but only for mild maneuvers.

For tracking multiple targets, several algorithms have been proposed, among which the joint probability data association (JPDA) algorithm is considered to be the best algorithm, since it provides good tracking performance when the targets are widely spaced. However, the tracking performance of this algorithm deteriorates, and coalescence of the tracks may occur, when the targets are closely spaced. Some efforts have been made to overcome the problem of tracking closely spaced targets by ignoring the target identity, but at the expense of very large complexity.

The work of this thesis is carried out in two parts. In the first part, two algorithms within the IMM framework are proposed to track a single maneuvering target, when the target turn

rate is not known *a priori*. In both the algorithms, the turn rate is dynamically estimated using noisy measurements. In the first algorithm, the turn rate at each time instant k is estimated based on the target speed and the radius of the circle formed by the measurement at that instant and the two previous consecutive noisy measurements, $(k-1)$ and $(k-2)$. The segment of this circle covered by these three noisy measurements is used to model the true track of the target at the instant time k. In the second algorithm, the accuracy of the turn rate estimated in the first algorithm is improved using the information on the level of the measurement noise.

In the second part of the thesis, a systematic study on the impact of the spacing between the targets as well as when the targets make abrupt turns with sharp angles on the tracking performance of the JPDA algorithm is conducted. Then, a new algorithm for tracking multiple targets based on the spatial distribution of the measurements for determining the weights for measurement-target association is proposed within the JPDA framework. The proposed algorithm for multiple target tracking is designed to deal with the problems of closely spaced targets and their abrupt sharp turns more effectively.

Effectiveness and superiority of the algorithms proposed for tracking single and multiple targets are demonstrated through extensive experiments with a wide variety of different scenarios for target motions.

I dedicate this work to the soul of my father, who passed away on August 29, 2020.

My father wished to see the day when I would receive my Ph.D. degree.

"Dear father, I have struggled hard to realize your dream and that day seems to have arrived, even though you are not physically with me.

May peace be upon you and see you when we meet again"

Acknowledgments

This work has resulted from the diligent hard work of three years, a period during which by the grace of Gad I have learned a great deal and built my background and knowledge in the field of target tracking. I would like to acknowledge many, who have supported, helped, and encouraged me through this journey.

First of all, I would like to thank my supervisors, Prof. M. Omair Ahmad and Prof. M.N.S. Swamy. It is my privilege and honor to be one of their Ph.D. students. My supervisors have provided me support and confidence through all the phases of my Ph.D. program. They gave me freedom to develop my ideas in this research with a professional guidance. Without their efforts and extensive experience in the field of my research and (in general) in signal processing, this work would not have been possible. I would like to thank the members of my supervisory committee, Prof. Wei-Ping Zhu, late Prof. Terry Fancott, Prof. Chun-Yi Su, and Prof. Chunyan Wang for their comments during the various stages of my study and for their reviewing my dissertation. In addition, I would like to thank Ms. Sheryl Tablan, the ECE program coordinator, and the staff of Concordia library for their help during the course of my study.

I would like to acknowledge the Government of Egypt and Ministry of Defense for giving me the opportunity to carry out my study at Concordia university and providing me with the financial support. I would also like to acknowledge the support and help of my colleague and friend, Ahmed Abouelfadl, in the preparation of the reports for my Ph.D. proposal and seminar.

I would like to thank my mother and late father, my parents in-law, my sister, and my brother for their encouragement and prayers during this stressful period. Last but not the least, I would like to thank my beloved wife, Rania, for her everlasting unbounded love, and scarifies. Finally, I would like to thank my lovely sons, Ahmed, Omar, and Fares, whose love has been a great source of inspiration for me to conduct my studies.

Contents

List of Figures

List of Tables

List of Abbreviations

STT	Single target tracking
MTT	Multi-target tracking
VD	Variable dimension
IMM	Interacting multiple model
GPB1	First order generalized pseudo-Bayesian
GPB2	Second order generalized pseudo-Bayesian
NN	Nearest neighbor
MHT	Multi-hypotheses tracking
PDA	Probabilistic data association
JPDA	Joint probabilistic data association
FISST	Finite set statistics
SetJPDA	Set joint probabilistic data association
KLSJPDA	Kullback-Liebler set JPDA
MOJPDA	Multiple objective data association
KF	Kalman filter
EKF	Exceeded Kalman filter
UKF	Unscented Kalman filter
CKF	Cubature Kalman filter
PF	Particle filter
2D	Two-dimensional
3D	Three-dimensional
CT	Coordinated turn
CTL	Coordinated turn for turning left

CTR	Coordinated turn for turning right
CV	Constant velocity
CA	Constant acceleration
3CTR	Three components turning rate
NEES	Normalized estimation error Squared
ANEES	Average normalized estimation error Squared
LANEES	Log average normalized estimation error Squared
RMSE	Root mean square error
ID	Target identity
MC	Monte-Carlo
STD	Standard deviation

List of Symbols

$\mathbf{x}(k)$	True state vector
$\hat{\mathbf{x}}(o)$	Initial state vector
$\hat{\mathbf{x}}(k)$	Estimate state vector
$\tilde{\mathbf{x}}(k)$	Predicted state vector
$\mathbf{z}(k)$	Received measurement vector
$\mathbf{\Gamma b}$	Process noise vector
$\mathbf{\Gamma Q \Gamma}'$	Covariance matrix of process noise
$\mathbf{m}$	Measurement noise vector
$\mathbf{U}$	Covariance matrix of measurement noise
$\mathbf{P}(k)$	Covariance matrix of the state vector
$\hat{\mathbf{P}}(k)$	Estimated covariance matrix
$\hat{\mathbf{P}}(o)$	Initial covariance matrix
$\tilde{\mathbf{P}}(k)$	Predicted covariance matrix
$\mathbf{S}(k)$	Innovation covariance matrix
$\mathbf{P}_{tr}$	Transition probability matrix
p_{ij}	Element (i, j) in $\mathbf{P}_{tr}$ of the IMM algorithm
$\hat{\mathbf{x}}_{oj}$	Input state vector of filter j in an IMM algorithm
$\hat{\mathbf{P}}_{oj}$	Input covariance matrix of filter j in an IMM algorithm
$\hat{\mathbf{x}}_{j}$	Output state vector of filter j in an IMM algorithm
$\hat{\mathbf{P}}_{j}$	Output covariance matrix of filter j in an IMM algorithm
$\bar{\mathbf{e}}$	Normalization vector
$\mathbf{F}_{CV}$	State transition matrix of the CV model
$\mathbf{F}_{CT}$	State transition matrix of the CT model

$\mathbf{F}_{3CTR}$	State transition matrix of the 3CTR model
E_{TH}	Threshold value
T	Sampling period
R	Radius of the turn
v	Speed of the target
A	Area of surveillance region
P_D	Probability of detection
z_ξ	Measurement position in x direction
z_η	Measurement position in y direction
I	Total number of targets
J	Total number of the gated-measurements
L	Total number of the hypotheses
$\tilde{\mathbf{z}}_i(k)$	Predicted position of the ith target
$\hat{\mathbf{z}}_i(k)$	Estimated position of the ith target
$\mathbf{z}_j(k)$	Measurement j received at the kth time step
h	Target-measurement association hypothesis
$\mathbf{Pr}_{h_l}$	Probability of lth hypothesis
$\mathbf{Pr}_{h_l n}$	Normalized probability of lth hypothesis
$d_{ij}(k)$	Mahalanobis distance between $\mathbf{z}_j(k)$ and $\tilde{\mathbf{z}}_i(k)$
$o_\xi(k)$	Center position in x direction
$o_\eta(k)$	Center position in y direction
$w_{ij}(k)$	Modified weight for $\mathbf{z}_j(k)$ being assigned to the ith target
$w_{ic}(k)$	Modified weight assigned to the clutters
$w_{ijn}(k)$	Normalized modified weight of $w_{ij}(k)$
$w_{icn}(k)$	Normalized modified weight of $w_{ic}(k)$
α	Multiplying factor
$\boldsymbol{\beta}(k)$	Innovation or residue vector of KF
$\boldsymbol{\beta}_{ij}(k)$	Residual between $\mathbf{z}_j(k)$ and $\tilde{\mathbf{z}}_i(k)$ of the target i

γ	Gate threshold
$\mathbf{\Gamma}$	Disturbance matrix
$\delta_{ij}(k)$	Adjustment parameter for the weight $\delta_{ij}(k)$
$\delta_{ijn}(k)$	Normalization of the parameter $\delta_{ij}(k)$
$\boldsymbol{\epsilon}_{ij}(k)$	Difference between a measurement j and the predicted position of the target i
ε_{il}	Target detection indicator
ξ	Position in x direction
$\dot{\xi}$	Velocity in x direction
$\hat{\xi}$	Estimated position in x direction
η	Position in y direction
$\dot{\eta}$	Velocity in y direction
$\hat{\eta}$	Estimated position in y direction
θ	Approaching angle of a target
λ	Intensity of the clutter
$\Lambda_j(k)$	Likelihood of the filter j
$\boldsymbol{\mu}(k-1)$	Mode probability vector
$\boldsymbol{\mu}(o)$	Initial mode probability vector
$\boldsymbol{\mu}_{i\|j}(k-1)$	Mixing probability matrix
$\varpi_{ij}(k)$	Weight of the measurement $\mathbf{z}_j(k)$ being assigned to the ith target
$\varpi_{ic}(k)$	Assigned weight the clutters
$\varpi_{ijn}(k)$	Normalization of the weight ϖ_{ij}
ρ_{ji}	Element in $\mathbf{\Omega}(k)$ matrix
σ_Q	Standard deviation of the process noise in the x and y directions
σ_s	Specified standard deviation of the measurement noise in the x and y directions
σ_a	Actual standard deviation of the measurement noise in the x and y directions
τ_{jl}	Measurement association indicator
ϕ	Number of unassigned measurements
ψ	Angle between two targets at the crossing point

$\mathbf{\Omega}(k)$	Validation matrix
$\omega(k)$	Turn rate
ω_{CT_L}	Left turning rate
ω_{CT_R}	Right turning rate
$\omega_{imp}(k)$	Improved turn rate

Chapter 1

Introduction

1.1 General

Target tracking is crucial in monitoring and controlling air traffic in civilian and military applications. Target tracking is a process of determining the position of a moving target as a function of time. Radar systems are used to search for the existence of targets in its region of coverage by providing a set of measurements on the target's position in space [1]. A radar system basically consists of a transmitter that transmits an electromagnetic wave, which travels through the space to the target. The radar antenna receives reflected signals from targets of interest as well as undesired signals, termed as clutters [2], reflected from buildings, ground, or sea. These reflected signals are sent to the receiver of the radar for processing [3]. The output of the receiver is a set of position measurements displayed on the radar display, and sent to a data processor for further processing in determining the tracks of the targets of interest using a tracking algorithm [4].

In order to design a tracking algorithm using radar measurements, we have to estimate the target kinematics such as position, velocity, and acceleration [5]. The coordinate system used in the tracking algorithm can be either Cartesian or polar. The former allows the use of linear filters such as Kalman filter (KF) [6–8], or nonlinear filters such as extended Kalman filter (EKF) [9–11], unscented Kalman filter (UKF) [12–14], cubature Kalman filter (CKF) [15–17], or particle filter (PF) [18–20], depending on whether the motion model used linear or nonlinear. On the other hand, the polar coordinate system requires to the filters to be nonlinear [21]. It is to be noted that the linear models and filters are preferred in view of

its lower computational time. In the next section, a brief review of the existing single and multiple target tracking algorithms is discussed.

1.2 A Review of Tracking Algorithms

Many algorithms have been proposed in the literature for target tracking. In this section, we first review the algorithms that have been proposed for tracking single target, and then for the multiple targets.

There are two main approaches to track a single maneuvering target. The first approach uses a maneuver detector [22–24] employing a statistical test [25]; the detector determines the times at which a maneuver starts and ends. An algorithm in this approach uses a single filter for tracking. The algorithm starts to track the target in a non-maneuvering mode, and switches to the maneuvering mode when the beginning of maneuver is detected, and switches back to the non-maneuvering mode when the maneuver ends. One of the algorithms in this approach is the one in [26] that employs a constant velocity (CV) motion model in its filter in which a low-level noise is used for non-maneuvering mode and a high-level noise for maneuvering mode. Another algorithm in this approach is the variable dimension (VD) algorithm [27]. This algorithm also uses a single filter, but employs two motion models, the CV model for the non-maneuvering mode, and the constant acceleration (CA) model for the maneuvering mode. Two other algorithms in this approach, namely those in [28] and [29], also use a single filter by employing a CV motion model for both the modes, but switch from Cartesian to polar coordinates for the maneuvering mode. In general, the algorithms in this approach have high peak position error at the beginning of a maneuver, and this error may become even higher than that obtained using the raw measurement data.

The second approach in target tracking is the one in which no maneuver detector is used. In this approach, a number of filters, each using a different motion model, are used in parallel. The outputs of these filters are weighted to obtain the overall output state estimate for the target of interest. The weight assigned to an individual filter output is proportional to the probability of its likelihood to estimate the target states accurately [30]. In this approach, the

problem of peak position error that appears at the beginning of the target maneuver using the algorithms of the first approach is overcome. In this approach, several algorithms have been proposed, such as the first order generalized pseudo-Bayesian (GPB1) algorithm [31], the second order GPB (GPB2) algorithm [32], and the interacting multiple model (IMM) algorithm [33]. The IMM algorithm provides better performance compared to that of GPB1, but with a slightly higher complexity. On the other hand, it has significantly lower complexity than GPB2 and with a comparable tracking performance [34]. The implementation of the IMM algorithm with larger number of filters does not necessarily improve the performance proportionally, despite its increased complexity [35]. The IMM algorithm provides a good performance when it employs three filters, one of which uses the constant velocity (CV) model and the other two use the coordinated turn (CT) model with known turn rates [36, 37]. However, the tracking performance deteriorates when the turn rates deviate from the maneuver performed by the target [38].

There exist a few methods for determining the turn rate when the turn rate is not known *a priori*. The first method obtains the turn rate by estimating the target acceleration and speed [39]. In general, the estimated accelerations are not precise and may cause an estimation of the turn rate, which is biased. In the second method, the state vector is augmented with the turn rate, which is estimated along with other components of the state vector [40, 41]. This results in a nonlinear CT model, which is computationally more expensive for estimating the state vector. In the recent years, it has been shown that the range rate information can be used for a better estimation of the turn rate [42–46]. However, this information may not be available in all radar types [38].

If clutters are taken into consideration, steps known as gating and data association are required in the algorithm to track a single target [47, 48]. Gating in a tracking algorithm refers to an area around the predicted position of the target at a time instant such that only the measurements received at that time instant falling inside that area are taken into consideration. Data association is the task of assigning each of the gated measurements a weight, which represents the probability of the measurement being associated with the target. The assigned measurements are then used to estimate the updated position of the target in the

current time instant. For this purpose, the nearest neighbor (NN) and probabilistic data association (PDA) techniques have been used. In the algorithm that uses NN technique [49], the measurement whose position is nearest to the predicted position of the target is the selected to estimate the position of the target and the remaining measurements are simply ignored. In the algorithm using the PDA technique [50,51], for each gated measurement, the estimate of the target position is first individually computed. Then, the overall estimate of the target position is computed as a weighted sum of all the individual estimates, in which a weight represents the probability of associating a measurement to the target. The tracking algorithm using PDA gives a tracking performance better than that of using NN in a heavy clutter environment [52]. It should be pointed out that these algorithms assume that the target is non-maneuvering. In order to track a maneuvering target in clutter, a combination of the PDA and IMM algorithms has been used in [53]. However, in this algorithm the turn rate assumed to be known.

In tracking multiple targets, all the target tracks need to be updated simultaneously. As in the case of single target tracking in the presence of clutters, gating and data association are essential steps of the tracking algorithm. In the literature, there exist different algorithms for tracking multiple targets. The simplest algorithm [54] uses the nearest neighbor technique in which the measurement within the gate of a target that is closest to the predicted position of the target is associated with it and the remaining measurements are ignored. However, the tracking performance of this algorithm degrades or may even fail when the clutter is heavy [55]. Another multiple target tracking algorithm is the one that uses multiple hypothesis tracking (MHT) technique [56,57], which employs the measurements received in all the scans up to the current one to construct all the possible hypotheses for target-measurement assignment. This technique exhibits a very good tracking performance even in the case of heavy clutter environment, but at the expense of a much larger computational complexity [58].

Another algorithm for multi-target tracking, known as the joint probability data association (JPDA) algorithm, with a complexity much lower than that of MHT, has been proposed in [59]. This algorithm computes the predicted position of the targets at a time instant using

the output estimate from the previous time instant, and creates a gate around the predicted position for each of the targets to find the gated measurements. The position estimate of a given target is first obtained individually for all the measurements falling inside its gate. Then, the overall position estimate for the target is computed as a weighted sum of all the individual estimates for the target [60], in which a weight represents the probability of associating a particular measurement to the target. The JPDA algorithm exhibits a good tracking performance when targets are widely spaced. When the targets are closely spaced, not only the performance of the JPDA algorithm deteriorates, but also even the tracks may coalesce [61, 62]. Many researchers have attempted to overcome this problem by proposing algorithms such as Kullback-Leibler set JPDA (KLSJPDA), Set JPDA (SJPDA) [63, 64] and multi-objective JPDA (MOJPDA) [65]. However, in addition to suffering from high computational complexity, these algorithms are not capable of preserving the identity of the targets. The above multi-target tracking algorithms assume that the targets are non-maneuvering. In order to track multiple maneuvering targets, a combination of the JPDA and IMM algorithms has been used in [66]. However, this algorithm uses a nonlinear CT model, which is computationally more expensive for estimating the position of the target.

From the above review of the literature in target tracking, it is seen that that the methods proposed for both single and multi-target tracking have certain limitations and disadvantages. In the case of single target tracking, most of the algorithms have been designed for specified turn rate for target maneuvering, and the performance of such algorithms deteriorates, and may even fail when the target maneuvers with a turn rate that is larger than that for which the algorithm has been designed. Even in algorithms in which the turn rate is estimated rather than being specified in advance, they cannot deal effectively for large values of the turn rate, or require the use of non-linear filters that adds to the complexity of the algorithm.

The most effective algorithm for multi-target tracking is JPDA, which provides very good tracking performance when the targets are widely spaced and the angles of sudden turns of the targets are not very large. No study has been undertaken of the impact on the performance of multi-target tracking algorithms, when the targets are closely spaced and/or when

the target sudden turns are large.

1.3 Objectives and Organization of the Thesis

Objectives of this thesis are to develop efficient algorithms for single target tracking, in which the turn rates of the target maneuver are not prespecified, and for multi-target tracking when the tracks of the targets are closely spaced as well as when the angles of sudden turns of the targets are large. In the first part of the thesis, two algorithms within the IMM framework are proposed to track a single maneuvering target, when the target turn rate is not known a priori. In both the algorithms, the turn rate is dynamically estimated using noisy measurements. In the first algorithm, the turn rate at each time instant k is estimated based on the target speed and the radius of the circle formed by the measurement at that instant and the two previous consecutive noisy measurements, $(k-1)$ and $(k-2)$. The segment of this circle covered by these three noisy measurements is used to model the true track of the target at the instant time k. In the second algorithm, the accuracy of the turn rate estimated in the first algorithm is improved using the information on the level of the measurement noise. In the second part of the thesis, a new multiple target tracking algorithm within JPDA framework is proposed that addresses the problems resulting from close spacing between the target tracks or sudden large turn angles, and is based on the spatial distribution of the gated measurements.

The thesis is organized as follows. In Chapter 2, the background material necessary for the development of the proposed schemes, is presented. In Chapter 3, an algorithm is proposed for tracking a single maneuvering target, when the turn rate is estimated without an *a priori* knowledge of the turn rate. Extensive experiments are performed under different scenarios of the target track to evaluate the performance of the algorithm and to compare its performance with that of the existing algorithms. The effect of the change in the level of measurement noise on the tracking performance is also studied. In Chapter 4, an algorithm is proposed in order to improve the accuracy of the estimated turn rate using the information on the level of the measurement noise. The impact on the tracking performance of this

algorithm, when the measurement noise level rises over the specified one for which the algorithm has been designed, is studied. In Chapter 5, a new algorithm for tracking multiple targets within the JPDA framework, is proposed. A systematic study on the performance of this algorithm as a function of the spacing between the targets as well as the turning angles of the targets is carried out. Experiments are performed to illustrate the effectiveness of the proposed scheme using various scenarios of target tracks and compare its performance with that of the existing JPDA algorithm. Finally, Chapter 6 represents some concluding remarks on the work carried out in this thesis and some possible research directions in target tracking are suggested.

Chapter 2

Background

Target tracking is essentially a process of following the evolution of the target motion with time. This process is generally represented by a state space model, which has two parts. The first part gives the evolution of the target state over time, whereas the second part relates the target state with the measurement. Since moving targets may have different kind of motions, these motions have to be modeled in order to accurately follow the evolution of the target state. In this chapter, the state space model is first described, and then some of the target motion models used in target tracking are presented. Finally, metrics to evaluate the performance of a tracking algorithm are briefly discussed.

2.1 State Space Model

Despite the fact that a real target is not a point in the space, it is conventionally treated as a point object without a shape in tracking applications [67]. For modeling the target movement over time and relate the target state with the measurement, the state space model with additive noise is used. We first introduce the state vector $\mathbf{x}(k)$ of a target, given by

$$\mathbf{x}(k) = [\xi(k), \dot{\xi}(k), \eta(k), \dot{\eta}(k)]^t \tag{2.1}$$

where $\xi(k)$ and $\eta(k)$ represent the position components of the target in the x and y directions, respectively, $\dot{\xi}(k)$ and $\dot{\eta}(k)$ are the velocity components of the target in the x and y directions, respectively, $(\cdot)^t$ denotes the transpose, and k is the time step. The state space model consists

of two parts. The first part is the dynamic equation, in which the evolution of the target state with respect to time is computed. The dynamic equation is given by [68]

$$\mathbf{x}(k) = \mathbf{F}\mathbf{x}(k-1) + \mathbf{\Gamma}\mathbf{b} \tag{2.2}$$

where $\mathbf{F}$ is the state transition matrix that relate the previous state vector to the current state vector, $\mathbf{\Gamma}$ is the disturbance matrix, and $\mathbf{b}$ is a vector containing the x and y components of a zero-mean Gaussian noise with a covariance matrix $\mathbf{Q}$. The matrices $\mathbf{\Gamma}$ and $\mathbf{Q}$ are given by [67]

$$\mathbf{\Gamma} = \begin{bmatrix} 0.5T^2 & 0 \\ T & 0 \\ 0 & 0.5T^2 \\ 0 & T \end{bmatrix} \tag{2.3}$$

and

$$\mathbf{Q} = \begin{bmatrix} \sigma_Q^2 & 0 \\ 0 & \sigma_Q^2 \end{bmatrix} \tag{2.4}$$

where T is the sampling period and σ_Q is the standard deviation of the process noise in the x and y directions. Generally, 2D-radar systems provide the measurements in polar coordinates, that is, in terms of azimuth and range. However, tracking using polar coordinates is not as accurate as tracking using Cartesian coordinates and hence the latter is recommended with a suitable measurement conversion [69,70]. Therefore, using a suitable conversion of the polar measurement such as used in [71], we can track the target in Cartesian coordinates for better tracking performance. Therefore, most of the algorithms that have been proposed for target tracking assume that suitable conversion has already been made of the polar measurement to Cartesian coordinates [37,39,72,73]. We have also followed the same practice as others have done. The radar assumed to provide only the positions information, i.e, range rate information is not provided. Then, the second equation of the state space model that is

the measurement equation, given by [74]

$$\mathbf{z}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{m} \tag{2.5}$$

where $\mathbf{z}(k)$ is the measurement received, $\mathbf{H}$ is the measurement matrix that relate the current state vector to the current received measurement, and $\mathbf{m}$ is the measurement noise, assumed to be zero-mean Gaussian with a covariance matrix $\mathbf{U}$. The matrices $\mathbf{H}$ and $\mathbf{U}$ given by [75]

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.6}$$

and

$$\mathbf{U} = \begin{bmatrix} \sigma_s^2 & 0 \\ 0 & \sigma_s^2 \end{bmatrix} \tag{2.7}$$

where σ_s is the specified standard deviation (STD) of the measurement noise in the x and y directions that is assumed to be equal to the actual STD, σ_a of the measurement noise. It is to be noted that both the process noise and the measurement noise are independent.

2.2 Target Motion Models

An aircraft has two types of motion: straight-line motion and maneuvering motion [76]. The straight-line motion can be modeled using a constant velocity (CV) model, the maneuver motion is customarily modeled using a coordinated turn (CT) model [77] or a three components turning rate (3CTR) model [39]. The selection of the motion model is based on the design of the tracking algorithm, i.e., the tracking algorithm is designed to track only targets with non-maneuvering motion or targets may perform maneuvers as well.

2.2.1 Constant velocity model

This model assumes that the target has a straight-line motion with a constant velocity in a two-dimensional space. The state transition matrix $\mathbf{F} = \mathbf{F}_{CV}$ of this model, given by [67]

$$\mathbf{F}_{CV} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

2.2.2 Coordinated turn model with known turn rate

This model assumes that the target moves at a constant speed and is preforming a maneuver with a constant turn rate ω in a two-dimensional space. The state transition matrix $\mathbf{F} = \mathbf{F}_{CT}$ of this model, given by [77]

$$\mathbf{F}_{CT} = \begin{bmatrix} 1 & \frac{\sin(\omega T)}{\omega} & 0 & -\frac{1-\cos(\omega T)}{\omega} \\ 0 & \cos(\omega T) & 0 & -\sin(\omega T) \\ 0 & \frac{1-\cos(\omega T)}{\omega} & 1 & \frac{\sin(\omega T)}{\omega} \\ 0 & \sin(\omega T) & 0 & \cos(\omega T) \end{bmatrix} \tag{2.9}$$

where ω is the target turn rate. When the sign of ω is positive, the model account for the left turning motion. Conversely, when the sign of ω is negative, the model account for the right turning motion [78]. In case the turn rate of the target is not known *a priori* and estimated, the turn rate varies each time step based on the estimation scheme and becomes $\omega(k)$.

2.2.3 Three component turn rate model

This model assumes that the target performs the maneuver motion with a constant turn rate in a 2D plane and use the target's position, velocity and acceleration to estimate the target

position. In this case, the state vector $\mathbf{x}(k)$ is expressed as

$$\mathbf{x}(k) = [\xi(k), \dot{\xi}(k), \ddot{\xi}(k), \eta(k), \dot{\eta}(k), \ddot{\eta}(k)]' \tag{2.10}$$

where $\ddot{\xi}(k)$ and $\ddot{\eta}(k)$ are the acceleration components in the x and y directions, respectively. The state transition matrix $\mathbf{F}_{3CTR}$ of this model is given by [39]

$$\mathbf{F}_{3CTR} = \begin{bmatrix} \mathbf{A}_f & 0 \\ 0 & \mathbf{A}_f \end{bmatrix} \tag{2.11}$$

where

$$\mathbf{A}_f = \begin{bmatrix} 1 & \omega^{-1}\sin(\omega T) & \omega^{-2}(1-\cos(\omega T)) \\ 0 & \cos(\omega T) & \omega^{-1}\sin(\omega T) \\ 0 & -\omega\sin(\omega T) & \cos(\omega T) \end{bmatrix} \tag{2.12}$$

and its corresponding disturbance matrix denoted as

$$\mathbf{\Gamma}_f = \begin{bmatrix} 0.167T^3 & 0 \\ 0.5T^2 & 0 \\ T & 0 \end{bmatrix} \tag{2.13}$$

2.3 Metrics for Performance Evaluation

To evaluate the performance of a tracking algorithm, Monte-Carlo simulations are used, and the following subsections show the metrics used for performance evaluation.

2.3.1 Tracking accuracy

To measure the tracking accuracy of a single or multiple target tracking algorithm, the root mean square error (RMSE) is used, which is given by [79]

$$RMSE(k) = \sqrt{\frac{1}{M}\sum_{m=1}^{M}(\hat{\xi}_m(k) - \xi_m(k))^2 + (\hat{\eta}_m(k) - \eta_m(k))^2} \tag{2.14}$$

where M is the total number of Monte-Carlo simulations, $\xi_m(k)$ and $\hat{\xi}_m(k)$ are, respectively, the true and estimated positions in the x direction, and $\eta_m(k)$ and $\hat{\eta}_m(k)$ are, respectively, the corresponding positions in the y direction, in the mth run of the Monte-Carlo simulation. The estimator performance is acceptable at the time step k, if the value of RMSE is less than the error obtained from the raw measurement of the radar, given by [72]

$$E_{TH} = \sigma_s \sqrt{2} \tag{2.15}$$

2.3.2 Consistency test

This test checks whether the state vector error is compatible with the estimated state covariance matrix of the estimator or not, and the state vector error defined as the difference between the true and estimated state of the target. The test is performed by calculating the average normalized estimation error squared (ANEES) and commonly used for the evaluation of a single target tracking algorithm [26]. The ANEES at the kth time step is given by

$$ANEES(k) = \frac{1}{Mn_x} \sum_{m=1}^{M} (\mathbf{x}_m(k) - \hat{\mathbf{x}}_m(k))^t (\hat{\mathbf{P}}_m(k))^{-1} (\mathbf{x}_m(k) - \hat{\mathbf{x}}_m(k)) \tag{2.16}$$

where $\mathbf{x}_m(k)$, $\hat{\mathbf{x}}_m(k)$, and $\hat{\mathbf{P}}_m(k)$ are the true, the estimated state vector, and the error covariance matrix of the target at the time step k on the mth Monte Carlo run, respectively, and n_x is the dimension of the state vector. The logarithmic ANEES (LANEES) is given by

$$LANEES(k) = \log_{10}(ANEES(k)) \tag{2.17}$$

is more expressive in terms of displaying the results [80]. When the value of LANEES test is equal or less than zero, the estimator is said to be consistent.

2.4 Summary

In this chapter, a brief review on the background material necessary for the development of the work in this thesis has been presented. The state space model used for target tracking has been introduced. Some of the target motion models that exist in the literature are also presented for a maneuvering target. Finally, the metrics generally used for the evaluation of the tracking performance have been presented.

Chapter 3

The First Proposed Algorithm for Tracking a Single Maneuvering Target

3.1 Introduction

To track a maneuvering target with an efficient tracking performance, the tracking algorithm has to precisely identify, when the target starts the maneuver motion, when the target ends that motion, and to be able to track the target during the maneuvering period. Moreover, the adopted motion models have to accurately match the predicted motion of the target and provide an accurate predicted kinematics as possible [21].

Two main approaches exist to track a maneuvering target; one of which use a maneuver detector and the other does not. The maneuver detector is a statistical test, which is formulated to decide whether the target maneuver has begun or not. Same framework is used to test whether the maneuver has ended or not [22]. In general, the algorithms in the first approach show a high peak position error at the beginning of the maneuver motion, which may be higher than the error obtained from the raw measurement data [27]. The other approach uses a number of filters that works in parallel. This approach overcomes the drawback in the first approach. The best tracking algorithm in this approach is achieved by the interacting multiple model (IMM) algorithm when it utilizes the CV and CT models with a prior knowledge about the target turn rate. However, when a target performs a maneuver with a turn rate that is larger than that of the one used in the design of the IMM

algorithm, the performance deteriorates, and track loss may happen. Turn rate estimation is an approach to overcome this problem. The only algorithm that estimates the turn rate using linear motion models and filters is the one given in [39]. In this algorithm, the IMM framework is used and the turn rate of the target is calculated from the estimated velocity and acceleration at each time step as the magnitude of the acceleration divided by the speed of the target. However, this algorithm shows accurate performance for only mild maneuvers due to the error in the calculated acceleration of the target when the maneuver is not mild. Another problem with the algorithm in [39] that needs to be pointed out is that the level of the actual measurement noise of the radar system may not remain the same as the ones used in the design of the algorithm. No study has been undertaken on the impact of the change in the level of the measurement noise from the specified one used in the design of the algorithm on the tracking performance.

In this chapter, we first review the framework of an IMM algorithm [36]. Then, we propose an algorithm within the IMM framework to track a single maneuvering target in which the turn rate is dynamically estimated. The turn rate is estimated at each time instant based on the target speed and the radius of the circle formed by the measurement at that instant and the two previous consecutive noisy measurements [81]. Extensive simulations are carried out to evaluate the performance of the proposed algorithm and compare it with that of the other algorithms designed in the IMM framework. Finally, we study the impact of the change in the level of the measurement noise on the tracking performance of the algorithm in [39] and that of the proposed algorithm.

3.2 IMM Algorithm

The IMM algorithm is quite effective in tracking a maneuvering target, wherein the target maneuver is modeled as a combination of different motion models. The output at each time step is a combination of the outputs of all the filters weighted by the corresponding mode probability, the mode probability being defined as the probability of the model matching the target motion. At each time step, the algorithm decides as to which model is suitable

to predict the motion of the target (non-maneuvering or maneuvering motion) based on the mode probability. The block diagram of a single cycle of the IMM algorithm, which uses two Kalman filters each using a different motion model, is shown in Fig. 3.1 [36].

The predicted state vector $\tilde{\mathbf{x}}_j(k)$ for a Kalman filter j is given by [7]

$$\tilde{\mathbf{x}}_j(k) = \mathbf{F}\hat{\mathbf{x}}_{oj}(k-1) \tag{3.1}$$

where $\hat{\mathbf{x}}_{oj}(k-1)$ is the input state vector for the filter j. The predicted error covariance matrix, $\tilde{\mathbf{P}}_j(k)$, is given by

$$\tilde{\mathbf{P}}_j(k) = \mathbf{F}\hat{\mathbf{P}}_{oj}(k-1)\mathbf{F}^t + \mathbf{\Gamma Q \Gamma}^t \tag{3.2}$$

where $\hat{\mathbf{P}}_{oj}(k-1)$ is the error covariance matrix of the input state vector for the filter j. The predicted position of the target, $\tilde{\mathbf{z}}_j(k)$ is given by

$$\tilde{\mathbf{z}}_j(k) = \mathbf{H}\tilde{\mathbf{x}}_j(k) \tag{3.3}$$

The measurement innovation or the residual of the filter $\boldsymbol{\beta}_j(k)$

$$\boldsymbol{\beta}_j(k) = \mathbf{z}(k) - \tilde{\mathbf{z}}_j(k) \tag{3.4}$$

where $\mathbf{z}(k)$ is the received measurement vector. The innovation error covariance matrix, $\mathbf{S}_j(k)$, is given by

$$\mathbf{S}_j(k) = \mathbf{H}\tilde{\mathbf{P}}_j(k)\mathbf{H}^t + \mathbf{U} \tag{3.5}$$

The Kalman gain $\mathbf{K}_j(k)$ is defined as

$$\mathbf{K}_j(k) = \tilde{\mathbf{P}}_j(k)\mathbf{H}^t(\mathbf{S}_j(k))^{-1} \tag{3.6}$$

The output state estimate $\hat{\mathbf{x}}_j(k)$ of the Kalman filter j and its error covariance matrix $\hat{\mathbf{P}}_j(k)$ are expressed as

$$\hat{\mathbf{x}}_j(k) = \tilde{\mathbf{x}}_j(k) + \mathbf{K}_j(k)\boldsymbol{\beta}_j(k) \tag{3.7}$$

$$\hat{\mathbf{P}}_j(k) = [\mathbf{I} - \mathbf{K}_j(k)\mathbf{H}]\tilde{\mathbf{P}}_j(k) \tag{3.8}$$

where $\mathbf{I}$ is the identity matrix.

Assuming the IMM algorithm to contain only two Kalman filters, each of which is used to find the output state estimate of the target according to a specific type of motion (e.g. non-maneuvering, referred to as mode 1, and maneuvering mode, referred to as mode 2). The algorithm starts with initial values for the mode probabilities $\mu_1(k-1)$ and $\mu_2(k-1)$. In addition, a fixed matrix $\mathbf{P}_{tr}$ is assumed, whose (i,j)th element, p_{ij}, $i,j = 1,2$, is the probability of transition from mode i to mode j, wherein the sum of the elements of each row in this matrix is unity. The mode probabilities as well as the transition probabilities are used to calculate the mixing probabilities $\mu_{i|j}(k-1)$ where $i,j = 1,2$. These mixing probabilities along with the filter outputs $\hat{\mathbf{x}}_j(k-1), j = 1,2$ of the previous cycle are utilized to calculate the inputs to the filters, $\hat{\mathbf{x}}_{oj}(k-1), j = 1,2$. In other words, each filter input is calculated as a weighted sum of all the filter outputs of the previous cycle. When the new measurement $\mathbf{z}(k)$ is received, each filter updates its output state, $\hat{\mathbf{x}}_1(k)$ and $\hat{\mathbf{x}}_2(k)$. Then, the likelihood probabilities of the filters, $\Lambda_1(k)$ and $\Lambda_2(k)$, are computed from the measurement innovation $\boldsymbol{\beta}(k)$, and the corresponding covariance matrix $\mathbf{S}(k)$ for each filter. Next, the mode probabilities are updated to $\mu_1(k)$ and $\mu_2(k)$, using $\Lambda_1(k)$ and $\Lambda_2(k)$, $\mu_1(k-1)$ and $\mu_2(k-1)$, and $\mathbf{P}_{tr}$. Finally, the output state estimate $\hat{\mathbf{x}}(k)$ is computed as a weighted sum of the outputs of the filters using the weights of the updated mode probabilities, $\mu_1(k)$ and $\mu_2(k)$. The outputs of the filters and the updated mode probabilities are set as the initial data for the next cycle. More details about the IMM algorithm can be found in [82]. The steps of the IMM algorithm with j filters are as follows

1. **State interaction**: In this step, the previous state estimates and their covariance matrices are mixed using the calculated mixing probabilities $\mu_{i|j}(k-1)$. The input state vector of the jth filter, $\hat{\mathbf{x}}_{oj}(k-1)$ and its covariance matrix $\hat{\mathbf{P}}_{oj}(k-1)$ are calculated as

$$\hat{\mathbf{x}}_{oj}(k-1) = \sum_{i=1}^{r} \hat{\mathbf{x}}_i(k-1)\mu_{i|j}(k-1) \qquad\qquad i,j = 1,\cdots,r \tag{3.9}$$

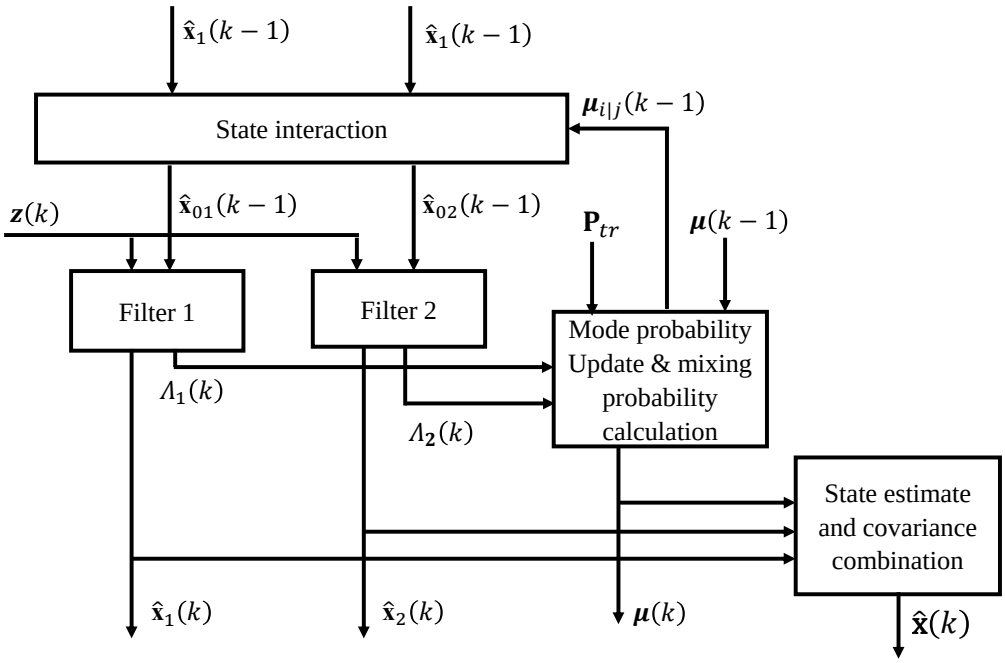


Figure 3.1: One cycle of the IMM algorithm when $r = 2$ [36].

and

$$\hat{\mathbf{P}}_{oj}(k-1) = \sum_{i=1}^{r} \mu_{i|j}(k-1)[\hat{\mathbf{P}}_i(k-1) + [\hat{\mathbf{x}}_i(k-1) - \hat{\mathbf{x}}_{oj}(k-1)] \\ [\hat{\mathbf{x}}_i(k-1) - \hat{\mathbf{x}}_{oj}(k-1)]^t], \quad i,j = 1,\cdots,r \tag{3.10}$$

where

$$\mu_{i|j}(k-1) = \frac{1}{\bar{e}_j} p_{ij}\mu_i(k-1), \qquad i,j = 1,\cdots,r \tag{3.11}$$

with the normalization constant

$$\bar{e}_j = \sum_{i=1}^{r} p_{ij}\mu_i(k-1), \qquad i,j = 1,\cdots,r \tag{3.12}$$

2. **Mode probability update**: when the measurement $\mathbf{z}(k)$ is received, each Kalman filter uses its input state and its error covariance matrix to calculate its output state $\hat{\mathbf{x}}_j(k)$ and its error covariance matrix $\hat{\mathbf{P}}_j(k)$. Moreover, both the innovation $\boldsymbol{\beta}_j(k)$ and its error covariance matrix $\mathbf{S}_j(k)$ [7] are used to calculate the likelihood of each filter, which is

given by

$$\Lambda_j(k) = \frac{1}{\sqrt{2\pi \mathbf{S}_j(k)}} \exp[-0.5(\boldsymbol{\beta}_j(k))^t (\mathbf{S}_j(k))(\boldsymbol{\beta}_j(k))], \qquad j = 1, \cdots, r \tag{3.13}$$

Then, the mode probability update for the jth filter is computed as

$$\mu_j(k) = \frac{1}{G} \Lambda_j(k) \bar{e}_j, \qquad j = 1, \cdots, r \tag{3.14}$$

and

$$G = \sum_{j=1}^{r} \Lambda_j(k) \bar{e}_j \tag{3.15}$$

3. **Fusion of the outputs**: The output state estimate $\hat{\mathbf{x}}(k)$ and its error covariance matrix $\hat{\mathbf{P}}(k)$ are computed as a fusion of all the filter output states and their covariance matrices weighted by the updated mode probabilities, respectively.

$$\hat{\mathbf{x}}(k) = \sum_{j=1}^{r} \hat{\mathbf{x}}_j(k) \mu_j(k) \tag{3.16}$$

and

$$\hat{\mathbf{P}}(k) = \sum_{j=1}^{r} \mu_j(k) [\mathbf{P}_j(k) + [\hat{\mathbf{x}}_j(k) - \hat{\mathbf{x}}(k)][\hat{\mathbf{x}}_j(k) - \hat{\mathbf{x}}(k)]^t] \tag{3.17}$$

3.3 First Proposed Algorithm

To avoid the calculation of the target acceleration, there exist another formula to calculate the turn rate of the target can be used. This formula considers that the target is moving in a circular path around a fixed center at a speed $v(k)$ in a 2-D space and requires the knowledge of the radius $R(k)$ of this circular path. The turn rate $\omega(k)$ formula is given by [83, 84]

$$\omega(k) = \frac{v(k)}{R(k)} \tag{3.18}$$

The speed $v(k)$ of the target can be calculated at each time step from the estimated velocity components in the output state estimate vector given by

$$v(k) = \sqrt{\dot{\xi}^2(k) + \dot{\eta}^2(k)} \tag{3.19}$$

where $\dot{\xi}(k)$ and $\dot{\eta}(k)$ are the velocity components in the x and y directions. Therefore, our aim is to estimate the radius $R(k)$ of the turn.

3.3.1 Determination of the radius of the turn

The idea behind the proposed scheme for dynamically estimating the turn rate of a target is depicted in Fig. 3.2. This figure shows an example of the actual track along with the noisy measurements. In order to estimate the track at each time instant k, we model the actual track by the segment of a circle formed by using the noisy measurements at time instant k and the two previous noisy measurements at time instants $(k-1)$ and $(k-2)$. The radius of the circle being very large, ideally infinite, is an indication that the target is moving in a straight-line segment. On the other hand, when it is finite and relatively much smaller, it is an indication that the target is performing a maneuver. In order to find the center and the radius of the circle at the time step k, we use the following equation.

$$R(k-2) = R(k-1) = R(k) \tag{3.20}$$

Then, construct two equations such as

$$R(k-2) = R(k-1) \tag{3.21}$$

$$R(k-1) = R(k) \tag{3.22}$$

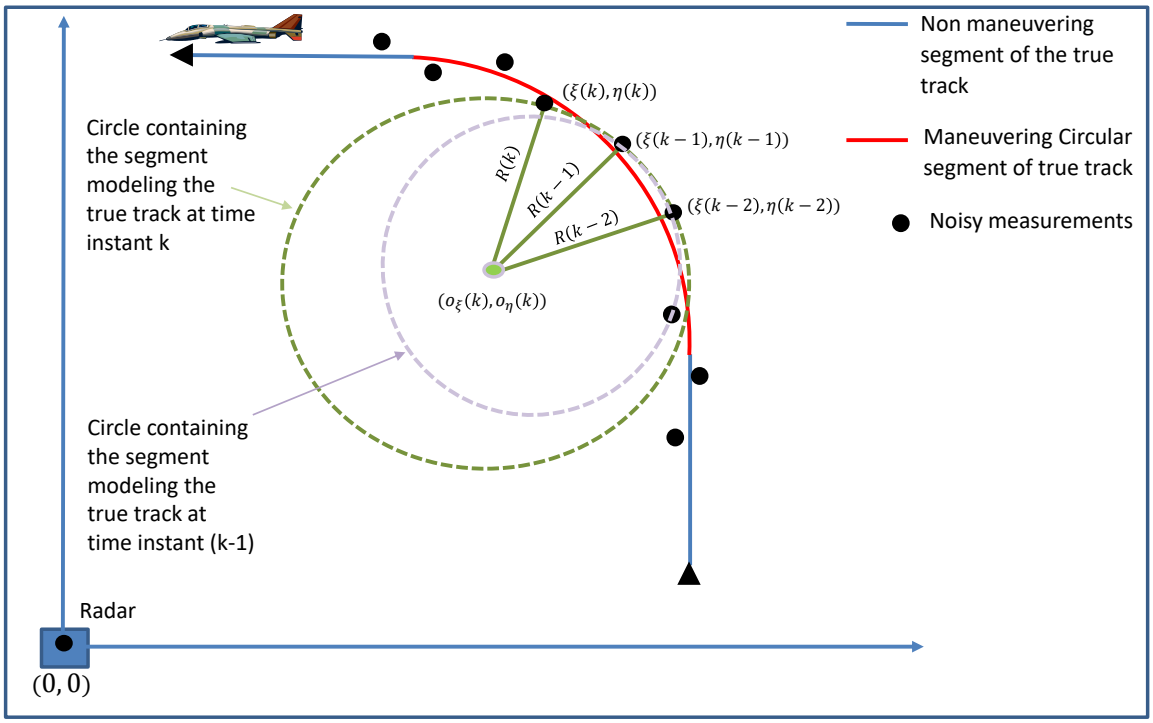


Figure 3.2: An illustration of the scheme used for dynamic estimation of the turn rate using noisy measurements.

From (3.22) we get

$$(\xi(k) - o_\xi(k))^2 + (\eta(k) - o_\eta(k))^2 = (\xi(k-1) - o_\xi(k))^2 + (\eta(k-1) - o_\eta(k))^2 \tag{3.23}$$

or

$$o_\xi(k) = \frac{(\xi^2(k-1) - \xi^2(k)) + (\eta^2(k-1) - \eta^2(k)) - 2o_\eta(k)(\eta(k-1) - \eta(k))}{2(\xi(k-1) - \xi(k))} \tag{3.24}$$

Similarly, from (3.21) we can derive that

$$o_\xi(k) = \frac{(\xi^2(k-2) - \xi^2(k-1)) + (\eta^2(k-2) - \eta^2(k-1)) - 2o_\eta(k)(\eta(k-2) - \eta(k-1))}{2(\xi(k-2) - \xi(k-1))} \tag{3.25}$$

Hence, from (3.24) and (3.25) we get

$$o_\eta(k) = \frac{(\xi^2(k-2)-\xi^2(k-1)+\eta^2(k-2)-\eta^2(k-1))(\xi(k-1)-\xi(k))-(\xi^2(k-1)-\xi^2(k)+\eta^2(k-1)-\eta^2(k))(\xi(k-2)-\xi(k-1))}{2((\eta(k-2)-\eta(k-1)))(\xi(k-1)-\xi(k))-2((\eta(k-1)-\eta(k)))(\xi(k-2)-\xi(k-1))} \tag{3.26}$$

Therefore, from (3.24) and (3.26), we get the center position $(o_{\xi}(k), o_{\eta}(k))$. We now calculate the radius $R(k)$ using $(o_{\xi}(k), o_{\eta}(k))$ and $(\xi(k), \eta(k))$ as the last received information about the target.

$$R(k) = \sqrt{(\xi(k) - o_{\xi}(k))^2 + (\eta(k) - o_{\eta}(k))^2} \tag{3.27}$$

3.3.2 Development of the algorithm

The block diagram of the first proposed tracking is shown in Fig. 3.3. The algorithm uses three Kalman filters, first of which uses the CV model to account for the straight-line motion of the target, and the second and third filters, denoted by CTL and CTR filters, use the CT model to account for the left turning and right turning motions, respectively. For left turns, the estimated turn rate takes a positive sign, while negative values are used for the right turns [78]. A multiplier is located at the input of the CTR filter to change the sign of the turn rate provided to the CT model in that filter to account for the right turn. The three measurements $\mathbf{z}(k)$, $\mathbf{z}(k-1)$, and $\mathbf{z}(k-2)$ are fed to the block that calculates the radius $R(k)$ at the current time step, as shown in Fig. 3.3. Furthermore, the speed of the target is calculated from the output state estimate of the algorithm. This output state vector contains the estimated components for both the position and velocity of the target, where the target speed is calculated using (3.19). The output of the last two mentioned blocks, $R(k)$ and $v(k)$, are then used to calculate the turn rate $\omega(k)$ as in Algorithm 1, which in turn is used in the next time step of the algorithm.

Algorithm 1 Determination of the turn rate $\omega(k)$

1: **Input** : $(\xi(k-2), \eta(k-2))$, $(\xi(k-1), \eta(k-1))$, $(\xi(k), \eta(k))$, and $(\dot{\xi}_k, \dot{\eta}_k)$
2: **Find** : $\omega(k)$
3: **begin**
4: Obtain $o_{\xi}(k)$ using (3.24)
5: Obtain $o_{\eta}(k)$ using (3.26)
6: Obtain $R(k)$ using (3.27)
7: Obtain $v(k)$ using (3.19)
8: Obtain $\omega(k)$ using (3.18)
9: **end**

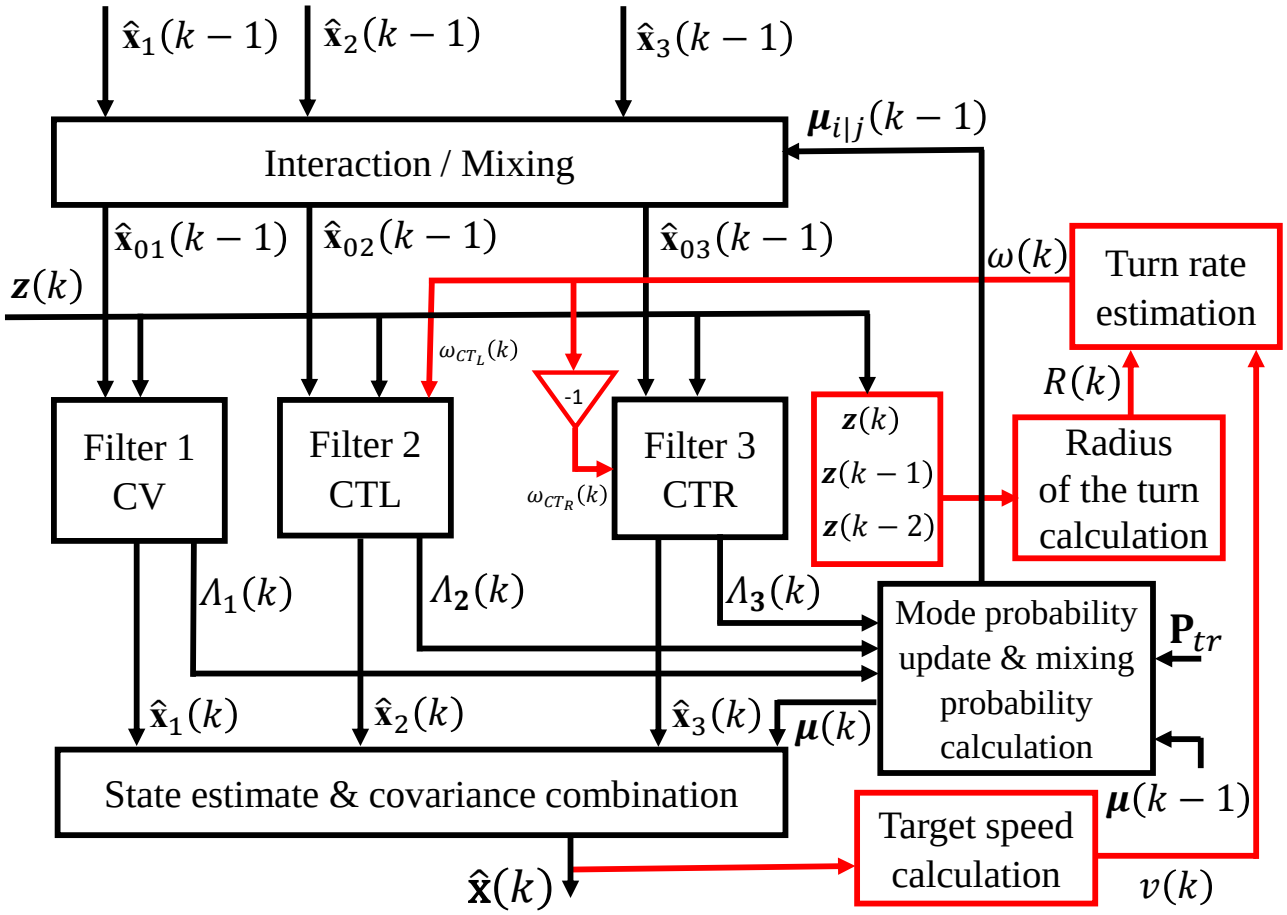


Figure 3.3: Block diagram of the first proposed tracking algorithm

$$\omega_{CT_L}(k) = \omega(k) \tag{3.28}$$

$$\omega_{CT_R}(k) = -\omega(k) \tag{3.29}$$

where $\omega_{CT_L}(k)$ and $\omega_{CT_R}(k)$ are the left and right turn rates, respectively. Once $\omega_{CT_L}(k)$ and $\omega_{CT_R}(k)$ have been determined and the new measurement $\mathbf{z}(k)$ is received, the measurement $\mathbf{z}(k-2)$ is discarded, and the other two previous measurements $\mathbf{z}(k-1)$ and $\mathbf{z}(k)$ are retained as $\mathbf{z}(k-2)$ and $\mathbf{z}(k-1)$, respectively, and the process is repeated. The complete algorithm is given as Algorithm 2.

Algorithm 2 The first proposed tracking algorithm

Input : $\mathbf{P}_{tr}^{PA1}$, $\mathbf{\Gamma}$, $\mathbf{Q}$, $\mathbf{U}$, $\boldsymbol{\mu}(o)$, $\omega_{CT_L}(o)$, $\omega_{CT_R}(o)$, N, and $\mathbf{z}(k)$
Find : $\hat{\mathbf{x}}(k)$ and $\hat{\mathbf{P}}(k)$
Initialization : using $\mathbf{z}(1)$ and $\mathbf{z}(2)$, find $\hat{\mathbf{x}}(o)$ and $\hat{\mathbf{P}}(o)$ $\forall$ KF from (3.32) and (3.33)
At k=3, Set :

$$\begin{aligned}
&\hat{\mathbf{x}}_1(k-1) = \hat{\mathbf{x}}_2(k-1) = \hat{\mathbf{x}}_3(k-1) = \hat{\mathbf{x}}(o)\\
&\hat{\mathbf{P}}_1(k-1) = \hat{\mathbf{P}}_2(k-1) = \hat{\mathbf{P}}_3(k-1) = \hat{\mathbf{P}}(o)\\
&\boldsymbol{\mu}(k-1) = \boldsymbol{\mu}(o)\\
&\omega_{CT_L}(k-1) = \omega_{CT_L}(o)\\
&\omega_{CT_R}(k-1) = \omega_{CT_R}(o)
\end{aligned}$$

begin the process at k = 3 to N
    for j=1:3 **do**
        Obtain $\bar{e}_j$ and $\mu_{i|j}(k-1)$ using (3.12) and (3.11)
        Obtain $\hat{\mathbf{x}}_{oj}(k-1)$ and $\hat{\mathbf{P}}_{oj}(k-1)$ using (3.9) and (3.10)
        Obtain $\hat{\mathbf{x}}_j(k)$ and $\hat{\mathbf{P}}_j(k)$ using (3.7) and (3.8) with $\omega_{CT_R}(k-1)$ and $\omega_{CT_L}(k-1)$
        Obtain $\boldsymbol{\beta}_j(k)$ and $\mathbf{S}_j(k)$ using (3.4) and (3.5)
        Obtain $\Lambda_j(k)$ using (3.13)
        Obtain G and $\mu_j(k)$ using (3.15) and (3.14)
    end for
    Compute $\hat{\mathbf{x}}(k)$ and $\hat{\mathbf{P}}(k)$ using (3.16) and (3.17)
    Compute $\omega(k)$ using Algorithm 1
    Compute $\omega_{CT_L}(k)$ and $\omega_{CT_R}(k)$ using (3.28) and (3.29)
end the process

3.4 Performance Evaluation

In this section, the performance of the proposed algorithm as well as that of the existing linear IMM algorithms are compared and evaluated using the mentioned metrics in Section 2.3.

3.4.1 Simulation parameters

For all the algorithms, let the first two consecutive received measurements be $\mathbf{z}(1)$ and $\mathbf{z}(2)$ given by

$$\mathbf{z}(1) = [z_{\xi}(1), z_{\eta}(1)]^t \tag{3.30}$$

$$\mathbf{z}(2) = [z_{\xi}(2), z_{\eta}(2)]^t \tag{3.31}$$

where $z_{\xi}(1)$ and $z_{\xi}(2)$ are the measured positions in the x direction at time $k = 1, 2$, respectively, while $z_{\eta}(1)$ and $z_{\eta}(2)$ are the measured positions in the y direction at time $k = 1, 2$, respectively. The Kalman filters for these algorithms are initialized by the difference between $\mathbf{z}(1)$ and $\mathbf{z}(2)$ [38]. Then, the initial estimate state vector $\hat{\mathbf{x}}(o)$ is given by

$$\hat{\mathbf{x}}(o) = \left[z_{\xi}(2), \frac{z_{\xi}(2) - z_{\xi}(1)}{T}, z_{\eta}(2), \frac{z_{\eta}(2) - z_{\eta}(1)}{T}\right]^t \tag{3.32}$$

where T is the sampling period. The initial error covariance matrix $\hat{\mathbf{P}}(o)$ is given by

$$\hat{\mathbf{P}}(o) = \begin{bmatrix} \sigma_s^2 & \sigma_s^2/T & 0 & 0 \\ \sigma_s^2/T & 2\sigma_s^2/T^2 & 0 & 0 \\ 0 & 0 & \sigma_s^2 & \sigma_s^2/T \\ 0 & 0 & \sigma_s^2/T & 2\sigma_s^2/T^2 \end{bmatrix} \tag{3.33}$$

where σ_s is the standard deviation of the measurement noise. Assuming that the target moves nearly in a straight-line at the beginning of the tracking process, the initial mode probability is given by [39]

$$\boldsymbol{\mu}(o) = \begin{cases} 0.6 & \textit{striaght line motion} \\ \frac{0.4}{r-1} & \textit{other motions} \end{cases} \tag{3.34}$$

where r is the total number of filters in each of the four IMM algorithms under consideration.

Monte-Carlo simulations of 100 runs is performed assuming the noise for all the algorithms to be the same, for fair comparison. The sampling period T is 1 second and the standard deviation σ_s of the measurement noise to be $10\,m$. All the experiments in this and other chapters are carried out in MATLAB R2018a using a PC with 4.3 GHz processor and 16 GB RAM.

The three algorithms that are used in the comparison are described below

The first algorithm, which we denote by A1, is an IMM algorithm that employs three filters; one of the filters uses the CV model and the other two the CT model with known turn rates of $\pm 2.5^\circ/s$. The standard deviation for the process noise for all the filters is assumed to be $0.003\,m/s^2$. The transition probability matrix of the algorithm is assumed as

$$\mathbf{P}_{tr}^{A1} = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{bmatrix} \tag{3.35}$$

The second algorithm, which we denote by A2, is the same as A1 except for the turn rates of the CT models that is assumed to be $\pm 3.5^\circ/s$.

The third algorithm [39], which we denote as A3, is also an IMM algorithm but employs only two filters; one uses a CV model, and the other use 3CTR model. The initial value of the turn rate is assumed to be $0.2^\circ/s$. The standard deviation for the process noise for both the filters that employ the CV model and the 3CTR model is assumed to be $0.1\,m/s^2$ and $10.5\,m/s^2$, respectively, and the transition probability matrix of the algorithm to be [39]

$$\mathbf{P}_{tr}^{A3} = \begin{bmatrix} 0.98 & 0.02 \\ 0.02 & 0.98 \end{bmatrix} \tag{3.36}$$

For the proposed algorithm, which we henceforth denote by PA1, the standard deviation for the process noise for all the filters is assumed to be $0.003\,m/s^2$. After having considered a number of probability transition matrices, it has been found that the best transition

probability matrix from the point of view of both RMSE and LANEES is given by

$$\mathbf{P}_{tr}^{PA1} = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \tag{3.37}$$

For details regarding the above choice, see Appendix A. We choose the initial values of the turn rates of the two CT models to be $0.2^\circ/s$ and $-0.2^\circ/s$.

3.4.2 Experimental results

The comparison of the proposed algorithm with that of the existing algorithms are carried out under seven different scenarios, in which the target performs various maneuvers. These scenarios along with the corresponding performance results in terms of RMSE and consistency, as measured by LANEES, are shown in the sub-figures (a), (b), and (c), respectively, in Fig. 3.4 to Fig. 3.10. The target moves at a speed of $100\,m/s$ with White Gaussian noise having zero mean and standard deviation of $0.1\,m/s^2$ in both the x and y directions.

(i) **Scenario 1**

In this scenario, the target starts with a straight-line motion for 15 seconds, then performs the first maneuver to the left with a turn rate of $1.5^\circ/s$ for 15 seconds. Afterwards, it moves again in a straight-line motion for 10 seconds, and then performs another turn to the left with a turn rate of $2.5^\circ/s$ for 30 seconds. Finally, it goes in a straight-line motion for a further 10 seconds. This target trajectory of this scenario is shown Fig. 3.4(a). The initial state vector is assumed to be

$$\mathbf{x}(o) = [12500\,m,\ -70.5\,m/s,\ 10000\,m,\ -70.5\,m/s]^t \tag{3.38}$$

It is seen from Fig. 3.4(b) that all the four algorithms exhibit acceptable performance in terms of RMSE. However, as expected, A1 exhibits the best performance, particularly at the second maneuver in view of the fact that the algorithm design matches the target

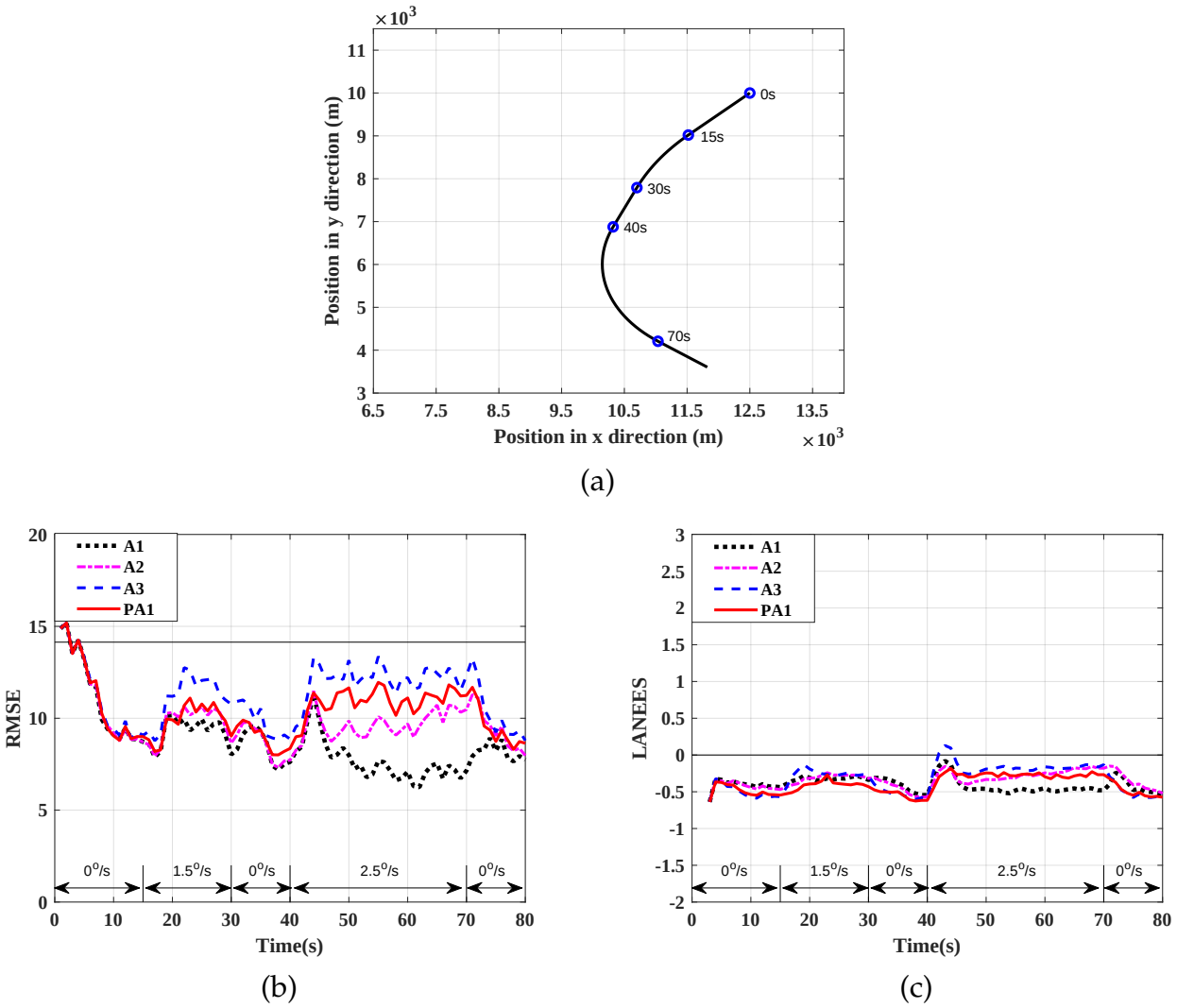


Figure 3.4: Results of scenario 1 for a target that performs turns of $1.5^\circ/s$, and $2.5^\circ/s$. (a) Actual target trajectory of scenario 1. (b) RMSE. (c) LANEES.

maneuver. In addition, it is seen from Fig. 3.4(c) that throughout the tracking period, LANEES value is always negative for all the algorithms and hence, the algorithms are consistent, except for A3, which has a positive value at the beginning of the second maneuver.

(ii) **Scenario 2**

This scenario is the same as scenario 1 except that the second turn rate of the target is increased to $3^\circ/s$. This scenario is depicted in Fig. 3.5(a). From Fig. 3.5(b), it is observed that the proposed algorithm and A2 exhibit acceptable performance throughout the tracking period. The performance of A1 deteriorates, as expected, compared to its

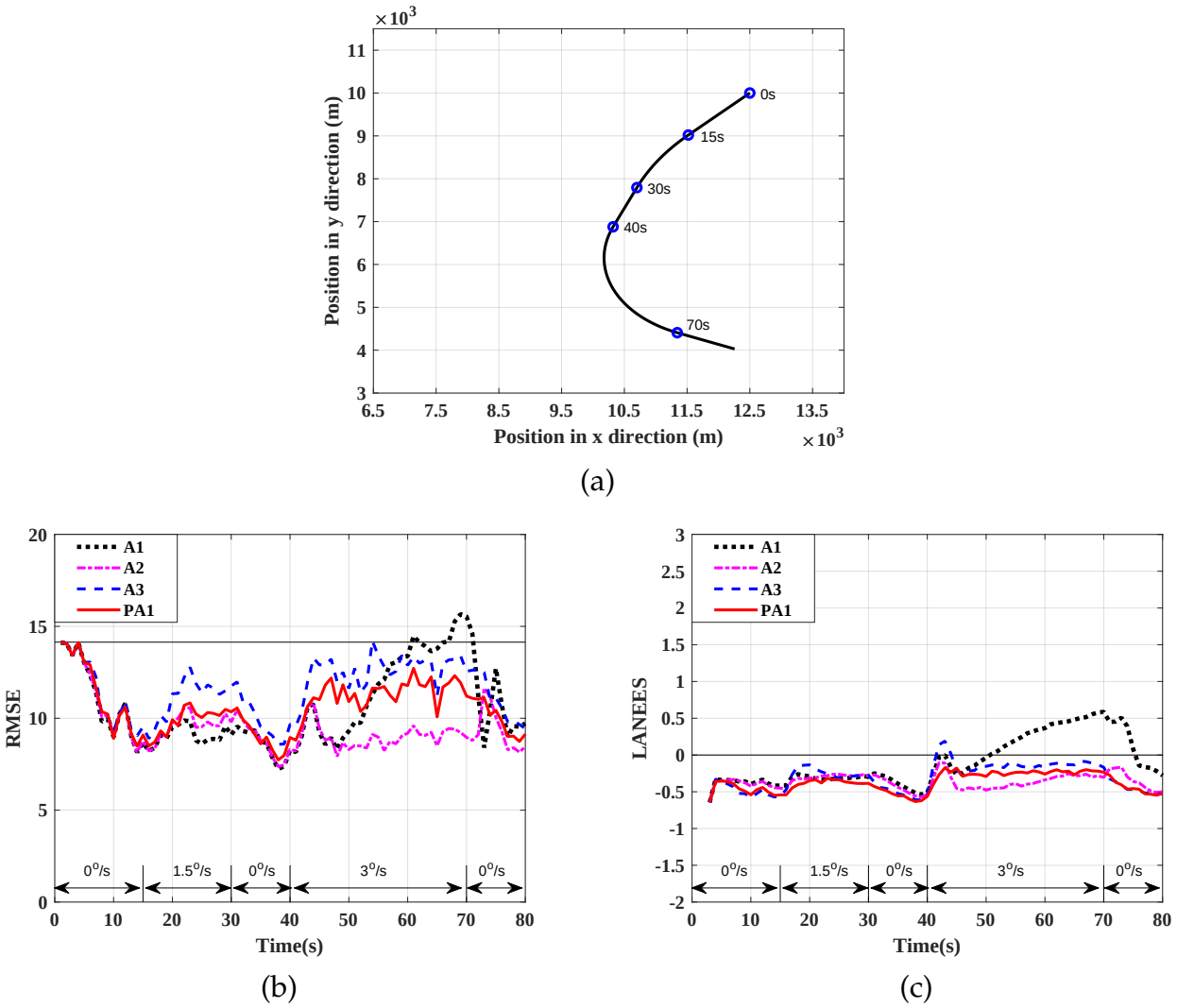


Figure 3.5: Results of scenario 2 for a target that performs turns of $1.5^\circ/s$, and $3^\circ/s$. (a) Actual target trajectory of scenario 2. (b) RMSE. (c) LANEES.

performance in the previous scenario, since the turn rate of the target is outside the range for which the algorithm is designed. Algorithm A2 provides the best performance, as expected, since the turn rate of the target is covered in the region for which the algorithm is designed. The performance of A3 and PA1 are acceptable throughout the tracking period. As seen from Fig. 3.5(c), the consistency of A1 is not acceptable during the second maneuver, as is to be expected. The other algorithms are consistent, but A3 again has a positive value at the beginning of the second maneuver.

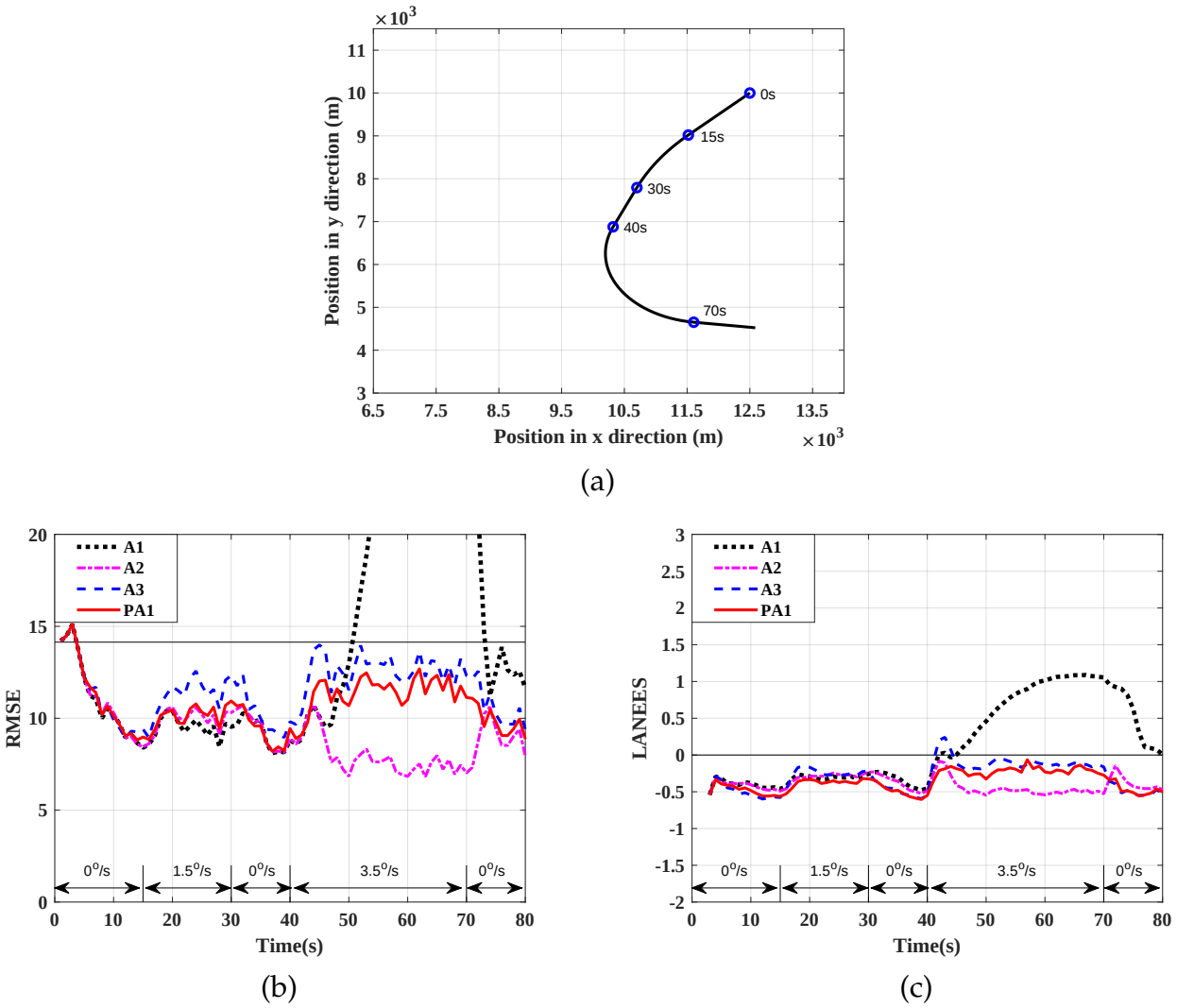


Figure 3.6: Results of scenario 3 for a target that performs turns of $1.5^\circ/s$, and $3.5^\circ/s$. (a) Actual target trajectory of scenario 3. (b) RMSE. (c) LANEES.

(iii) **Scenario 3**

In this scenario, the second turn rate is increased to $3.5^\circ/s$. This scenario is shown in Fig. 3.6(a). It is seen from Figs. 3.6(b) and 3.6(c) that A1 cannot track the second maneuver. As expected, A2 gives the best performance, since the turn rate of the target matches with one of the turn rates for which the algorithm has been designed. The performance of A3 as well as that of PA1 in terms of RMSE and LANEES are acceptable; however, as in the previous scenarios, A3 exhibits positive value for LANEES at the beginning of the second maneuver.

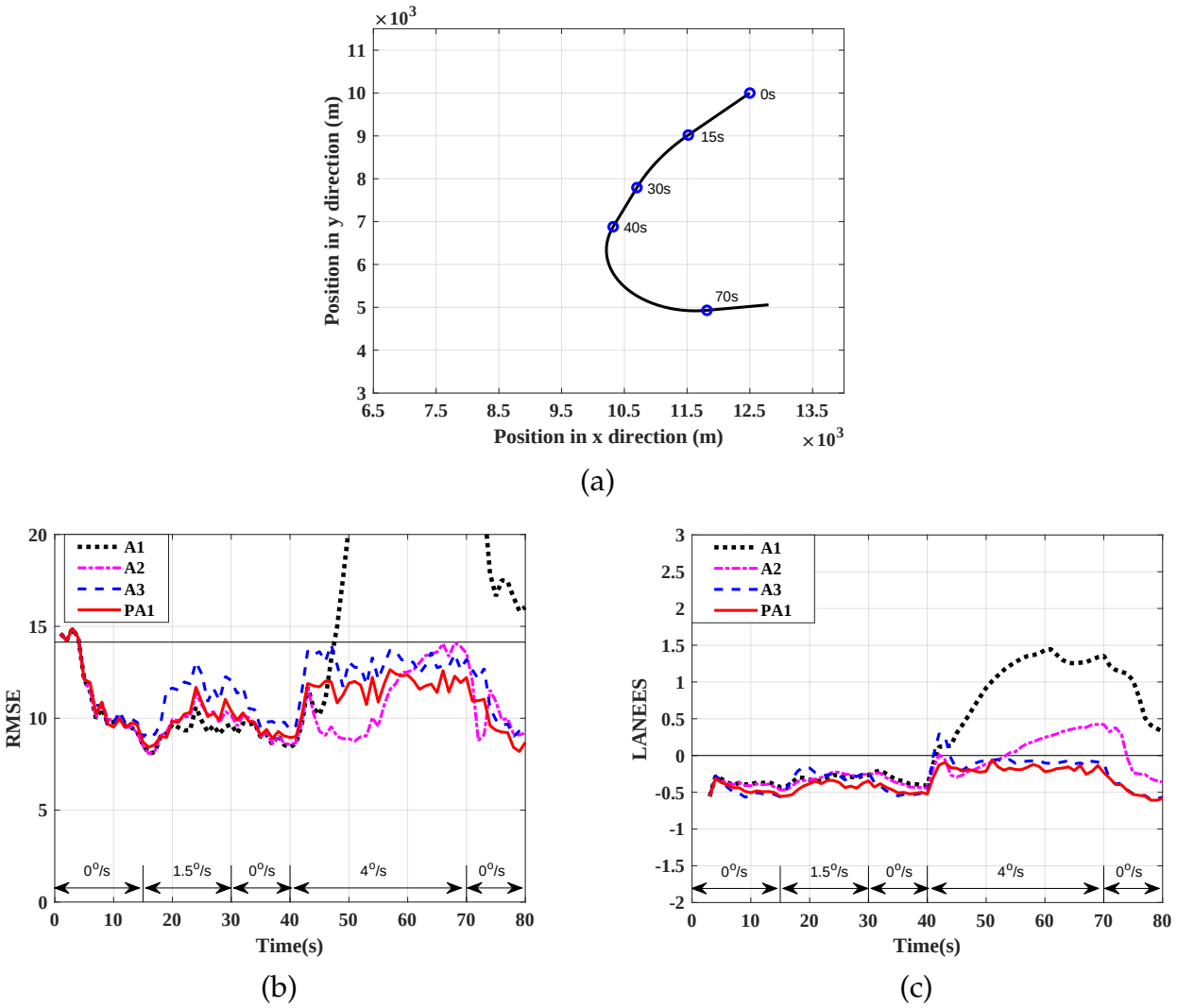


Figure 3.7: Results of scenario 4 for a target that performs turns of $1.5^\circ/s$, and $4^\circ/s$. (a) Actual target trajectory of scenario 4. (b) RMSE. (c) LANEES.

(iv) **Scenario 4**

In this scenario, the second turn rate is increased to $4^\circ/s$. This scenario is shown in Fig. 3.7(a). As expected, A1 is unable to track the target during the second maneuver, as seen in Figs. 3.7(b) and 3.7(c) for the reason pointed out in the previous scenario. Again, as expected the performance of A2 in terms of RMSE is worse than its performance in the previous scenario, since the turn rate of the second maneuver is outside the range for which the algorithm is designed. Moreover, its performance in terms of LANEES is not acceptable during the second maneuver. Algorithms A3 and PA1 give acceptable performance in terms of both the metrics; however, as in the previous scenarios, A3 exhibits positive value for LANEES at the beginning of the second

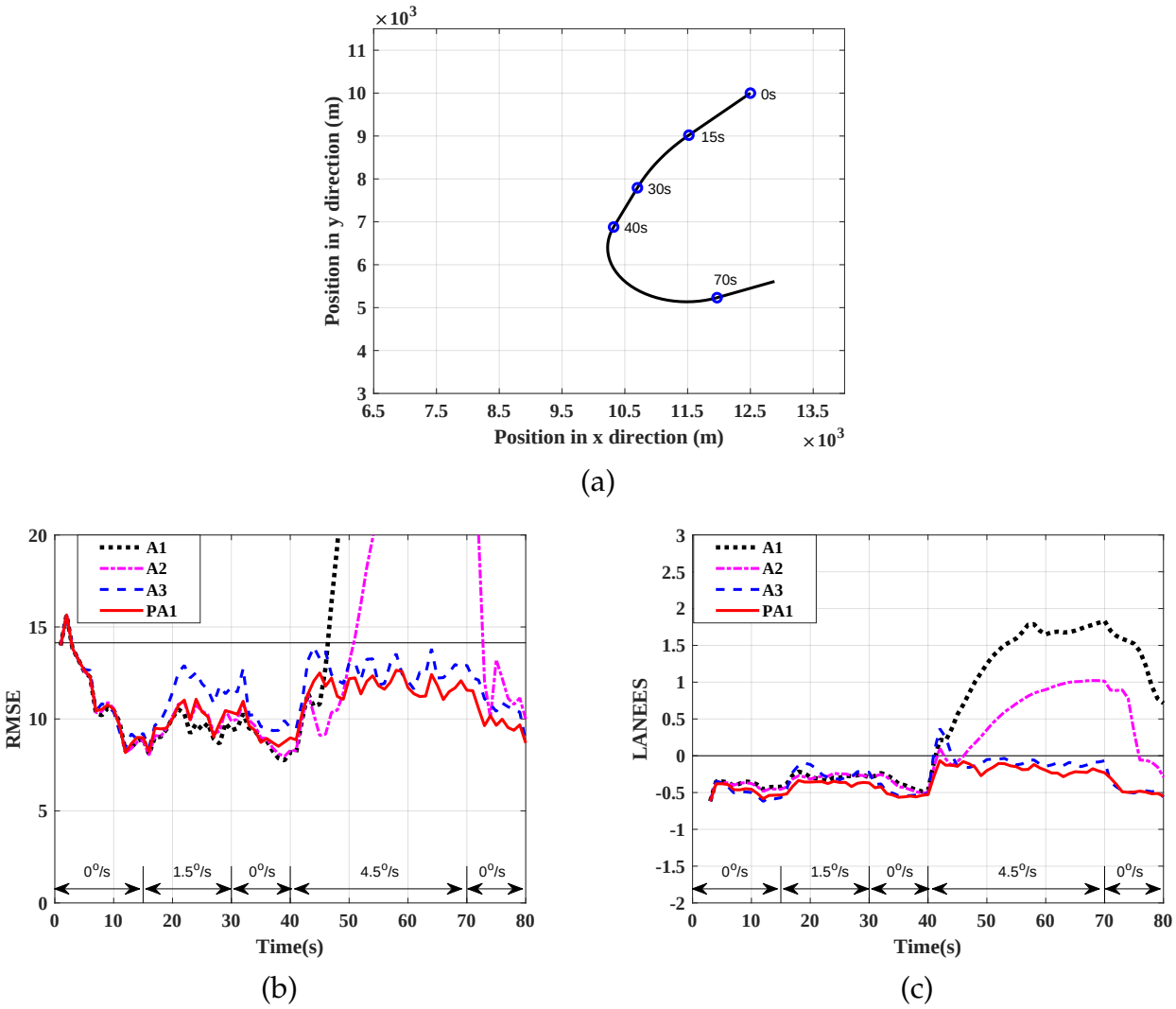


Figure 3.8: Results of scenario 5 for a target that performs turns of $1.5^\circ/s$, and $4.5^\circ/s$. (a) Actual target trajectory of scenario 5. (b) RMSE. (c) LANEES.

maneuver.

(v) **Scenario 5**

This scenario is the same as scenario 4 except that the turn rate of the second maneuver is $4.5^\circ/s$. This scenario is depicted in Fig. 3.8(a). It is seen from Figs. 3.8(b) and 3.8(c) that both A1 and A2 fail to track the target, since the turn rate of the target is outside the range for which these algorithms are designed; moreover, the performance of these algorithms in terms of LANEES is not acceptable during the second maneuver. Both A3 and PA1 show satisfactory performance in terms of the two metrics; however, as in the previous scenarios, A3 exhibits positive value for LANEES at the beginning of the

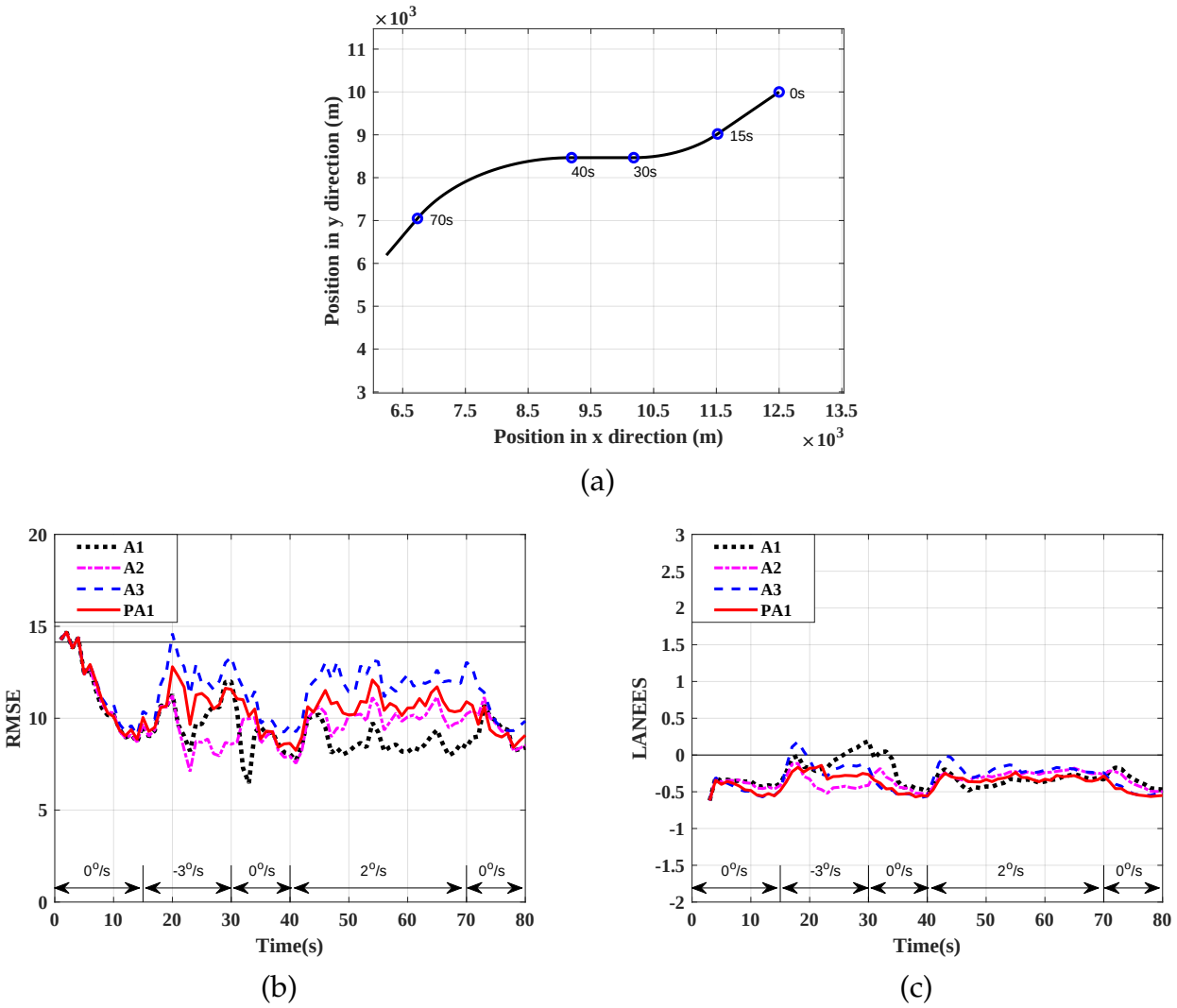


Figure 3.9: Results of scenario 6 for a target that performs turns of $-3^\circ/s$, and $2^\circ/s$. (a) Actual target trajectory of scenario 6. (b) RMSE. (c) LANEES.

second maneuver.

(vi) **Scenario 6**

In this scenario, the target performs two maneuvers the first to the right at a turn rate of $-3^\circ/s$ and the second to the left at a turn rate of $2^\circ/s$. This scenario is shown in Fig. 3.9(a). From Fig. 3.9(b), it is seen that the performance of both A1 and A2 is better than that of A3 and PA1. As is to be expected, the performance of A1 is the best during the second maneuver, since the target turn rate is close to one of the turn rates employed in the design of the algorithm. However, the performance of A2 is the best during the first maneuver because the target turn rate is close to one of turn rates used in the design

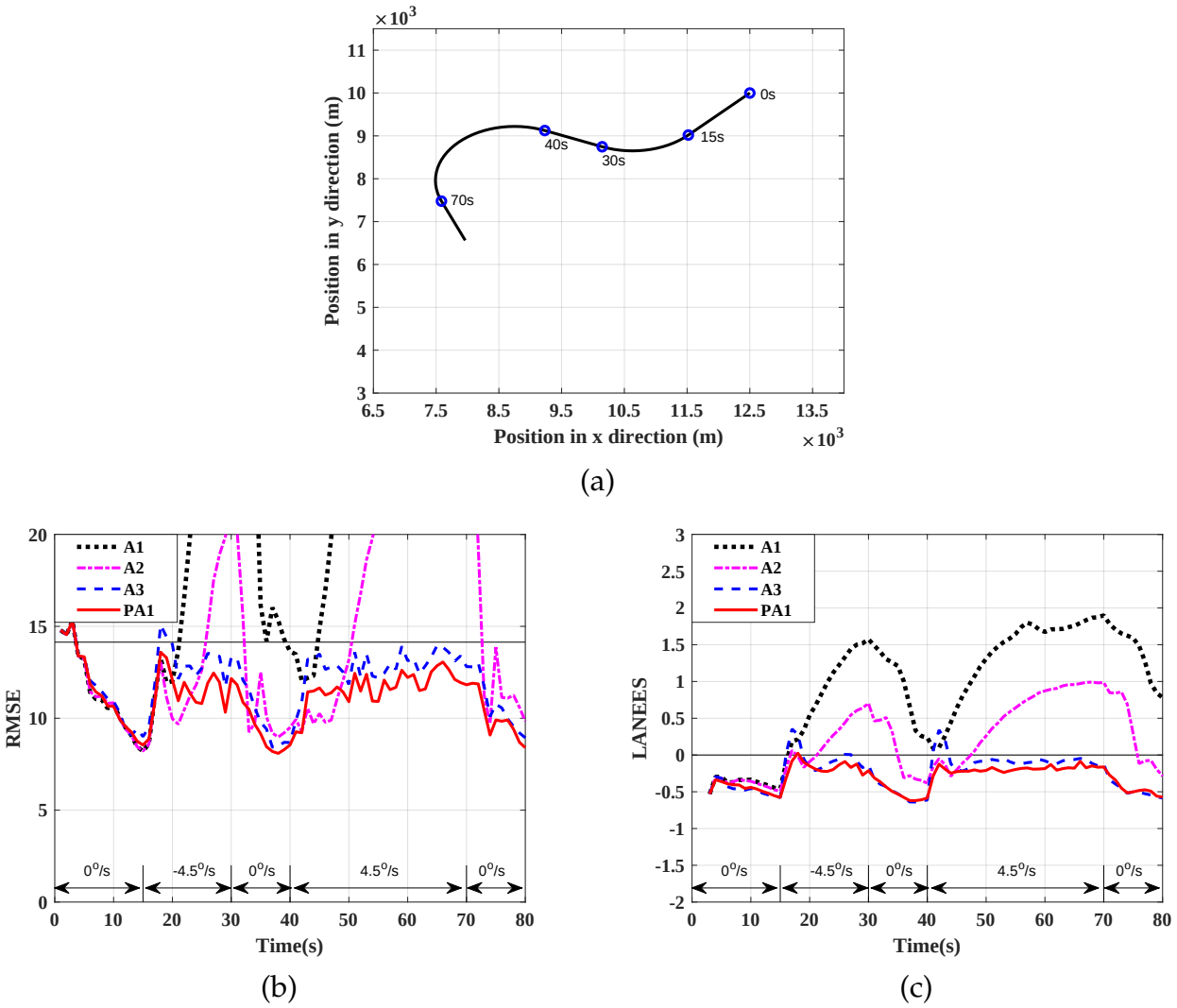


Figure 3.10: Results of scenario 7 for a target that performs turns of $-4.5^\circ/s$, and $4.5^\circ/s$. (a) Actual target trajectory of scenario 7. (b) RMSE. (c) LANEES.

of this algorithm. Both A3 and PA1 exhibit acceptable performance in terms of RMSE; however, A3 has error value more than the error obtained from the raw measurement data for RMSE at the beginning of the first maneuver. It is seen from Fig. 3.9(c) that all the algorithms have acceptable consistency, except that A1 and A3 have positive values for LANEES at the end and the beginning of the first maneuver, respectively.

(vii) **Scenario 7**

This scenario is the same as scenario 6 except that the turn rate of the first maneuver is $-4.5^\circ/s$ and that of the second is $4.5^\circ/s$. This scenario is depicted in Fig. 3.10(a). From Figs. 3.10(b) and 3.10(c) it is seen that, as expected, both A1 and A2 are unable

to track the target for either of the maneuvers, since the target turn rate is outside the ranges for which these algorithms are designed. Both A3 and PA1 show satisfactory performance in terms of the two metrics. However, A3 exhibits a positive value for LANEES at the beginning of both maneuvers, whereas the proposed algorithm has acceptable consistency throughout the tracking process.

It is noticed from these scenarios that the algorithms A1 and A2 exhibit a performance better than that of A3 or PA1 the proposed algorithm, when the target turn rate is covered in the region for which these algorithms have been designed. However, when the turn rate is outside the range, then their performance may not be acceptable or they may even fail to track the target. Further, it is observed that A3 and PA1 exhibit acceptable performance in all the scenarios. But, the performance of PA1 is always better than that of A3 during all the scenarios in terms of both RMSE and the consistency, whereas the consistency of A3 may have a positive value at the beginning of a maneuver, when the turn rate is outside the range of $\pm 2^{\circ}/s$. Based on these results, the proposed algorithm PA1 represents a promising performance in realistic scenarios, where prior information about the target turn rate is rarely available.

3.4.3 Effect of an increased level of the measurement noise on the tracking accuracy

In the previous subsection, it was shown that both A3 and PA1 provide good tracking performance when the target turn rate is unknown, with the latter exhibiting a superior performance. This comparison was made under the assumption that the actual measurement noise level is the same as the one specified in the design of the algorithms. In this subsection, we study the effect on the performance of these two algorithms, when the standard deviation σ_a of the actual measurement noise rises over the specified standard deviation σ_s by 20%. The study is conducted for three values of σ_s, namely, $\sigma_s = 10\,m$, $\sigma_s = 50\,m$ and $\sigma_s = 100\,m$, each under three different scenarios of the target track.

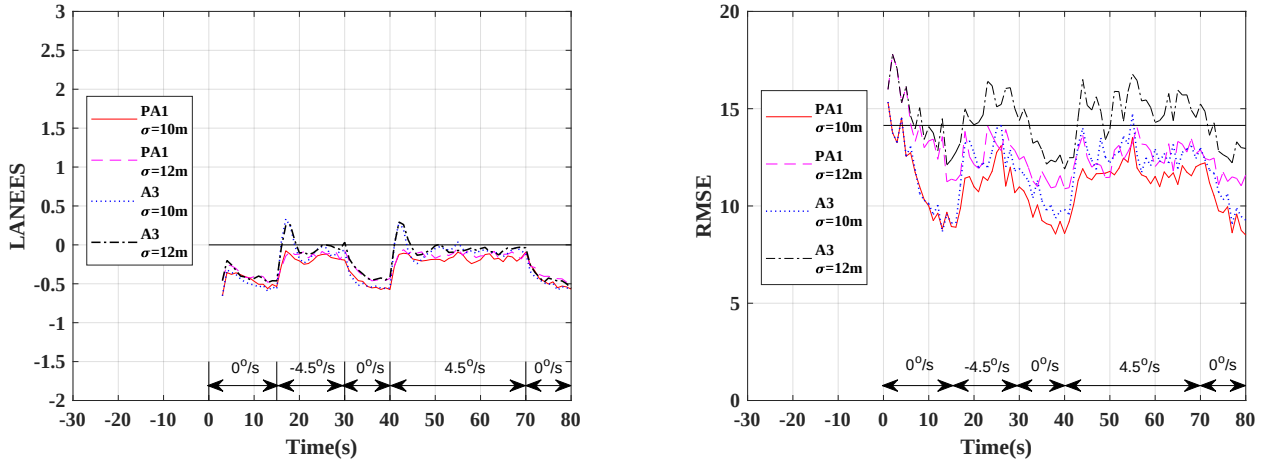


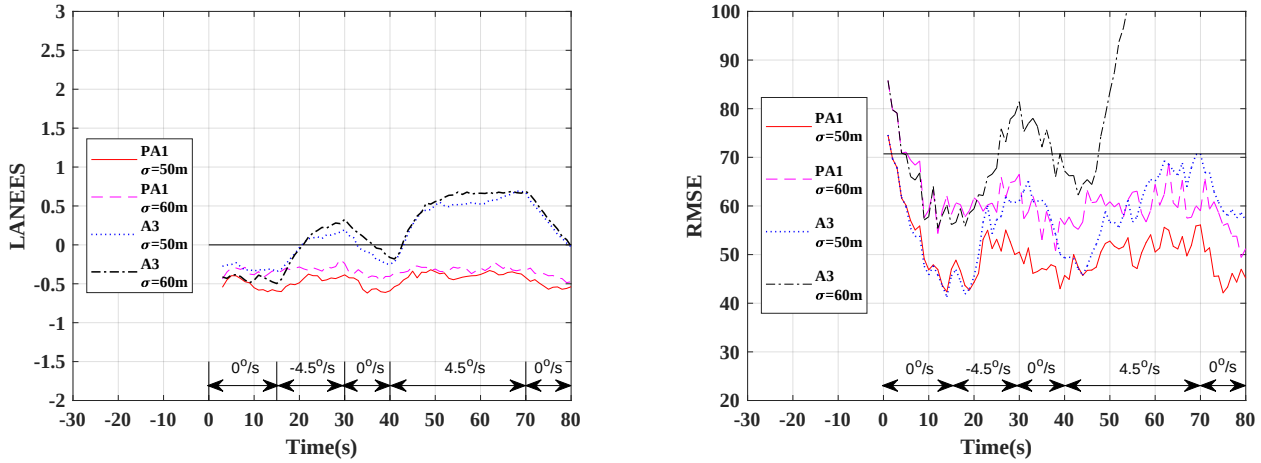
Figure 3.11: Performance of A3 and PA1 with $\sigma_s = 10\,m$ for scenario 8, when $\sigma_a = 10\,m$ and $12\,m$. (a) LANEES. (b) RMSE.

(ix) **Scenario 8**

This scenario is similar to scenario 7, and it is depicted in Fig. 3.10(a).

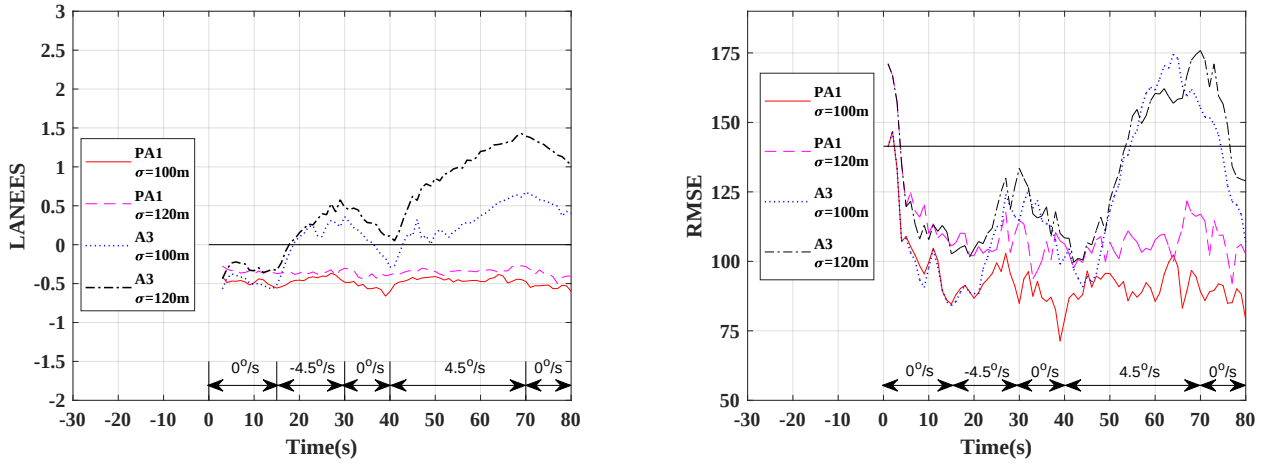
Case 1: $\sigma_s = 10\,m$ *and* $\sigma_a = 12\,m$. The performance of each of the two algorithms, for both $\sigma_a = 10\,m$ and $\sigma_a = 12\,m$, in terms of LANEES is shown in Fig. 3.11(a). It is seen from this figure that PA1 is remains consistent when σ_a rises from $10\,m$ to $12\,m$, whereas A3 is not consistent for both the values of σ_a, since its LANEES has positive values at the beginning of the two maneuvers. The tracking accuracy of A3 and PA1 in terms of RMSE is shown in Fig. 3.11(b). It is seen from this figure that the performance of PA1 is still acceptable when σ_a rises to $12\,m$, whereas the performance of A3 does not remains so, since RMSE values become larger than the threshold value $E_{TH} = 14.14\,m$ during both the maneuvers.

Case 2: $\sigma_s = 50\,m$ *and* $\sigma_a = 60\,m$. The performance of each of the two algorithms, for both $\sigma_a = 50\,m$ and $\sigma_a = 60\,m$, in terms of LANEES is shown in Fig. 3.12(a). It is seen from this figure that PA1 still remains consistent when σ_a rises from $50\,m$ to $60\,m$, whereas A3, as in case 1, is not consistent for both the values of σ_a of this case during the two maneuvers. As a matter of fact, the consistency of A3 becomes worse than in case 1. The tracking accuracy of A3 and PA1 in terms of RMSE is shown in Fig. 3.12(b). It is seen from this figure that even though the performance of PA1 for $\sigma_a = 60\,m$ is



(a) (b)

Figure 3.12: Performance of A3 and PA1 with $\sigma_s = 50\,m$ for scenario 8, when $\sigma_a = 50\,m$ and $60\,m$. (a) LANEES. (b) RMSE.



(a) (b)

Figure 3.13: Performance of A3 and PA1 with $\sigma_s = 100\,m$ for scenario 8, when $\sigma_a = 100\,m$ and $120\,m$. (a) LANEES. (b) RMSE.

not as good as for $\sigma_a = 50\,m$, it is still remains acceptable. The performance of A3 for $\sigma_a = 50\,m$ is acceptable, however, it is not remains so for $\sigma_a = 60\,m$, since RMSE values become much larger than the threshold value $E_{TH} = 70.71\,m$ during the two maneuvers. It also to be noted that for A3, the rise of RMSE values above the threshold value is now much larger than in case 1.

Case 3: $\sigma_s = 100\,m$ *and* $\sigma_a = 120\,m$. The performance of each of the two algorithms, for both $\sigma_a = 100\,m$ and $\sigma_a = 120\,m$, in terms of LANEES is shown in Fig. 3.13(a). It is seen from this figure that PA1 remains consistent when σ_a rises by 20% even at such a

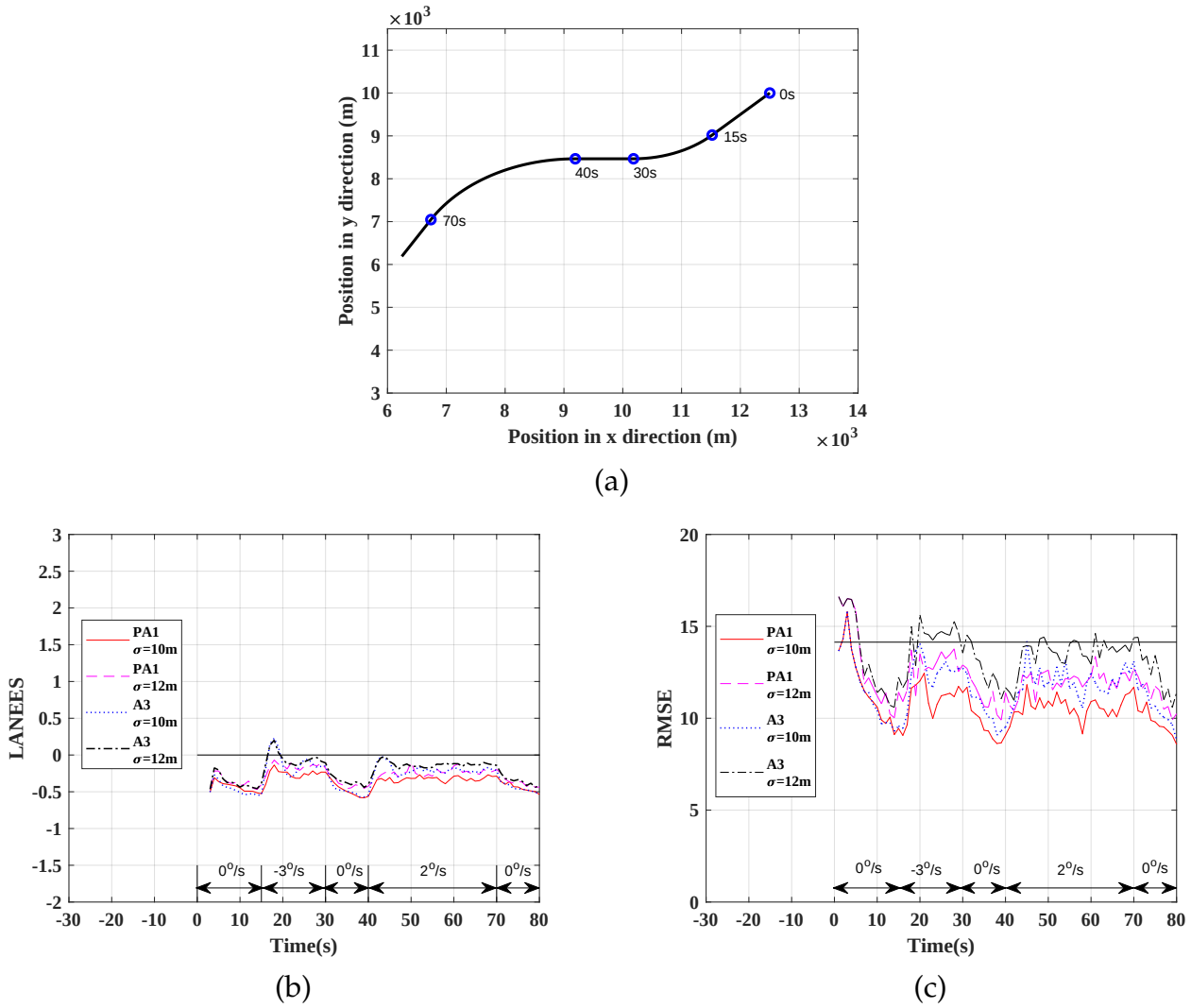
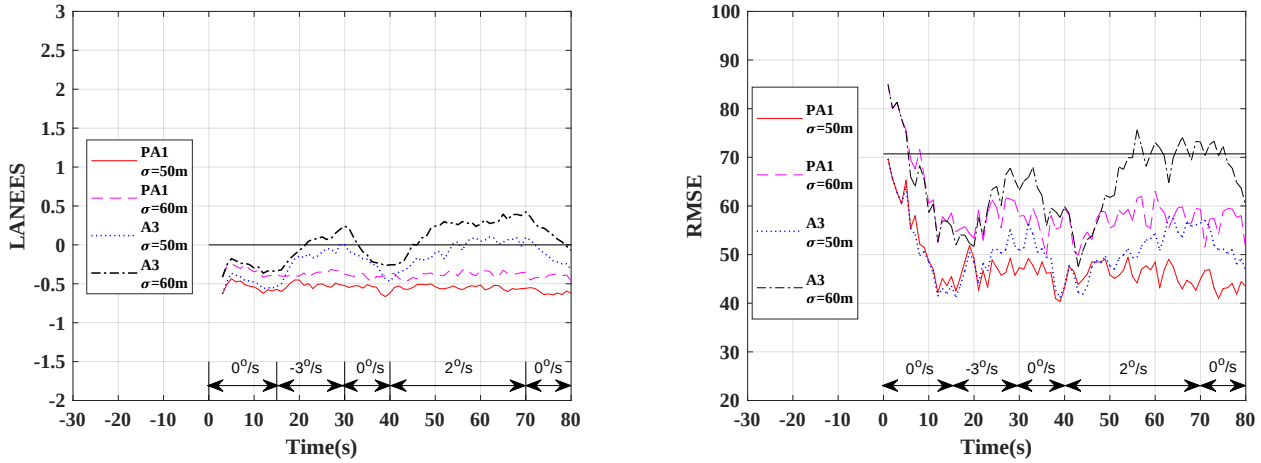


Figure 3.14: Performance of A3 and PA1 with $\sigma_s = 10\,m$ for scenario 9, when $\sigma_a = 10\,m$ and $12\,m$. (a) Actual target trajectory of scenario 9 (b) LANEES. (c) RMSE.

high level of measurement noise, whereas A3 in this case now exhibits inconsistency almost for the entire tracking period for both the values of σ_a. The tracking accuracy of A3 and PA1 in terms of RMSE is shown in Fig. 3.13(b). It is seen from this figure that even though the performance of PA1 for $\sigma_a = 120\,m$ is not as good as for $\sigma_a = 100\,m$, it is still remain acceptable for this high level of measurement noise. On the other hand, the performance of A3 is not acceptable neither for $\sigma_a = 100\,m$ nor $\sigma_a = 120\,m$ during the second maneuver, since RMSE values are greater than $E_{TH} = 141.42\,m$.



(a) (b)

Figure 3.15: Performance of A3 and PA1 with $\sigma_s = 50\,m$ for scenario 9, when $\sigma_a = 50\,m$ and $60\,m$. (a) LANEES. (b) RMSE.

(x) **Scenario 9**

This scenario is similar to scenario 8 except that the turn rate of the first maneuver is $-3^\circ/s$ and that of the second is $2^\circ/s$. This scenario is depicted in Fig. 3.14(a).

Case 1: $\sigma_s = 10\,m$ *and* $\sigma_a = 12\,m$. The performance of each of the two algorithms, for both $\sigma_a = 10\,m$ and $\sigma_a = 12\,m$, in terms of LANEES is shown in Fig. 3.14(b). It is seen from this figure that PA1 is remains consistent when σ_a rises from $10\,m$ to $12\,m$, whereas A3 is not consistent for both the values of σ_a, since its LANEES has positive values at the beginning of the two maneuvers. The tracking accuracy of A3 and PA1 in terms of RMSE is shown in Fig. 3.14(c). It is seen from this figure that the performance of PA1 is still acceptable when σ_a rises to $12\,m$, whereas the performance of A3 does not remains so, since RMSE values become larger than the threshold value $E_{TH} = 14.14\,m$ during both the maneuvers.

Case 2: $\sigma_s = 50\,m$ *and* $\sigma_a = 60\,m$. The performance of each of the two algorithms, for both $\sigma_a = 50\,m$ and $\sigma_a = 60\,m$, in terms of LANEES is shown in Fig. 3.15(a). It is seen from this figure that PA1 still remains consistent when σ_a rises from $50\,m$ to $60\,m$, whereas A3, as in case 1, is not consistent for both the values of σ_a of this case during the two maneuvers. As a matter of fact, the consistency of A3 becomes worse than in case 1. The tracking accuracy of A3 and PA1 in terms of RMSE is shown in Fig. 3.15(b).

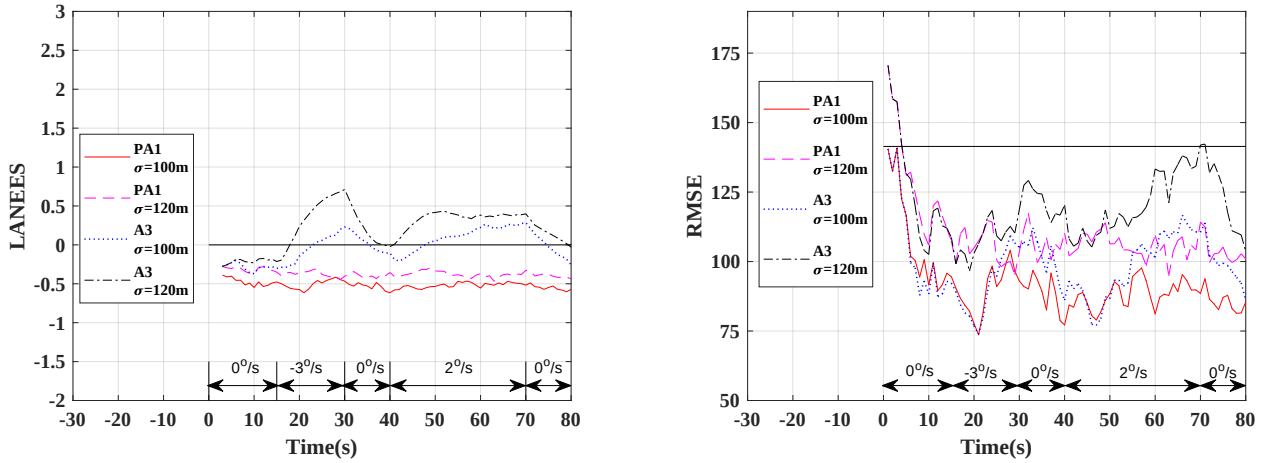


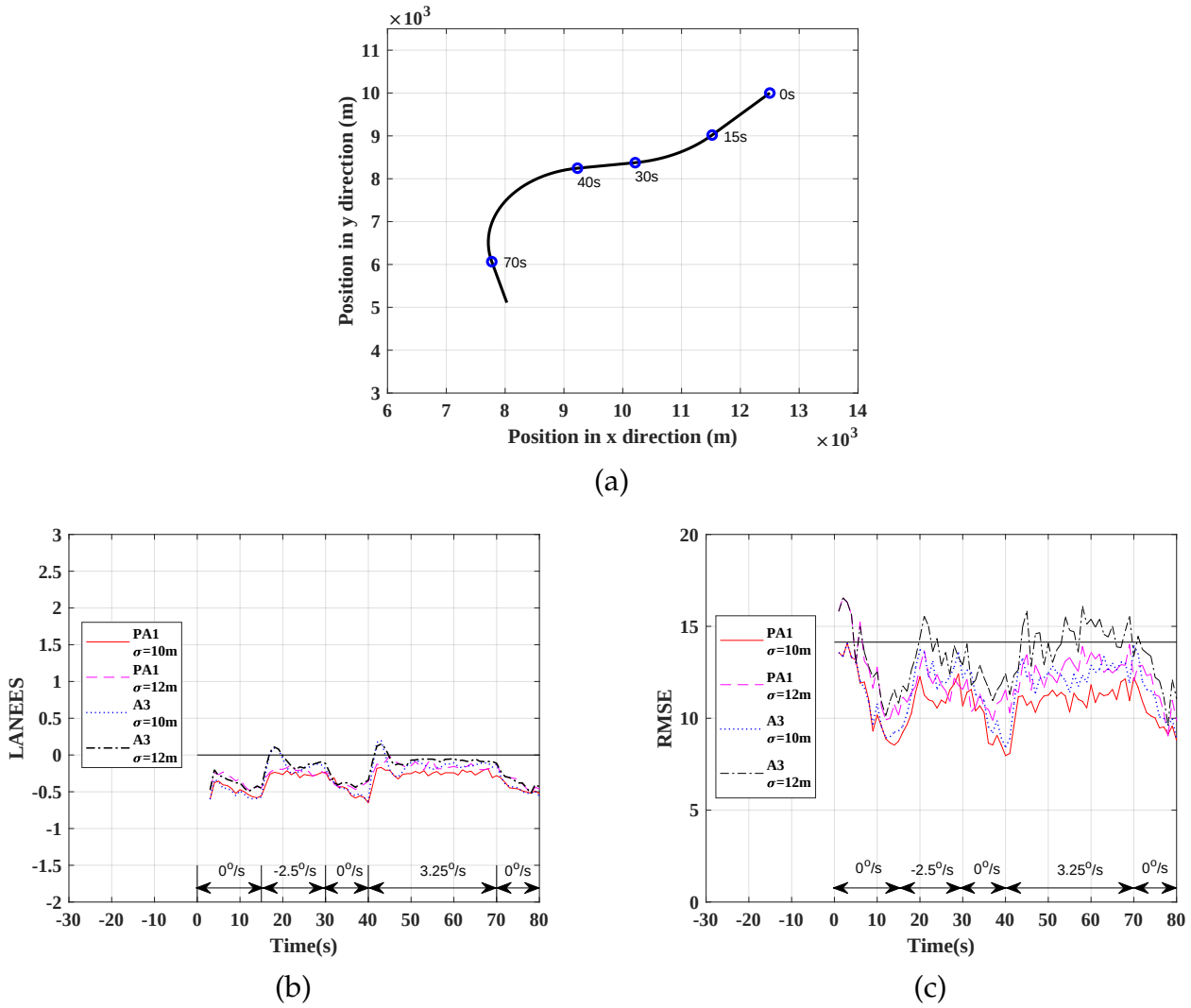
Figure 3.16: Performance of A3 and PA1 with $\sigma_s = 100\,m$ for scenario 9, when $\sigma_a = 100\,m$ and $120\,m$. (a) LANEES. (b) RMSE.

It is seen from this figure that even though the performance of PA1 for $\sigma_a = 60\,m$ is not as good as for $\sigma_a = 50\,m$, it is still remains acceptable. The performance of A3 for $\sigma_a = 50\,m$ is acceptable, however, it is not remains so for $\sigma_a = 60\,m$, since RMSE values become much larger than the threshold value $E_{TH} = 70.71\,m$ during the two maneuvers. It also to be noted that for A3, the rise of RMSE values above the threshold value is now much larger than in case 1.

Case 3: $\sigma_s = 100\,m$ *and* $\sigma_a = 120\,m$. The performance of each of the two algorithms, for both $\sigma_a = 100\,m$ and $\sigma_a = 120\,m$, in terms of LANEES is shown in Fig. 3.16(a). It is seen from this figure that PA1 remains consistent when σ_a rises by 20% even at such a high level of measurement noise, whereas A3 in this case now exhibits inconsistency almost for the entire tracking period for both the values of σ_a. The tracking accuracy of A3 and PA1 in terms of RMSE is shown in Fig. 3.16(b). It is seen from this figure that even though the performance of A1 for $\sigma_a = 120\,m$ is not as good as for $\sigma_a = 100\,m$, it is still remain acceptable for this high level of measurement noise. On the other hand, the performance of A3 is not acceptable neither for $\sigma_a = 100\,m$ nor $\sigma_a = 120\,m$ during the second maneuver, since RMSE values are greater than $E_{TH} = 141.42\,m$.

(xi) **Scenario 10**

This scenario is similar to scenario 8 except that the turn rate of the first maneuver is



(a) (b) (c)

Figure 3.17: Performance of A3 and PA1 with $\sigma_s = 10\,m$ for scenario 10, when $\sigma_a = 10\,m$ and $12\,m$. (a) Actual target trajectory of scenario 10. (b) LANEES. (c) RMSE.

$-2.5^\circ/s$ and that of the second is $3.25^\circ/s$. This scenario is depicted in Fig. 3.17(a).

Case 1: $\sigma_s = 10\,m$ and $\sigma_a = 12\,m$. The performance of each of the two algorithms, for both $\sigma_a = 10\,m$ and $\sigma_a = 12\,m$, in terms of LANEES is shown in Fig. 3.17(b). It is seen from this figure that PA1 is remains consistent when σ_a rises from $10\,m$ to $12\,m$, whereas A3 is not consistent for both the values of σ_a, since its LANEES has positive values at the beginning of the two maneuvers. The tracking accuracy of A3 and PA1 in terms of RMSE is shown in Fig. 3.17(c). It is seen from this figure that the performance of PA1 is still acceptable when σ_a rises to $12\,m$, whereas the performance of A3 does not remains so, since RMSE values become larger than the threshold value $E_{TH} = 14.14\,m$

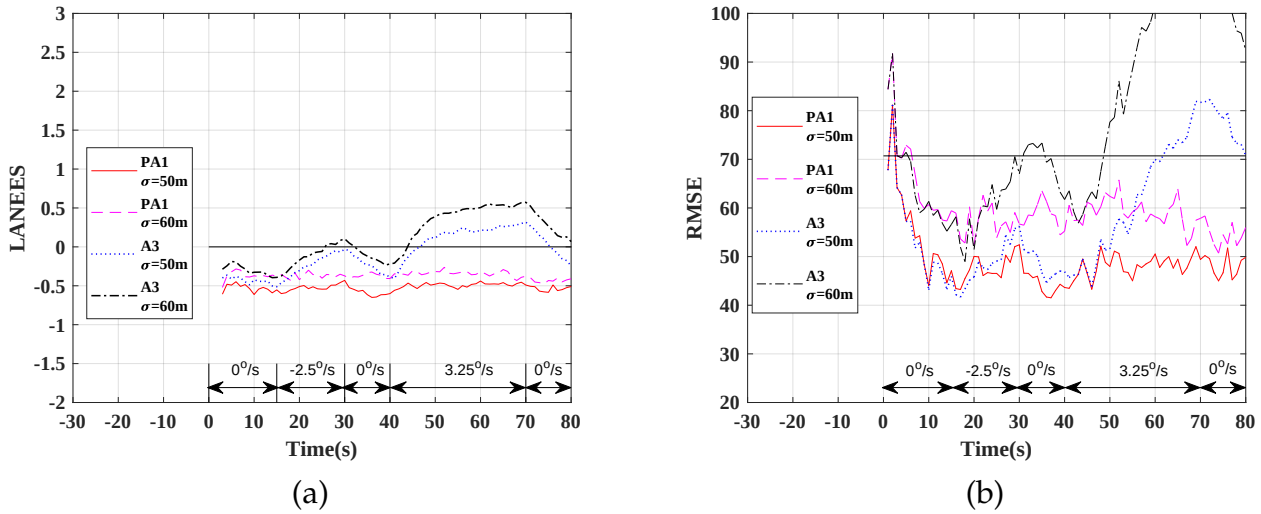


Figure 3.18: Performance of A3 and PA1 with $\sigma_s = 50\,m$ for scenario 10, when $\sigma_a = 50\,m$ and $60\,m$. (a) LANEES. (b) RMSE.

during both the maneuvers.

Case 2: $\sigma_s = 50\,m$ and $\sigma_a = 60\,m$. The performance of each of the two algorithms, for both $\sigma_a = 50\,m$ and $\sigma_a = 60\,m$, in terms of LANEES is shown in Fig. 3.18(a). It is seen from this figure that PA1 still remains consistent when σ_a rises from $50\,m$ to $60\,m$, whereas A3, as in case 1, is not consistent for both the values of σ_a of this case during the two maneuvers. As a matter of fact, the consistency of A3 becomes worse than in case 1. The tracking accuracy of A3 and PA1 in terms of RMSE is shown in Fig. 3.18(b). It is seen from this figure that even though the performance of PA1 for $\sigma_a = 60\,m$ is not as good as for $\sigma_a = 50\,m$, it is still remains acceptable. The performance of A3 for $\sigma_a = 50\,m$ is acceptable, however, it is not remains so for $\sigma_a = 60\,m$, since RMSE values become much larger than the threshold value $E_{TH} = 70.71\,m$ during the two maneuvers. It also to be noted that for A3, the rise of RMSE values above the threshold value is now much larger than in case 1.

Case 3: $\sigma_s = 100\,m$ and $\sigma_a = 120\,m$. The performance of each of the two algorithms, for both $\sigma_a = 100\,m$ and $\sigma_a = 120\,m$, in terms of LANEES is shown in Fig. 3.19(a). It is seen from this figure that PA1 remains consistent when σ_a rises by 20% even at such a high level of measurement noise, whereas A3 in this case now exhibits inconsistency almost for the entire tracking period for both the values of σ_a. The tracking accuracy of

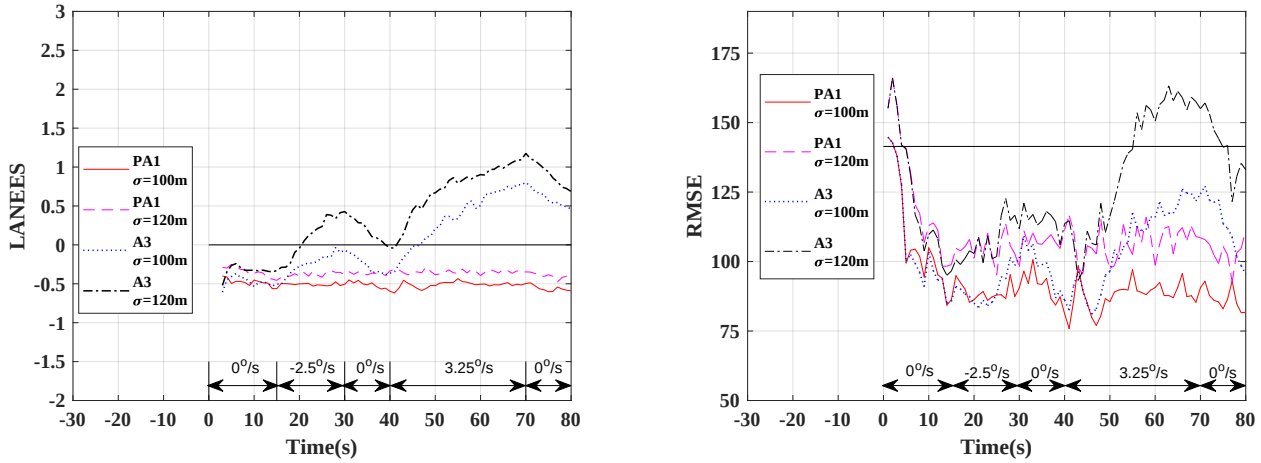


Figure 3.19: Performance of A3 and PA1 with $\sigma_s = 100\,m$ for scenario 10, when $\sigma_a = 100\,m$ and $120\,m$. (a) LANEES. (b) RMSE.

A3 and PA1 in terms of RMSE is shown in Fig. 3.19(b). It is seen from this figure that even though the performance of PA1 for $\sigma_a = 120\,m$ is not as good as for $\sigma_a = 100\,m$, it is still remain acceptable for this high level of measurement noise. On the other hand, the performance of A3 is not acceptable neither for $\sigma_a = 100\,m$ nor $\sigma_a = 120\,m$ during the second maneuver, since RMSE values are greater than $E_{TH} = 141.42\,m$.

Based on these observations, we can draw the following conclusions.

(i) The performance of A3, in terms of the two metrics, is acceptable only when $\sigma_a = \sigma_s$ does not have a very high value, i.e., only when $\sigma_a = \sigma_s = 10\,m$ or $50\,m$. Its performance becomes unacceptable both when σ_s is very large, i.e., when $\sigma_s = 100\,m$, or when regardless of the value of σ_s, σ_a rises over that of σ_s by 20%.

(ii) Algorithm PA1 performs very well, in terms of both the metrics, for $\sigma_a = \sigma_s$ irrespective of the three values of σ_s considered, even though its performance somewhat deteriorates for higher values of σ_s. When σ_a rises over σ_s by as much as 20%, PA1 remains consistent and its RMSE values remain lower than the acceptable threshold value irrespective of the three values of σ_s. However, the performance of PA1 in terms of RMSE becomes lower, when the value of σ_a rises by 20% over that of σ_s.

3.5 Summary

In this chapter, we have proposed an algorithm to track a single maneuvering target, when the information on the target turn rate is not known *a priori* and the radar provides only information on the position of the target using noisy measurements. The turn rate is obtained dynamically using these noisy measurements. A detailed study has been carried out to choose an appropriate transition probability matrix for better tracking performance. A comparison of the proposed algorithm with two existing algorithms has been carried out. One of these two algorithms assumes that the target turn rate is known and the other one estimates it during the tracking process. Based on a number of scenarios with various target turn rates, it has been shown that the proposed algorithm exhibits satisfactory performance in all of the scenarios considered, and its performance is superior to that of the adaptive algorithm. Further, when an unexpected maneuver is performed by the target, and the target turn rate is not covered by the algorithms that have been designed with a prior information on the target turn rate, the performance of these algorithms deteriorates or the algorithms may even fail to track the target, whereas the proposed algorithm always exhibits a satisfactory performance.

Finally, we have studied the performance of the proposed algorithm as well as on that of the algorithm in [39] in situations when the value of the actual standard deviation of the measurement noise rises, due to some reason, over that of the specified one for which these algorithms have been designed. It has been shown that the performance of both the algorithms deteriorates, particularly for large values of σ_s. However, the proposed algorithm still exhibits acceptable performance, whereas the performance of the other algorithm becomes unacceptable from the point of view of the consistency and/or in terms of RMSE for large values of σ_s.

Chapter 4

The Second Proposed Algorithm for Tracking a Single Maneuvering Target

4.1 Introduction

In Chapter 3, we proposed an algorithm for tracking a single maneuvering target in the IMM framework by introducing a scheme for estimating the turn rate of the target dynamically. The turn rate at each instant of time was estimated by using three consecutive noisy measurements. In this chapter, we develop a scheme to estimate the target turn rate more accurately using the information of the level of the measurement noise, and employ this scheme in the algorithm for target tracking proposed in Chapter 3 in an effort to achieve a better performance [85]. The performance of this algorithm is evaluated and compared with that of the algorithm proposed in Chapter 3. The performance of the algorithm is also studied when the level of the measurement noise exceeds the level used in the design of the algorithm.

4.2 Second Proposed algorithm

4.2.1 Development of the algorithm

We have seen in Chapter 3 that the received measurement data do not lie exactly on the true target track, the reason for this being that these measurements are noisy. In the algorithm

of Chapter 3, we modeled the true track of the target dynamically at each time instant by a segment of the circle formed by the noisy measurement received at that instant and the two previous consecutive measurements. Note that the curvature of the circle at that time instant is larger than that of the true track in view of the scatteredness of the measurement data around the true track. The result of this larger curvature affects the estimated radius as well as the estimated target speed. Thus, the estimated value of the target turn rate is always larger than the actual turn rate of the target. In order to compensate for the inaccuracy that results from a higher value of the estimated turn rate, we refine the estimated turn rate by introducing a multiplying factor α as

$$\omega_{imp}(k) = \omega(k)\alpha, \qquad 0 < \alpha < 1 \tag{4.1}$$

where $w_{imp}(k)$ is the refined value of the estimated turn rate. The block diagram of this algorithm is shown in Fig. 4.1, which is the same as Fig. 3.3 in Chapter 3, except for the introduction of a multiplier with its inputs being the multiplying factor α and $\omega(k)$. The output $\omega_{imp}(k)$ of the multiplier is used in the next iteration of the algorithm. Henceforth, this algorithm is referred to as PA2.

In the next subsection, we empirically determine the value of the multiplying factor α used in this algorithm, designed for a specified value of the standard deviation of the measurement noise and compare its performance with that of the algorithm of Chapter 3, PA1.

4.2.2 Experimental results and performance evaluation

In this subsection, experiments are performed to find the optimal value of α for PA2 for a radar system with a specified measurement noise standard deviation σ_s. For this purpose, we assume that the actual value of the standard deviation, σ_a, is the same as σ_s that is used in the design of the algorithm. We vary the value of α in the range 0.05 to 0.95 in steps of 0.05 and obtain the performance of the algorithm in terms of the two metrics that are mentioned in Section 2.3, namely LANEES and RMSE.

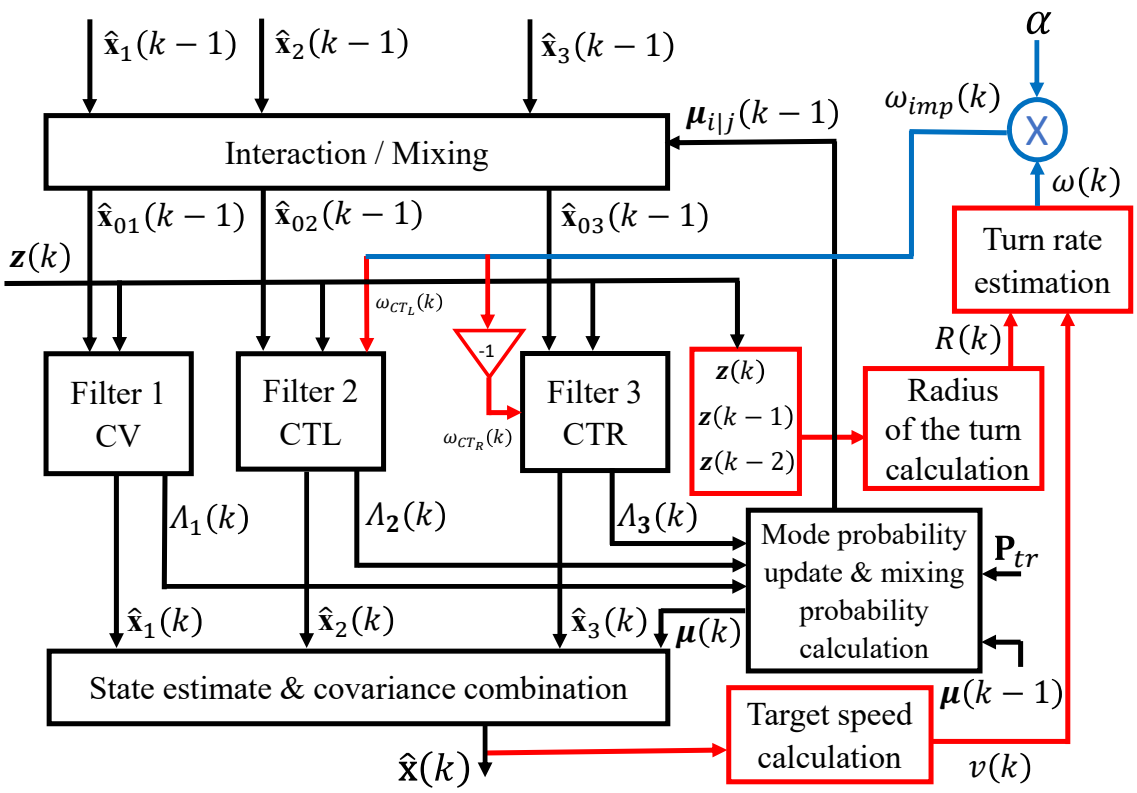


Figure 4.1: Block diagram of the second proposed algorithm

The value of α that satisfies the consistency condition and gives the best performance in terms of RMSE is chosen as the optimal value of α at this specified measurement noise level.

For a given value of the specified standard deviation, σ_s, of the measurement noise, experiments are carried out under three different scenarios to find the optimal value of α. In this study, we examine three different radar systems each with a different value for the specified standard deviation of the measurement noise, $\sigma_s = 10\,m$, $\sigma_s = 50\,m$, and $\sigma_s = 100\,m$.

Scenario 1

In this scenario, the target moves with a constant speed of $100\,m/s$ with white Gaussian process noise having a zero mean and standard deviation of $0.1\,m/s^2$ in both the x and y directions. The initial position and velocity of the target are assumed to be $(12.5, 10)\,Km$ and $(-70.5, -70.5)\,m/s$, respectively. The target starts with a straight-line motion, then performs a right maneuver of $-4.5°/s$. Afterwards, the target returns to a straight-line motion, followed by a left maneuver of $4.5°/s$ and finally returns to a straight-line motion. The durations of the five segments of the motion, are respectively, $15\,s$, $15\,s$, $10\,s$, $30\,s$, and $10\,s$. This

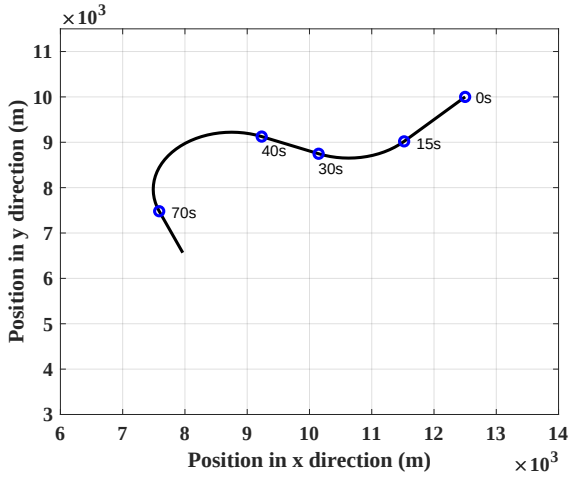
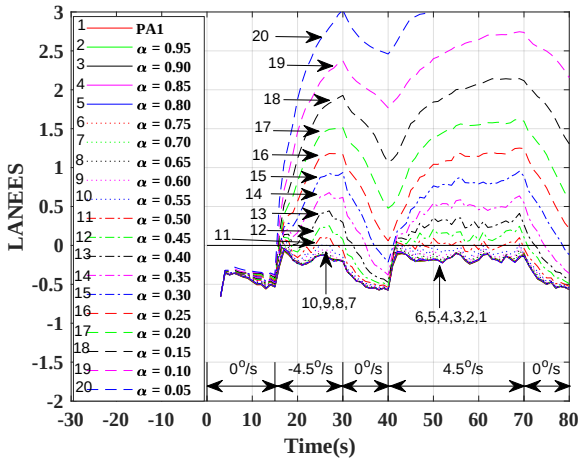
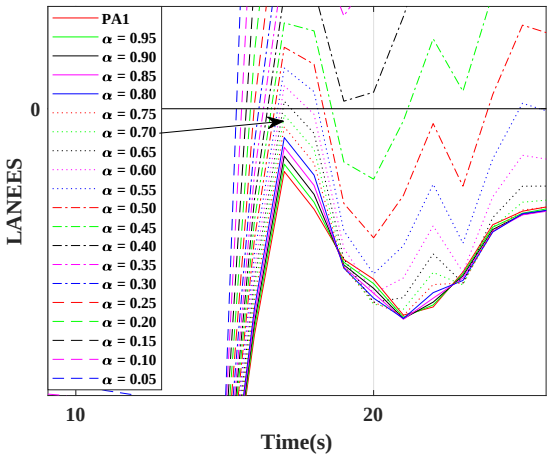


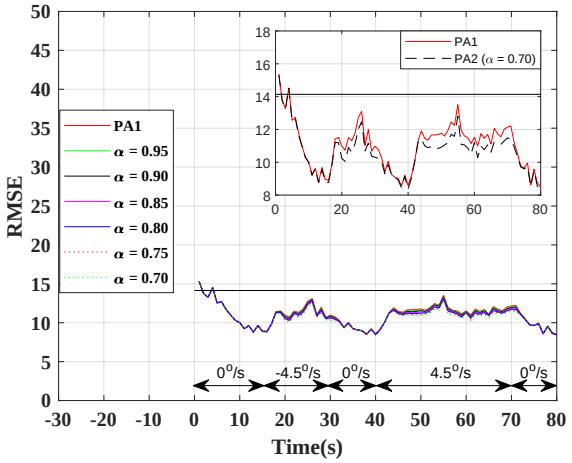
Figure 4.2: Actual target trajectory of scenario 1.



(a)



(b)



(c)

Figure 4.3: Performance of PA2 with different values of α compared to PA1 for scenario 1, when $\sigma_s = 10\,m$. (a) LANEES. (b) Zoomed version of (a) around LANEES=0. (c) RMSE.

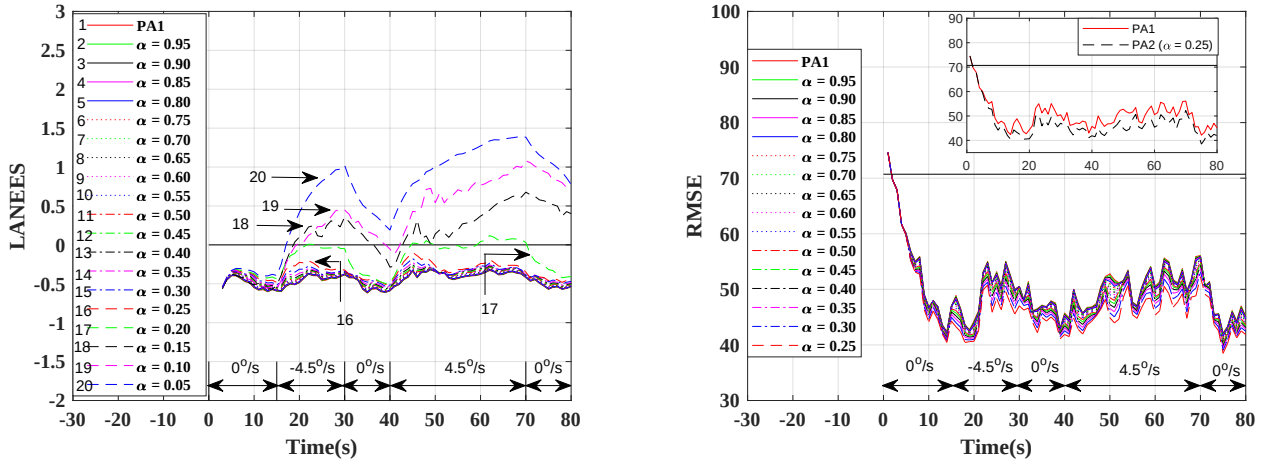


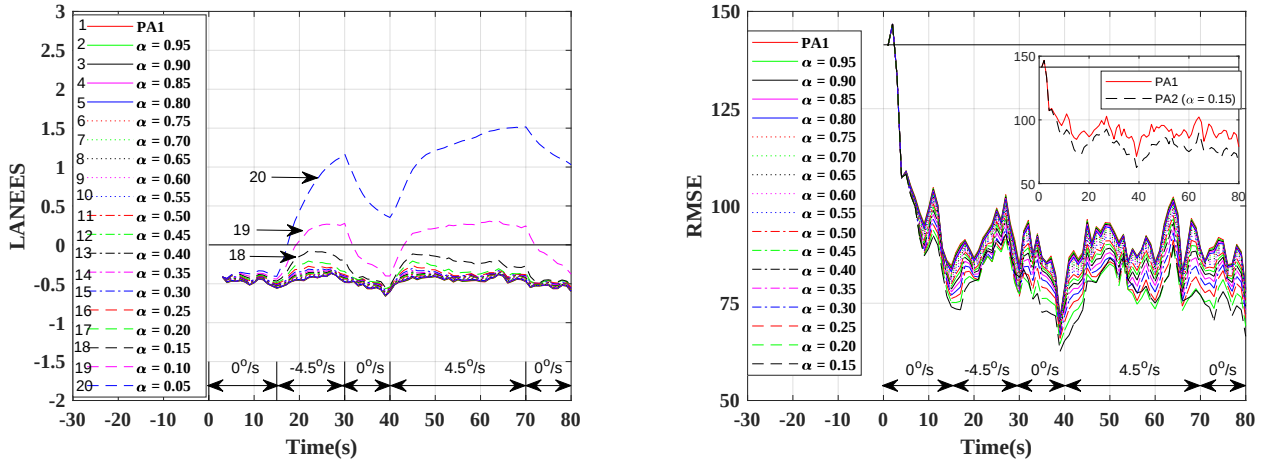
Figure 4.4: Performance of PA2 with different values of α compared to PA1 for scenario 1, when $\sigma_s = 50\,m$. (a) LANEES. (b) RMSE.

scenario is depicted in Fig. 4.2.

Case 1: $\sigma_s = 10\,m$. Fig. 4.3(a) shows the performances, in terms of LANEES, of PA2 for different values for α and of PA1. Fig. 4.3(b) shows a zoomed version of Fig. 4.3(a) around the value of LANEES equal to zero. From Figs. 4.3(a) and 4.3(b), it is observed that the range of α for which PA2 is consistent is from 0.7 to 0.95. The tracking accuracies, in terms of RMSE, of PA2 with this range of α and of PA1 are shown in Fig. 4.3(c). It is seen from this figure that PA2 in this range of α provides an accuracy better than that provided by PA1 and the best tracking accuracy is achieved when $\alpha = 0.7$.

Case 2: $\sigma_s = 50\,m$. Fig. 4.4(a) shows the performances, in terms of LANEES, of PA2 for different values for α and of PA1. It is observed from this figure that the range of α for which PA2 is consistent is from 0.25 to 0.95. The tracking accuracies, in terms of RMSE, of PA2 with this range of α and of PA1 are shown in Fig. 4.4(b). It is seen from this figure that PA2 in this range of α provides an accuracy better than that provided by PA1 and the best tracking accuracy is achieved when $\alpha = 0.25$.

Case 3: $\sigma_s = 10\,m$. Fig. 4.5(a) shows the performances, in terms of LANEES, of PA2 for different values for α and of PA1. It is observed from this figure that the range of α for which PA2 is consistent is from 0.15 to 0.95. The tracking accuracies, in terms of RMSE, of PA2 with this range of α and of PA1 are shown in Fig. 4.5(b). It is seen from this figure that PA2 in



(a) (b)

Figure 4.5: Performance of PA2 with different values of α compared to PA1 for scenario 1, when $\sigma_s = 100\,m$. (a) LANEES. (b) RMSE.

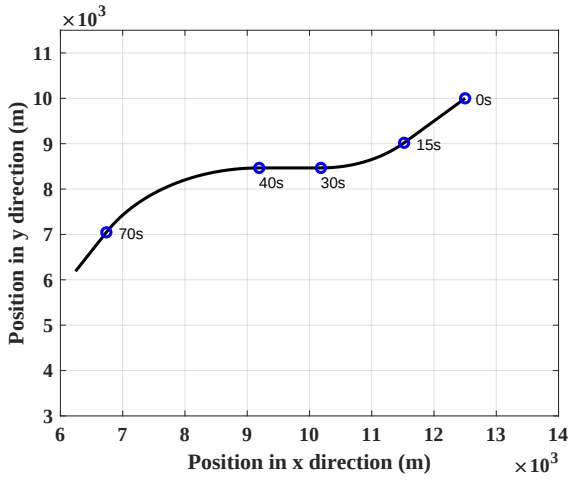
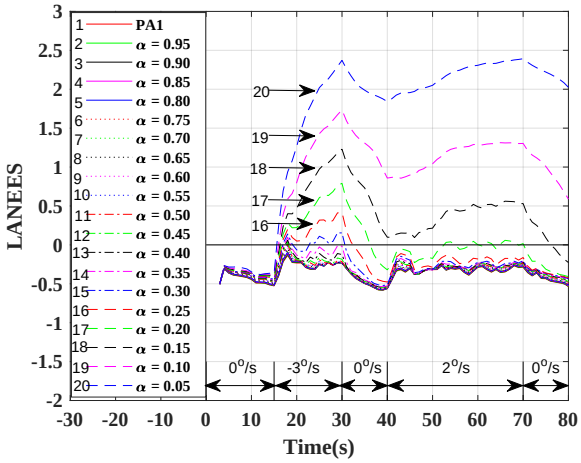


Figure 4.6: Actual target trajectory of scenario 2.

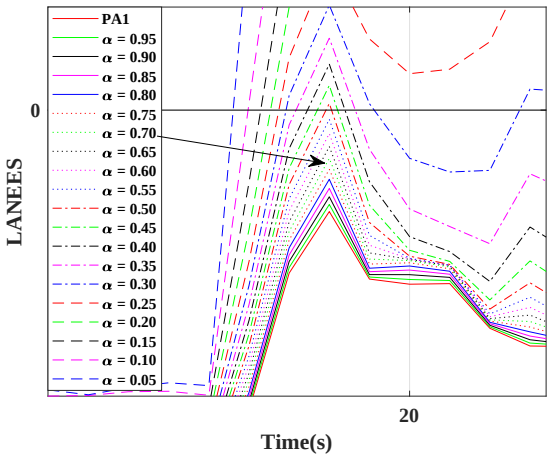
this range of α provides an accuracy better than that provided by PA1 and the best tracking accuracy is achieved when $\alpha = 0.15$.

Scenario 2

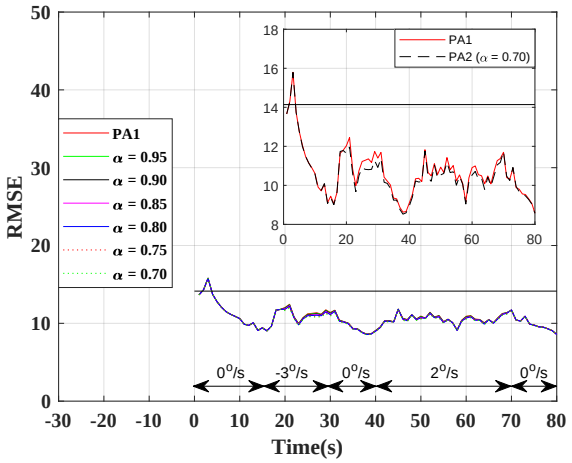
This scenario is the same as scenario 1 except that the first and second turn rates in the former are $-3^\circ/s$ and $2^\circ/s$, respectively. This scenario is depicted in Fig. 4.6. As in scenario 1, we again examine three different radar systems each with a different value for the specified standard deviation of the measurement noise, $\sigma_s = 10\,m$, $\sigma_s = 50\,m$, and $\sigma_s = 100\,m$.



(a)



(b)

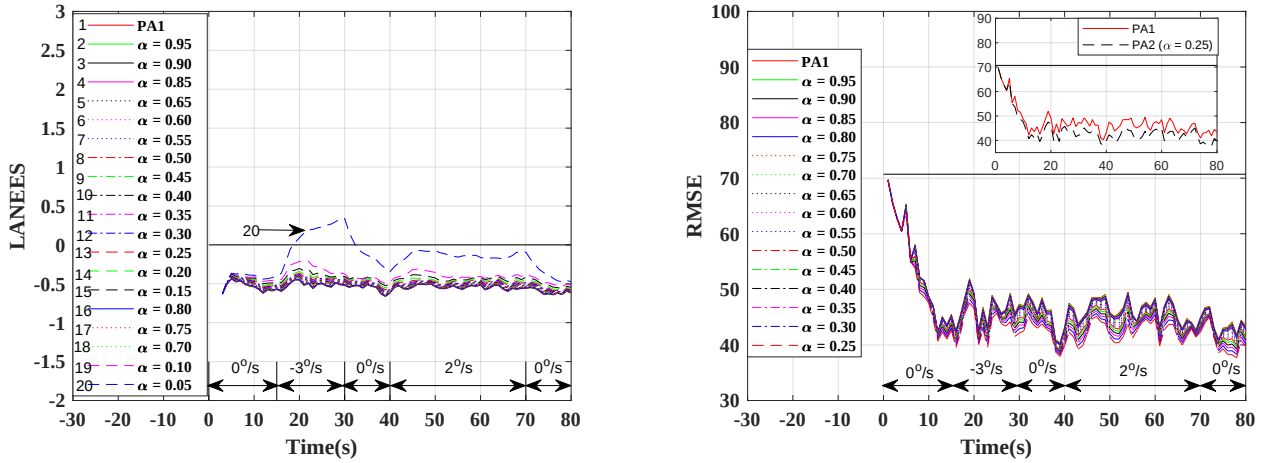


(c)

Figure 4.7: Performance of PA2 with different values of α compared to PA1 for scenario 2, when $\sigma_s = 10\,m$. (a) LANEES. (b) Zoomed version of (a) around LANEES=0. (c) RMSE.

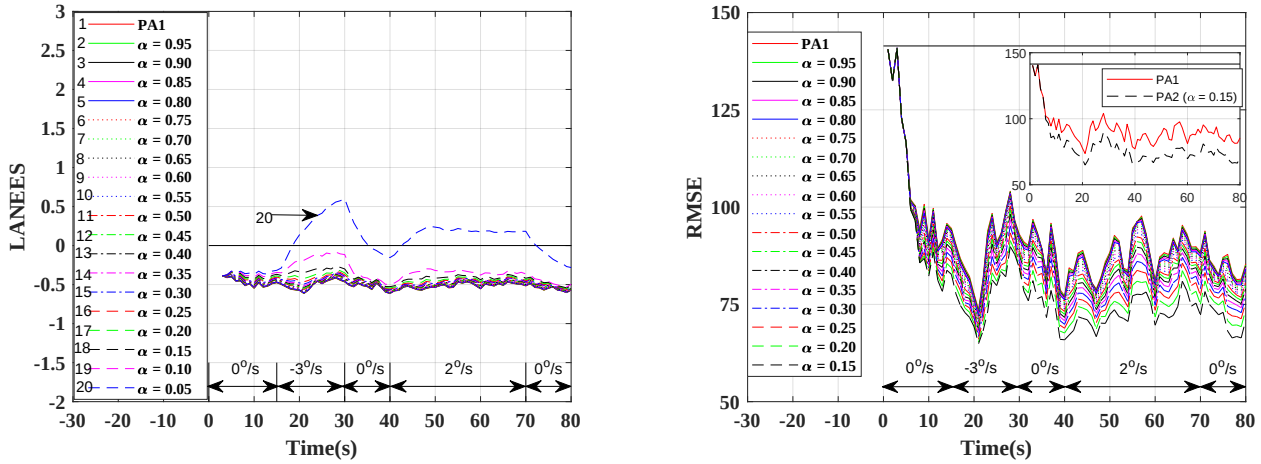
Case 1: $\sigma_s = 10\,m$. Fig. 4.7(a) shows the performances, in terms of LANEES, of PA2 for different values for α and of PA1. Fig. 4.7(b) shows a zoomed version of Fig. 4.7(a) around the value of LANEES equal to zero. From Figs. 4.7(a) and 4.7(b), it is observed that the range of α for which PA2 is consistent is from 0.7 to 0.95. The tracking accuracies, in terms of RMSE, of PA2 with this range of α and of PA1 are shown in Fig. 4.7(c). It is seen from this figure that PA2 in this range of α provides an accuracy better than that provided by PA1 and the best tracking accuracy is achieved when $\alpha = 0.7$.

Case 2: $\sigma_s = 50\,m$. Fig. 4.8(a) shows the performances, in terms of LANEES, of PA2 for different values for α and of PA1. It is observed from this figure that the range of α for which



(a) (b)

Figure 4.8: Performance of PA2 with different values of α compared to PA1 for scenario 2, when $\sigma_s = 50\,m$. (a) LANEES. (b) RMSE.



(a) (b)

Figure 4.9: Performance of PA2 with different values of α compared to PA1 for scenario 2, when $\sigma_s = 100\,m$. (a) LANEES. (b) RMSE.

PA2 is consistent is from 0.25 to 0.95. The tracking accuracies, in terms of RMSE, of PA2 with this range of α and of PA1 are shown in Fig. 4.8(b). It is seen from this figure that PA2 in this range of α provides an accuracy better than that provided by PA1 and the best tracking accuracy is achieved when $\alpha = 0.25$.

Case 3: $\sigma_s = 10\,m$. Fig. 4.9(a) shows the performances, in terms of LANEES, of PA2 for different values for α and of PA1. It is observed from this figure that the range of α for which PA2 is consistent is from 0.15 to 0.95. The tracking accuracies, in terms of RMSE, of PA2 with this range of α and of PA1 are shown in Fig. 4.9(b). It is seen from this figure that PA2 in

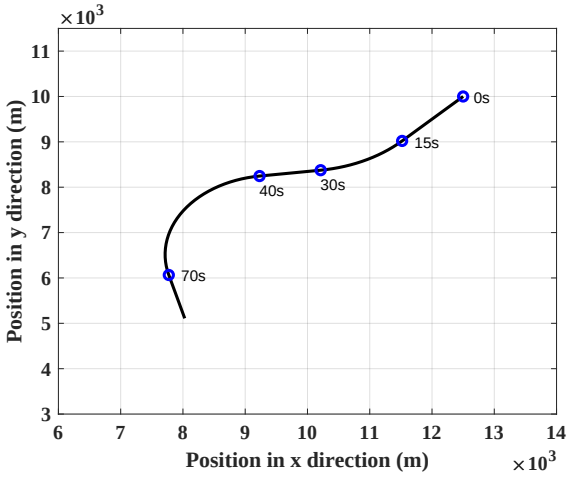


Figure 4.10: Actual target trajectory of scenario 3.

this range of α provides an accuracy better than that provided by PA1 and the best tracking accuracy is achieved when $\alpha = 0.15$.

Scenario 3

This scenario is the same as scenario 1 except that the first and second turn rates in the former are $-2.5^\circ/s$ and $3.25^\circ/s$, respectively. This scenario is depicted in Fig. 4.10. As in the previous two scenarios, we again examine three different radar systems each with a different value for the specified standard deviation of the measurement noise, $\sigma_s = 10\,m$, $\sigma_s = 50\,m$, and $\sigma_s = 100\,m$.

Case 1: $\sigma_s = 10\,m$. Fig. 4.11(a) shows the performances, in terms of LANEES, of PA2 for different values for α and of PA1. Fig. 4.11(b) shows a zoomed version of Fig. 4.11(a) around the value of LANEES equal to zero. From Figs. 4.11(a) and 4.11(b), it is observed that the range of α for which PA2 is consistent is from 0.7 to 0.95. The tracking accuracies, in terms of RMSE, of PA2 with this range of α and of PA1 are shown in Fig. 4.11(c). It is seen from this figure that PA2 in this range of α provides an accuracy better than that provided by PA1 and the best tracking accuracy is achieved when $\alpha = 0.7$.

Case 2: $\sigma_s = 50\,m$. Fig. 4.12(a) shows the performances, in terms of LANEES, of PA2 for different values for α and of PA1. It is observed from this figure that the range of α for which PA2 is consistent is from 0.25 to 0.95. The tracking accuracies, in terms of RMSE, of PA2 with this range of α and of PA1 are shown in Fig. 4.12(b). It is seen from this figure that PA2 in

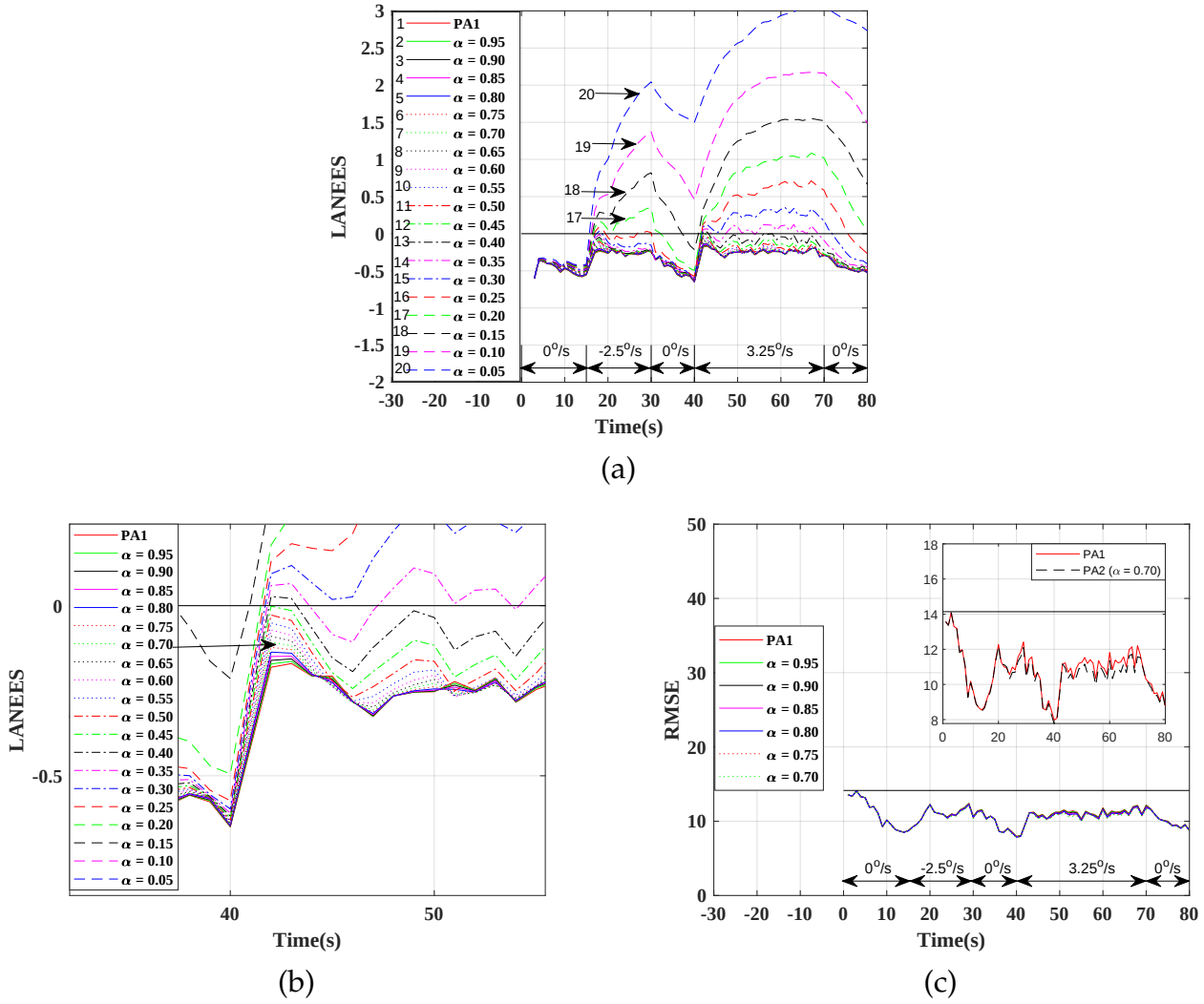
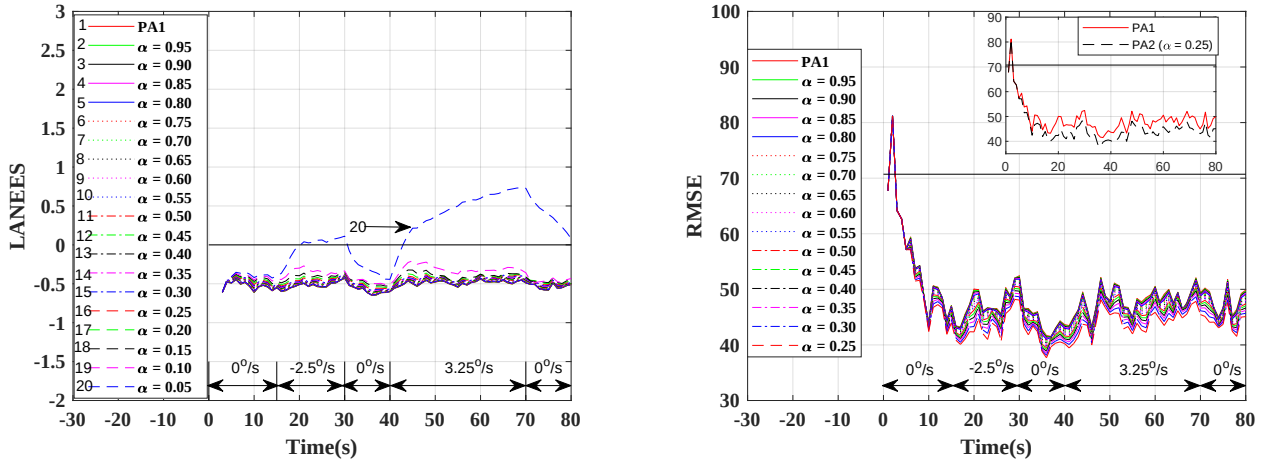


Figure 4.11: Performance of PA2 with different values of α compared to PA1 for scenario 3, when $\sigma_s = 10\,m$. (a) LANEES. (b) Zoomed version of (a) around LANEES=0. (c) RMSE.

this range of α provides an accuracy better than that provided by PA1 and the best tracking accuracy is achieved when $\alpha = 0.25$.

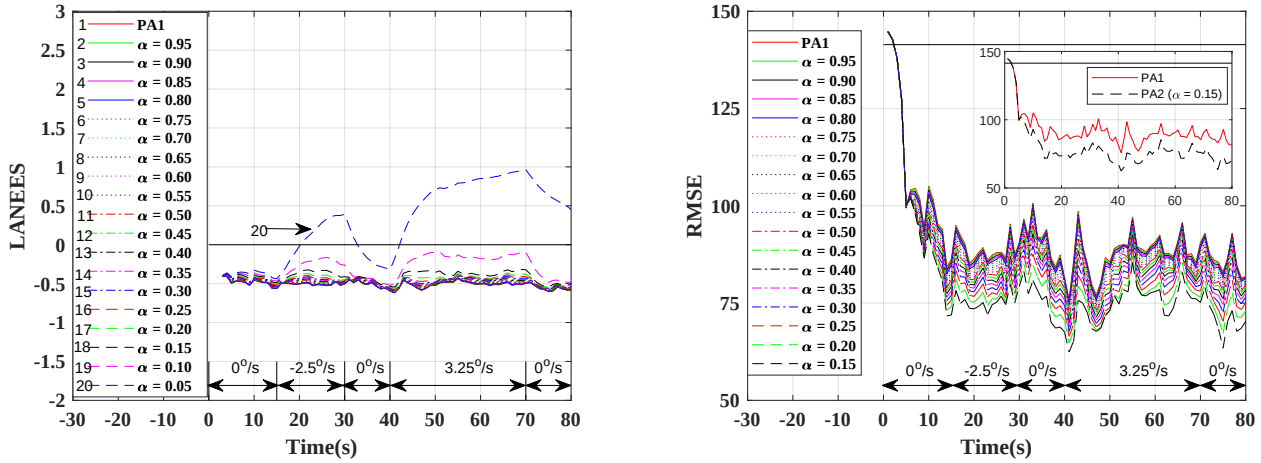
Case 3: $\sigma_s = 10\,m$. Fig. 4.13(a) shows the performances, in terms of LANEES, of PA2 for different values for α and of PA1. It is observed from this figure that the range of α for which PA2 is consistent is from 0.15 to 0.95. The tracking accuracies, in terms of RMSE, of PA2 with this range of α and of PA1 are shown in Fig. 4.13(b). It is seen from this figure that PA2 in this range of α provides an accuracy better than that provided by PA1 and the best tracking accuracy is achieved when $\alpha = 0.15$.

From the above results, we observe that the values of α for the best tracking performance



(a) (b)

Figure 4.12: Performance of PA2 with different values of α compared to PA1 for scenario 3, when $\sigma_s = 50\,m$. (a) LANEES. (b) RMSE.



(a) (b)

Figure 4.13: Performance of PA2 with different values of α compared to PA1 for scenario 3, when $\sigma_s = 100\,m$. (a) LANEES. (b) RMSE.

of PA2 are 0.7, 0.25, and 0.15, respectively, for the three cases of the specified standard deviation values of the measurement noise, namely, $\sigma_s = 10\,m$, $\sigma_s = 50\,m$, and $\sigma_s = 100\,m$, irrespective of the scenarios used in our experiments. In other words, the value of α for the best performance of PA2 depends on the value of σ_s and it does not depend on the scenario used. Consequently, we chose the optimal values of the correction factor α to be 0.7, 0.25, and 0.15 for $\sigma_s = 10\,m$, $\sigma_s = 50\,m$, and $\sigma_s = 100\,m$, respectively. In addition, in all the experiments the performance of PA2 is found to be superior to that of PA1.

In Table 4.1, we present the mean RMSE values for the target position obtained from

Table 4.1: Mean RMSE for the three scenarios individually and over all the scenarios considered together when $\sigma_s = 10\,m$

	Mean RMSE for scenario 1	Mean RMSE for scenario 2	Mean RMSE for scenario 3	Mean RMSE value averaged over the means of all the scenarios
$PA1$	11.0146	10.6704	10.7414	10.8088
$PA2(\alpha = 0.7)$	10.6414	10.5293	10.5366	10.5691
Percentage reduction in RMSE	3.39%	1.32%	1.91%	2.2%

Table 4.2: Mean RMSE for the three scenarios individually and over all the scenarios considered together when $\sigma_s = 50\,m$

	Mean RMSE for scenario 1	Mean RMSE for scenario 2	Mean RMSE for scenario 3	Mean RMSE value averaged over the means of all the scenarios
$PA1$	50.3493	47.4176	48.9359	48.9009
$PA2(\alpha = 0.25)$	46.8720	44.1363	45.4655	45.4913
Percentage reduction in RMSE	6.91%	6.92%	7.09%	6.97%

both PA1 and PA2 (with $\alpha = 0.7$) for each of the three scenarios individually, as well as the mean RMSE value averaged over all the scenarios for the case of $\sigma_s = 10\,m$. The mean RMSE value for a given scenario is obtained as the sum of the RMSE values at the various sampling instants divided by the total number of samples over the tracking period. The mean RMSE value averaged over the three scenarios is obtained as the average of the mean RMSE values for the three scenarios. The last row in the table gives the percentage reduction in RMSE resulting from taking the measurement noise into account in the estimation of the turn rate. This percentage is calculated as the difference between the mean RMSE value for PA1 and PA2 divided by the former mean RMSE value. Tables 4.2 and 4.3 give the

Table 4.3: Mean RMSE for the three scenarios individually and over all the scenarios considered together when $\sigma_s = 100\,m$

	Mean RMSE for scenario 1	Mean RMSE for scenario 2	Mean RMSE for scenario 3	Mean RMSE value averaged over the means of all the scenarios
$PA1$	93.4634	91.8183	92.0903	92.4573
$PA2(\alpha = 0.15)$	82.8196	79.0223	79.4214	80.4211
Percentage reduction in RMSE	11.39%	13.94%	13.76%	13.03%

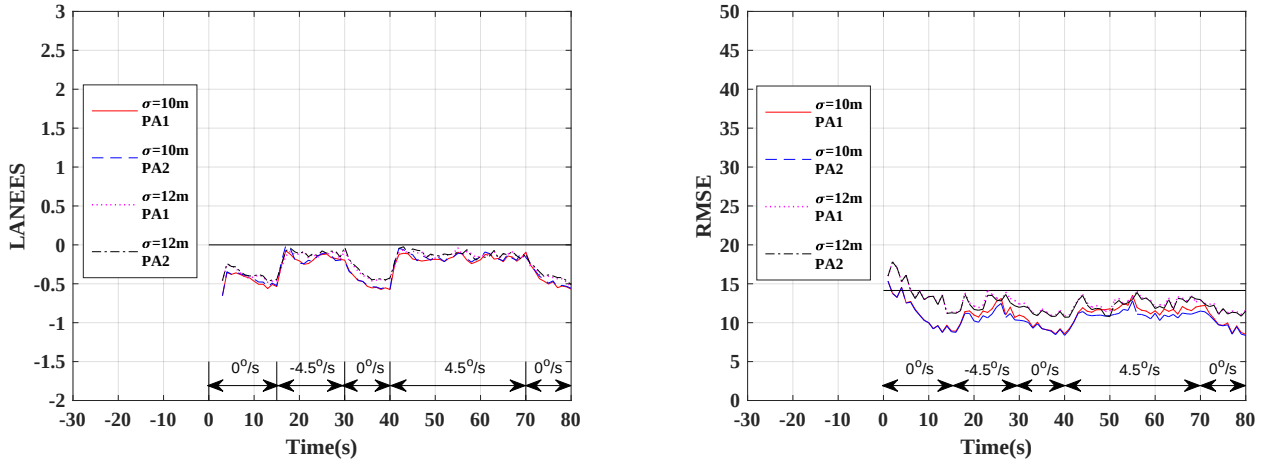
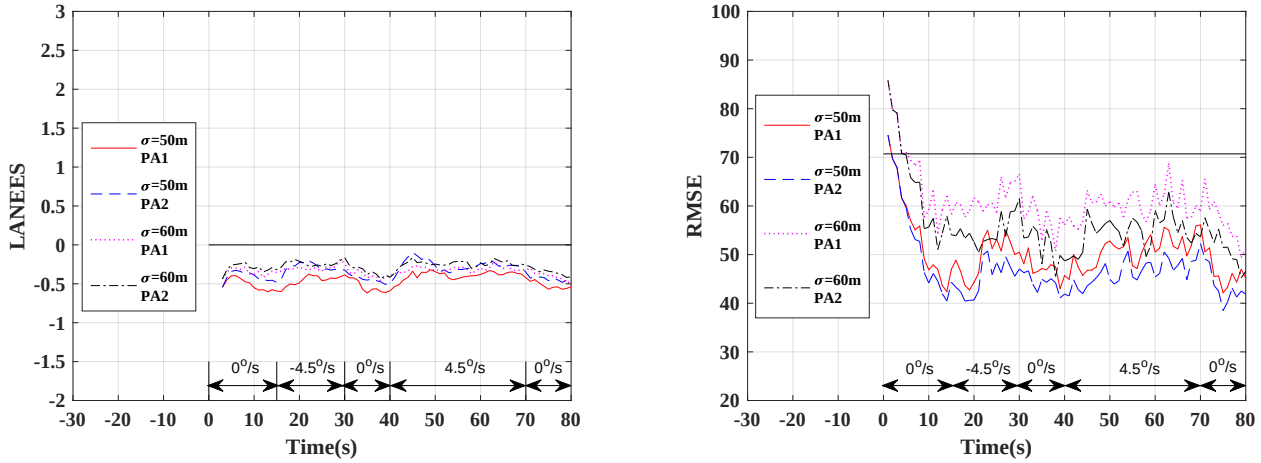


Figure 4.14: Performance of PA1 and PA2 for scenario 1, when $\sigma_a = 10\,m$ and $12\,m$. (a) LANEES. (b) RMSE.

corresponding results when $\sigma_s = 50\,m$ and $100\,m$, respectively. It is seen from results of the three tables that there are reductions in the RMSE values both for individual scenarios and for all the scenarios taken together irrespective of the value of the standard deviation, σ_s, of the measurement noise. However, this reduction becomes more substantial as the value of σ_s increases. In other words, a significant improvement in the tracking accuracy can be achieved by using PA2 with a value of α suitably determined, particularly when the value of σ_s is large.

4.3 Impact of the Actual Measurement Noise Level Rising above the Specified Level

In this section, we study the impact of an increase in the value of σ_a over that of σ_s on the tracking performance of PA2. The same three cases of σ_s, namely, $\sigma_s = 10\,m$, $\sigma_s = 50\,m$, and $\sigma_s = 100\,m$ are considered for the same three scenarios, when σ_a rises above σ_s by 20%.



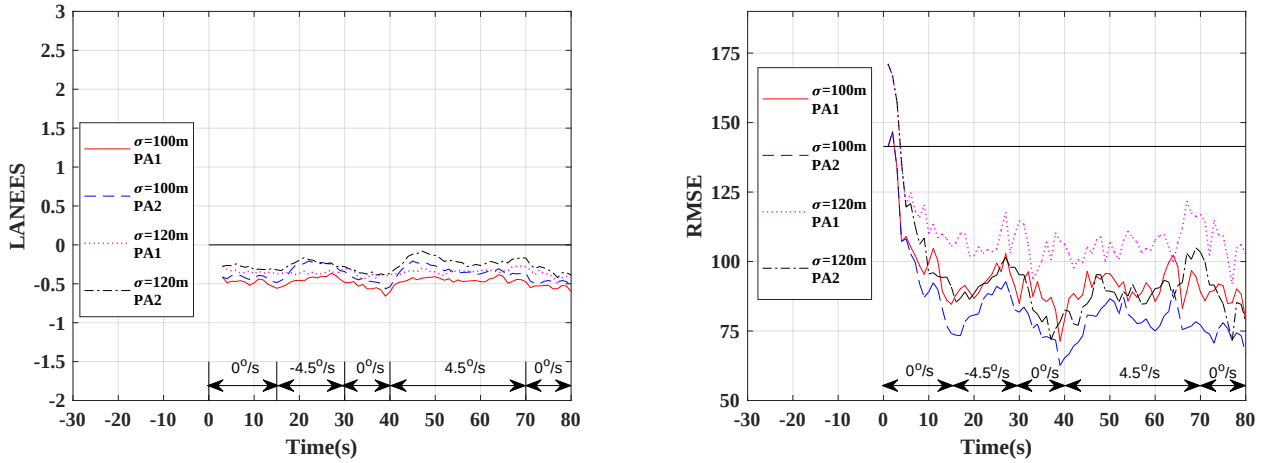
(a) (b)

Figure 4.15: Performance of PA1 and PA2 for scenario 1, when $\sigma_a = 50\,m$ and $60\,m$. (a) LANEES. (b) RMSE.

Scenario 1

Case 1: $\sigma_s = 10\,m$ *and* $\sigma_a = 12\,m$. For this case, PA2 is used with $\sigma_s = 10\,m$, for which the value of α was determined to be 0.7 in Section 4.2. The performance of PA2 in terms of LANEES and RMSE for $\sigma_a = 12\,m$ is shown in Figs. 4.14(a) and 4.14(b), respectively. For the sake of comparison, these figures also include the performance results of PA1 and PA2 corresponding to $\sigma_a = \sigma_s = 10\,m$ and PA1 corresponding to $\sigma_s = 10\,m$ and $\sigma_a = 12\,m$. It is seen from Fig. 4.14(a) that both the algorithms are consistent throughout the tracking period. It is seen from the RMSE plot in Fig. 4.14(b) that the tracking accuracy of PA2 benefits from the use of α for $\sigma_a = 12\,m$, as it did for $\sigma_a = 10\,m$.

Case 2: $\sigma_s = 50\,m$ *and* $\sigma_a = 60\,m$. For this case, PA2 is used with $\sigma_s = 50\,m$, for which the value of α was determined to be 0.25 in Section 4.2. The performance of PA2 in terms of LANEES and RMSE for $\sigma_a = 60\,m$ is shown in Figs. 4.15(a) and 4.15(b), respectively. For the sake of comparison, these figures also include the performance results of PA1 and PA2 corresponding to $\sigma_a = \sigma_s = 50\,m$ and PA1 corresponding to $\sigma_s = 50\,m$ and $\sigma_a = 60\,m$. It is seen from Fig. 4.15(a) that both the algorithms are consistent throughout the tracking period. It is seen from the RMSE plot in Fig. 4.15(b) that the tracking accuracy of PA2 benefits from the use of α for $\sigma_a = 60\,m$, as it did for $\sigma_a = 50\,m$. The improvement in the mean value of



(a) (b)

Figure 4.16: Performance of PA1 and PA2 for scenario 1, when $\sigma_a = 100\,m$ and $120\,m$. (a) LANEES. (b) RMSE.

RMSE in this case is larger than that in Case 1 for the same 20% increase in the value of σ_a over that of σ_s.

Case 3: $\sigma_s = 100\,m$ *and* $\sigma_a = 120\,m$. For this case, PA2 is used with $\sigma_s = 100\,m$, for which the value of α was determined to be 0.15 in Section 4.2. The performance of PA2 in terms of LANEES and RMSE for $\sigma_a = 120\,m$ is shown in Figs. 4.16(a) and 4.16(b), respectively. For the sake of comparison, these figures also include the performance results of PA1 and PA2 corresponding to $\sigma_a = \sigma_s = 100\,m$ and PA1 corresponding to $\sigma_s = 100\,m$ and $\sigma_a = 120\,m$. It is seen from Fig. 4.16(a) that both the algorithms are consistent throughout the tracking period. It is seen from the RMSE plot in Fig. 4.16(b) that the tracking accuracy of PA2 benefits from the use of α for $\sigma_a = 120\,m$, as it did for $\sigma_a = 100\,m$. The improvement in the mean value of RMSE in this case is even larger than that in Case 2 for the same 20% increase in the value of σ_a over that of σ_s.

Scenario 2

Case 1: $\sigma_s = 10\,m$ *and* $\sigma_a = 12\,m$. For this case, PA2 is used with $\sigma_s = 10\,m$, for which the value of α was determined to be 0.7 in Section 4.2. The performance of PA2 in terms of LANEES and RMSE for $\sigma_a = 12\,m$ is shown in Figs. 4.17(a) and 4.17(b), respectively. For the sake of comparison, these figures also include the performance results of PA1 and PA2 corresponding to $\sigma_a = \sigma_s = 10\,m$ and PA1 corresponding to $\sigma_s = 10\,m$ and $\sigma_a = 12\,m$. It is

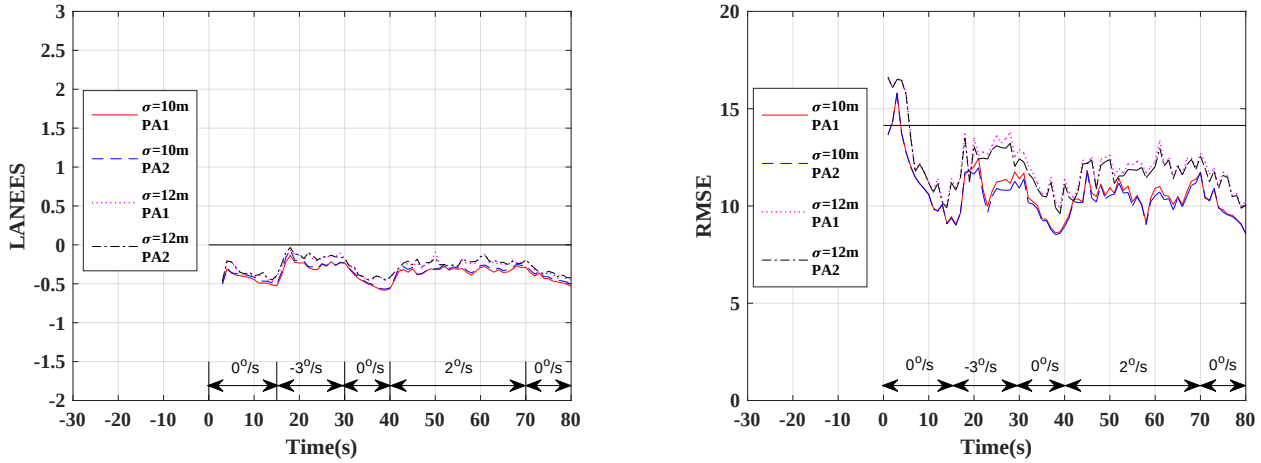


Figure 4.17: Performance of PA1 and PA2 for scenario 2, when $\sigma_a = 10\,m$ and $12\,m$. (a) LANEES. (b) RMSE.

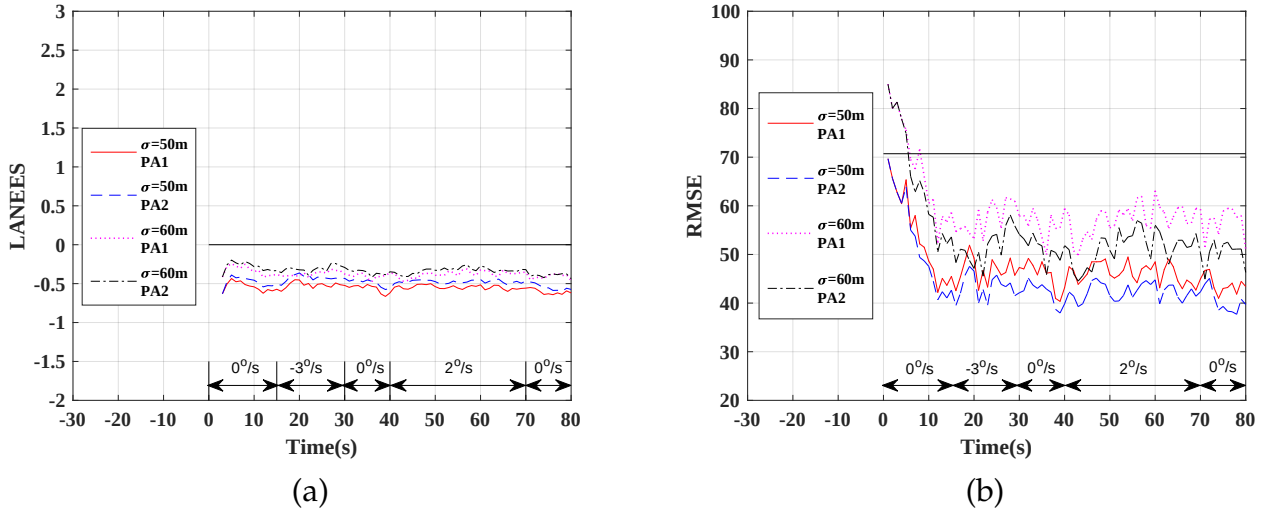


Figure 4.18: Performance of PA1 and PA2 for scenario 2, when $\sigma_a = 50\,m$ and $60\,m$. (a) LANEES. (b) RMSE.

seen from Fig. 4.17(a) that both the algorithms are consistent throughout the tracking period. It is seen from the RMSE plot in Fig. 4.17(b) that the tracking accuracy of PA2 benefits from the use of α for $\sigma_a = 12\,m$, as it did for $\sigma_a = 10\,m$.

Case 2: $\sigma_s = 50\,m$ *and* $\sigma_a = 60\,m$. For this case, PA2 is used with $\sigma_s = 50\,m$, for which the value of α was determined to be 0.25 in Section 4.2. The performance of PA2 in terms of LANEES and RMSE for $\sigma_a = 60\,m$ is shown in Figs. 4.18(a) and 4.18(b), respectively. For the sake of comparison, these figures also include the performance results of PA1 and PA2 corresponding to $\sigma_a = \sigma_s = 50\,m$ and PA1 corresponding to $\sigma_s = 50\,m$ and $\sigma_a = 60\,m$. It is

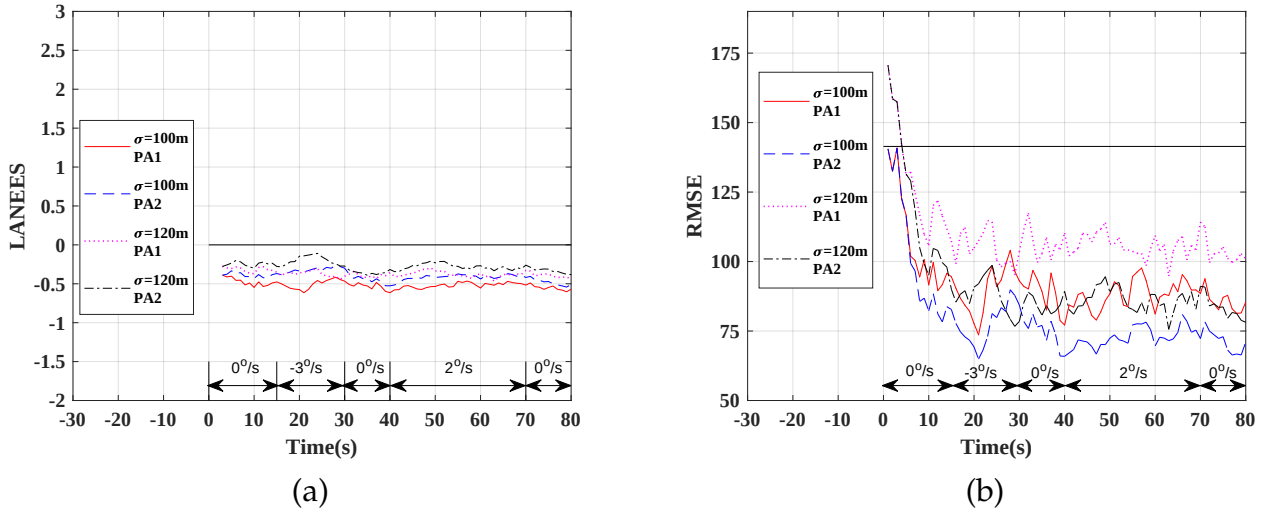


Figure 4.19: Performance of PA1 and PA2 for scenario 2, when $\sigma_a = 100\,m$ and $120\,m$. (a) LANEES. (b) RMSE.

seen from Fig. 4.18(a) that both the algorithms are consistent throughout the tracking period. It is seen from the RMSE plot in Fig. 4.18(b) that the tracking accuracy of PA2 benefits from the use of α for $\sigma_a = 60\,m$, as it did for $\sigma_a = 50\,m$. The improvement in the mean value of RMSE in this case is larger than that in Case 1 for the same 20% increase in the value of σ_a over that of σ_s.

Case 3: $\sigma_s = 100\,m$ *and* $\sigma_a = 120\,m$. For this case, PA2 is used with $\sigma_s = 100\,m$, for which the value of α was determined to be 0.15 in Section 4.2. The performance of PA2 in terms of LANEES and RMSE for $\sigma_a = 120\,m$ is shown in Figs. 4.19(a) and 4.19(b), respectively. For the sake of comparison, these figures also include the performance results of PA1 and PA2 corresponding to $\sigma_a = \sigma_s = 100\,m$ and PA1 corresponding to $\sigma_s = 100\,m$ and $\sigma_a = 120\,m$. It is seen from Fig. 4.19(a) that both the algorithms are consistent throughout the tracking period. It is seen from the RMSE plot in Fig. 4.19(b) that the tracking accuracy of PA2 benefits from the use of α for $\sigma_a = 120\,m$, as it did for $\sigma_a = 100\,m$. The improvement in the mean value of RMSE in this case is even larger than that in Case 2 for the same 20% increase in the value of σ_a over that of σ_s.

Scenario 3

Case 1: $\sigma_s = 10\,m$ *and* $\sigma_a = 12\,m$. For this case, PA2 is used with $\sigma_s = 10\,m$, for which the value of α was determined to be 0.7 in Section 4.2. The performance of PA2 in terms of

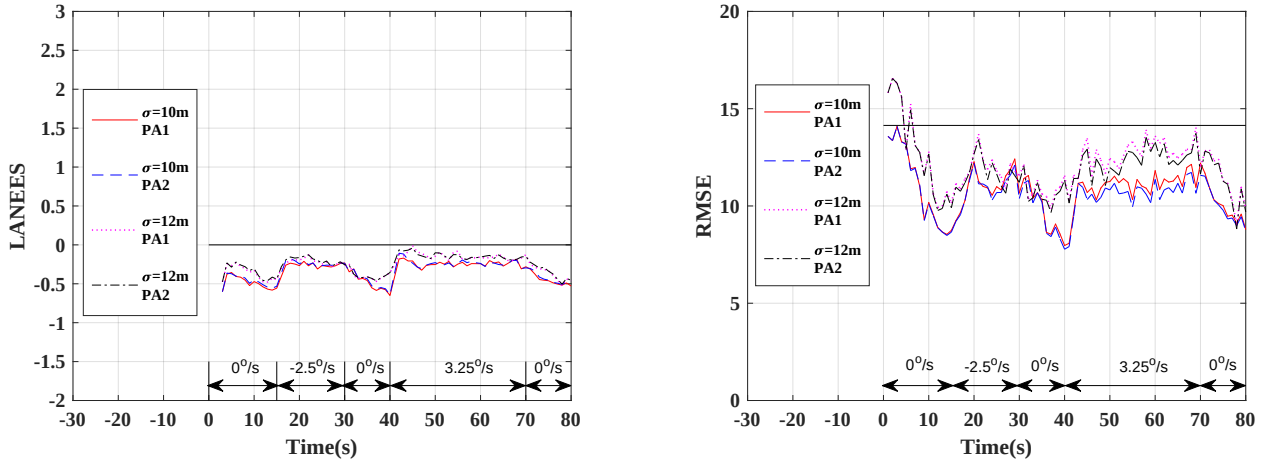


Figure 4.20: Performance of PA1 and PA2 for scenario 3, when $\sigma_a = 10\,m$ and $12\,m$. (a) LANEES. (b) RMSE.

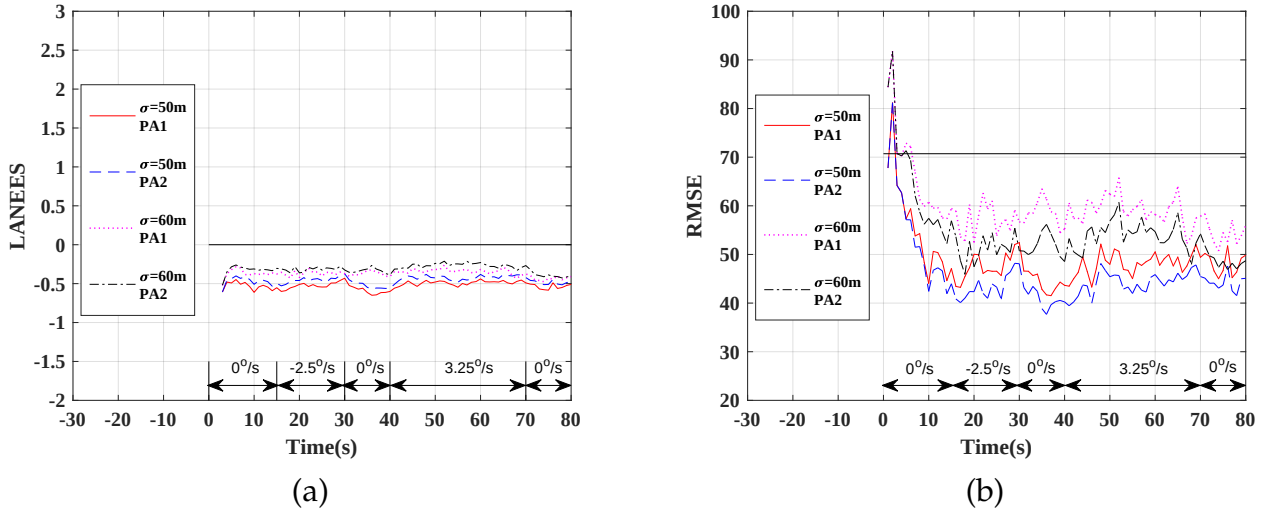
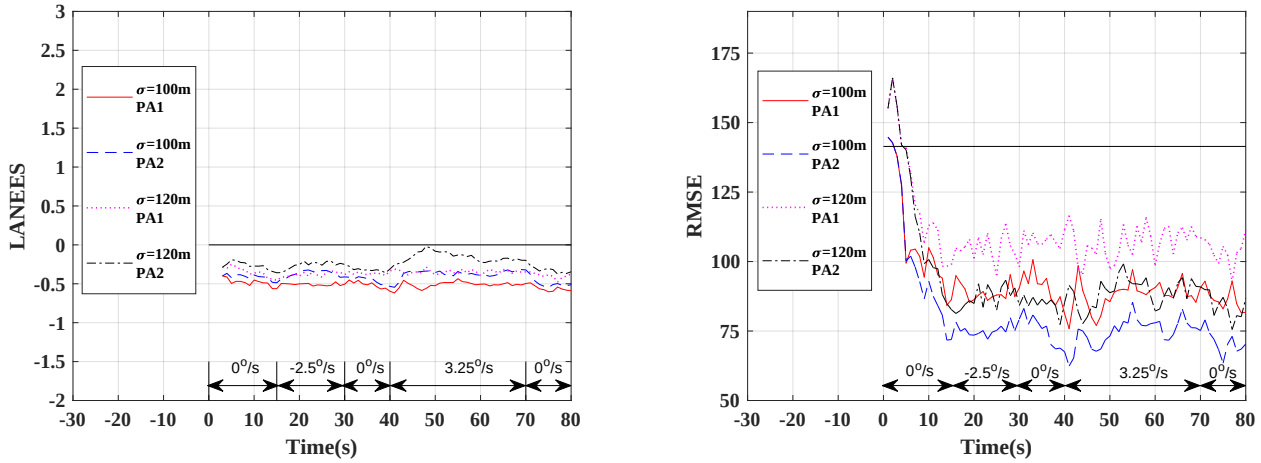


Figure 4.21: Performance of PA1 and PA2 for scenario 3, when $\sigma_a = 50\,m$ and $60\,m$. (a) LANEES. (b) RMSE.

LANEES and RMSE for $\sigma_a = 12\,m$ is shown in Figs. 4.20(a) and 4.20(b), respectively. For the sake of comparison, these figures also include the performance results of PA1 and PA2 corresponding to $\sigma_a = \sigma_s = 10\,m$ and PA1 corresponding to $\sigma_s = 10\,m$ and $\sigma_a = 12\,m$. It is seen from Fig. 4.20(a) that both the algorithms are consistent throughout the tracking period. It is seen from the RMSE plot in Fig. 4.20(b) that the tracking accuracy of PA2 benefits from the use of α for $\sigma_a = 12\,m$, as it did for $\sigma_a = 10\,m$.

Case 2: $\sigma_s = 50\,m$ *and* $\sigma_a = 60\,m$. For this case, PA2 is used with $\sigma_s = 50\,m$, for which the value of α was determined to be 0.25 in Section 4.2. The performance of PA2 in terms



(a) (b)

Figure 4.22: Performance of PA1 and PA2 for scenario 3, when $\sigma_a = 100\,m$ and $120\,m$. (a) LANEES. (b) RMSE.

of LANEES and RMSE for $\sigma_a = 60\,m$ is shown in Figs. 4.21(a) and 4.21(b), respectively. For the sake of comparison, these figures also include the performance results of PA1 and PA2 corresponding to $\sigma_a = \sigma_s = 50\,m$ and PA1 corresponding to $\sigma_s = 50\,m$ and $\sigma_a = 60\,m$. It is seen from Fig. 4.21(a) that both the algorithms are consistent throughout the tracking period. It is seen from the RMSE plot in Fig. 4.21(b) that the tracking accuracy of PA2 benefits from the use of α for $\sigma_a = 60\,m$, as it did for $\sigma_a = 50\,m$. The improvement in the mean value of RMSE in this case is larger than that in Case 1 for the same 20% increase in the value of σ_a over that of σ_s.

Case 3: $\sigma_s = 100\,m$ *and* $\sigma_a = 120\,m$. For this case, PA2 is used with $\sigma_s = 100\,m$, for which the value of α was determined to be 0.15 in Section 4.2. The performance of PA2 in terms of LANEES and RMSE for $\sigma_a = 120\,m$ is shown in Figs. 4.22(a) and 4.22(b), respectively. For the sake of comparison, these figures also include the performance results of PA1 and PA2 corresponding to $\sigma_a = \sigma_s = 100\,m$ and PA1 corresponding to $\sigma_s = 100\,m$ and $\sigma_a = 120\,m$. It is seen from Fig. 4.22(a) that both the algorithms are consistent throughout the tracking period. It is seen from the RMSE plot in Fig. 4.22(b) that the tracking accuracy of PA2 benefits from the use of α for $\sigma_a = 120\,m$, as it did for $\sigma_a = 100\,m$. The improvement in the mean value of RMSE in this case is even larger than that in Case 2 for the same 20% increase in the value of σ_a over that of σ_s.

Table 4.4: Mean RMSE for the three scenarios individually and over all the scenarios considered together, when $\sigma_s = 10\,m$ and $\sigma_a = 12\,m$

	Mean RMSE for scenario 1	Mean RMSE for scenario 2	Mean RMSE for scenario 3	Mean RMSE value averaged over the means of all the scenarios
$PA1$	13.2750	12.1387	12.1570	12.5236
$PA2(\alpha = 0.7)$	12.9456	11.9376	11.9252	12.2695
Percentage reduction in RMSE	2.48%	1.66%	1.91%	2.2%

Table 4.5: Mean RMSE for the three scenarios individually and over all the scenarios considered together, when $\sigma_s = 50\,m$ and $\sigma_a = 60\,m$

	Mean RMSE for scenario 1	Mean RMSE for scenario 2	Mean RMSE for scenario 3	Mean RMSE value averaged over the means of all the scenarios
$PA1$	60.9185	59.1473	59.5658	58.8772
$PA2(\alpha = 0.25)$	55.8272	53.9228	54.5296	54.7599
Percentage reduction in RMSE	8.36%	8.83%	8.45%	8.55%

The above analysis of the performance results of PA2 with three different scenarios shows that the use of α improves the RMSE performance while still preserving its consistency even in the case when the value of the standard deviation, σ_a, of the actual measurement noise rises over the value of the standard deviation, σ_s, of the specified measurement noise of the radar system by as much as 20%, irrespective of the three values of σ_s considered.

In Table 4.4, we present the mean RMSE values for the target position obtained from both PA1 and PA2 (with $\alpha = 0.7$) for each of the three scenarios individually, as well as that averaged over all the scenarios for the case when the level of σ_a is 20% higher than $\sigma_s = 10\,m$. Tables 4.5 and 4.6 give the corresponding results when the level of σ_a is 20% higher than

Table 4.6: Mean RMSE for the three scenarios individually and over all the scenarios considered together, when $\sigma_s = 100\,m$ and $\sigma_a = 120\,m$

	Mean RMSE for scenario 1	Mean RMSE for scenario 2	Mean RMSE for scenario 3	Mean RMSE value averaged over the means of all the scenarios
$PA1$	109.9822	109.6458	109.5877	109.7386
$PA2(\alpha = 0.15)$	94.0818	92.2254	92.8827	93.0633
Percentage reduction in RMSE	14.46%	15.89%	15.24%	15.20%

$\sigma_s = 50\,m$ and $\sigma_s = 100\,m$, respectively. It is seen from the results of these three tables that there are reductions in RMSE values both for individual scenarios and for all the scenarios taken together irrespective of the value of the standard deviation, σ_s, of the measurement noise. However, this reduction becomes more substantial as the value of σ_s increases. In other words, a significant improvement in the tracking accuracy can be achieved by using PA2 with a value of α suitably determined even when the value of σ_a becomes larger than that of σ_s by 20%, particularly when the value of σ_s is large.

4.4 Summary

In this chapter, an algorithm has been developed within the IMM framework for tracking a single maneuvering target that takes into account the level on the measurement noise in estimating the target turn rate by augmenting the estimated turn rate using a multiplying factor. It has been shown that this algorithm provides a tracking performance that is much superior to that provided by the algorithm of Chapter 3, since the estimated value of the turn rate is more accurate for a given level of the measurement noise. It has also been shown that this algorithm can handle an increase in the measurement noise level better than the algorithm of Chapter 3 does.

The algorithm proposed in this chapter is well suited in real life situations in which the turn rate of the target is generally not known *a priori*, and the level of the measurement noise may rise above the level for which the algorithm is designed.

Chapter 5

Tracking Closely-Spaced non-Maneuvering Multiple Targets

5.1 Introduction

In an algorithm for tracking multiple targets, a track is constructed for each of the targets based on the previous measurements and suitably updated after the arrival of new measurements [86, 87]. At a given time instant k, the algorithm stars with finding the predicted position of each target using a motion model and its estimated position in the previous time instant. Since at a given time instant, a number of measurements are received, each measurement needs to be suitably assigned to the various targets. This task is called measurement assignment, which is performed in two steps, a gating step and a data association step. Gating at time instant k refers to specifying for each target an area centered at the predicted position of the targets so that only those measurements received at time instant k and fall inside the gates are used by the algorithm [88]. Data association is the task of assigning each of the gated measurements the probability of being associated with the various targets [89]. The assigned measurements are then used to estimate the updated position of the targets in the current time instant. These estimates of the target positions are used to find the predicted position of the targets for the next time instant [90].

The most used algorithm to track a number of targets known as the joint probability date association (JPDA) [59]. This algorithm provides a very good tracking performance when targets are widely spaced; however, the tracking performance deteriorates and coalescence may occurs when the targets are closely spaced. It is to be noted that when the targets are closely spaced, the spatial distribution of the measurements is such that one or more measurements lie in the overlapped regions of the target gates. The JPDA algorithm allocates weights to the measurements by finding the probability of all possible hypotheses for measurement-target association regardless of a measurement falling inside or outside of the overlapped regions of the gates.

In this chapter, we conduct a systematic study on the impact of the spacing between the targets as well as when the targets make abrupt turns with sharp angles on the tracking performance of the JPDA algorithm. Then, we propose a new multiple target tracking algorithm within the JPDA framework to track the targets no matter whether they are widely or closely-spaced based on the spatial distribution of the measurements within a target gate [91]. Extensive experiments are performed and demonstrate the effectiveness of the proposed algorithm over the conventional JPDA algorithm.

5.2 Problem Formulation

In this section, the problem of tracking multiple targets in a clutter environment is formulated in a two-dimensional space, assuming the number of targets is known *a priori*. The radar system is assumed to provide only the position measurements, and that each target can generate only one measurement per sampling period. The target state and the corresponding measurement for the time step k are modeled using the state space model, as mentioned in section 2.1. The receiver detects each measurement with a probability of detection P_D. These received measurements originate not only from the targets but also from the clutter. The clutter is assumed to have a Poisson distribution with intensity λ over the surveillance region with an area A. These clutters are uniformly distributed over A . The

average number of clutters C per scan is given by [65]

$$C = \lambda A \tag{5.1}$$

For the time step k, the measurements that fall inside the gates of the targets are denoted by $\mathbf{z}_1(k), \mathbf{z}_2(k), \cdots, \mathbf{z}_{J_k}(k)$, where J_k is their total number.

5.3 Proposed Algorithm for Tracking Multiple Targets

In the JPDA algorithm of [59], a weight is assigned to each of the measurements falling within the gate of a target in order to associate this measurement to the target. When the targets are closely spaced, one or more of the measurements may lie inside an overlapped region of the gates. In this case, a measurement that lies inside the overlapped region needs to be associated with more than one target, i.e., with all the targets forming the overlapped region. This is different from a non-overlapped measurement, which is associated with only one target, i.e., with the target of the gate in which this measurement falls. However, in the JPDA algorithm, the same procedure is used for determining the weights for all the measurements within the gate regardless of whether a measurement lies in an overlapped region or not, that is, the procedure of finding the probabilities of all possible hypotheses for the measurement-target association under consideration and adding them up to find the value of the weight of this measurement-target pair. Since a measurement that is closer to the predicted position of the target is more likely to have been resulted from this target, by assigning a larger weight for the association of this measurement-target pair can improve the tracking performance. However, in the JPDA algorithm the spatial distribution of the measurements falling inside the gate with respect to the predicted position of the targets is not taken into consideration for determining the weight values for the measurement-target pairs.

In this section, we propose an algorithm for tracking multiple targets within the JPDA framework, in which the notion of distance between a measurement and the predicted

position of the target is incorporated in the method of determining the weights for the measurement-target pairs so that a higher weight is assigned to the measurement that is closer to the predicted position of the target. This can reduce the tracking error, which is extremely important, when the targets become closely spaced and the estimated tracks tend to merge. The proposed algorithm for tracking multiple targets is now described below.

1. **Finding gated measurements**: As in the JPDA algorithm [92], the output state of the ith target is used from the previous time step to find its predicted position in the kth time step, $\tilde{\mathbf{z}}_i(k)$. Then, a gate around the predicted position of each of the targets is formed to find the gated measurements for each of the targets. The gating is based on Mahalanobis distance [93] between the predicted target position $\tilde{\mathbf{z}}_i(k)$ and the measurement position $\mathbf{z}_j$. A measurement $\mathbf{z}_j$ is considered to be gated for the target i when it satisfies the gate test given by [38]

$$(\boldsymbol{\epsilon}_{ij}(k))^t(\mathbf{S}_i(k))^{-1}\boldsymbol{\epsilon}_{ij}(k) \leq \gamma \tag{5.2}$$

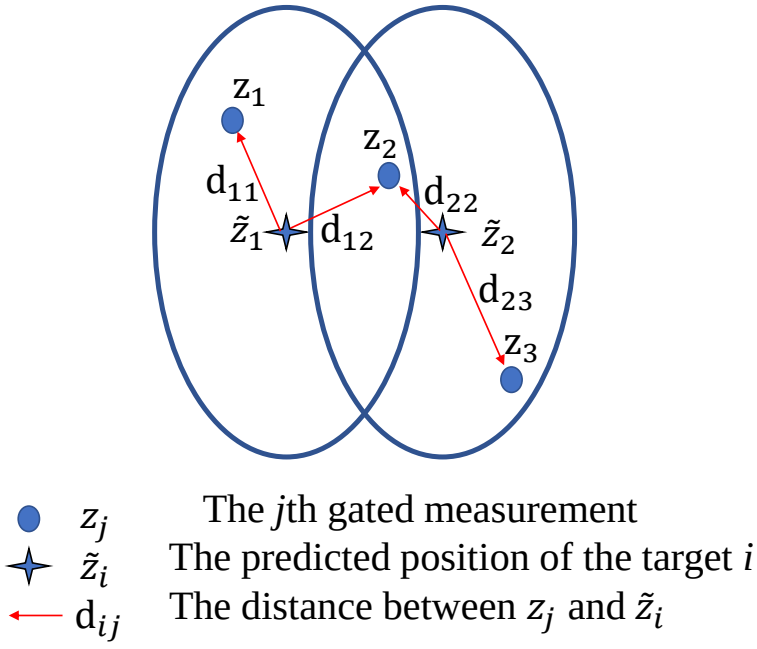


Fig. 5.1: Two overlapped gates and four gated measurements.

where $\epsilon_{ij}(k)$ is the difference between a received measurement and the predicted position of the target i , $\mathbf{S}_i(k)$ is the innovation covariance matrix of the target i at the time step k, and γ is the gate threshold. This gating process is performed for each of the targets.

In order to illustrate the gating procedure, we consider a case of two targets and four measurements, as shown in Fig.5.1, where the gates are overlapping. Note that in this example it is assumed that the measurements $\mathbf{z}_1$ and $\mathbf{z}_3$ lie in the two different regions, whereas $\mathbf{z}_2$ is in the overlapped region.

2. **Distance calculation**: For each target i we track, we compute the Mahalanobis distance $d_{ij}(k)$, between the predicted position $\tilde{\mathbf{z}}_i(k)$ of the target i and each of its gated measurements $\mathbf{z}_j$, given by

$$d_{ij}(k) = \sqrt{(\mathbf{z}_j(k) - \tilde{\mathbf{z}}_i(k))^t (\mathbf{S}_i(k))^{-1} (\mathbf{z}_j(k) - \tilde{\mathbf{z}}_i(k))} \tag{5.3}$$

The various distances are marked in Fig.5.1. The distances between $\tilde{\mathbf{z}}_1(k)$ and its gated measurements $\mathbf{z}_1$ and $\mathbf{z}_2$ are d_{11} and d_{12}, respectively, whereas the distances between $\tilde{\mathbf{z}}_2(k)$ and its gated measurements $\mathbf{z}_2$ and $\mathbf{z}_3$ are d_{22} and d_{23}, respectively.

3. **Construction of validation matrix**: A validation matrix $\mathbf{\Omega}(k)$ of order $J \times (I+1)$, where the rows $1, \cdots, J$ correspond to the gated measurements $\mathbf{z}_1$, $\mathbf{z}_2$, $\cdots$, $\mathbf{z}_J$, and the first column corresponds to the clutters and the columns $2, \cdots, (I+1)$ correspond to the targets $1, \cdots, I$. The elements ρ_{ji}, $j = 1, \cdots, J$; $i = 0, \cdots, I$ ($i = 0$ corresponds to "no target", i.e., clutters) of the matrix $\mathbf{\Omega}(k)$ are defined as follows.

$$\rho_{ji} = \begin{cases} 1 & \textit{when } i = 0 \\ 1 & \textit{when measurment } j \textit{ is associated with target } i \\ 0 & \textit{otherwise} \end{cases} \tag{5.4}$$

For the case considered in Fig.5.1, the validation matrix is given by

$$\mathbf{\Omega}(k) = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \tag{5.5}$$

4. **Hypotheses generation**: All possible hypotheses h_l for measurement-target association are enumerated, where $l = 1, \cdots, L$, and L is the total number of the hypotheses. Each hypothesis is a binary matrix, where the rows correspond to the measurements $1, \cdots, J$ and columns to targets $i = 0, 1, \cdots, I$, where $i = 0$ refers to "no target", i.e., clutters, that satisfies two constraints. The first constraint is that each row must have one and only one non-zero element. The second constraint is that each column can have only one non-zero element except for the first column, which must have at least one non-zero element in order to account for the possibility that more than one of the gated measurements may have resulted from clutters. For the case considered in Fig.5.1, there are 8 hypotheses, and these are given below.

$$\begin{cases} h_1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} h_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} h_3 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} h_4 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \\ h_5 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} h_6 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} h_7 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} h_8 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \end{cases} \tag{5.6}$$

5. **Probability computation for the generated hypotheses**: To compute the probability Pr_{h_l} for each hypothesis h_l, we introduce two binary indicators, ε_{li} and τ_{lj}. The indicator ε_{li} is given a value of 1 or 0 depending on whether or not to target i any of the

measurements has been assigned. The indicator τ_{lj} is given a value 1 or 0 depending on whether the measurement j is assigned to any target. The number of the gated measurements that are not assigned to any of the targets is denoted by $\phi_l = J - \sum_{j=1}^{J} \tau_{lj}$. The probability of the hypothesis h_l is given by [59]

$$Pr_{h_l} = \frac{\phi_l\,!}{A^{\phi_l}} \prod_{j=1}^{J} (\Lambda_{ij})^{\tau_{lj}} \prod_{i=1}^{I} (P_D)^{\varepsilon_{li}} (1 - P_D)^{1-\varepsilon_{li}} \tag{5.7}$$

where

$$\Lambda_{ij} = \frac{1}{\sqrt{2\pi \mathbf{S}_i(k)}} e^{-\frac{1}{2}(\boldsymbol{\beta}_{ij}(k))^t \mathbf{S}_i(k) \boldsymbol{\beta}_{ij}(k)} \tag{5.8}$$

and $\boldsymbol{\beta}_{ij}(k)$ is the residual between the measurement $\mathbf{z}_j(k)$ and the predicted position $\tilde{\mathbf{z}}_i(k)$ of the target i. For the example considered in Fig. 5.1, the probability of the hypothesis h_7 is given by

$$Pr_{h_7} = \frac{1}{A} \Lambda_{11} \Lambda_{23} (P_D)^2 \tag{5.9}$$

The normalized probability $Pr_{h_{ln}}$ for the hypothesis h_l is

$$Pr_{h_{ln}} = \frac{Pr_{h_l}}{\sum_{l=1}^{L} Pr_{h_l}} \tag{5.10}$$

6. **Allocation of measurement weights**: The weight ϖ_{ij} of the gated measurement $\mathbf{z}_j(k)$ assigned to target i is computed as the sum of the normalized probabilities of all those hypotheses in which their ρ_{ji} elements are unity, and it is given by

$$\varpi_{ij}(k) = \sum_{l \; s.t. \; \rho_{ji}=1} Pr_{h_{ln}} \tag{5.11}$$

Since the sum of all the normalized weights must equal unity, the remaining weight $\varpi_{ic}(k)$ is assigned to the clutters and is given by

$$\varpi_{ic}(k) = 1 - \sum_{j=1}^{J} \varpi_{ij}(k) \tag{5.12}$$

For the example considered in Fig.5.1, for instance, the measurement z_2 is assigned to target 1 only in two out of the eight hypotheses, namely, h_5 and h_6.Thus, the weight ϖ_{12} is given by

$$\varpi_{12}(k) = Pr_{h_{5n}} + Pr_{h_{6n}} \tag{5.13}$$

7. **Computation of modified weights**: The tracking performance of a target i depends on the positions of the various measurements associated with this target. The weight $\varpi_{ij}(k)$ as computed in Step 5 does not take this point into consideration, namely, how near or how far the position of the measurement $\mathbf{z}_j(k)$ is from the predicted position of the target i. In order to take this factor into consideration, we define the parameter $\delta_{ij}(k)$ to be the inverse of $d_{ij}(k)$ so that a heavier weight is attached when the measurement $\mathbf{z}_j(k)$ is closer to the predicted position of the target $\tilde{\mathbf{z}}_i(k)$, that is

$$\delta_{ij}(k) = \frac{1}{d_{ij}(k)} \tag{5.14}$$

For a given target i, the value of the parameter $\delta_{ij}(k)$ depends on $\mathbf{z}_j(k)$. The value of the parameter $\delta_{ij}(k)$ is made dimensionless by normalizing it over all the values of this parameter for a given target i. The normalized value $\delta_{ijn}(k)$ is given by

$$\delta_{ijn}(k) = \frac{\delta_{ij}(k)}{\sum_j \delta_{ij}(k)} \tag{5.15}$$

Each weight $\varpi_{ij}(k)$ is multiplied by its corresponding parameter $\delta_{ijn}(k)$ to obtain the modified weight as

$$w_{ij}(k) = \varpi_{ij}(k)\delta_{ijn}(k) \tag{5.16}$$

The modified value of the weight $\varpi_{ic}(k)$ is given by

$$w_{ic}(k) = 1 - \sum_{j=1}^{J} w_{ij}(k) \tag{5.17}$$

Finally, the normalized modified weights are given by

$$w_{ijn}(k) = \frac{w_{ij}(k)}{w_{ic}(k) + \sum_j w_{ij}(k)} \tag{5.18}$$

$$w_{icn}(k) = \frac{w_{ic}(k)}{w_{ic}(k) + \sum_j w_{ij}(k)} \tag{5.19}$$

8. **Output state estimation**: To find the output state estimate $\hat{\mathbf{x}}_i(k)$ of the target i, we first find the output state estimate $\hat{\mathbf{x}}_{ij}(k)$ individually for each measurement j falling inside the gate of the target i. Then, the output state estimate $\hat{\mathbf{x}}_i(k)$ of target i is a weighted sum of $\hat{\mathbf{x}}_{ij}(k)$, given by

$$\hat{\mathbf{x}}_i(k) = w_{icn}(k)\tilde{\mathbf{x}}_i(k) + \sum_{j=1}^{J} w_{ijn}(k)\hat{\mathbf{x}}_{ij}(k) \tag{5.20}$$

where $\tilde{\mathbf{x}}_i(k)$ is the predicted state of the ith target at the kth time step.

9. **Output covariance matrix**: To find the output covariance matrix $\hat{\mathbf{P}}_i(k)$ of the target i, we first find the output covariance matrix $\hat{\mathbf{P}}_{ij}(k)$ individually for each measurement j falling inside the gate of the target i. Then, the output covariance matrix $\hat{\mathbf{P}}_i(k)$ of target i is given by

$$\begin{aligned}\hat{\mathbf{P}}_i(k) = w_{icn}(k)(\tilde{\mathbf{P}}_i(k) + [\tilde{\mathbf{x}}_i(k) - \hat{\mathbf{x}}_i(k)][\tilde{\mathbf{x}}_i(k) - \hat{\mathbf{x}}_i(k)]^t]) \\ + \sum_{j=1}^{J} w_{ijn}(k)[\hat{\mathbf{P}}_{ij}(k) + [\hat{\mathbf{x}}_{ij}(k) - \hat{\mathbf{x}}_i(k)][\hat{\mathbf{x}}_{ij}(k) - \hat{\mathbf{x}}_i(k)]^t]\end{aligned} \tag{5.21}$$

where $\tilde{\mathbf{P}}_i(k)$ is the predicted covariance matrix of the ith target at the kth time step.

The output state $\hat{\mathbf{x}}_i(k)$ and the covariance matrix $\hat{\mathbf{P}}_i(k)$ are used to find the predicted state $\hat{\mathbf{x}}_i(k+1)$ and covariance $\hat{\mathbf{P}}_i(k+1)$ of the ith target for the next time step $(k+1)$.

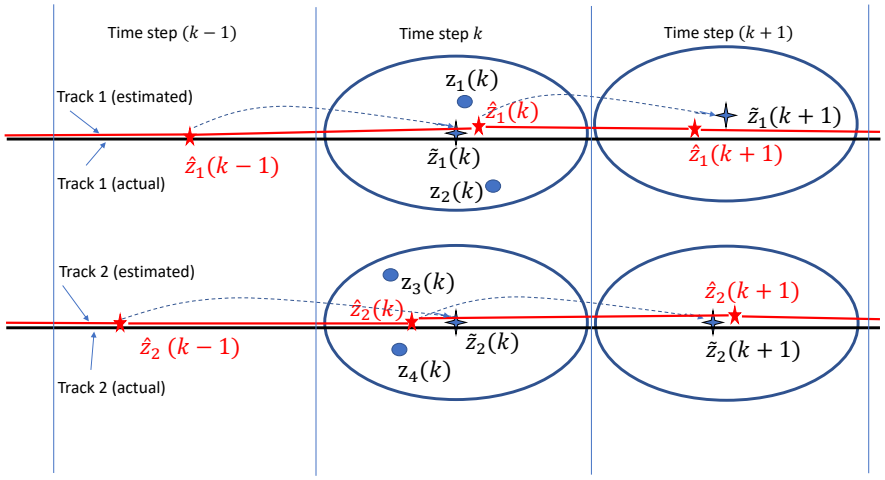
5.4 Rational Behind the Functioning of the Proposed Algorithm

As the targets become close to one another, so do the respective gates and eventually the gates overlap. The number of measurements falling inside the overlapped region represents the degree of overlap. These measurements are of special significance, since such measurements need to be associated to more than one target. The way the measurements are associated with a target affects the computation of the weights, since the estimated output is a weighted sum of the state estimates resulting from the individual measurements falling inside the gate. The relative values of the different weights impact the tracking performance of a JPDA-based algorithm, specially when the targets are close to one another. In this section, we explain as to how the proposed method of weight computation incorporating the information on spatial distribution of the measurements can significantly improve the tracking performance of the proposed algorithm over that of JPDA [59], when the targets are closely spaced.

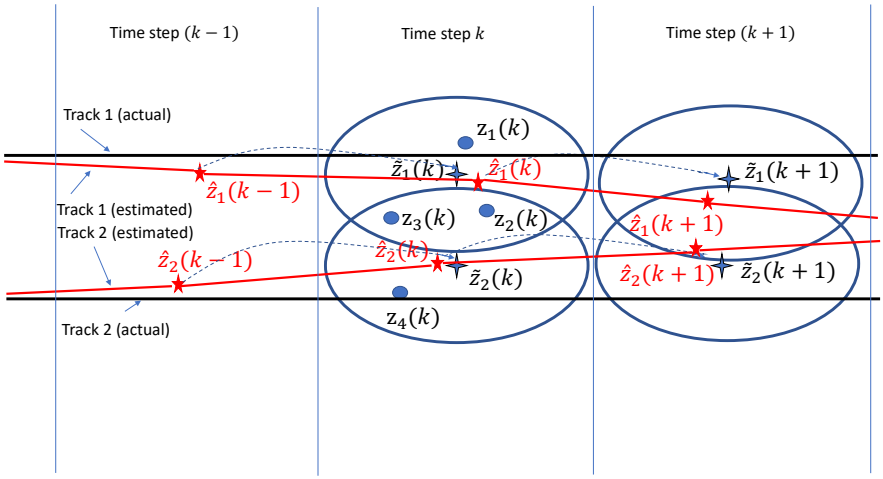
The performance of a JPDA-based algorithm depends very much on the residual between the predicted target position and the corresponding measurements. The idea behind such an algorithm is to obtain an estimate of the target position so that this new estimate is closer to the measurements than the predicted target position. Specifically, in the JPDA algorithm of [59], the computation of the weight ϖ_{ij}, using relevant hypotheses, involves the residual between the predicted position of target i and the position of measurement j. Overlapped gates add more uncertainty to the system, since an overlapped measurement can result from any of the targets involved in the formation of the overlapped region, in addition to being resulted from the clutter. In this case, an overlapped measurement is used to find the output state estimate for all the targets involved in the overlap. Hence, a method that uses

solely the probabilities of the hypotheses for the computation becomes less accurate when the gates overlap, that is, when the targets are closely spaced. Since that particular target among the targets involved in the overlap that is closer to the overlapped measurement has a greater likelihood of originating this particular measurement, the weight allocated to this measurement for computing the estimate of the position of this target should have a larger weight. It is for this reason that in the computation of the weights in the proposed scheme, the weights are adjusted by suitably modifying the hypothesis probabilities based on the spatial distribution of the measurements. However, this is done individually for each target involved in the overlap so that the constraint of all the hypothesis probabilities adding up to unity as well as the sum of the modified weights being unity is not violated. The use of the weights so modified effectively reduces the adverse impact of the gate overlaps.

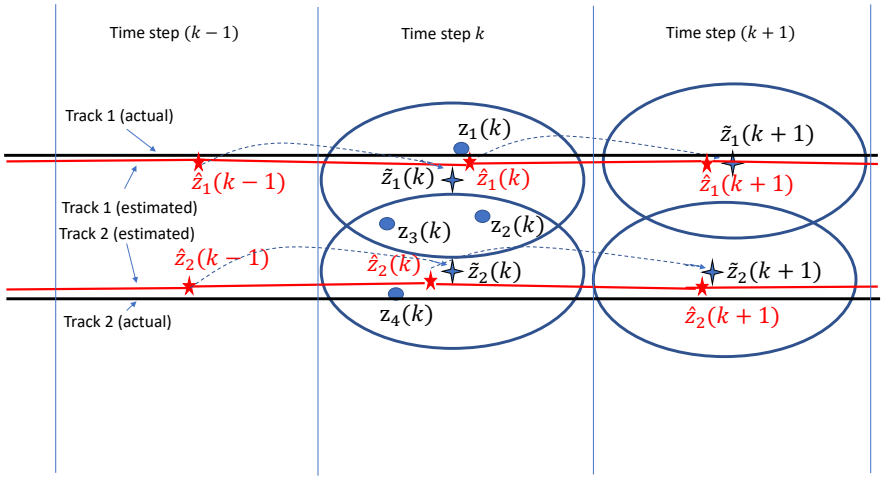
We now illustrate as to how the tracking performance of the JPDA algorithm is impacted as the targets become closely spaced and how the proposed scheme reduces this adverse impact by taking an example of tracking two targets. Figure 5.2(a) shows the case in which the two targets are far apart. Since in this case, overlap of the gates will not occur, i.e., there is no common measurement between the two targets, both the JPDA algorithm in [59] and the proposed one will provide a very similar good tracking performance. The gates of the two targets overlap when the spacing between them is reduced below a certain level. Figure 5.2(b) shows the tracking performance of the JPDA algorithm. It is seen from this figure that at the time step k, there are two measurements $\mathbf{z}_2$ and $\mathbf{z}_3$ that lie in the overlapped region, and therefore, shared between the two targets. This causes larger errors in the estimated positions, $\hat{\mathbf{z}}_1$ and $\hat{\mathbf{z}}_2$, of the two targets than in the case of no overlap, and thus, brings these two estimates closer to each other. The tracking performance of the algorithm will get worse in the succeeding time steps and may eventually result in the coalescence of the two estimated tracks. Figure 5.2(c) shows the tracking performance of the proposed algorithm. It is seen from this figure that even though at the time step k, there are still two measurements that fall in the overlapped region, in view of the modified weights, the estimated positions of the two targets have smaller errors and each estimated track is closer to its actual track. As a result, the estimated tracks of the two targets have very little tendency of getting closer,



(a)

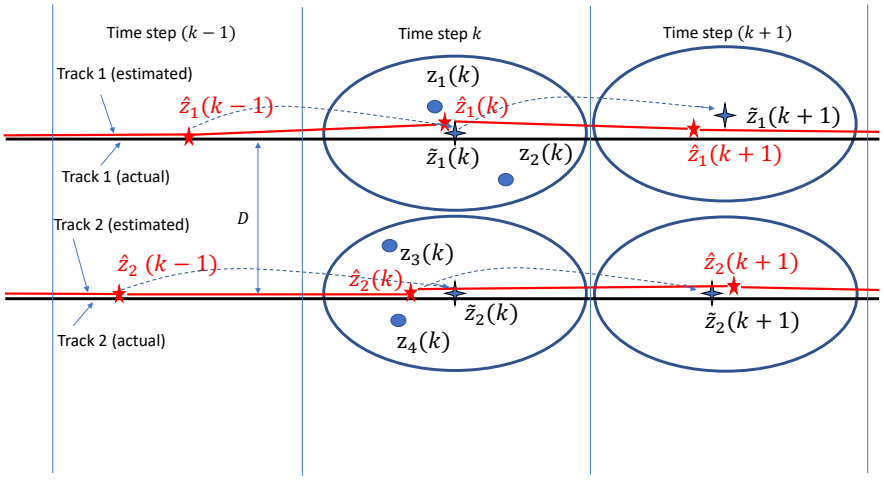


(b)

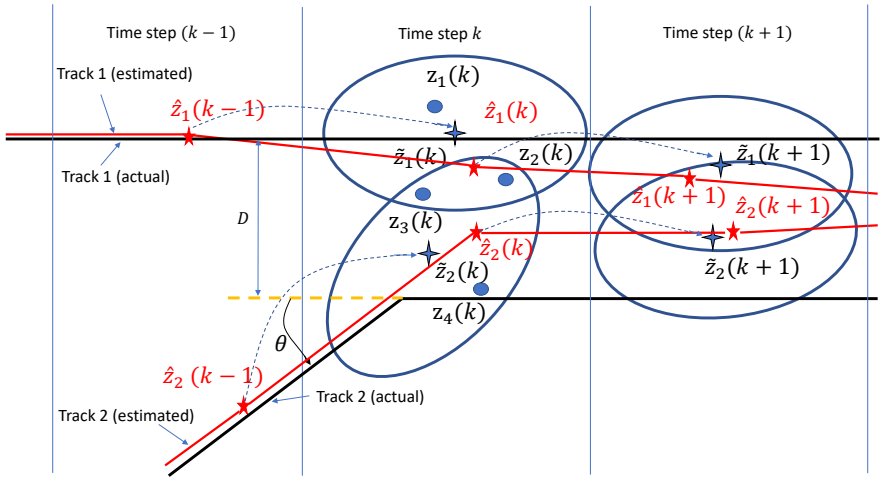


(c)

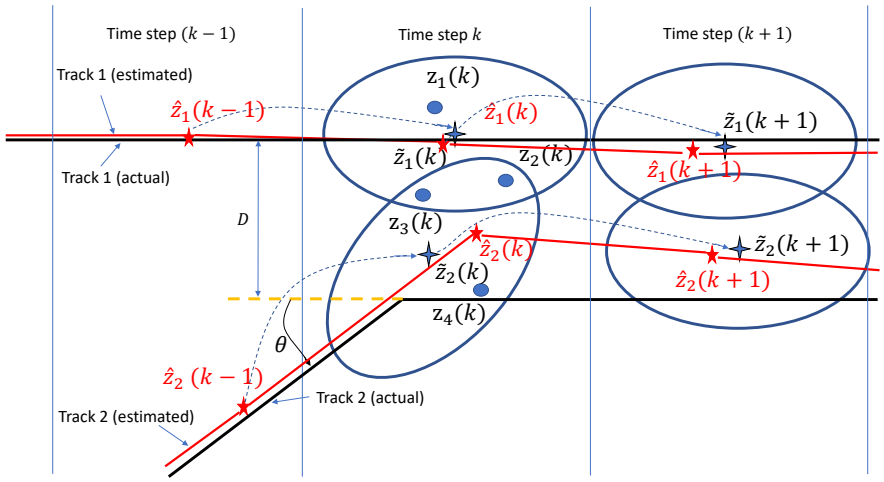
Figure 5.2: (a) Tracking of two widely-spaced targets using the JPDA algorithm of [59] or the proposed algorithm. Tracking of two closely-spaced targets using (b) JPDA algorithm [59] and (c) proposed algorithm.



(a)



(b)



(c)

Figure 5.3: (a) Tracking of two targets with a spacing of D between them using the JPDA algorithm of [59] or the proposed algorithm. Tracking of two targets when one target approaches the other at an angle θ and then continues in parallel with the other one with a spacing of D between them using (b) JPDA algorithm [59] and (c) proposed algorithm.

and therefore, successfully avoid the coalescence of the two estimated tracks.

Next, we demonstrate how the performance of the JPDA algorithm is also affected when one or more of the targets take a sharp turn. We illustrate this again be considering an example of two targets. Figure 5.3(a) shows two targets moving in parallel with a spacing D between them. It is assumed that the spacing D is large enough as not to cause an overlap of the gates of the two targets. Hence, in this case both the JPDA and the proposed algorithms will track the targets satisfactorily. Now, let us assume that initially target 2 is approaching target 1 at an angle θ, and then takes a sharp turn at the time step k when the spacing between the two targets becomes D; then from that point on, target 2 continues to move in parallel to the track of target 1. In a JPDA-based algorithm, the gate for the target at the time step k is constructed based on the target state at the time step $(k-1)$. Hence, the gate is constructed with the assumption that target 2 will still be moving with an angle θ. Consequently, the gates of the two targets will necessarily overlap if the angle θ has a large value and the spacing D is not too large. It is because of this overlap, the performance of the JPDA algorithm deteriorates and eventually coalescence between the two tracks may occur in the succeeding time steps as shown in Fig.5.3(b). On the other hand, the proposed algorithm will be able to handle this situation effectively, and thus, avoid coalescence of the two tracks, as seen from Fig.5.3(c). However, it should be pointed out that if the spacing D is further reduced but without causing overlap, then the proposed algorithm will still work if the angle θ is smaller, even in those cases where the JPDA algorithm will fail to track the targets satisfactorily.

Finally, it must to pointed out that in a situation when the two targets become very close to each other, most if not all the measurements, will fall in the overlapped region. The method of obtaining the weights based on the spatial distribution of the measurements will not help, since the distributions of the measurements with respect to both the targets will be almost identical. Hence, in the case when the targets are very closely spaced, even the proposed algorithm will fail to track the targets.

5.5 Performance Evaluation

In this section, the performance of the proposed algorithm is evaluated and compared with that of the JPDA algorithm of [59] by considering the example of two targets being tracked. The performance of both the algorithms is evaluated using the root mean square error metric as mentioned in Section 2.3.1.

5.5.1 Simulation parameters

In all our experiments, we assume that the two targets are moving at a constant speed of $100\,m/s$. Further, it is assumed that the sampling period $T = 1\,s$, the measurement noise standard deviation, $\sigma_s = 10\,m$, the probability of detection $P_D = 0.9$, and the gate threshold parameter $\gamma = 16$. Clutter is generated by assuming it to have a Poisson distribution with the intensity parameter $\lambda = 2 \times 10^{-4}\,m^{-2}$, over the surveillance region with an area $A = 140 \times 10^3\,m^2$. The standard deviation σ_Q of the process noise is assumed to be $35\,m$. Mote-Carlo simulations with $M = 100$ runs are carried out to obtain the estimates of the target states.

5.5.2 Experimental results

The performance of the algorithms is organized in four categories. In the first category, examples are considered to illustrate the impact of the spacing between the two targets, which are moving in parallel. In the second category, examples are considered in which the targets are moving in straight lines but not in parallel. The examples in the third category illustrate the impact of one target approaching at a certain angle the other one moving in a straight-line, and then taking a sudden turn to move in parallel with the other. In the last category, a number of examples are taken by considering scenarios with various combinations of the examples in the three categories.

In all the figures depicting the tracking performance, part (a) of the figures shows the true target motion and estimated tracks using the JPDA algorithm of [59], part (b) the true

target motion and estimated tracks using the proposed algorithm, parts (c) and (d) RMSEs of the positions of targets 1 and 2, respectively, for both the algorithms.

5.5.2.1 Category 1

In this category, the two targets move in parallel for a period of $40\,s$ with different values of the spacing D between them.

(i) **Example 1**

In this example, the spacing between the two targets is taken to be $D = 150\,m$. From Fig. 5.4, it is seen that for this spacing both the algorithms provide very good and similar performance.

(ii) **Example 2**

In this example, the spacing between the two targets is reduced by $5\,m$ to $145\,m$. Although the JPDA algorithm was providing a very good performance when D was $150\,m$, it seen form Fig. 5.5(a) that when the spacing is reduced only by $5\,m$, the performance of the JPDA algorithm gets severely deteriorated and coalescence occurs. On the other hand, it is seen from Fig. 5.5(b) that the proposed algorithm continues to exhibit a very good tracking performance. From Figs. 5.5(c) and 5.5(d), it is observed that for both the targets the RMSE values obtained from JPDA exceed the threshold value, and hence, its performance is unacceptable. On the other hand, the RMSE values obtained from the proposed algorithm for both the targets are below the threshold value, resulting in very good tracking performance.

(iii) **Example 3**

In this example, the spacing between the two targets is set to $70\,m$. Form Figs. 5.6(a) and 5.6(b), it is observed that, when the spacing is reduced from $145\,m$ to $70\,m$, as expected, the performance of the JPDA algorithm severely deteriorates and coalescence occurs, whereas the proposed algorithm still provides a very good tracking performance. From Figs. 5.6(c) and 5.6(d), it is observed that for both the targets the RMSE

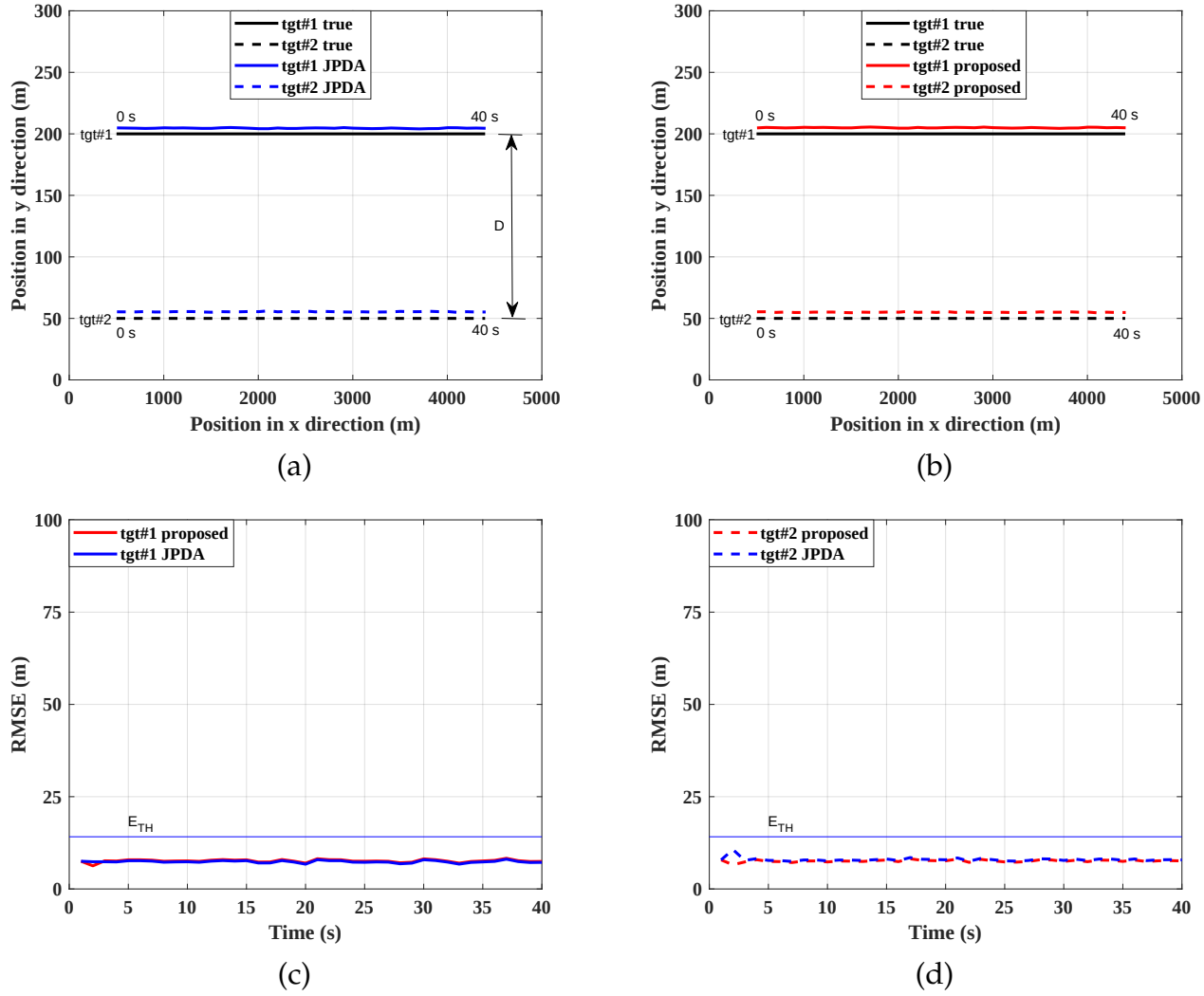


Figure 5.4: Tracking of two targets moving in parallel with $D = 150\,m$. True and estimated tracks using (a) JPDA algorithm and (b) proposed algorithm. Position RMSEs for (c) target 1 and (d) target 2.

values obtained from JPDA are higher than the threshold value, thus causing its performance to become unacceptable. On the other hand, in the case of the proposed algorithm, the RMSE values for either of the two targets remain below the threshold value, thus indicating that the performance of the proposed algorithm is still acceptable.

(iv) **Example 4**

In this example, the spacing between the two targets is further reduced by $2\,m$ to $68\,m$. Form Figs. 5.7(a) and 5.7(b), it is observed that, as expected, the performance of the JPDA algorithm continues to be deteriorated along with the coalescence of the two

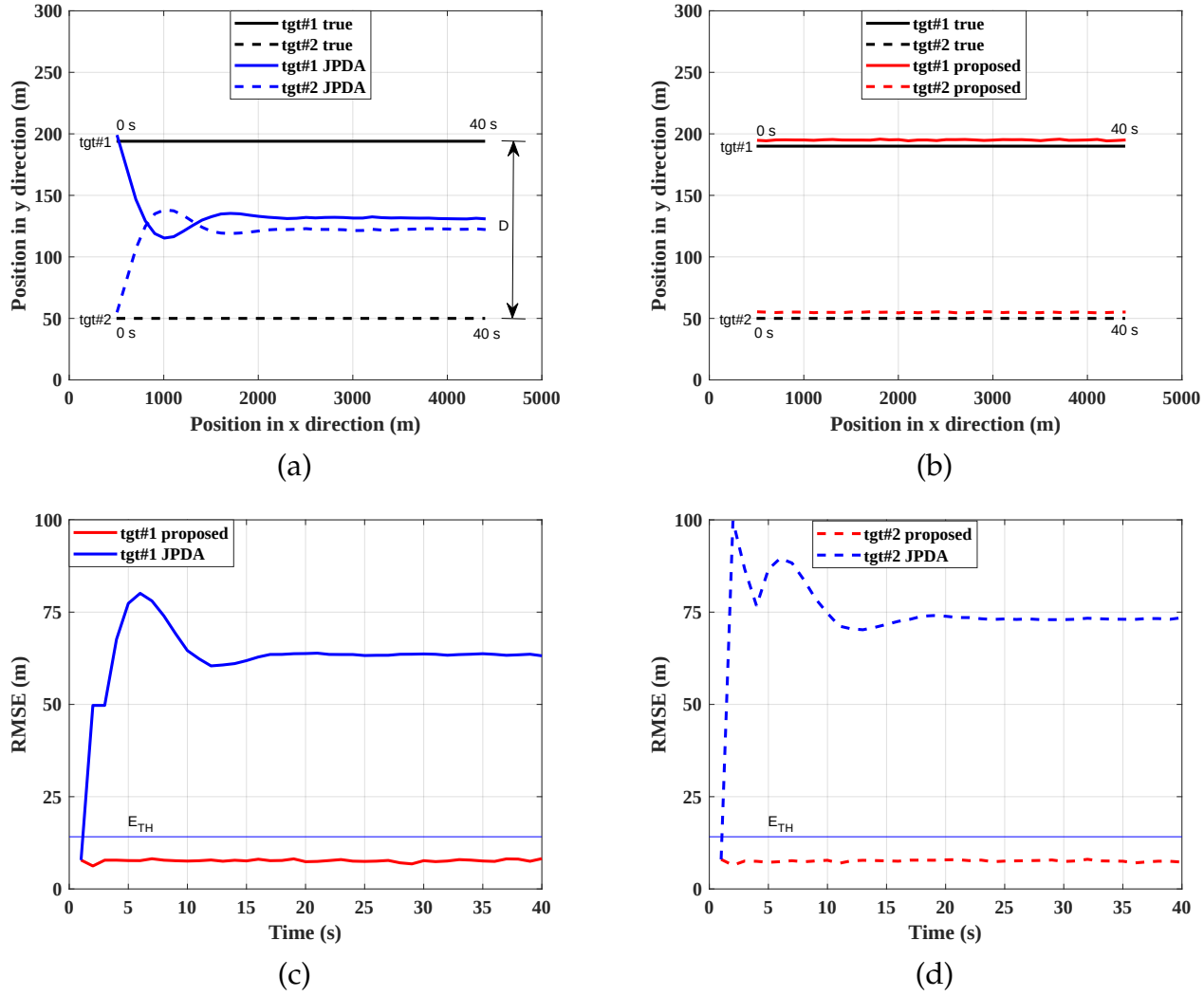


Figure 5.5: Tracking of two targets moving in parallel with $D = 145\,m$. True and estimated tracks using (a) JPDA algorithm and (b) proposed algorithm. Position RMSEs for (c) target 1 and (d) target 2.

tracks, whereas the proposed algorithm does not suffer from any coalescence. From Figs. 5.7(c) and 5.7(d), it is observed that for both the targets the RMSE values obtained from JPDA as well as from the proposed algorithm exceeds the threshold value, thus causing their performance to become unacceptable. It is to be pointed out that even though the performance of the proposed algorithm is not acceptable, coalescence of the tracks does not occur.

We can conclude from the results of the above experiments that the effect of the spacing between the targets on the performance of the two algorithms. The two algorithms exhibit very good performance when the spacing is large. However, both the algorithms have their

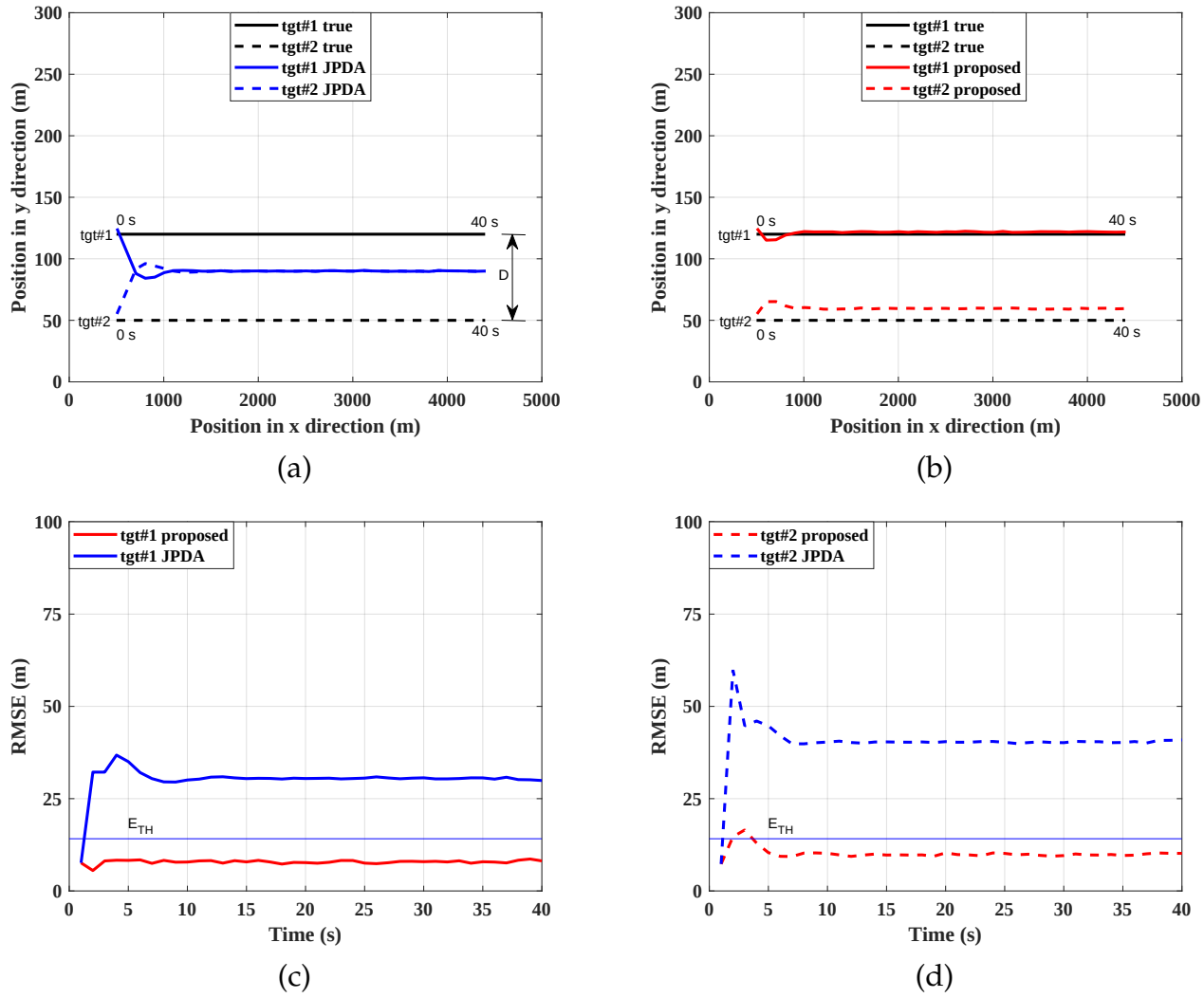


Figure 5.6: Tracking of two targets moving in parallel with $D = 70\,m$. True and estimated tracks using (a) JPDA algorithm and (b) proposed algorithm. Position RMSEs for (c) target 1 and (d) target 2.

own minimum spacing below which they fail to perform satisfactorily. This minimum spacings for the proposed algorithm is much lower than that for the JPDA algorithm. In other words, the performance of the proposed algorithm is much less sensitive to the spacing between the targets, in that the proposed algorithm continues to perform satisfactorily even for small values of the spacing between the targets, whereas the JPDA algorithm fails to provide satisfactory tracking performance even for much larger values.

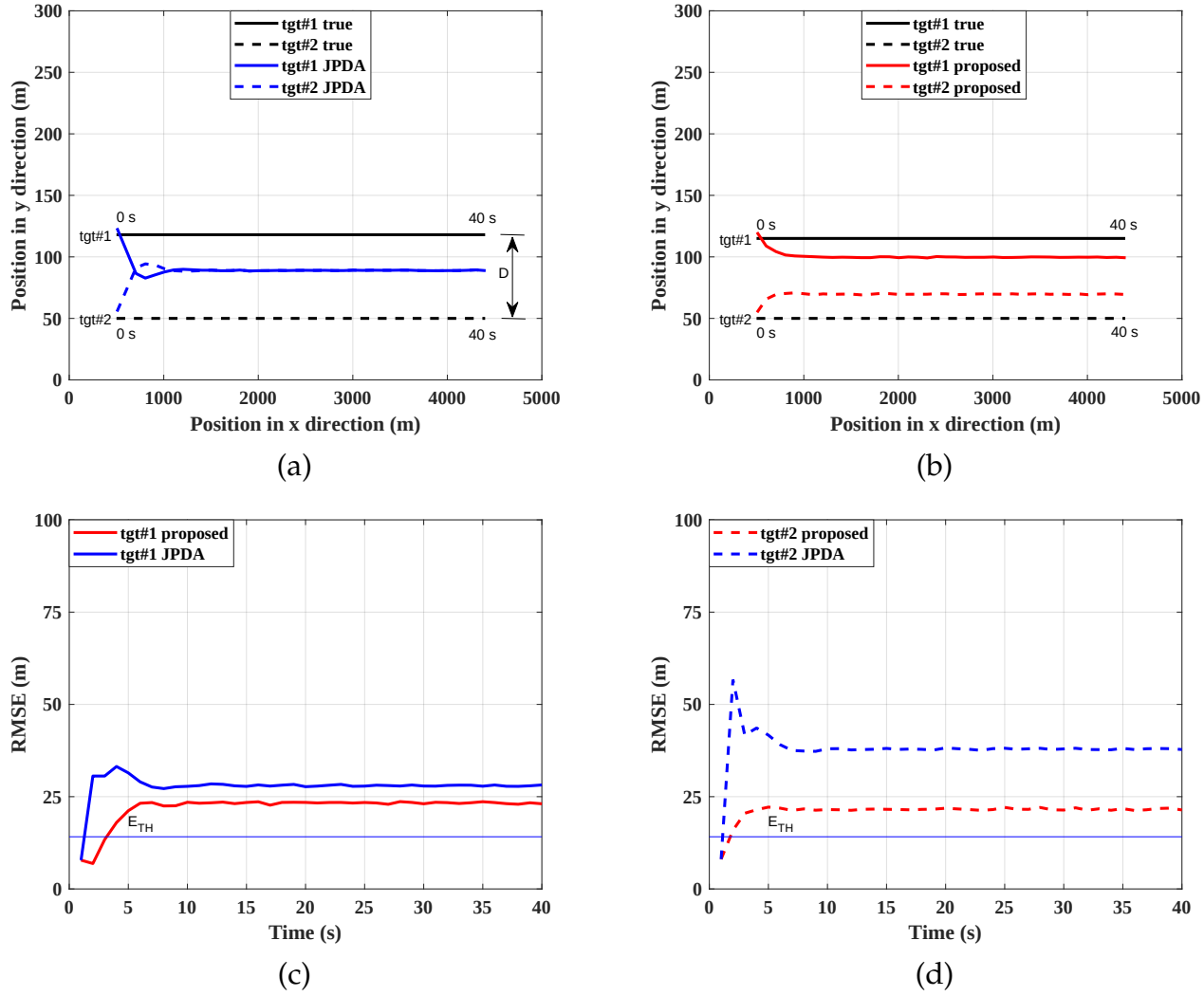


Figure 5.7: Tracking of two targets moving in parallel with $D = 68\,m$. True and estimated tracks using (a) JPDA algorithm and (b) proposed algorithm. Position RMSEs for (c) target 1 and (d) target 2.

5.5.2.2 Category 2

In this category, we consider examples in which the targets are moving in straight lines with an angle ψ between the two tracks.

(vi) **Example 5**

In this example, the two targets are moving with an angle $\psi = 60^{\circ}$ between their tracks. From Figs. 5.8(a) and 5.8(b), it appears that the tracking performance of both the algorithms is satisfactory. However, from Figs. 5.8(c) and 5.8(d), it is noted that the RMSE values of JPDA exceed the threshold value in an appreciable region around the point where the two tracks intersect. On the other hand, the RMSE values of the

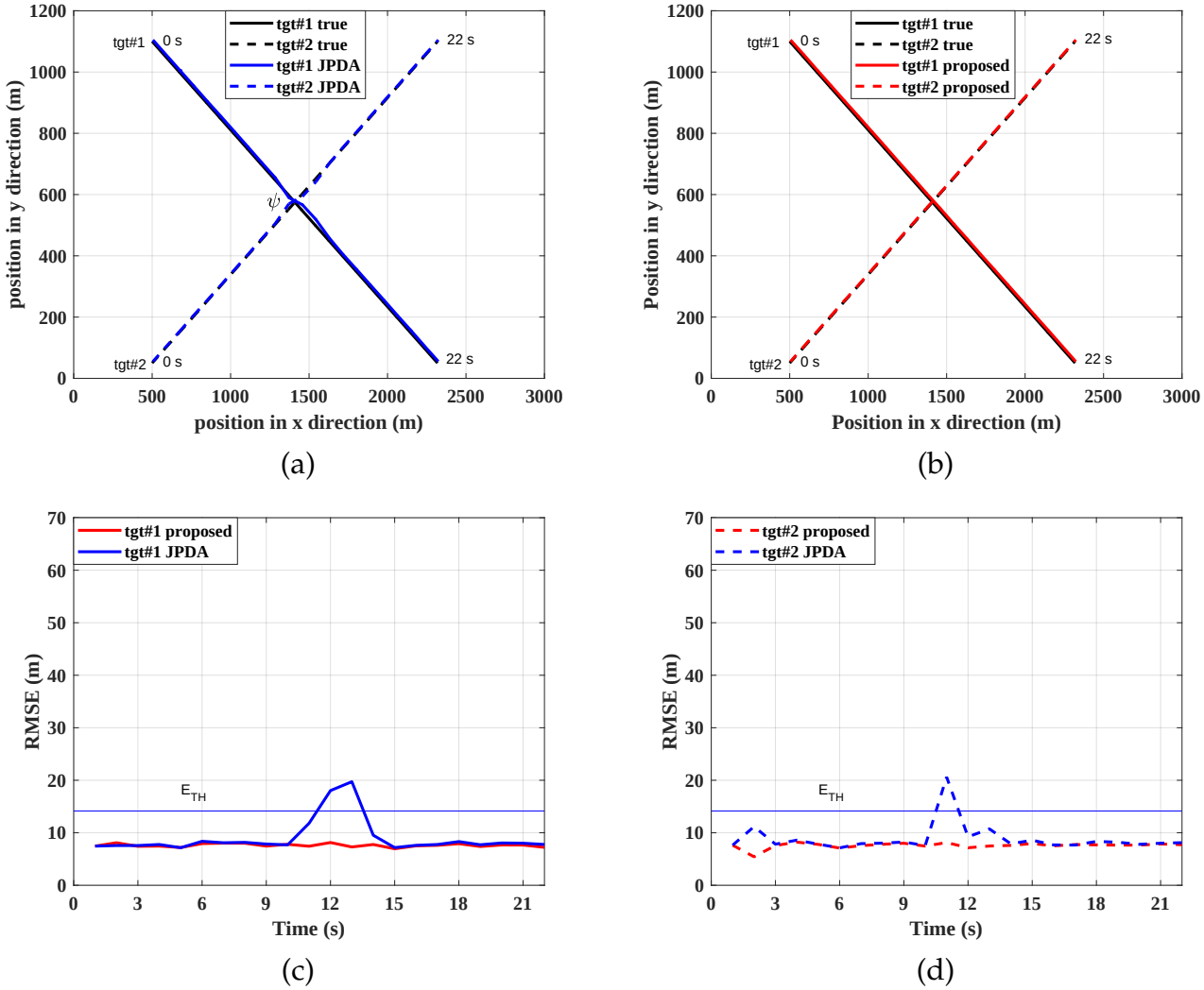


Figure 5.8: Tracking of two targets moving with an angle $\psi = 60^\circ$ between the tracks. True and estimated tracks using (a) JPDA algorithm and (b) proposed algorithm. Position RMSEs for (c) target 1 and (d) target 2.

proposed algorithm are less than the threshold value even in the intersecting region, thus exhibiting a very good tracking performance throughout the tracking period of the targets.

(vii) **Example 6**

In this example, we reduce the angle ψ from 60° to 30°. From Fig. 5.9(a), it is seen that the tracking performance of the JPDA algorithm gets severely deteriorated beyond the point of intersection of the two targets, when ψ is decreased to 30°, whereas the proposed algorithm continues to exhibit an acceptable tracking performance, as seen in 5.9(b). Form Figs. 5.9(c) and 5.9(d), it is noted that the RMSE values of JPDA exceed

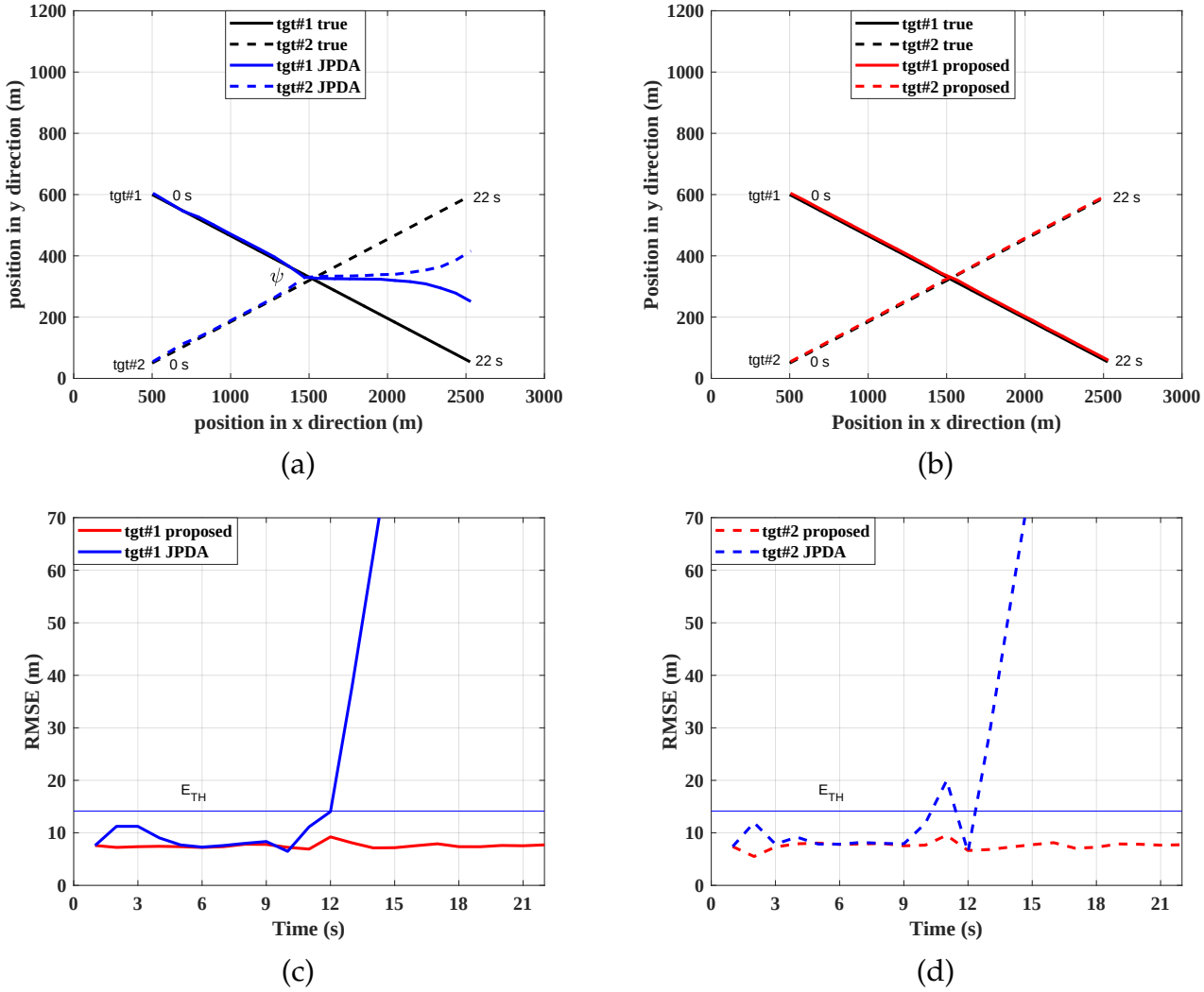


Figure 5.9: Tracking of two targets moving with an angle $\psi = 30^\circ$ between the tracks. True and estimated tracks using (a) JPDA algorithm and (b) proposed algorithm. Position RMSEs for (c) target 1 and (d) target 2.

the threshold value around the point of intersection. On the other hand, the RMSE values of the proposed algorithm are less than the threshold value throughout the tracking period, thus exhibiting a very good tracking performance.

However, it can be shown that the proposed algorithm would also fail to track the targets satisfactorily when ψ is decreased below 17°.

In summary, regardless of the value of the intersection angle ψ, the RMSE values of JPDA exceeds the threshold value in the neighborhood of the point of intersection and the region of this violation widens as the value of ψ is reduced. The JPDA algorithm completely fails to track the targets beyond the point of intersection for an angle ψ below a certain value,

say ψ_1. On the other hand, the RMSE values of the proposed algorithm never exceed the threshold value throughout the tracking period for a value of ψ that is larger than a value, say ψ_2, which is much smaller than ψ_1.

5.5.2.3 Category 3

In this category, one target approaches the other at an angle θ for a period of $10\,s$, and then takes a sudden turn to continue to move in parallel with the other one for a period of $30\,s$ with a spacing D between them. Since both the algorithms provide a very good tracking performance when D is $150\,m$, we set D to this value for all the examples in this category, whereas the angle θ is varied in order to study the effect of varying the angle θ on the performance of the algorithms.

(viii) **Example 7**

In this example, target 2 approaches target 1 at an angle θ of 30°. From Fig. 5.10, it clear that both the algorithms exhibit very good tracking performance.

(ix) **Example 8**

In this example, the angle θ is increased to 45°. It is clear from Fig. 5.11(a) that this increase in the angle adversely affects the tracking performance of JPDA in that it leads to severe deterioration and track coalescence. On the other hand, the proposed algorithm provides a very good tracking performance throughout the tracking period, as seen from Fig. 5.11(b). It is observed from Figs. 5.11(c) and 5.11(d) that the RMSE values for JPDA abruptly exceed the threshold value by a large amount, after the true tracks of the two targets become parallel, thus leading to its unsatisfactory tracking performance. On the other hand, the RMSE values for the proposed algorithm do not exceed the threshold value and the performance is satisfactory.

(x) **Example 9**

In this example, the angle θ is further increased to 90°. It seen from Figs. 5.12(a), 5.12(c) and 5.12(d) that, as expected, JPDA badly fails to track the targets. It seen from Figs. 5.12(b), 5.12(c) and 5.12(d) that the proposed algorithm still provides very satisfactory

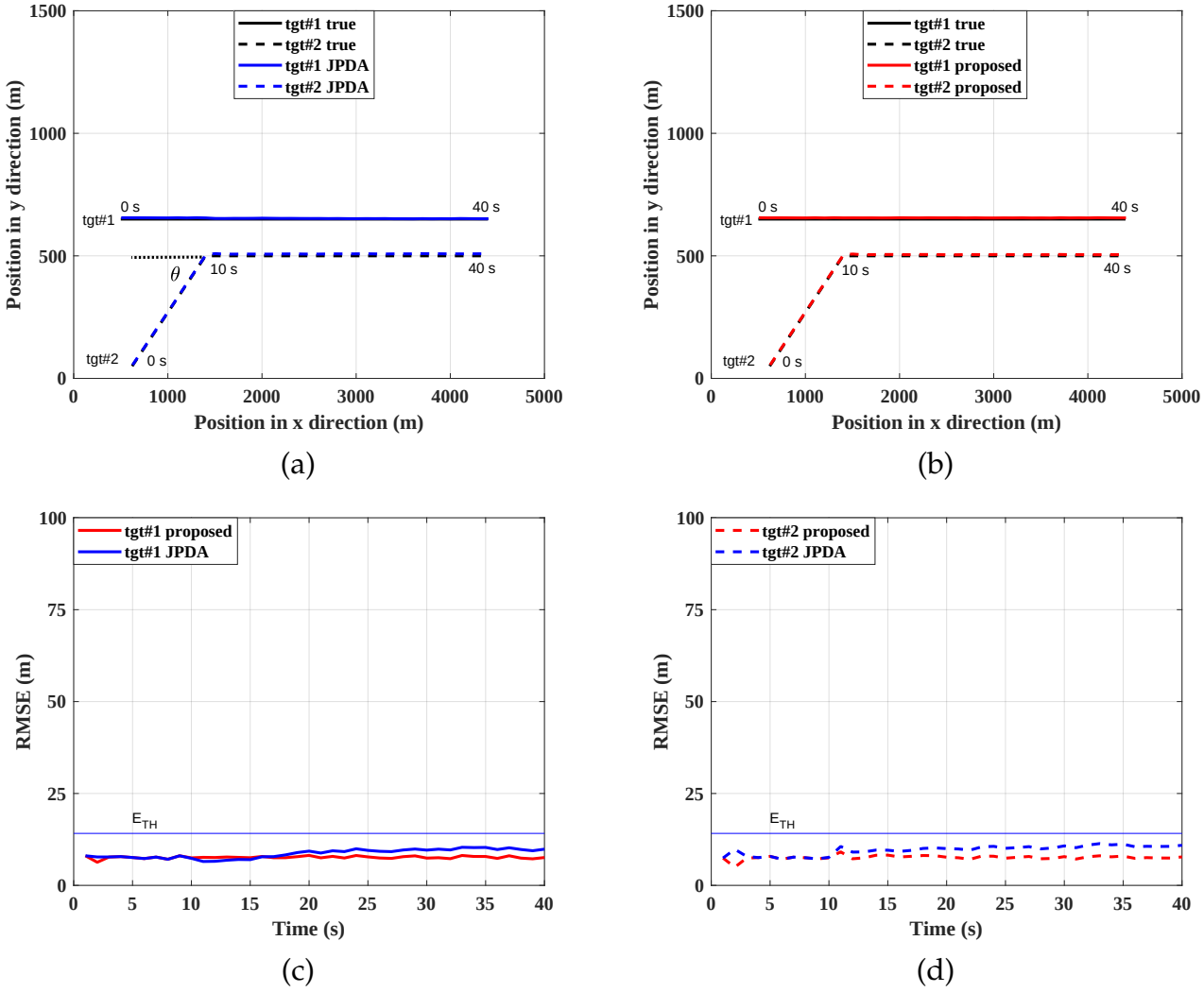


Figure 5.10: Tracking of two targets when one target approaches the other at an angle $\theta = 30^\circ$ and then moving in parallel with $D = 150\,m$. True and estimated tracks using (a) JPDA algorithm and (b) proposed algorithm. Position RMSEs for (c) target 1 and (d) target 2.

tracking performance even though the RMSE value for target 2 momentarily exceeds the threshold value at the turning point of its track.

Although the performance of both the algorithms have been satisfactory when the targets move in parallel to each other with a spacing of $150\,m$, JPDA fails to track the targets properly when target 2 approaches target 1 at an angle θ that is $\theta = 45^\circ$ or higher, and moves in parallel to target 1 for the same spacing of 150 m, as in the case when the two targets moving in parallel throughout. This is due to the fact that the gates of the two targets at the turning points overlap for this value of the angle.

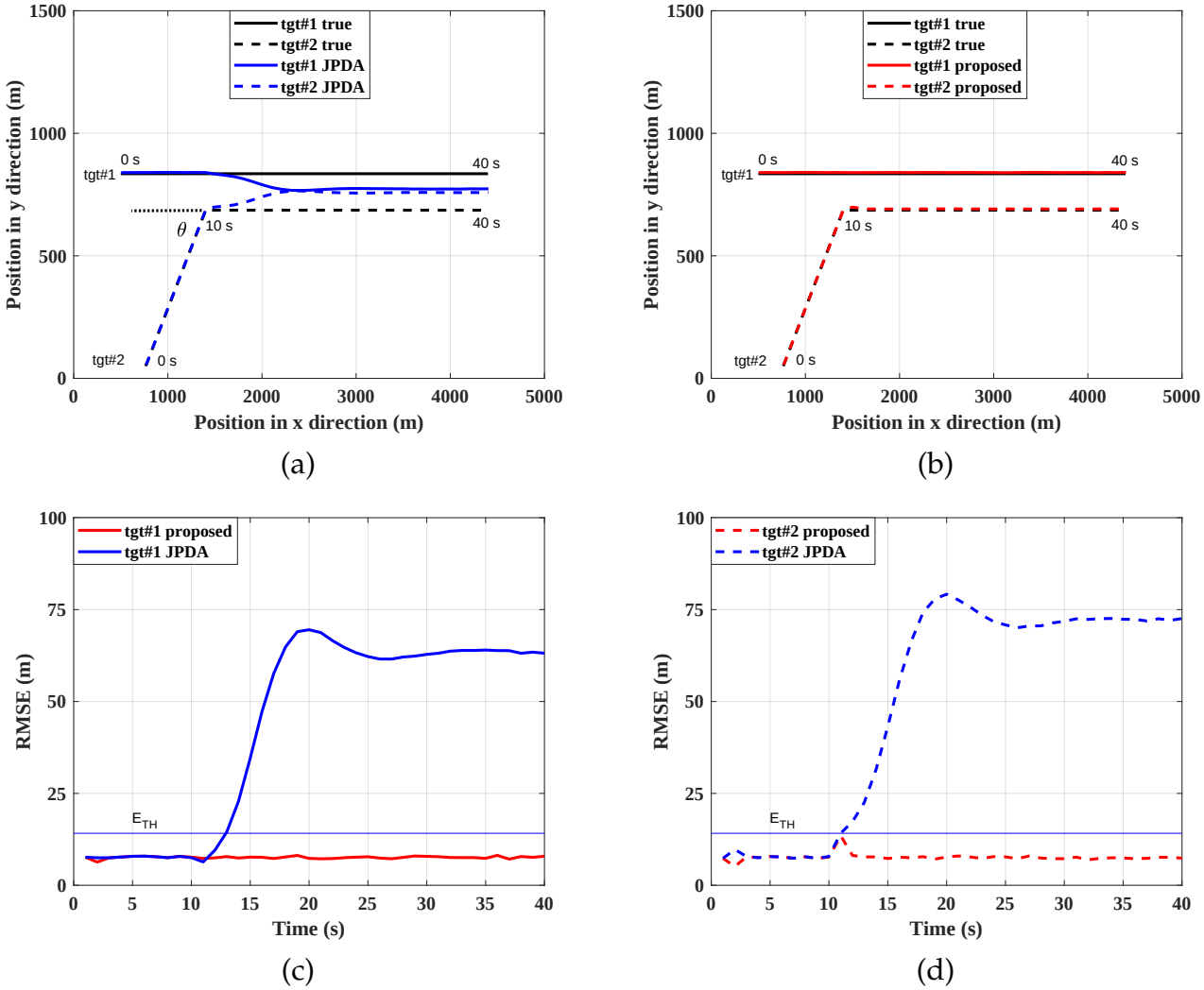


Figure 5.11: Tracking of two targets when one target approaches the other at an angle $\theta = 45^\circ$ and then moving in parallel with $D = 150\,m$. True and estimated tracks using (a) JPDA algorithm and (b) proposed algorithm. Position RMSEs for (c) target 1 and (d) target 2.

On the other hand, even though the gates of the two targets overlap in the case of the proposed algorithm also, the tracking performance continues to be satisfactory in view of the modified weights introduced in the proposed algorithm.

It can be concluded from the results of the experiments in this category that for both the algorithms, as the angle of approach θ of target 2 increases, the spacing between the targets needs to be increased accordingly for satisfactory performance of the algorithms. However, for a given spacing between the targets for which the two targets moving in parallel are capable of performing satisfactorily, the proposed algorithm can handle a larger angle θ than

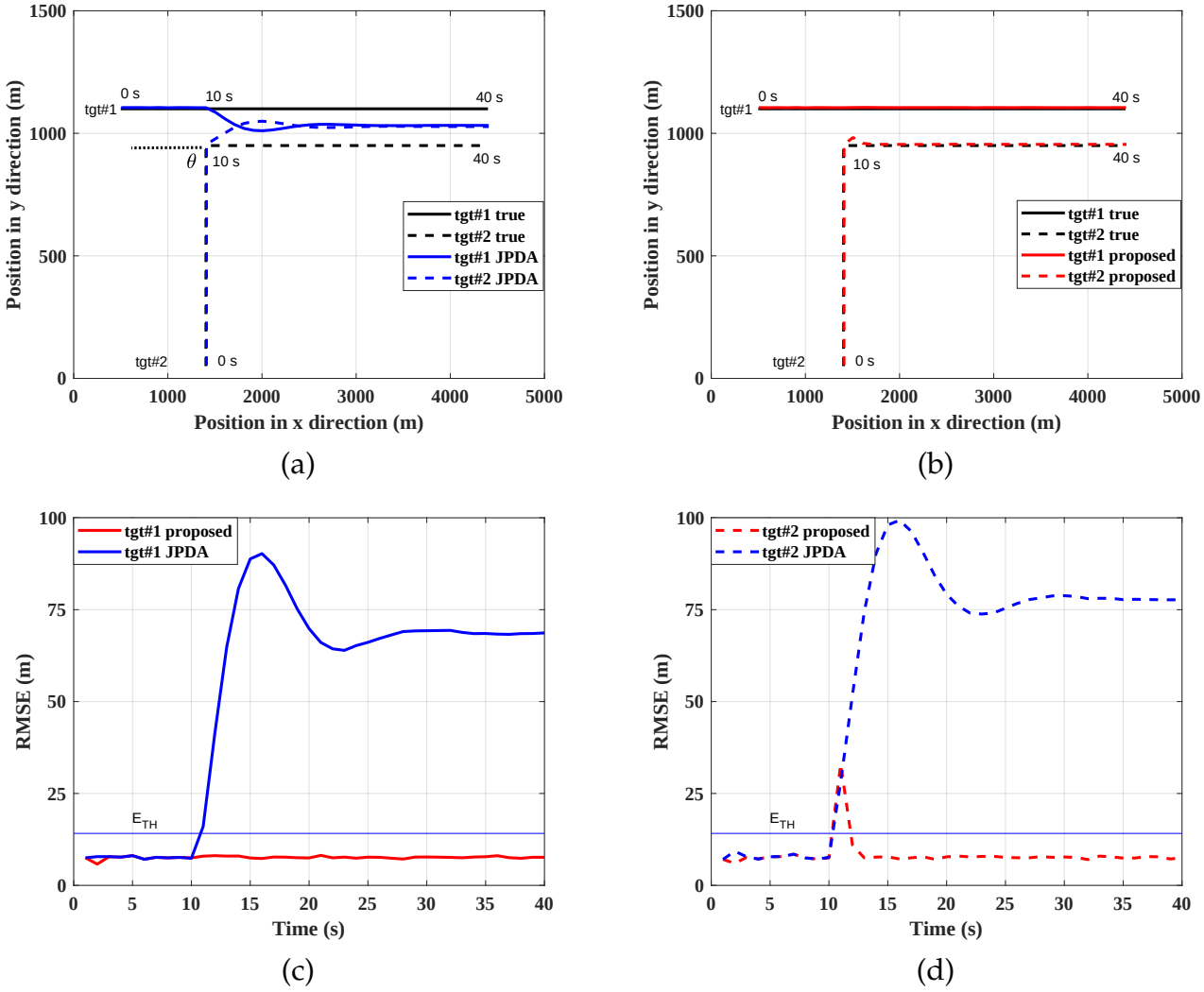


Figure 5.12: Tracking of two targets when one target approaches the other at an angle $\theta = 90^\circ$ and then moving in parallel with $D = 150\,m$. True and estimated tracks using (a) JPDA algorithm and (b) proposed algorithm. Position RMSEs for (c) target 1 and (d) target 2.

JPDA, and for a given angle θ, the minimum spacing needed by the proposed algorithm is much lower than that required by JPDA for satisfactory performance.

5.5.2.4 Category 4

In this category, we consider a number of examples having various combinations of the examples taken in categories 1, 2, and 3.

(xi) **Example 10**

This example is the same as Example 7, except that target 1 also is approaching target 2

at the same angle, i.e., 30°. It is observed from Fig. 5.13(a) that coalescence of the tracks occurs at the turning point and continues beyond in the case of JPDA. It is to be pointed out that even though JPDA was performing satisfactorily when only one target was approaching the other at an angle of 30° with $D = 150\,m$, as in Example 7, it now fails to track properly when the other target also is approaching at the same angle. This has happened in view of the fact that overlap of the gates necessarily occurs at the turning point. On the other hand, the proposed algorithm gives a very good performance throughout the tracking period, as seen from Fig. 5.13(b). It is observed from Figs. 5.13(c) and 5.13(d) that the RMSE values for JPDA abruptly exceed the threshold value by a large amount, after the true tracks of the two targets become parallel, thus leading to its unsatisfactory tracking performance. On the other hand, the RMSE values for the proposed algorithm do not exceed the threshold value at any point throughout the tracking period, just as in Example 7.

It can be easily shown that when D is increased from $150\,m$ to $200\,m$, JPDA will also function satisfactorily.

(xii) **Example 11**

This example is the same as the previous one but with the spacing D decreased by $50\,m$ to $100\,m$. Form Figs. 5.14(a) and 5.14(b), it is observed that, as expected, the performance of the JPDA algorithm continues to be deteriorated coupled with the coalescence of the two tracks, whereas the proposed algorithm does not suffer from any coalescence. From Figs. 5.14(c) and 5.14(d), it is observed that for both the targets the RMSE values obtained from JPDA as well as from the proposed algorithm when the targets move in parallel exceeds the threshold value, thus causing their performance to become unacceptable during this period. It is to be pointed out that even though the performance of the proposed algorithm is not acceptable, coalescence of the tracks does not occur during the parallel motion of the targets.

From Examples 10 and 11, it can be concluded that even when JPDA algorithm preforms

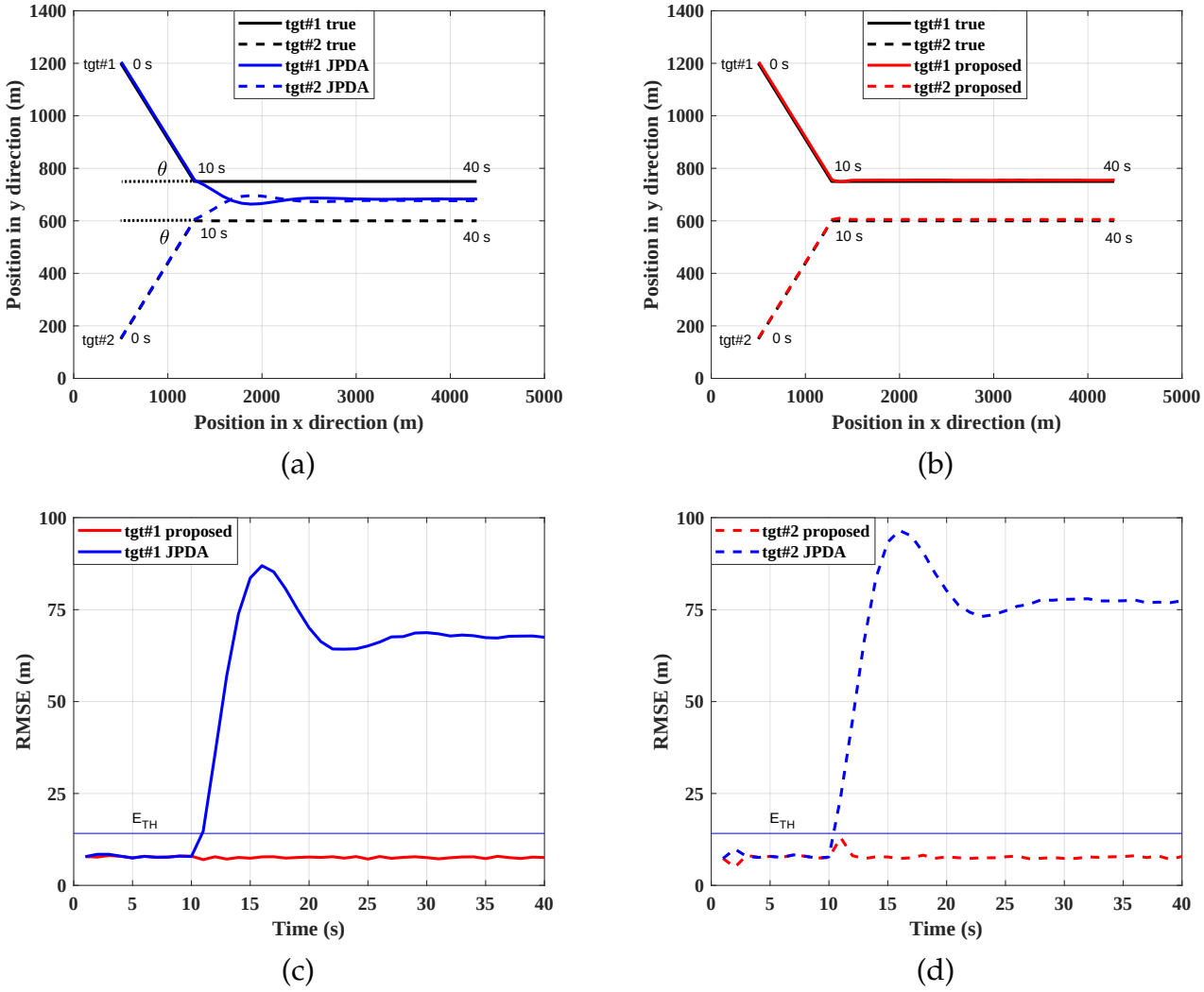


Figure 5.13: Tracking two targets each approaching at an angle $\theta = 30^\circ$ and then moving in parallel with $D = 150\,m$. True and estimated tracks using (a) JPDA algorithm and (b) proposed algorithm. Position RMSEs for (c) target 1 and (d) target 2.

well in the case of only one of the targets taking a sudden turn and then moving in parallel with the other, it fails to track the targets properly when each of the targets takes a turn simultaneously. Even though the same conclusion can be drawn for the proposed algorithm, it happens for a spacing that is much lower than in the case of JPDA. Further it can be conclude that for a given spacing for which both the algorithm perform satisfactorily if they were to move in parallel, the proposed algorithm can sustain a much sharper angle of approach than the JPDA algorithm can, before the performance becomes unsatisfactory.

(xiii) **Example 12**

In this example, the two targets approach each other, each at an angle 30° for $10\,s$, then

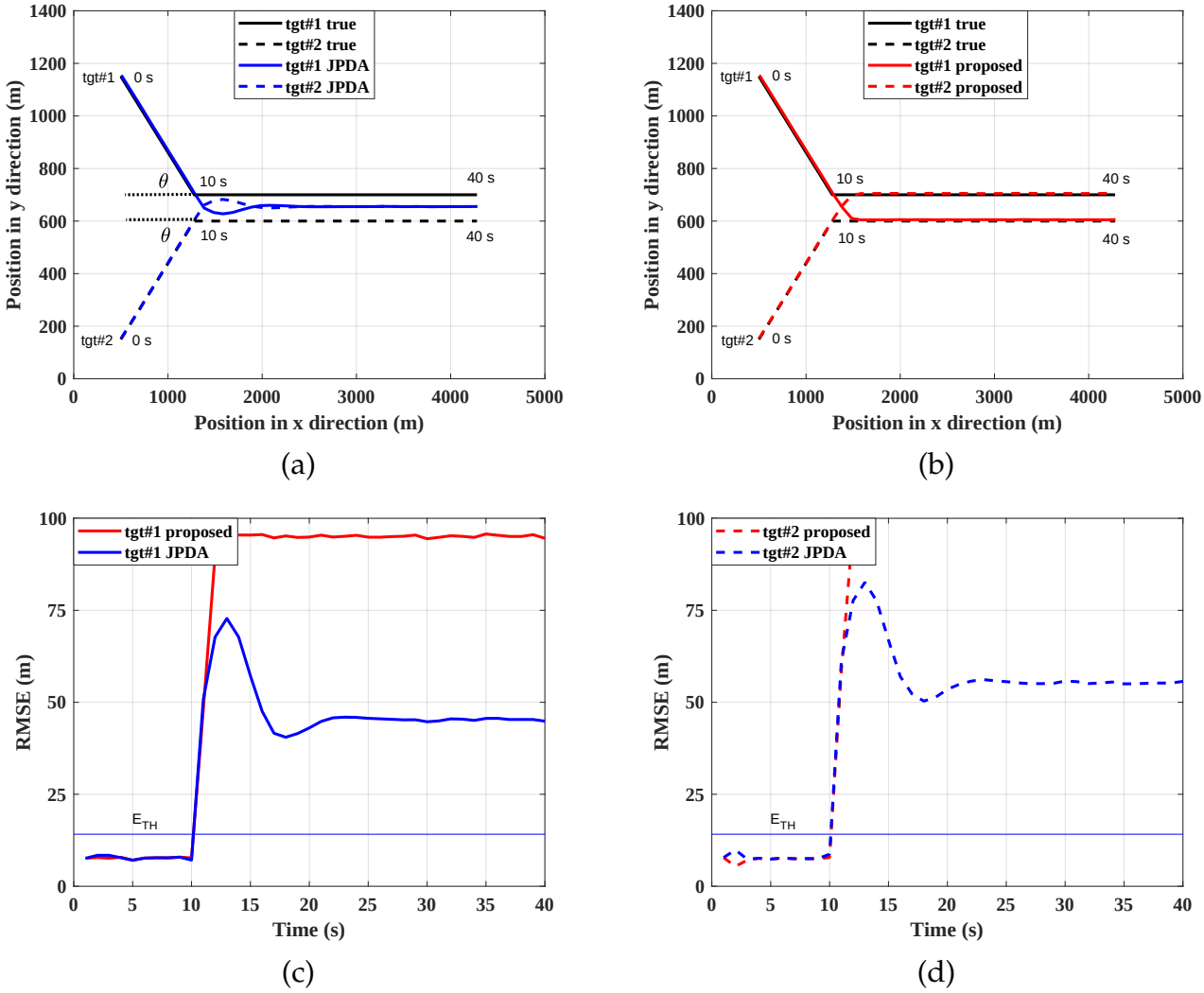


Figure 5.14: Tracking two targets each approaching at an angle $\theta = 30^\circ$ and then moving in parallel with $D = 100\,m$. True and estimated tracks using (a) JPDA algorithm and (b) proposed algorithm. Position RMSEs for (c) target 1 and (d) target 2.

move in parallel with $D = 140\,m$, and finally, at $30\,s$ one of them suddenly diverges with an angle of 30°. From Fig. 5.15(a), as expected, the coalescence of the tracks of the two targets occurs for JPDA during the parallel part of the tracks and this coalescence continues even after target 2 makes another turn at $30\,s$. On the other hand, as seen from Fig. 5.15(b), the proposed algorithm provides a very good performance throughout the tracking period. It is seen from Figs. 5.15(c) and 5.15(d) that the RMSE values for JPDA, as expected, abruptly exceed the threshold value by large amounts at $10\,s$ and continue with such large errors during the parallel part of the tracks. At $30\,s$, when target 2 takes a turn, the RMSE value experiences another abrupt increase

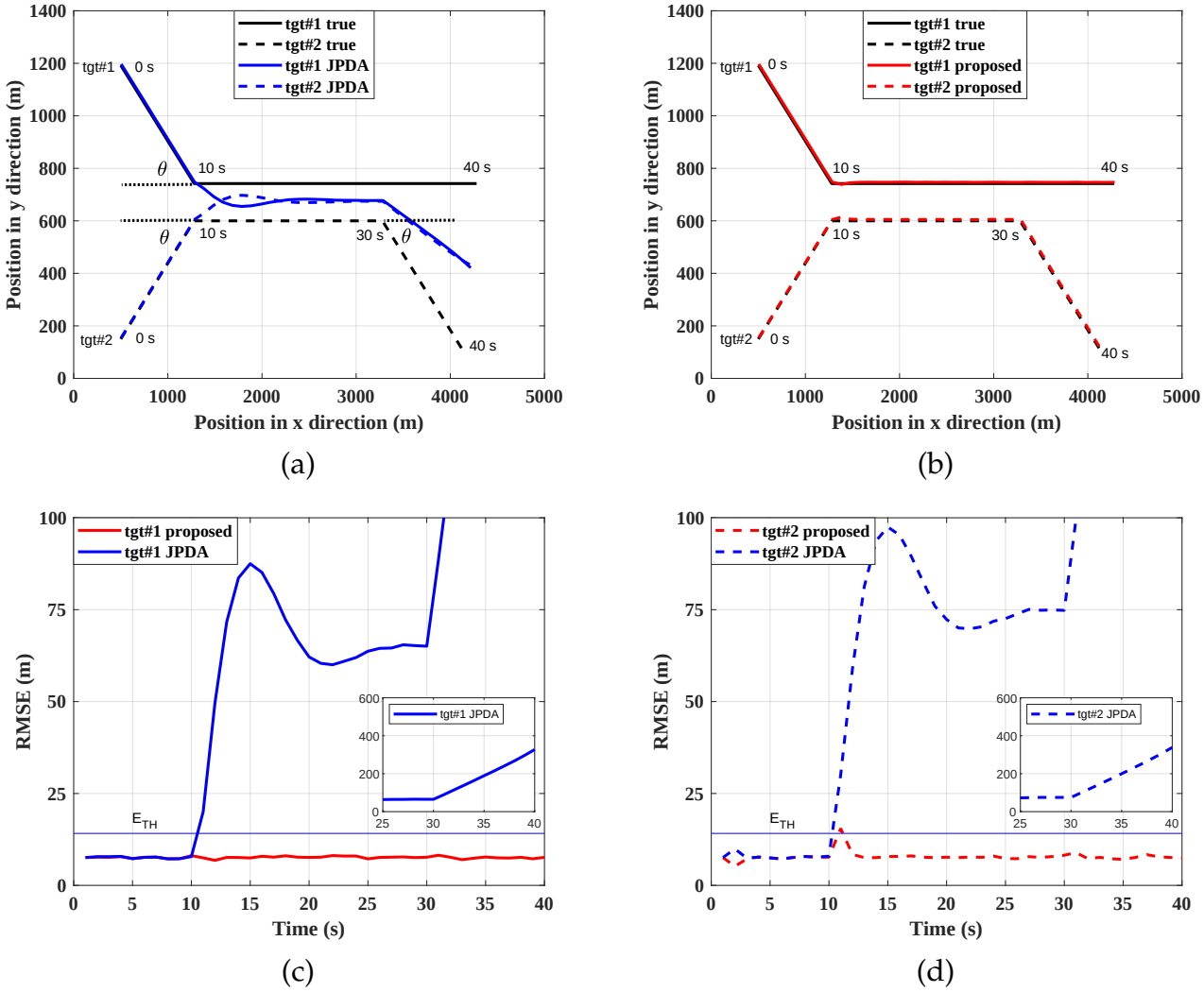


Figure 5.15: Tracking two targets each approaching at an angle $\theta = 30^\circ$, and then moving in parallel with $D = 140\,m$, and target 2 diverging at $\theta = 30^\circ$. True and estimated tracks using (a) JPDA algorithm and (b) proposed algorithm. Position RMSEs for (c) target 1 and (d) target 2.

and the two estimated tracks remain almost merged for the remainder of the tracking period. However, the RMSE values for the proposed algorithm do not exceed the threshold value at any point throughout the tracking period.

(xiv) **Example 13**

In this example, the two targets are approaching each other at an angle 30° for $10\,s$, then continue to move in parallel with $D = 140\,m$, and finally, at $30\,s$ they suddenly diverge with an angle of 30°. From Fig. 5.16(a), as expected, the coalescence of the tracks of

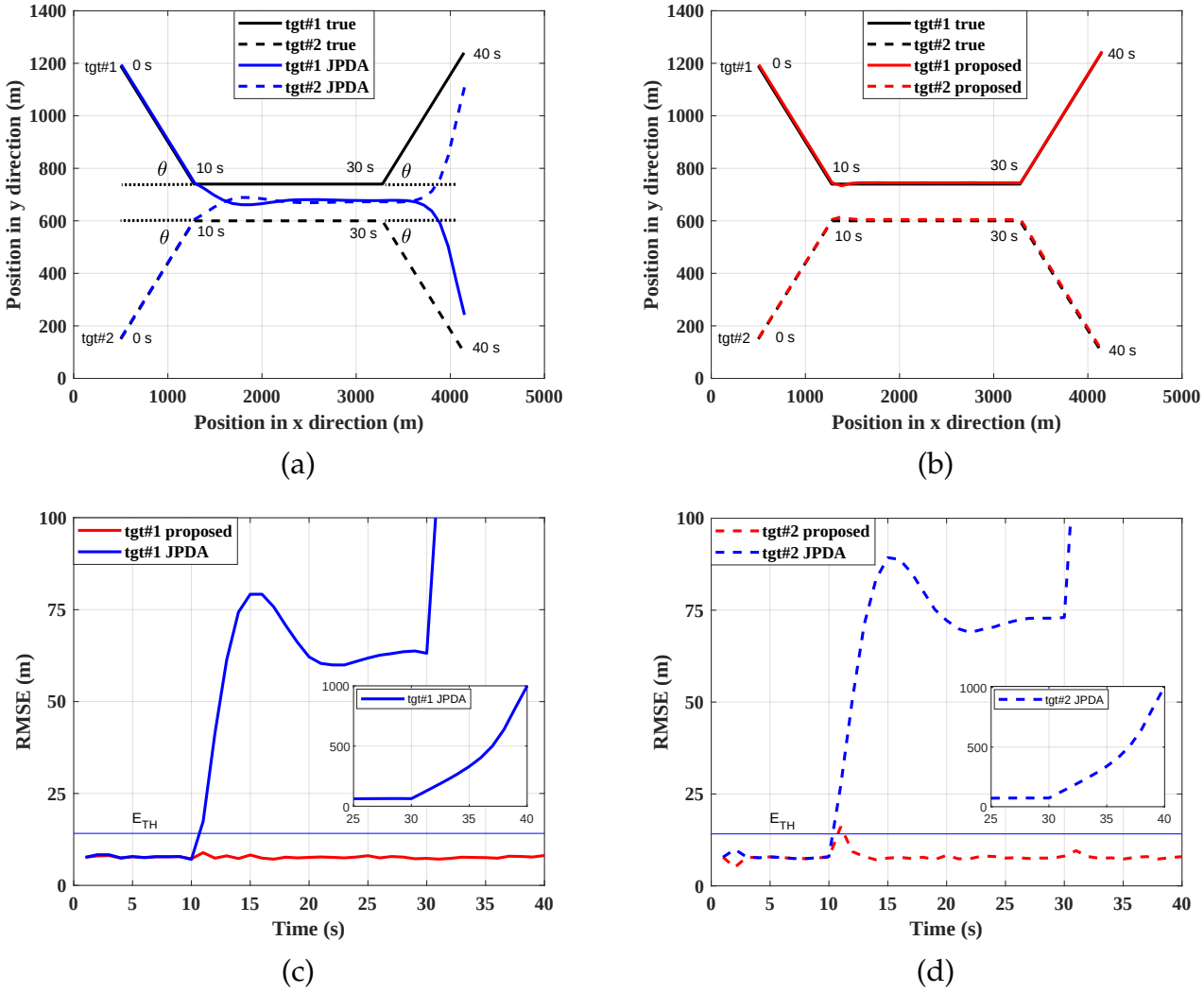


Figure 5.16: Tracking two targets each approaching at an angle $\theta = 30^\circ$, and then moving in parallel with $D = 140\,m$, and each diverging at $\theta = 30^\circ$. True and estimated tracks using (a) JPDA algorithm and (b) proposed algorithm. Position RMSEs for (c) target 1 and (d) target 2.

the two targets occurs for JPDA during the parallel part of the tracks; however, mis-identity of the targets takes place beyond the parallel part. On the other hand, the proposed algorithm gives a very good performance throughout the tracking period, as seen from Fig. 5.16(b). From Figs. 5.16(c) and 5.16(d), it seen that the RMSE values for JPDA, as expected, abruptly exceed the threshold value by a large amount during the parallel part, but now they get much worse beyond the parallel part of the tracks, thus leading to its complete failure in tracking the targets after $10\,s$. On the other hand, the RMSE values for the proposed algorithm do not exceed the threshold value at any point throughout the tracking period.

For JPDA, if the performance is unsatisfactory during the parallel part, it is seen from Examples 12 and 13 that it becomes worse when one or both the targets diverge, in the sense that the tracking errors get much worse along with the coalescence or mis-identity problem. On the other hand, the proposed algorithm continues to perform satisfactorily for the same divergence angle for which the JPDA algorithm fails. However, the problem encountered by JPDA would eventually occur for a spacing much smaller, or the divergence angle much larger than in the case of JPDA.

It is to be noted that the proposed algorithm for multiple target tracking is equally applicable in case of no clutter exists.

5.6 Summary

In this chapter, we have made a systematic and thorough study on the limitation of the JPDA algorithm with regard to the impact of the spacing between the targets as well as that of the abrupt turning angles of the targets on its tracking performance. In addition, we have proposed a new algorithm for the tracking of multiple targets within the JPDA framework based on the spatial distribution of the measurements falling inside the gate of a target. It has been shown that by considering the spatial distribution of the measurements, more appropriate values to the weights for the measurement- target association can be obtained. This has led to an enormously improved tracking performance of the proposed algorithm even when the targets are much more closely spaced and make much sharper turns, the cases in which the existing JPDA algorithm fails. A number of examples using two targets, with varied scenarios of different spacing between them and their turn angles, have been considered to demonstrate the effectiveness of the proposed algorithm and its superiority over the JPDA algorithm.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Target tracking is crucial in monitoring and controlling air traffic in civilian and military applications. One of the challenges in tracking single targets is during its maneuvering segments. Most of the algorithm that have been designed under the assumption that the maneuvering angles are known, and their performance deteriorates or may even fail when the angle of maneuver exceeds the assumed angle. In tracking multiple targets, tracking closely-spaced targets and keeping their identities is a major challenge.

This thesis has been concerned with the problems of accurately tracking single maneuvering targets when the target turn rates are not known and tracking multiple targets when they are closely spaced and/or make abrupt turns with large angles. These two problems have been investigated in two parts.

In the first part of the thesis, two algorithms have been proposed for tracking a single maneuvering target, when the information on the target turn rate is not known *a priori*, in an interacting multiple model (IMM) framework. In both the algorithms, the turn rate has been estimated dynamically using noisy measurements. In the first algorithm, the turn rate of the target has been estimated based on the target speed and the radius of a circle formed by three consecutive noisy measurements. It has been shown that this algorithm provides a more accurate value of the turn rate compared to that provided by the only other algorithm existing in the literature, and thus, yields a better tracking performance.It has also been shown that the performance of this algorithm is less sensitive than the existing algorithm to

increases in the levels of the measurement noise over the specified one used in the design of the algorithms. In the second algorithm, the accuracy of the estimated turn rate is improved by using the information on the level of the measurement noise. It has been shown that the tracking performance of this second algorithm is substantially superior to that of the first one, particularly when the measurement noise level is high.

In the second part of the thesis, the problem of tracking closely-spaced targets, while preserving their identities has been considered. A systematic and thorough study on the limitations of the JPDA algorithm in tracking closely spaced targets as well as when the targets make sudden turns with large angles has been conducted. A new algorithm for tracking multiple targets in a JPDA framework based on the spatial distribution of the measurements falling inside the gates of the targets has been proposed. It has been shown that by taking the spatial distribution of the target measurements into consideration, more appropriate weights can be assigned to the target-measurement associations, and hence, the tracking performance of the algorithm can be improved significantly.

This thesis has made important contributions in developing methods for accurately estimating the target turn rate for tracking single target and in developing a new method for computing weights for the target-measurement associations for multiple target tracking. The methods developed have been used to provide very good performance in some specific situations of target tracking. However, it can be expected that these methods are sufficiently general for them to be used in other realistic scenarios of target tracking.

6.2 Future Work

In this thesis, a number of algorithms for tracking single and multiple targets have been proposed based on new ideas for estimating the angles of target maneuvers in the case of single target tracking and measurement-target associations in the case of multiple target tracking. Based on these new ideas, there is scope for further investigation in target tracking.

- The algorithm for tracking a single maneuvering target proposed in Chapter 4 determines

empirically the multiplying factor for each measurement noise level. It would be worthwhile to develop a scheme for determining this multiplying factor adaptively rather than empirically.

- The algorithms for tracking a single maneuvering target proposed in Chapters 3 and 4 assume that there is no clutter. Hence, it is of interest to investigate if these algorithms can be extended to track a single maneuvering target in a clutter environment.

- A new multiple target tracking algorithm to track multiple targets, whether they are widely or closely spaced, has been proposed in Chapter 5; however, the algorithm assumes that the targets are non-maneuvering. This assumption may not hold good in real life situations, since the targets may perform maneuvers. Hence, it should be of great interest to study the tracking of multiple maneuvering targets, particularly when they are closely spaced.

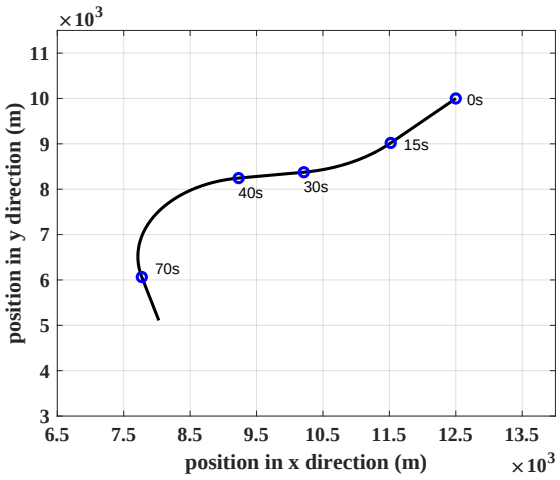
Appendix A

Determination of the State Transition Matrix

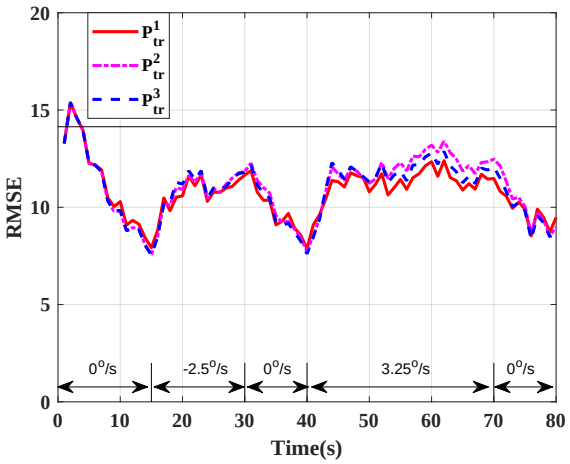
As mentioned earlier, the target has three modes of motion, namely, straight-line motion (SL), left turning motion (LT), and right turning motion (RT), which will be referred to as modes 1, 2, and 3, respectively. The element p_{ij}, $i \neq j$ in the transition probability matrix $\mathbf{P}_{tr}$ represents the probability of the target transiting from mode i to mode j, while p_{ii} represents the probability of the target continuing to be in mode i. It has been found in [26] that p_{ii} should be between 0.8 and 0.98 for good tracking results.

We assume that p_{22} and p_{33} should have values less than p_{11} in view of the fact that we adaptively estimate ω and the predicted position by the CT models are not precise. The other elements of this matrix are adjusted such that the sum of the elements in each row is equal to unity. A number of different transition matrices and various scenarios with different maneuvers have been considered, and it has been found that the best performance in terms of both RMSE and LANEES metrics is given by

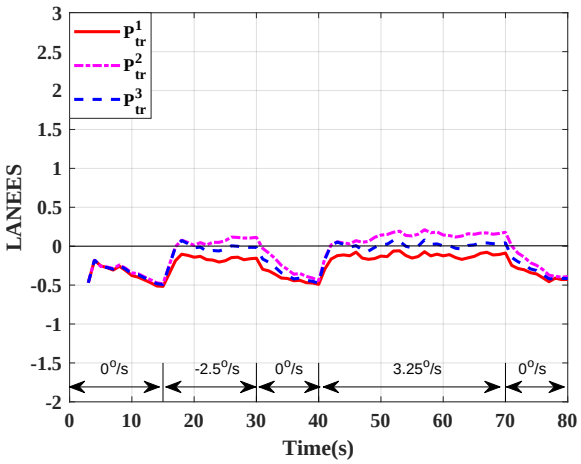
$$\mathbf{P}_{tr}^{1} = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \tag{A.1}$$



(a)



(b)



(c)

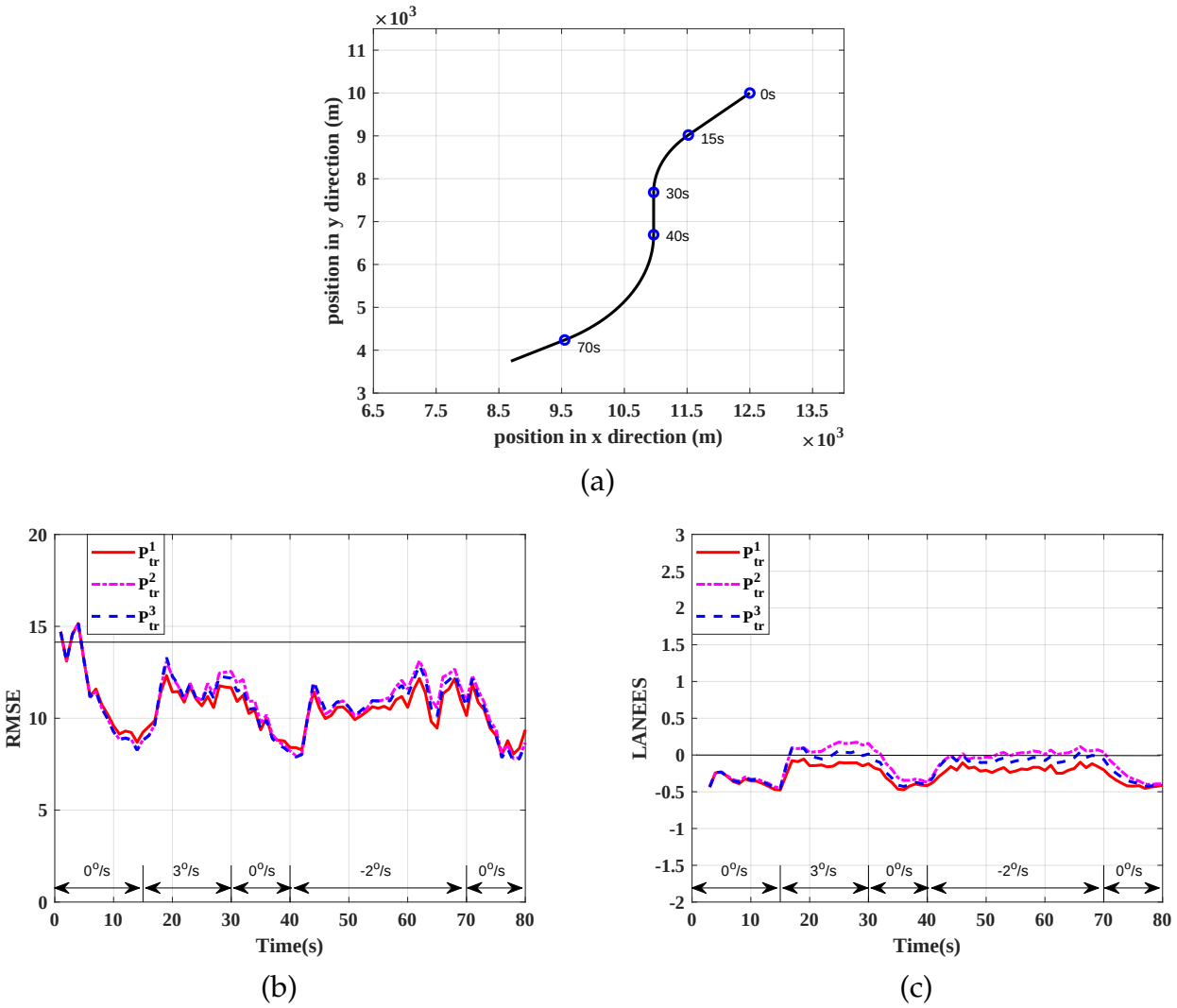
Figure A.1: Results of scenario 1 for a target that performs turns of $-2.5^\circ/s$, and $3.25^\circ/s$. (a) Actual target trajectory of scenario 1. (b) RMSE. (c) LANEES.

For purpose of illustration, we consider the following two other transition matrices

$$\mathbf{P}_{tr}^{2} = \begin{bmatrix} 0.95 & 0.025 & 0.025 \\ 0.2 & 0.8 & 0 \\ 0.2 & 0 & 0.8 \end{bmatrix} \tag{A.2}$$

and

$$\mathbf{P}_{tr}^{3} = \begin{bmatrix} 0.95 & 0.025 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.15 & 0.05 & 0.8 \end{bmatrix} \tag{A.3}$$



(a)

(b)

(c)

Figure A.2: Results of scenario 2 for a target that performs turns of $3^\circ/s$, and $-2^\circ/s$. (a) Actual target trajectory of scenario 2. (b) RMSE. (c) LANEES.

and compare their performance with that of $\mathbf{P}^1_{tr}$ under two different scenarios.

The first scenario is shown in Fig.A.1(a). The target starts with a straight-line motion for 15 seconds, then performs the first maneuver to the right with a turn rate of $-2.5^\circ/s$ for 15 seconds. Afterwards, it moves again in a straight-line motion for 10 seconds, and then performs another maneuver to the left with a turn rate of $3.25^\circ/s$ for 30 seconds. Finally, it goes in a straight-line motion for a further 10 seconds. The initial state vector used is

$$\mathbf{x}(o) = [12500\,m,\ -70.5\,m/s,\ 10000\,m,\ -70.5\,m/s]^t \tag{A.4}$$

The algorithm with $\mathbf{P}^1_{tr}$ shows a better tracking performance during the maneuvers in terms

of RMSE, as shown in Fig.A.1(b), compared to that of the algorithm with $\mathbf{P}_{tr}^{2}$ or $\mathbf{P}_{tr}^{3}$. It is clear from Fig.A.1(c) the algorithm with $\mathbf{P}_{tr}^{1}$ preserves its consistency throughout the tracking period, while the other two do not in the sense that LANEES for these two becomes positive during the maneuvers.

The second scenario is the same as the first one except for the maneuvering turn rates; the first maneuver is performed to the left with a turn rate of $3^{\circ}/s$ and the second maneuver to the right with a turn rate of $-2^{\circ}/s$, as shown in Fig.A.2(a). It is see from Figs.A.2(b) and A.2(c) that the conclusions made regarding RMSE and consistency of the algorithm with $\mathbf{P}_{tr}^{1}$, $\mathbf{P}_{tr}^{2}$ and $\mathbf{P}_{tr}^{3}$ for the previous scenario hold good for the present scenario also.

From the above two scenarios, it is clear that the performance of the algorithm in terms of both RMSE and LANEES is better with $\mathbf{P}_{tr}^{1}$ as the transition probability matrix than with $\mathbf{P}_{tr}^{2}$ or $\mathbf{P}_{tr}^{3}$. Similar performance of $\mathbf{P}_{tr}^{1}$ has been observed in other scenarios and with other transition probability matrices, but are not reported here. In view of these findings, we choose $\mathbf{P}_{tr}^{1}$ as the proposed transition probability matrix and denote it by $\mathbf{P}_{tr}^{PA1}$.

References

[1] Skolnik, *Introduction to radar system*, 3rd ed. New Delhi, India: Tata McGraw-Hill, 2001.

[2] D. K. Barton and S. A. Leonov, *Radar technology encyclopedia*. Norwood, MA: Artech house, 1998.

[3] M. A. Richards, *Fundamentals of Radar Signal Processing*, 2nd ed. New York, USA: McGraw-Hill, 2005.

[4] S. Challa, M. Morelande, D. Mušicki, and R. Evans, *Fundamentals of Object Tracking*. New York, USA: Cambridge University Press, 2011.

[5] C.-B. Chang and K.-P. Dunn, *Applied State Estimation and Association*. MA, USA: MIT press, 2016.

[6] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[7] M. B. Rhudy, R. A. Salguero, and K. Holappa, "A kalman filtering tutorial for undergraduate students," *International Journal of Computer Science and Engineering Survey*, vol. 8, no. 1, pp. 1–18, 2017.

[8] S. Haykin, *Adaptive Filter Theory*, 3rd ed. NJ, USA: Prentice Hall, 1996.

[9] H. Khazraj, F. Faria da Silva, and C. L. Bak, "A performance comparison between extended kalman filter and unscented kalman filter in power system dynamic state estimation," in *Proc. of the 51st International Universities Power Engineering Conference*, Sep. 2016, pp. 1–6.

[10] J. Shen, Y. Liu, S. Wang, and Z. Sun, "Evaluation of unscented kalman filter and extended kalman filter for radar tracking data filtering," in *Proc. of The 2014 European Modelling Symposium*, Oct 2014, pp. 190–194.

[11] F. Daowang, L. Teng, and H. Z. Tao, "Square-root second-order extended kalman filter and its application in target motion analysis," *IET Radar, Sonar Navigation*, vol. 4, no. 3, pp. 329–335, June 2010.

[12] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Trans. Autom. Control*, vol. 45, no. 3, pp. 477–482, March 2000.

[13] M. Zarei-Jalalabadi and S. M. Malaek, "Modification of unscented kalman filter using a set of scaling parameters," *IET Signal Proc.*, vol. 12, no. 4, pp. 471–480, 2018.

[14] A. F. Garcia-Fernandez, M. R. Morelande, and J. Grajal, "Truncated unscented kalman filtering," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3372–3386, July 2012.

[15] I. Arasaratnam and S. Haykin, "Cubature kalman filters," *IEEE Trans. Autom. Control*, vol. 54, no. 6, pp. 1254–1269, June 2009.

[16] S. Jie, Q. Guoqing, L. Yinya, and S. Andong, "Stochastic convergence analysis of cubature kalman filter with intermittent observations," *J. Syst. Eng. Electron.*, vol. 29, no. 4, pp. 823–833, Aug 2018.

[17] M. He and J. He, "A dynamic enhanced robust cubature kalman filter for the state estimation of an unmanned autonomous helicopter," *IEEE Access*, vol. 7, pp. 148 531–148 540, 2019.

[18] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb 2002.

[19] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, July 2010.

[20] Y. Xu, K. Xu, J. Wan, Z. Xiong, and Y. Li, "Research on particle filter tracking method based on kalman filter," in *Proc. of the 2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference*, May 2018, pp. 1564–1568.

[21] A. Farina and S. Pardini, "Survey of radar data-processing techniques in air-traffic-control and surveillance systems," in *Proc. of IEE conference of Communications, Radar and Signal Processing*, vol. 127, no. 3, 1980, pp. 190–204.

[22] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking: decision-based methods," in *Proc. of SPIE conference on Signal and Data Processing of Small Targets*, vol. 4728, 2002, pp. 511 – 534.

[23] J. Ru, A. Bashi, and X.-R. Li, "Performance comparison of target maneuver onset detection algorithms," in *Proc. of SPIE conference on Signal and Data Processing of Small Targets*, vol. 5428, 2004, pp. 419 – 428. [Online]. Available: https://doi.org/10.1117/12.553360

[24] H. Liu, Z. Zhou, and C. Lu, "Maneuvering detection using multiple parallel CUSUM detector," *Mathematical Problems in Engineering*, 2018.

[25] K. Ramachandra, *Kalman Filtering Techniques for Radar Tracking*. NewYork, USA: Marcel Dekker, 2000.

[26] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. New York, USA: John Wiley & Sons, 2004.

[27] Y. Bar-Shalom and K. Birmiwal, "Variable dimension filter for maneuvering target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-18, no. 5, pp. 621–629, Sep. 1982.

[28] J. A. Roecker and C. D. McGillem, "Target tracking in maneuver-centered coordinates," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 25, no. 6, pp. 836–843, 1989.

[29] J. A. Roecker and C. D. McGillem, "Target tracking in maneuver centered coordinates," in *Proc. of the 1988 IEEE National Radar Conference*, April 1988, pp. 68–73.

[30] T. Kirubarajan and Y. Bar-Shalom, "Kalman filter versus IMM estimator: when do we need the latter?" *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1452–1457, Oct 2003.

[31] G. Ackerson and K. Fu, "On state estimation in switching environments," *IEEE Trans. Autom. Control*, vol. 15, no. 1, pp. 10–17, February 1970.

[32] C. B. Chang and M. Athans, "State estimation for discrete systems with switching parameters," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-14, no. 3, pp. 418–425, May 1978.

[33] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with markovian switching coefficients," *IEEE Trans. Autom. Control*, vol. 33, no. 8, pp. 780–783, Aug 1988.

[34] K. Watanabe and S. G. Tzafestas, "Generalized pseudo-bayes estimation and detection for abruptly changing systems," *J. Intell. Rob. Syst.*, vol. 7, no. 1, pp. 95–112, Feb 1993.

[35] Xiao-Rong Li and Y. Bar-Shalom, "Multiple-model estimation with variable structure," *IEEE Trans. Autom. Control*, vol. 41, no. 4, pp. 478–493, April 1996.

[36] A. F. Genovese, "The interacting multiple model algorithm for accurate state estimation of maneuvering targets," *Johns Hopkins APL technical digest*, vol. 22, no. 4, pp. 614–623, 2001.

[37] A. Munir, J. A. Mirza, and A. Q. Khan, "Parameter adjustment in the turn rate models in the interacting multiple model algorithm to track a maneuvering target," in *Proc. of the IEEE International Multi Topic Conference, 2001. Technology for the 21st Century*, Dec 2001, pp. 262–266.

[38] H. You, X. Jianjuan, and G. Xin, *Radar data processing with applications*. Singapore: Wiley, 2016.

[39] A. Munir and D. P. Atherton, "Maneuvring target tracking using different turn rate models in the interacting multiple model algorithm," in *Proc. of 1995 34th IEEE Conference on Decision and Control*, vol. 3, Dec 1995, pp. 2747–2751.

[40] H. Wang, T. Kirubarajan, and Y. Bar-Shalom, "Precision large scale air traffic surveillance using IMM assignment estimators," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 35, no. 1, pp. 255–266, Jan 1999.

[41] R. Visina, Y. Bar-Shalom, and P. Willett, "Multiple-model estimators for tracking sharply maneuvering ground targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 3, pp. 1404–1414, June 2018.

[42] D. F. Bizup and D. E. Brown, "Maneuver detection using the radar range rate measurement," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 1, pp. 330–336, Jan 2004.

[43] X. Yuan, C. Han, Z. Duan, and M. Lei, "Adaptive turn rate estimation using range rate measurements," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 4, pp. 1532–1541, October 2006.

[44] V. B. Frencl and J. B. R. do Val, "Tracking with range rate measurements: Turn rate estimation and particle filtering," in *Proc. of the 2012 IEEE Radar Conference*, May 2012, pp. 0287–0292.

[45] L. Zhu and X. Cheng, "High manoeuvre target tracking in coordinated turns," *IET Radar, Sonar Navigation*, vol. 9, no. 8, pp. 1078–1087, 2015.

[46] V. B. Frencl, J. B. do Val, R. S. Mendes, and Y. C. Zuñiga, "Turn rate estimation using range rate measurements for fast manoeuvring tracking," *IET Radar, Sonar Navigation*, vol. 11, no. 7, pp. 1099–1107, 2017.

[47] J. B. Collins and J. K. Uhlmann, "Efficient gating in data association with multivariate gaussian distributed states," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, no. 3, pp. 909–916, July 1992.

[48] L. D. Stone, R. L. Streit, T. L. Corwin, and K. L. Bell, *Bayesian multiple target tracking*. Norwood, MA: Artech House, 2013.

[49] X. R. Li and Y. Bar-Shalom, "Tracking in clutter with nearest neighbor filters: analysis and performance," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 32, no. 3, pp. 995–1010, 1996.

[50] Y. Bar-Shalom and X. Li, *Multitarget-multisensor Tracking: Principles and Techniques*. Storrs, CT: YBS, 1995.

[51] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no. 5, pp. 451 – 460, 1975. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0005109875900217

[52] L. R. Kenari and M. R. Arvan, "Comparison of nearest neighbor and probabilistic data association methods for non-linear target tracking data association," in *Proc. of the 2014 Second RSI/ISM International Conference on Robotics and Mechatronics*, Oct 2014, pp. 047–052.

[53] Y. Bar-Shalom, *Multitarget-multisensor Tracking: Application and Advances*. Storrs, CT: YBS, 1996.

[54] E. Maggio and A. Cavallaro, *Video Tracking Theory and Practice*. West Sussex, UK: John Wiley & Sons, 2011.

[55] S. Stergiopoulos, *Advanced signal processing handbook: theory and implementation for radar, sonar, and medical imaging real time systems*. Florida, USA: CRC press, 2001.

[56] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, December 1979.

[57] W. L. Melvin and J. Scheer, Eds., *Principles of Modern Radar: Advanced techniques*. Institution of Engineering and Technology, 2012. [Online]. Available: https://digital-library.theiet.org/content/books/ra/sbra020e

[58] S. S. Blackman, *Multiple-target tracking with radar applications*. Dedham, MA: Artech House, 1986.

[59] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE J. Ocean. Eng.*, vol. 8, no. 3, pp. 173–184, July 1983.

[60] B.-S. Yaakov, D. Fred, and H. Jim, "The probabilistic data association filter: estimation in the presence of measurement origin un-certainty," *IEEE Control Syst. Mag.*, vol. 29, no. 6, pp. 82–100, 2009.

[61] D. F. Crouse, Y. Bar-Shalom, P. Willett, and L. Svensson, "The JPDAF in practical systems: computation and snake oil," in *Proc. of SPIE conference on Signal and Data Processing of Small Targets*, vol. 7698, 2010, pp. 428 – 442. [Online]. Available: https://doi.org/10.1117/12.848895

[62] R. J. Fitzgerald, "Track biases and coalescence with probabilistic data association," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-21, no. 6, pp. 822–825, Nov 1985.

[63] L. Svensson, D. Svensson, and P. Willett, "Set JPDA algorithm for tracking unordered sets of targets," in *Proc. of the 2009 12th International Conference on Information Fusion*, July 2009, pp. 1187–1194.

[64] L. Svensson, D. Svensson, M. Guerriero, and P. Willett, "Set JPDA filter for multitarget tracking," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4677–4691, Oct 2011.

[65] Y. Zhu, J. Wang, and S. Liang, "Efficient joint probabilistic data association filter based on kullback–leibler divergence for multi-target tracking," *IET Radar, Sonar Navigation*, vol. 11, no. 10, pp. 1540–1548, 2017.

[66] D. Mao, A. Xue, D. Peng, and Y. Guo, "An improved IMMJPDA algorithm for tracking multiple maneuvering targets in clutter," in *Proc. of the 2006 6th World Congress on Intelligent Control and Automation*, vol. 1. IEEE, 2006, pp. 4317–4320.

[67] X. Rong Li and V. P. Jilkov, "Survey of maneuvering target tracking. part I. dynamic models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, Oct 2003.

[68] J. R. Raol, *Data fusion mathematics: theory and practice*. London , UK: CRC Press, 2016.

[69] D. Lerro and Y. Bar-Shalom, "Tracking with debiased consistent converted measurements versus EKF," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 29, no. 3, pp. 1015–1022, July 1993.

[70] Y. Bar-Shalom, *Multitarget-multisensor tracking: advanced applications*. Norwood, MA: Artech House, 1990.

[71] S. V. Bordonaro, P. Willett, and Y. Bar-Shalom, "Unbiased tracking with converted measurements," in *Proc. of the 2012 IEEE Radar Conference*, May 2012, pp. 0741–0745.

[72] X. R. Li and Y. Bar-Shalom, "Design of an interacting multiple model algorithm for air traffic control tracking," *IEEE Trans. Control Syst. Technol.*, vol. 1, no. 3, pp. 186–194, Sep. 1993.

[73] G. Xie, L. Sun, T. Wen, X. Hei, and F. Qian, "Adaptive transition probability matrix-based parallel IMM algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, pp. 1–10, 2019.

[74] M. A. Richards, J. A. Scheer, and W. A. Holm, Eds., *Principles of Modern Radar: Basic principles*. Institution of Engineering and Technology, 2010. [Online]. Available: https://digital-library.theiet.org/content/books/ra/sbra021e

[75] X. R. Li and V. P. Jilkov, "A survey of maneuvering target tracking. part III: Measurement models," in *Proc. of SPIE Conference on Signal and Data Processing of Small Targets, San Diego, CA, USA*, Jul. 2001.

[76] J. S. Duncan, *pilot Handbook of aeronautical knowledge*. Oklahoma, USA: U.S Department of transportation, 2016.

[77] R. A. Best and J. P. Norton, "A new model and efficient tracker for a target with curvilinear motion," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 3, pp. 1030–1037, July 1997.

[78] B. Han, H. Huang, L. Lei, C. Huang, and Z. Zhang, "An improved IMM algorithm based on STSRCKF for maneuvering target tracking," *IEEE Access*, vol. 7, pp. 57 795–57 804, 2019.

[79] J. Wang, T. Zhang, X. Xu, and Y. Li, "A variational Bayesian based strong tracking interpolatory cubature kalman filter for maneuvering target tracking," *IEEE Access*, vol. 6, pp. 52 544–52 560, 2018.

[80] X. R. Li, Z. Zhao, and V. P. Jilkov, "Practical measures and test for credibility of an estimator," in *Proc. of Workshop on Estimation, Tracking, and Fusion - A Tribute to Yaakov Bar-Shalom*, 2001, pp. 481–495.

[81] M. Eltoukhy, M. O. Ahmad, and M. N. S. Swamy, "An adaptive turn rate estimation for tracking a maneuvering target," *IEEE Access*, vol. 8, pp. 94 176–94 189, 2020.

[82] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: a survey," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 1, pp. 103–123, Jan 1998.

[83] A. Ruina and R. Pratap, *introduction to statics and dynamics*. Oxford, U: Oxford University Press, 2015.

[84] D. S. Pietro, "Relating angular and regular motion variables," https://www.khanacademy.org/science/physics/torque-angular-momentum/rotational-kinematics/v/relating-angular-and-regular-motion-variables, November 2019.

[85] M. Eltoukhy, M. O. Ahmad, and M. N. S. Swamy, "An improved adaptive interacting multiple model algorithm for tracking a maneuvering target," *submitted for journal publication*.

[86] H. Kamel and M. Eltoukhy, "Implementation of storing data algorithm for tracking targets using labview," *International journal of advances in engineering and technology*, Feb 2017.

[87] J. R. Raol, *Multi-Sensor Data Fusion with MATLAB*, 1st ed. New York, USA: CRC Press, 2009.

[88] D. Musicki and M. Morelande, "Gate volume estimation for target tracking," in *Proc. of the International Conference on Information Fusion. Paris*, 2004.

[89] T. Kirubarajan and Y. Bar-Shalom, "Probabilistic data association techniques for target tracking in clutter," in *Proc. of the IEEE*, vol. 92, no. 3, March 2004, pp. 536–557.

[90] M. Eltoukhy, H. Kamel, and M. Hassan, "Implementation of a proposed multiple target tracking algorithm using labview," *International journal of advances in engineering and technology*, Sep 2015.

[91] M. Eltoukhy, M. O. Ahmad, and M. N. S. Swamy, "A new JPDA algorithm for tracking closely-spaced multiple targets," *submitted for journal publication*.

[92] B.-N. Vo, M. Mallick, Y. bar shalom, S. Coraluppi, R. III, R. Mahler, and B.-T. Vo, "Multitarget tracking," *Wiley Encyclopedia*, pp. 1–25, 09 2015.

[93] G. J. McLachlan, "Mahalanobis distance," *Resonance*, vol. 4, no. 6, pp. 20–26, 1999.