Target tracking in the range-Doppler space

Prepared by:

S.F. Middleton

MDDSTE003

Supervised by:

M.R. Inggs

Department of Electrical Engineering

August 2012

A dissertation submitted to the Department of Electrical Engineering,

University of Cape Town,

in partial fulfilment of the requirements

for the degree of

Master of Engineering specialising in *Radar*.

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this project report from the work(s) of other people, has been attributed and has been cited and referenced.
3. This project report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Signature of Author .....

Cape Town
31 August 2012

Abstract

Commensal radars make use of transmitters of opportunity to detect, locate and track targets in a manner that is non-detrimental to these transmitters. The target detections are displayed on amplitude-range-Doppler plots and follow curved trajectories with time. In the case of FM band commensal radars, these detections are subject to low and fluctuating range resolution making them difficult to follow visually. This complication can be alleviated by tracking the targets in the range-Doppler space. This project compares the Kalman, polynomial and recursive Gauss-Newton tracking filters for this purpose by using simulated and real data. The filter performance is evaluated on: tracking errors, computational load, data association statistics and real data tracking. All three filters perform well, however the recursive Gauss-Newton filter tracks tracks the most targets in the real data, achieves low errors and is the most efficient in terms of computational load.

Acknowledgements

I would like to thank my supervisor, Michael Inggs for the excellent guidance given throughout the project. For their support, I thank my parents and family. I thank Project Ledger for the funding provided, and also, for their help and guidance, I thank my colleagues Craig Tong and Roaldje Nadjiasngar.

Contents

List of Figures

List of Tables

List of Symbols

β_{fa}	—	Clutter density
B	—	Signal bandwidth
c	—	Speed of light
δR	—	Range resolution
λ	—	Forgetting factor of the recursive Gauss-Newton filter
P_d	—	Probability of detection
P_{ftc}	—	Probability of false track confirmation
P_{ttc}	—	Probability of true track confirmation
σ_a	—	Standard deviation of the acceleration noise
σ_R	—	Standard deviation of the range measurement noise
σ_d	—	Standard deviation of the Doppler measurement noise
θ	—	Fading parameter of the fading memory polynomial filter

Nomenclature

ARD plot—Amplidute-range-Doppler plot which shows the output of the matched filter.

Bistatic dissector—The dissector of the angle between the transmitter and receiver from the target.

Bistatic Doppler frequency—The Doppler frequency shift measured by a bistatic radar as a result of a target's radial motion with respect to the bistatic dissector.

Bistatic radar—A radar where the transmitter and receiver are at separate locations.

Bistatic range—The measurement of range made by a bistatic radar, i.e. the distance from the transmitter to the target to the receiver.

Centroiding—The process of resolving a range smeared target into a single point.

Constant False Alarm Rate (CFAR)—A detector that maintains a constant false alarm rate by adaptively adjusting the threshold.

Commensal radar—A radar system that makes use of transmitters of opportunity to detect and range targets without adversely affecting the operation of these transmitters.

Coherent Processing Interval (CPI)—In the context of commensal radar, the CPI refers to the integration time of the matched filter and is the time

between successive updates.

Data association—The process of determining which measurements to assign to a particular track.

Doppler frequency—A shift in the radio frequency of the return from a target or other object as a result of the object's radial motion relative to the radar.

Filtering—The estimation problem when the number of measurements and the time index are the same.

FMCR—FM band Commensal Radar.

GNSS—Global Navigation Satellite Systems.

Prediction—The estimation problem when the number of measurements is less than the time index.

Range smear—The spreading of a target across several range cells as a result of low range resolution.

RMS error—Root mean square error.

RRSG—Radar Remote Sensing Group.

Smoothing—The estimation problem when the number of measurements exceeds the time index, i.e. all the measurements are available immediately.

UTM—Universal Transverse Mercator.

Wi-Fi—Wireless local area network based on the IEEE 802.11 standards.

Chapter 1

Introduction

A commensal radar makes use of one or more transmitters of opportunity to detect and range targets. This is done in a manner that is non-detrimental to the transmitters in question and so allows for undetectable operation. As demand for bandwidth in the electromagnetic spectrum grows, a system that makes use of existing transmissions, such as commensal radar, presents itself as a clear advantage. Making use of transmitters of opportunity also allows commensal radars to operate at frequencies which would previously have been unavailable to radars because of spectrum licensing [1]. There is also a cost benefit: transmitter cost can be in the region of one third of the overall cost of a radar system. Reduced power consumption and longer integration times are also advantages.

Transmitters of opportunity can be anything from cell phone tower transmissions [2], analogue [3] and digital [4] television broadcasts and even space-borne (Global Navigation Satellite Systems (GNSS), etc.) broadcasts [5]. There have even been demonstrations of Wi-Fi based systems [6]. FM radio broadcasts are favoured because of their high average transmit powers, wide deployment, noise-like waveform and simple receiver structure [7].

UCT's Radar Remote Sensing Group (RRSG) has a FM band commensal radar (FMCR) system under development [8]. Targets are located by cross-correlating

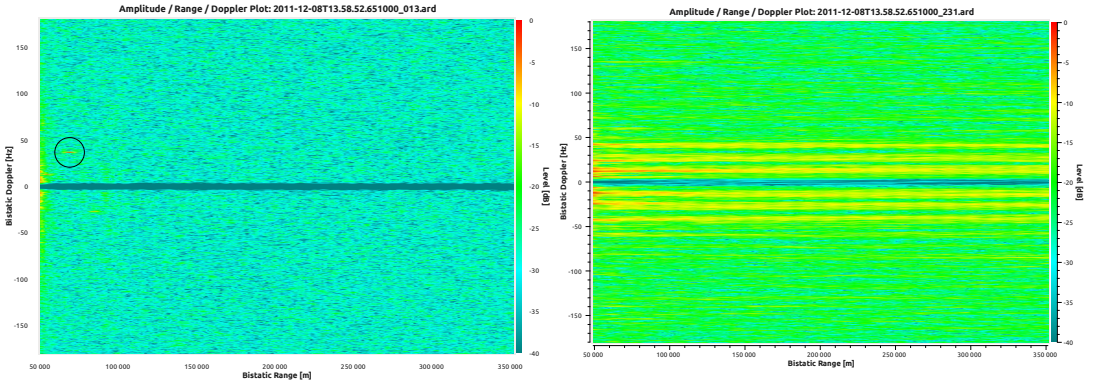

Figure 1.1: Example ARD plots produced by the RRSG's commensal radar system. Left: high bandwidth transmit signal producing a plot showing the target as a small orange blip (circled). Right: plot produced while transmit signal bandwidth is low resulting in range smear.

a reference signal with the reflected signal, producing amplitude-range-Doppler (ARD) plots. Targets appear as peaks at positions relating to their bistatic range and bistatic Doppler frequency. Displaying successive ARD plots allows one to observe the target progressing along a trajectory. However, the bandwidth of FM radio signals is relatively low (100 kHz at best), which gives rise to poor range resolution. In addition to this, the bandwidth varies with the content being broadcast [9] which results in 'range smear'. This range smear results in the target spreading out across most of the range cells for the duration of the low bandwidth transmission. Figure 1.1 shows two ARD plots produced by the RRSG's commensal radar system, one depicting a well defined target and the other showing an example of range smear.

From these ARD plots, target localization (by multilateration) and tracking needs to be performed [10]. This process is complicated, however, by the poor range resolution as well as range smear. While Doppler-only tracking has been proposed [11, 12, 13] to circumvent the issue of poor range resolution, there remain various data association issues that have to be solved. For example: whether or not all the receivers are observing the same target; and in the more general case of multiple targets per receiver, determining which return belongs to which target for each receiver. As a staring point, target tracking in the

range-Doppler space to produce smoother target trajectories and eliminate clutter is desirable. These plots could then serve as inputs to data association and localization techniques.

1.1 Problem statement

UCT's RRSG has a working commensal radar which can detect targets at bistatic ranges of up to 150 km. These targets appear on amplitude-range-Doppler plots which are formed by cross-correlating the surveillance signal with multiple Doppler-shifted copies of the reference signal [7]. After applying a constant false alarm rate (CFAR) detector, a plot of '1's (corresponding to detections) and '0's is obtained.

This plot displays the detections of true targets as well as false targets. True targets will move in the range and Doppler space from one scan to another. However, with the variations in instantaneous bandwidth, the range measurements can be very inaccurate. This, in addition to missed detections, make it difficult to follow the target visually on the CFAR'd range-Doppler plots. Thus, distinguishing true targets from false targets can be quite troublesome.

1.2 Objectives

The problem of identifying true targets amid numerous false detections, described in Section 1.1, can be dealt with by using linear tracking filters. The trajectories of true targets can be tracked in range-Doppler space making the targets easily distinguishable from the clutter points. In addition to this, a smoothed trajectory of the target will also be obtained, eliminating some of the errors introduced by poor range measurements.

The objectives of this project are then to:

- Identify, from the literature, suitable tracking filters that can be applied to

range-Doppler space tracking.

- Create a simulation environment in MATLAB where targets can be detected by a commensal radar and tracked in the range-Doppler space using some of the tracking filters identified from the literature.
- Compare the performance of the selected tracking filters by evaluating computational load, RMS position error and track initiation, true track confirmation and true track deletion statistics.
- Further asses the performance of the filters by applying them to real commensal radar data.

1.3 Overview

This section section describes the approach followed to meet the aforementioned objectives and also serves as an overview of the project report to follow.

Chapter two presents a concise theoretical background of the three filters being used. The aim is not to provide a comprehensive coverage of the theory, but rather to discuss each filter's merits and also examine their differences.

The chapter begins with the Kalman filter which is derived using probabilistic methods and can be computed recursively. Given its widespread application and trusted performance, the Kalman filter is used as the 'gold standard' for the project.

Next, the recursive form of the Gauss-Newton filter is considered. Unlike the Kalman filter's probabilistic approach, the Gauss-Newton filter uses statistical methods for its derivation. The linear, recursive form, however, leads to a filter very similar to the Kalman filter, differing only by a forgetting factor which makes the filter more adaptable to manoeuvring targets. This trait could be useful in the application to Commensal radars.

Following this, the polynomial filter is discussed. Similar to the recursive Gauss-

Newton filter, it is derived from statistical methods, where a polynomial is fitted to the observation data in the least squares sense. The expanding and fading memory variants are focused on, which are combined into the composite polynomial filter. This filter takes advantage of the self-initialising property of the expanding memory polynomial, as well as the adaptiveness of the fading memory polynomial at the cost of increased computational load. The polynomial appearance of target trajectories in ARD plots suggests that the polynomial filter might be particularly advantageous.

The chapter ends off by summarising the differences between the filters and forecasts how these differences might be either beneficial or detrimental to this particular estimation problem.

The filters discussed in Chapter 2 are then tested in simulations. This simulation environment is built in Chapter 3 where a target is injected into a two dimensional space and travels along a trajectory for a given period of time.

This target is then sensed by a commensal radar. The commensal radar in this scenario consists of a transmitter and a single receiver. The transmitter is modelled as an FM band, omni-directional transmitter. The receiver is separated from the transmitter by a base line of 20 km. The system measures the bistatic range and bistatic Doppler of the target at 1 s intervals.

Figure 1.2 shows the three filters tracking a target in range-Doppler space with accurate observations. As this is still a single target scenario with no clutter, applying the smoothing filters is a trivial matter requiring only track initiation and subsequent updates.

However, the issues of poor range resolution and range smear, which are inherent in commensal sensor systems, must also be simulated. In an ideal environment, centroiding the smeared data would produce a peak corresponding to the true target range value. The addition of noise, however, means that this peak could be anywhere within the range smeared Doppler cells.

Thus, a simple way to simulate range smear is to simulate a random, time varying bandwidth. This bandwidth relates to the range resolution by $\delta R = c/2B$, which

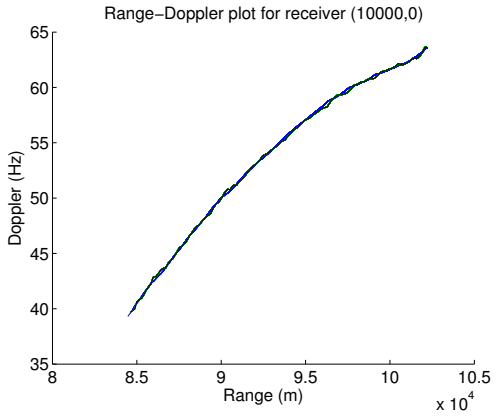

Figure 1.2: The polynomial, Kalaman and recursive Gauss-Newton filters tracking target in ARD-space.

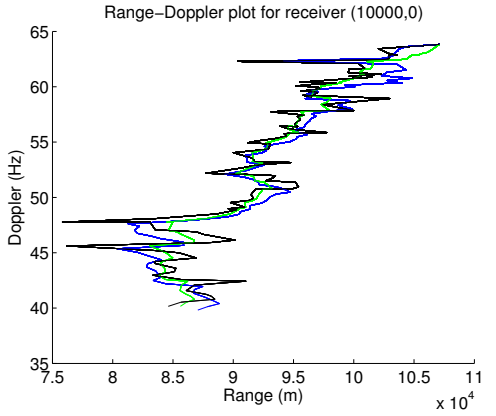

Figure 1.3: The polynomial, Kalaman and recursive Gauss-Newton filters tracking target in ARD-space with large range errors.

in turn gives the extent of the range smear. And so, by randomly choosing a point in this extent, a centroided peak is simulated.

As might have been expected however, these large range errors (up to 10 km) throw even the best tracking filter. This is demonstrated in Figure 1.3. One can see that by ignoring these occasional, large errors, the filter can continue to perform adequately.

The simulation environment goes on to include clutter. The clutter is inserted as false detections, the addition of which adds the requirement that the filters per-

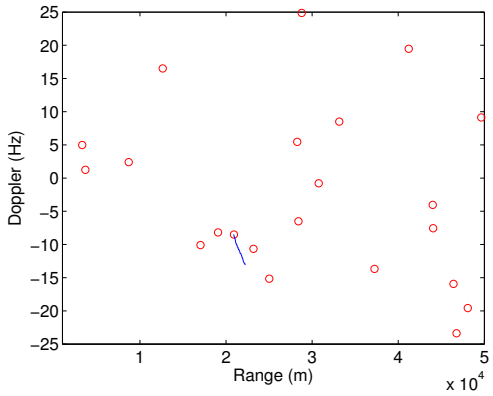

Figure 1.4: The Kalman filter tracking a target's bistatic range and bistatic Doppler amongst false detections.

form data association and gating. The clutter is added as uniformly distributed points in the range and Doppler dimensions. The number of false detections for each CPI is a Poisson distributed random variable. Nearest neighbour data association is used for the detections that fall within a given filter's gate. Conventional elliptical gating techniques are used. Figure 1.4 shows the Kalman filter's track (blue line) amongst false detections (red 'o's).

Having built the simulation environment, the next task is to evaluate the performance of the three different filters. This is done in Chapter 4, where the performance metrics considered are computational load, RMS error, peak errors and track initialisation and maintenance statistics.

Given the real-time requirements of target tracking, it is essential that the computational load of the filters be compared. The computational load is first estimated by counting the number of floating point additions and multiplications required for each filter's initialisation and update. Table 1.1 shows the tallies for the filter initialisations.

These results can then be compared to timing values from the simulations. Although the simulations are run in the infamously slow MATLAB, the filters can still be compared with each other to identify the least expensive from the computational load standpoint. The results for the filter update show that the RGN

Table 1.1: Number of arithmetic operations per filter initialisation.

Filter	Multiplications	Additions	Inverses
Kalman	240	260	1
Polynomial	30	21	1
RGN	257	260	1

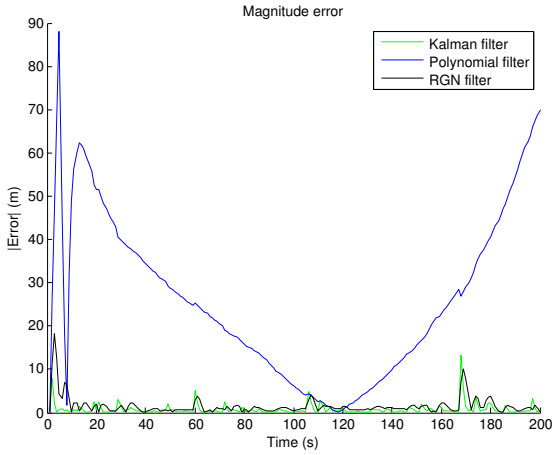

Figure 1.5: The magnitude of the range errors for the three different filters are plotted against time.

filter, with its simpler covariance matrix computation, is the fastest filter.

The position errors of the filters must also be considered. Figure 1.5 plots the magnitude of the position error against time for the three different filters and shows that the polynomial filter starts off with a poor position error, converges and then diverges. This behaviour of the polynomial filter comes as a trade-off with smoothness. The polynomial filter attempts to adaptively fit a polynomial to the data leading to a much smoother, but in some cases less accurate, plot. The Kalman and RGN filters maintain low errors.

Monte Carlo simulations give aggregate RMS position errors and produce figures similar to Figure 1.5. These results indicate that the Kalman filter obtains the lowest RMS range and Doppler errors. Increasing the forgetting factor of the RGN filter improves the range and Doppler errors. The polynomial filter on the other hand exhibits a trade-off with its fading parameter, with higher values

improving range tracking while lower values give better Doppler tracking.

Track confirmation statistics for each filter are determined by Monte Carlo simulation. A target is inserted into a clutter free space for a number of observations. The number of times the track is confirmed is counted and so the track confirmation statistic is obtained. The polynomial filter, implemented in this case as a composite EMP/FMP filter, has a particularly cumbersome initialisation process which was expected to degrade the track confirmation statistics, despite the conjectures of the literature. However, the results show that it performs the best out of the three in this regard.

True track and false track deletion statistics are also obtained by similar methods and once again the polynomial filter comes out best.

Although the results are more complicated and are subject to the chosen forgetting factor, fading parameter, etc., Table 1.2 summarises the key findings of this chapter.

Table 1.2: Summary indicating which filters performed the best in the three aspects considered.

Filter	Computational load	Tracking errors	Data association
Kalman		○	
Polynomial			○
RGN	○		

The next step in comparing the filters, is to apply them to real data. In Chapter 5, the filters are integrated into the processing chain and run on the output of the CFAR detector.

In order to mitigate the effects of the poor range resolution, the CFAR'd data is centroided to resolve multiple detections for a target into just one.

The chapter focuses on a target of interest which appears at the beginning of the dataset. The three different filters are run on this target. The measurement noise covariance matrices and gates are tweaked until satisfactory tracking is achieved. The effects of the fading parameter and forgetting factor of the the polynomial

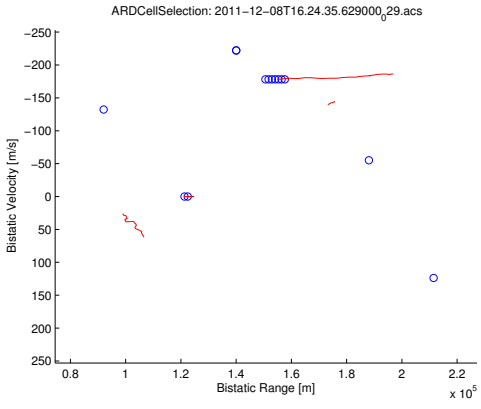

Figure 1.6: The RGN filter tracking the target of interest for the duration of its persistence with $\lambda = 0.8$.

and RGN filters respectively are also investigated.

In comparing the three filters against each other, it was found that the RGN filter, with a forgetting factor of 0.8, performed the best. This was found to be the case for the target of interest as well as the other targets which appeared later in the dataset. Figure 1.6 shows the RGN filter tracking the target of interest for the duration of its persistence.

The final chapter concludes that the RGN filter performs the best out the three filters considered. This is arrived at by taking all of the results into consideration. The RGN filter is computationally efficient, tracks accurately and performs the best on the real data.

The polynomial filter is not written off however, and it is acknowledged that with further work, focusing on a 1st degree polynomial with better data association techniques, it might still be able to outperform the RGN filter.

The Kalman filter's reliable performance was noted, as was the trade-off between tracking accuracy and data association performance that was governed by the filter and measurement covariance matrices.

Also recommended for future work is a consistent set of performance metrics to allow for better assessment of tracking filter performance.

These conclusions, however, lie at the end of the road and we must commence our investigation by reviewing the theory.

Chapter 2

Reviewing the theory

This chapter serves as a literature review in which the derivations of the various filters are summarised. Two main theoretical approaches exist in filtering theory which lead to various filtering techniques. The approaches considered are the statistical methods, comprising of least squares and maximum likelihood estimators, and the probabilistic methods.

The filters that will be discussed in fair detail, and taken further in this project, are the Kalman, polynomial and recursive Gauss-Newton filters. It is worth noting here that our problem is not a conventional tracking problem in Cartesian coordinates (or any other conventional coordinate system), but concerns rather the tracking of range and Doppler processes each governed by their own differential equation.

Each of these filters is well described in the literature. The Kalman filter is perhaps the most renowned filter for radar tracking applications, though generally for Cartesian tracking problems as opposed to our case of range-Doppler space tracking. The Kalman filter and its theory is described in various levels of detail in many publications [14, 15, 16, 17, 18]. Out of these, Jazwinski's "Stochastic processes and filtering theory" [16] gives the most in-depth derivations and will thus form the basis of our discussions on Kalman filtering theory.

The recursive Gauss-Newton filter is a recursive form of the Gauss-Newton filter,

a filter particularly good at non-linear estimation problems. However, the Gauss-Newton filter, based on the weighted least squares method and Newton's method of local linearisation [17, 19] , is computationally expensive [20]. Thus a recursive form was developed by Nadjiasngar [21] and further adapted to the Levenberg-Marquardt [22] method to improve convergence [23]. The filter is characterised by its forgetting factor, which sets the effective memory length of the filter. A shorter memory length makes the filter more adaptive, making it suitable to manoeuvring target tracking. In linearising the filter, one arrives at a filter very similar to that of the Kalman filter, differing only by the presence of the forgetting factor. Setting this forgetting factor to unity leads one to the Kalman filter.

Polynomial filters take a least squares approach to state estimation by fitting a polynomial to the observation data in the least squares sense. The estimate will be either good or bad, depending on how well the process can be estimated by the polynomial (of chosen degree m). Polynomial filtering theory is described in Morrison's books [17, 19], with the former serving as the reference for this component of the literature review. The expanding memory and fading memory polynomial filters (EMP and FMP respectively) are considered for the implementation of the composite polynomial filter [24]. The EMP is self-initialising and is intended to cope with situations where all the data is not immediately available and can be formulated recursively. FMP filters, on the other hand derive from weighted least squares where the weight factor decays exponentially. Thus, the effects of anomalous events will be decreased as time moves on; the filter is more adaptable. The FMP is not self initialising and so the composite polynomial filter is used to take advantage of both the FMP and EMP.

Next up, we look at the formulation of the Kalman filter.

2.1 Kalman filter

The Kalman filter - and its non-linear form, the extended Kalman filter - have been used extensively over the years in a variety of applications. These appli-

cations range from conventional tracking to target tracking using Doppler and bearings information only [25, 26], to passive television [27] and Wi-Fi [6] based target tracking and range-Doppler tracking [7] to name but a few.

There have also been modifications to the non-linear form to prevent premature covariance collapse and subsequent filter divergence. This is done by re-evaluating the linearisation technique. The unscented Kalman filter [28], the finite difference extended Kalman filter [29] as well as the strong tracking finite difference extended Kalman filter [30] are but a few of the many examples. Needless to say, the Kalman filter has been the focus of much research and has been widely applied practically.

We consider only the linear Kalman filter. Jazwinski pursues the probabilistic filtering approach in his book, a concise summary of which is presented here. Opposed to the probabilistic approach, statistical methods make use of maximum likelihood, least squares or recursive least squares to form state estimates. They assume that the process noise and measurement noise are not random inputs with well defined statistics but rather errors of an unknown character. Thus,

$$x_{k+1} = \phi(x_k, t_{k+1}, t_k) + \Gamma(x_k, t_k) w_{k+1} \tag{2.1}$$

which describes the discrete stochastic dynamical system, can be thought of as an ordinary difference equation that can be solved if the errors are known. x_k is the n-vector state at time t_k ($k = 0, 1, 2...$) and ϕ is the n-vector transition function. Γ is the noise coefficient matrix and w_k is the white Gaussian sequence $w_k \sim N(0, Q_k)$. Given the noisy observations,

$$y_k = h(x_k, t_k) + v_k \tag{2.2}$$

where h is the measurement function and $v_k \sim N(0, R_k)$ is the measurement noise with covariance R_k, x_k can be estimated with small dynamical system and observation errors by taking the least squares approach. Recursive least squares methods prevent the *ad infinitum* accumulation of observations by solving the

solution in terms of the previous estimate. However, Jazwinski argues that these least squares methods do nothing but re-derive the work of the probabilistic methods and have no real probabilistic meaning. While these methods are simpler, the results can be more difficult to interpret.

Probabilistic methods, while involving stochastic differential equations (and hence greater complexity), present formal approaches to the probabilistic problem with clear meaning and significance. The probabilistic methods interpret the errors in Equations 2.1 and 2.2 as random sequences with well defined statistics. It is shown that maximising the conditional probability density function:

$$p(x_0, ...x_N|y_1, ..., y_N) \tag{2.3}$$

with respect to $\{x_0, ..., x_N\}$ is equivalent to the least squares problem (provided Γ in 2.1 is a function of time only). This is referred to as joint maximum likelihood estimation where the estimate is the peak of the joint conditional density (2.3).

The derivation of the filtering algorithm, which is essentially an equation for the conditional mean, is quite complex. The non-linear case is first derived and then specialised to the linear case. Making use of stochastic calculus theory such as Brownian motion processes, Itô integrals and Kolmogorov's forward equation, the time evolution of the conditional density, its moments as well as mode are analysed.

The evolution of the conditional density leads to:

$$p(x, t_k|Y_{t_k}) = \frac{p(y_k|x)p(x, t_k|Y_{t_k}^-)}{\int p(y_k|\zeta)p(\zeta, t_k|Y_{t_k}^-)d\zeta} \tag{2.4}$$

where $Y_{t_k}^-$ is the general representation of the previous measurements at any time $t_k \leq t < t_{k+1}$ and $Y_{t_k} = \{y_{t_1}, ...y_{t_k}\}$ is a sequence of measurements. Using this result, along with the conditional mean and covariance obtained from the evolution of moments and knowledge that $p(x, t_k|Y_{t_k}) \sim N(\hat{x}_{t_k}^{t_k^+}, P_{t_k}^{t_k^+})$, it is concluded that:

$$\hat{x}_{t_k}^{t_k^+} = (M^T(t_k)R_k^{-1}M(t_k) + (P_{t_k}^{t_k^-})^{-1})^{-1}(M^T(t_k)R_k^{-1}y_k + (P_{t_k}^{t_k^-})^{-1}\hat{x}_{t_k}^{t_k^-}) \tag{2.5}$$

and

$$(P_{t_k}^{t_k^+})^{-1} = (P_{t_k}^{t_k^-})^{-1} + M^T(t_k)R_k^{-1}M(t_k) \tag{2.6}$$

$M(t_k)$ is the generalisation of the measurement matrix $h(x_k, t_k)$ for the linear case and $P_{t_k}^{t_k^+}$ is the covariance. The notation here (eg. $\hat{x}_{t_k}^{t_k^+}$) refers to the filtered quantity (in this case $\hat{x}$) in the non-discrete time interval from t_k to t_k^+. Recasting the above equations yields the set of equations:

$$P_{t_k}^{t_k^+} = P_{t_k}^{t_k^-} + P_{t_k}^{t_k^-}M^T(t_k)[M(t_k)P_{t_k}^{t_k^-}M^T(t_k) + R_k]^{-1}M(t_k)P_{t_k}^{t_k^-} \tag{2.7}$$

$$\hat{x}_{t_k}^{t_k^+} = \hat{x}_{t_k}^{t_k^-} + P_{t_k}^{t_k^-}M^T(t_k)[M(t_k)P_{t_k}^{t_k^-}M^T(t_k) + R_k]^{-1}(y_k - M(t_k)\hat{x}_{t_k}^{t_k^-}) \tag{2.8}$$

and defining the Kalman gain

$$K(t_k) \triangleq P_{t_k}^{t_k^-}M^T(t_k)[M(t_k)P_{t_k}^{t_k^-}M^T(t_k) + R_k]^{-1} \tag{2.9}$$

gives the simpler forms:

$$P_{t_k}^{t_k^+} = P_{t_k}^{t_k^-} + K(t_k)M(t_k)P_{t_k}^{t_k^-} \tag{2.10}$$

$$\hat{x}_{t_k}^{t_k^+} = \hat{x}_{t_k}^{t_k^-} + K(t_k)(y_k - M(t_k)\hat{x}_{t_k}^{t_k^-}) \tag{2.11}$$

which are easily recognisable as the Kalman filter equations in the continuous-discrete form. Obtaining the discrete form and adopting the notation that will be used from here on results in the equations:

$$K_k \triangleq P_{k|k-1} M_k^T [M_k P_{k|k-1} M_k^T + R_k]^{-1} \tag{2.12}$$

$$P_{k|k} = P_{k|k-1} + K_k M_k P_{k|k-1} \tag{2.13}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - M_k \hat{x}_{k|k-1}) \tag{2.14}$$

where $P_{k|k}$ is the filter covariance matrix which gives an indication as to how well the filter is performing. $\hat{x}_{k|k}$ is the filtered estimate of the state x_k. The quantity $(y_k - M_k \hat{x}_{k|k-1})$ is referred to as the innovation and has the covariance matrix $S = M_k P_{k|k-1} M_k^T + R_k$ [14]. The inverse of the innovation covariance matrix (S^{-1}) is used in the gating procedure which is discussed in the next chapter.

In the event that the Kalman gain is not exact (due to computation error, for example), the stabilised form of the filter covariance matrix:

$$P_{k|k} = [I - K_k M_k] P_{k|k-1} [I - K_k M_k]^{-1} + K_k R_k K_k^T \tag{2.15}$$

should be used instead of Equation 2.13 [15].

But the Kalman filter is a 'predictor-corrector' [18] and thus far we have only 'corrected'. Prediction is performed with the transition matrix Φ giving:

$$\hat{x}_{k+1|k} = \Phi \hat{x}_{k|k} \tag{2.16}$$

$$P_{k+1|k} = \Phi P_{k|k} \Phi^T + Q_k \tag{2.17}$$

The above equations, along with Equations 2.12 and 2.14 form the basis of the Kalman filter that is implemented and tested in the following chapters. The stabilised computation of the covariance matrix (2.15) is used throughout. The filter, well established in the field of tracking, is used in the remainder of this project report as the 'gold standard'. The RGN and polynomial filters will be pitched against the Kalman filter, implemented in the form above.

2.2 Recursive Gauss-Newton filter

The Gauss-Newton filter, in its non-recursive form, was invented by Gauss [20] and is discussed thoroughly in Morrison's books [17, 19]. Several strengths over the extended Kalman filter are discussed by Morrison [20, 31], however the computational load remains high. The recursive form was developed to reduce the computational requirements leading to a filter similar to the iterated extended Kalman filter, differing by the addition of a forgetting factor [21].

We summarise the derivation of the GN filter from minimum variance estimation and the subsequent development of the recursive form developed by Nadjiasngar [21]. This essentially involves weighted least squares estimation and local linearisation of the process and measurement functions. Hence, the Gauss-Newton filter. The equations arrived at, however, are for non-linear problems and so specialisations to the linear case must be made for our linear problem.

Firstly, a process state is considered which is governed by the non-linear differential equation:

$$DX(t) = F(X(t)) \tag{2.18}$$

and observed by the non-linear function:

$$Y(t) = G(X(t)) + v(t) \tag{2.19}$$

F is a non-linear vector function and G is a non-linear function which describe the process and observation schemes respectively. $v(t)$ is a random Gaussian vector. We should note the lack of process noise in Equation 2.18 in comparison to Equation 2.1.

Using Newton's local linearisation method, a perturbation vector ($\delta X(t)$) is arrived at from the expression $X(t) = \bar{X}(t) + \delta X(t)$. This expression results from the fact that the given initial condition will result in infinitely many solutions to

the differential equation. $\bar{X}(t)$ is one solution which satisfies the same differential equation as, and is close to, $X(t)$.

This perturbation vector, which is essentially an error in the state estimate, is also governed by a differential equation:

$$D(\delta X(t)) = A(\bar{X}(t))\delta X(t) \tag{2.20}$$

where

$$A(\bar{X}(t)) = \frac{\partial F(X(t))}{\partial (X(t))}|_{\bar{X}(t)} \tag{2.21}$$

is known as the sensitivity matrix and means that Equation 2.20 is a linear differential equation with the transition equation:

$$\delta X(t+\zeta) = \Phi(t_n + \zeta, t_n, \bar{X})\delta X(t) \tag{2.22}$$

From here, the perturbation vector for the non-linear observation scheme is obtained, firstly by defining a simulated noise free vector $\bar{Y}_n = G(\bar{X}_n)$, which subtracted from the actual observation Y_n gives:

$$\delta Y_n = Y_n - \bar{Y}_n \tag{2.23}$$

This can be shown to be related to the state perturbation vector by $\delta Y_n = M(\bar{X}_n)\delta X_n + v_n$, where $M(\bar{X}_n)$ is the Jacobean of G evaluated at $\bar{X}_n$.

Assembling $L+1$ observations, the equation $\delta \mathbf{Y}_n = \mathbf{T}_n \delta X_n + \mathbf{V}_n$ is obtained, where $\mathbf{T}_n$ is referred to as the total observation vector and is defined as:

$$\mathbf{T}_n = \begin{bmatrix} M(\bar{X}_n) \\ M(\bar{X}_{n-1})\Phi(t_{n-1}, t_n; \bar{X}) \\ \vdots \\ M(\bar{X}_{n-L})\Phi(t_{n-L}, t_n; \bar{X}) \end{bmatrix} \tag{2.24}$$

Minimum variance estimation is then used to estimate the perturbation vector

$$\delta\hat{X}_n = (\mathbf{T}_n^T \mathbf{R}_n^{-1} \mathbf{T}_n)^{-1} \mathbf{T}_n^T \mathbf{R}_n^{-1} \delta\mathbf{Y}_n \tag{2.25}$$

with covariance matrix $S_n = (\mathbf{T}_n^T \mathbf{R}_n^{-1} \mathbf{T}_n)^{-1}$ and the least squares weight matrix $\mathbf{R}_n^{-1}$. This weight matrix can be arbitrarily assigned and is the measure of our belief in the observation model [16]. The values can be fiddled to obtain the best least squares fit. In this case, $\mathbf{R}_n^{-1}$ is defined as a matrix of the inverses of the observation error covariance matrices R_n.

The above equations derive a filter where the corrected state estimate is obtained through the relationship $\hat{X}_n^c = \hat{X}_n + \delta\hat{X}_n$. However the memory requirements of the total observation vector $\mathbf{T}_n$ make the filter computationally expensive.

The recursive formulation, intended to mitigate this computational load, starts by introducing a forgetting factor $\lambda \leq 1$ to the least squares weight matrix $\mathbf{R}_n^{-1}$ as follows:

$$\mathbf{R}_n^{-1} = \begin{bmatrix} R^{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \lambda R^{-1} & & \vdots \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \cdots & \cdots & \lambda^n R^{-1} \end{bmatrix} \tag{2.26}$$

Defining $\delta\hat{X}_n = \mathbf{W}_n^{-1}\xi_n$, where $\mathbf{W}_n = \mathbf{T}_n^T \mathbf{R}_n^{-1} \mathbf{T}_n$ and $\xi = \mathbf{T}_n^T \mathbf{R}_n^{-1} \delta\mathbf{Y}_n$, the recursive update is obtained, after some fiddling, from the equations:

$$\delta\hat{X}_n = K_n[Y_n - G(\bar{X}_n) - M(\bar{X}_n)(\hat{X}_{n|n-1} - \bar{X}_n)] \tag{2.27}$$

and

$$W_{n|n}^{-1} = \lambda^{-1}[I - K_n M(\bar{X}_n)]W_{n|n-1}^{-1} \tag{2.28}$$

where an observer gain has been defined as:

$$K_n = W_{n|n-1}^{-1} M(\bar{X}_n)^T [R_n + M(\bar{X}_n) W_{n|n-1}^{-1} M(\bar{X}_n)^T]^{-1} \tag{2.29}$$

Thus, a corrected state estimate $\bar{X}_n = \hat{X}_{n-1} + \delta\hat{X}_n$ can be obtained iteratively for a predetermined number of runs when $\bar{X}_n = \hat{X}_{n-1}$.

Equations 2.27, 2.28 and 2.29 can immediately be recognised as the iterative extended Kalman filter (IEKF) equations [14, 16] with the addition of the forgetting factor λ. Thus, setting λ to unity leaves one with the IEKF, arrived at from minimum variance methods as opposed to the probabilistic methods of Section 2.1.

However, Equations 2.27, 2.28 and 2.29 are for non-linear processes and non-linear observation schemes. Thus we abandon the iterative least squares approach, which is necessary for the non-linear least squares problem. The non-linear process and observation functions and their linearised approximates are replaced by linear process and observation functions.

Thus, instead of searching for δX_n to minimise the weighted sum of squared deviations in the equation $\delta\mathbf{Y}_n = \mathbf{T}_n\delta X_n + \mathbf{V}_n$, we rather look at $Y_n = HX_n + v_n$. H is the linear observation scheme. Noting these similarities, the following equations are arrived at:

$$\hat{X}_{n|n} = \hat{X}_{n|n-1} + K_n[Y_n - H\hat{X}_{n|n-1}] \tag{2.30}$$

$$W_{n|n}^{-1} = \lambda^{-1}[I - K_n H]W_{n|n-1}^{-1} \tag{2.31}$$

$$K_n = W_{n|n-1}^{-1} H^T [R_n + H W_{n|n-1}^{-1} H^T]^{-1} \tag{2.32}$$

which, apart from the presence of the forgetting factor and slightly different notation, are exactly the same as the Kalman filter equations (2.12, 2.13, 2.14) arrived at in Section 2.1.

Given the strong similarities between the equations arrived at here and the Kalman filter equations in Section 2.1, it might seem a bit convoluted and inaccurate to refer to Equations 2.30, 2.31 and 2.32 as the recursive Gauss-Newton filter when they simply appear as slightly modified forms of the Kalman filter, arrived at by the statistical methods instead of probabilistic methods. However, we will continue refer to these equations as the recursive Gauss-Newton filter for two reasons: simplicity in differentiating whether or not a forgetting factor is being used and; to emphasise the fact that this filter, while very similar to the Kalman filter, was derived from statistical methods.

Now we look at the polynomial filter which also uses the least squares statistical methods, but with a different approach.

2.3 Polynomial filter

Polynomial filtering is concerned with fitting a polynomial, in the least squares sense, to a sequence of L observations and was introduced in Morrison's book [17]. The Kalman filter overshadowed it, however, perhaps because of superior computational efficiency. A possible advantage of the polynomial filter over the Kalman filter is in high clutter and low probability of detection environments where the Kalman filters's covariance matrix becomes small [24]. This small covariance matrix affects the data association techniques which in turn leads to track deletion. Given the often parabolic trajectory of targets in the range-Doppler space, using polynomial filters to track targets in the range-Doppler space might be of particular interest.

This section summarizes the relevant aspects of polynomial filtering from Morrison's book [17], starting with the fixed-memory polynomial filter in its simplest form. The classical least squares method of polynomial filtering begins by first

choosing a polynomial in r of first degree, for example:

$$[p^*(r)]_n = (\beta_0)_n + (\beta_1)_n r \tag{2.33}$$

where the $*$ indicates that we are dealing with an estimate and n indicates that the total observation vector $Y_n = \{y_{n-L}, ..., y_{n-1}, y_n\}$ of length L is being considered. The coefficients (β's) remain arbitrary, to be estimated by the least squares method.

The sum of squared residuals is then:

$$e_n = \sum_{r=0}^{L} \left[y_{n-L+r} - \sum_{j=0}^{1} (\beta_j)_n r^j \right]^2 \tag{2.34}$$

which is minimised by setting $\partial e_n / \partial (\beta_i)_n = 0$ for $i = 0, 1$. The solution obtained cannot be easily represented in functional form and has to be obtained numerically. This problem is circumvented by making use of orthogonal polynomials to obtain a functional form. This is not shown here as the basic concept of polynomial filtering has been presented and we now move on to the polynomial filters that concern us: the expanding memory and fixed memory polynomial filters.

2.3.1 Expanding memory polynomial

The expanding memory polynomial filter allows for filtering to commence upon arrival of the first observation. This is of particular importance in tracking applications where all the data is not immediately available.

In the same manner as above, a least squares polynomial fit will be performed on the data, except that in this case, the filter memory will be expanding with each new observation. In addition to this, a recursive formulation can be obtained reducing the memory requirements of the algorithm. The increasing database of observations results in the filter becoming smoother as time progresses.

The method makes use of Legendre polynomials and assumes that the observations are equally spaced in time (a limitation which is addressed by Reyneke [32]). The same least squares fit is performed and the coefficients found to minimize e_n are:

$$(\beta_j)_n = \frac{\sum_{k=0}^{n} p(k;j,n)y_k}{[c(j,n)]^2} \tag{2.35}$$

where $0 \leq j \leq m$, m is the polynomial degree, and n is the temporarily frozen time index which gives the orthogonality range of the polynomial ($L = n$). $[c(j,n)]^2 = \sum_{r=0}^{j}[p(r;j,n)]^2$, and $p(r;j,n)$ are the discrete polynomials satisfying the orthogonality condition and are given by:

$$p(r;j,n) \equiv \sum_{\nu=0}^{j}(-1)^{\nu}\binom{j}{\nu}\binom{j+\nu}{\nu}\frac{r^{(\nu)}}{n^{(\nu)}} \tag{2.36}$$

It should be noted that n is both the data counter as well as the memory length - hence expanding memory.

The above expressions, however, depend on all observations being available in memory. Thus, to alleviate the computational load, a recursive formulation can be obtained. The recursive derivation, which starts by setting $r = n + 1$ and involves much algebra, results in the equations (for a 1st degree polynomial):

$$(z_1^*)_{n+1,n} = (z_1^*)_{n,n-1} + \frac{6}{(n+2)(n+1)}\epsilon_n \tag{2.37}$$

$$(z_0^*)_{n+1,n} = (z_0^*)_{n,n-1} + (z_1^*)_{n+1,n} + \frac{2(2n+1)}{(n+2)(n+1)}\epsilon_n \tag{2.38}$$

where $\epsilon_n \equiv y_n - (z_0^*)_{n,n-1}$ is the error or innovation and $(z_k^*)_{n+1,n}$ is the updated estimate of the k-th derivative ($0 \leq k \leq m$) of the m-th degree polynomial based on the n-th observation as well as the previous estimate. The equations for the 0th and 2nd degree EMP filters, which are used later, can be found in the appendices.

The filter's covariance matrix, donated here as $S_{n+1,n}^*$ (and not to be confused

with the innovation covariance matrix S for the Kalman and RGN filters) depends on the covariance of the input errors as well as the memory length n and is thus time varying. The covariance matrix for the degree one EMP is presented below (the 0th and 2nd degree covariance matrices can be found in the appendices).

$$S^*_{n+1,n} = \sigma_v^2 \begin{pmatrix} \frac{2(n+3)}{(n+1)n} & \frac{6}{(n+1)n} \\ \frac{6}{(n+1)n} & \frac{12}{(n+2)(n+1)n} \end{pmatrix} \tag{2.39}$$

A useful property of the EMP, which is highlighted in [17], is its self-initialisation. Thus, one needs only the first measurement to start the filtering without having to set any other parameters. This applies to any degree of EMP filter, a property which will be exploited by the composite polynomial filter which is discussed later.

However, the expanding memory nature of the filter makes it less adaptable to large errors or changes in system processes. Thus, such anomalies can lead the filter to diverge. To overcome this, the fading memory polynomial filter is considered next which considers all measurements in its estimating, with a significance, however, that decays as time moves forward.

2.3.2 Fading memory polynomial

As the name suggests, the fading memory polynomial (FMP) filter makes use of weighted least squares estimation to fit a polynomial to equally spaced observations with a fading memory effect. The weight factor introduced has the effect of tapering old datum as it recedes into the past.

The FMP considers the Laguerre polynomials satisfying the orthonormal condition:

$$\phi_j(r) \equiv \left(\frac{1-\theta}{\theta^j}\right)^{1/2} \theta^j \sum_{\nu=0}^{j} (-1)^\nu \binom{j}{\nu} \binom{1-\theta}{\theta}^\nu \binom{r}{\nu} \tag{2.40}$$

to minimize the scalar:

$$e_n \equiv \sum_{r=0}^{\infty} \{y_{n-r} - [p^*(r)]_n\}^2 \theta^t \tag{2.41}$$

where $[p^*(r)]_n = \sum_{j=0}^{m} (\beta_j)_n \phi_j(r)$. The polynomial is in r and of degree m. θ^t is the exponential weight. Solving for the coefficients $(\beta_j)_n$ leads to the polynomial which minimizes Equation 2.41:

$$[p^*(r)]_n = \sum_{j=0}^{m} \left[\sum_{k=0}^{\infty} y_{n-k} \phi_j(k) \theta^k \right] \phi_j(r) \tag{2.42}$$

This polynomial now serves as our estimate of the process and is based on all the observations up to y_n. To diminish some of the computational load, a recursive formulation can be found. The recursive equations for the 1st degree FMP are shown below:

$$(z_1^*)_{n+1,n} = (z_1^*)_{n,n-1} + (1-\theta)^2 \epsilon_n \tag{2.43}$$

$$(z_0^*)_{n+1,n} = (z_0^*)_{n,n-1} + (z_1^*)_{n+1,n} + (1-\theta^2) \epsilon_n \tag{2.44}$$

where $\epsilon_n \equiv y_n - (z_0^*)_{n,n-1}$ is the error or innovation and $(z_k^*)_{n+1,n}$ is the updated estimate of the k-th derivative ($0 \leq k \leq m$) of the m-th degree polynomial based on the n-th observation as well as the previous estimate. The equations for the 0th and 2nd degree FMP filters, which are used later, can be found in the appendices.

Unlike the EMP in Section 2.3.1, the FMP's covariance matrix does not vary with time. It is a function of the fading parameter θ and can thus be precomputed. The covariance matrix for the first degree FMP is given as:

$$S_{n,n}^* = \sigma_v^2 \frac{1-\theta}{(1+\theta)^3} \begin{pmatrix} 1+4\theta+5\theta^2 & (1-\theta)(1+3\theta) \\ (1-\theta)(1+3\theta) & 2(1-\theta)^2 \end{pmatrix} \tag{2.45}$$

which suggests that the variances of the estimates can be made as small as desired by setting the fading parameter θ close enough to unity. This however does nothing to mitigate transient errors and so a compromise between smoothing and persistent transient errors must be made.

Unlike the EMP, the FMP filter is not self-initialising, however, the fact that the FMP can track changing trajectories makes it more favourable than the EMP, which would simply diverge. Thus, a compromise in the form of the composite polynomial filter is used.

The composite polynomial filter is initialised with the EMP filter. After a defined number of measurements, a transition to the FMP is made. This transition cannot take place at an arbitrary point in time. Instead, the transition occurs when the filter covariance matrices are approximately equal. This means that the time of transition is dependant on the fading parameter as well as the degree of the filter. Table 2.1 gives the number of samples after which the transition can occur for the 0th, 1st and 2nd degree [24].

Table 2.1: Transition sample number as a function of θ for different degree polynomials.

Degree	Ns
0	$2/(1-\theta)$
1	$3.2/(1-\theta)$
2	$4.36/(1-\theta)$
3	$5.51/(1-\theta)$

When transitioning to the FMP, its precomputed weights and covariance matrix are used. This acts as a further advantage as the computational load is reduced somewhat.

The polynomial filtering equations look very much like the alpha-beta-gamma filters. In fact, it can be shown that the alpha-beta-gamma filters are a generalised form of the polynomial filter [24]. Richards et al [18] discuss a form of the alpha-beta filter with gains dependent on the number of measurements which is very similar to the EMP discussed in Section 2.3.1.

Nonetheless, the polynomial filters provide some interesting advantages over the other filters discussed in this chapter and will be examined further in the following chapters.

2.4 Conclusions

This chapter focused on three tracking filters. The filters considered were: the Kalman filter for its widespread use and versatility; a linear form of the RGN filter which has only recently appeared in the literature and; the polynomial filter for its perceived suitability to the trajectories of the targets.

The three filters are derived using different methods. The Kalman and RGN filter, which take the probabilistic and statistical approaches respectively, end up as two very similar filters, differing only by the forgetting factor of the RGN filter. The polynomial filter uses the least squares method like the RGN filter, but by fitting a m degree polynomial instead.

The polynomial filters of greatest interest are the expanding and fading memory polynomial filters. The filters complement each other nicely with the FMP being adaptable to target manoeuvre changes and also offering better rejection of transient errors. The EMP on the other hand is self initialising. The strengths of each of these filters are combined into the composite polynomial filter which initialises using the EMP and then switches to the FMP for better tracking. We are interested in the polynomial filter as target trajectories in range-Doppler space are often parabolic in appearance and would thus seem well approximated by an appropriately fitted polynomial. This comes at a cost of higher computational load in comparison to the other filters.

The RGN filter offers an interesting modification to the Kalman filter by providing a forgetting factor. This forgetting factor shortens the effective memory length of the filter. In the presence of large errors, which can be expected in FM band commensal radar tracking, a shorter memory length might make the filter more adaptable to the large variations in measurements.

The Kalman filter will then act as a reference or 'gold standard'. Widely implemented and much studied, it was found to work well by Howland for range-Doppler tracking [7]. Thus the polynomial and RGN filters can be compared to the Kalman filter. The effects of their fading parameter and forgetting factor can be assessed.

These investigations are carried out in the following chapters, starting with simulations and then using real data.

Chapter 3

Building the simulations

This chapter details the simulation environment that was built to test the tracking filters. The target models are described, as well as the characteristics of the FMCR system. An effort was made to simulate the RRSG's commensal radar. Initially, the filters run in a clutter-free scenario and are then tested with clutter added. The gating techniques used and data association methods are also described. This chapter only goes as far as developing the simulations. In the next chapter, the performance of the filters is analysed and parameters are tweaked and finally, in the penultimate chapter, real data tests are performed.

3.1 Simulation environment

A very simple simulation environment is built in MATALB. The details of this environment are laid out in this section and are divided into two parts. Firstly the target models are descried and then the commensal radar system that observes this target. The false detections of the simulations are also described.

3.1.1 Target model

The target is simulated as a point in two dimensional space and is initialised with a state vector $X_0 = [x\ y\ \dot{x}\ \dot{y}]^T$. As the tracking takes place on two dimensional range-Doppler plots, the additional information gained from a three dimensional observation space was deemed to be minimal.

The target persists for 100 s and progresses along a trajectory governed by a transition matrix. The models used are the nearly constant velocity model:

$$X_k = \begin{pmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} X_{k-1} + v_{k-1} \tag{3.1}$$

where v_{k-1} is the white acceleration noise of the target with a standard deviation σ_a. Increasing the intensity of the noise (by increasing σ_a) results in the white-noise accelerations model, a simple manoeuvring target model [33]. A target with a known turn-rate target model is also simulated:

$$X_k = \begin{pmatrix} 1 & 0 & \frac{a}{\omega} & -\frac{(1-b)}{\omega} \\ 0 & 1 & \frac{(1-b)}{\omega} & \frac{a}{\omega} \\ 0 & 0 & b & -a \\ 0 & 0 & a & b \end{pmatrix} X_{k-1} + v_{k-1} \tag{3.2}$$

As the anticipated end use of most commensal radars is air-traffic control [8], the linear non-manoeuvring and known turn-rate (ω) target models are of greatest interest. Example trajectories of a linear and turning target are shown in Figure 3.1.

3.1.2 Commensal radar model

This target is then sensed by a commensal radar system. The commensal radar system in this scenario consists of a transmitter and a single receiver. The transmitter is modelled as an FM band, omni-directional transmitter with a center frequency of 98 MHz. The receiver is separated from the transmitter by a base line of 20 km. The system measures the bistatic range and bistatic Doppler of the target at 1 s intervals, referred to as a coherent processing interval (CPI). The noise on the Doppler measurements is normally distributed with a standard deviation of 0.1 Hz [12]. This commensal radar setup can can be seen in Figure 3.1.

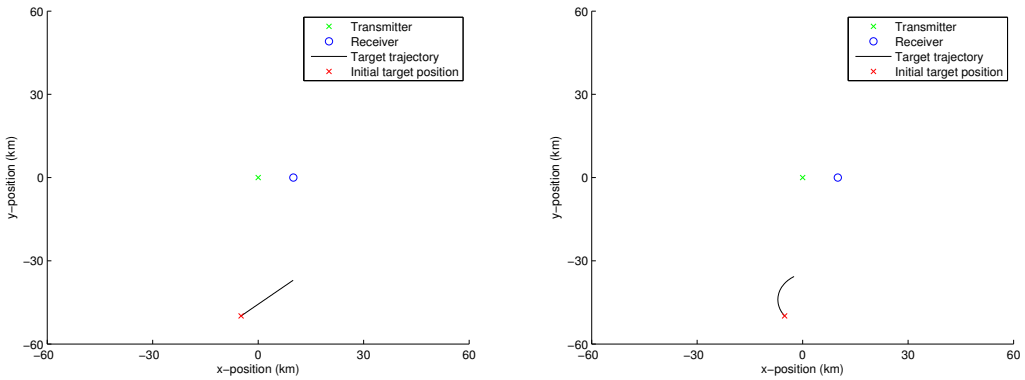

Figure 3.1: Trajectories for linear (left) and turning (right) targets in the observation space. Transmitter and receiver are also plotted.

The bandwidth of the FM radio broadcasts, which varies with the content being broadcast, is analysed by Griffiths et al [9, 34]. The best achievable bandwidth is in the order of 100 kHz while speech sees the effective bandwidth dropping to 9.1 kHz. Pauses will see the instantaneous bandwidth dropping close to 0 Hz. This results in a time varying range resolution which is related to the instantaneous bandwidth by $\delta R = c/2B$.

The simulated environment created here concerns the output of a CFAR detector after centroiding has been performed. Thus, unlike the range-Doppler tracking performed by Demming et al [35], our detections are well resolved and are marked

by ‘o’s in the range-Doppler plots.

The fluctuations in range resolution lead to range smear and need to be accommodated in the simulations. This is done by considering three scenarios.

Firstly, by centroiding the range-smeared Doppler cell, the true peak, and thus the true range, is located. In this event, the range smear really has no adverse effects, except in the event where multiple targets fall in the same range bin and become indistinguishable.

However, the presence of noise (at very poor SNRs) negate this theory. This is demonstrated in Figure 3.2 where two signals are generated with 200 Hz and 20 kHz of bandwidth respectively. The matched filtered response of each signal is the calculated and plotted. In the noiseless case, the peaks of both signals are discernible. The noisy case, however, sees the lower bandwidth signal’s peak being lost. This demonstrates how the targets true range might be falsely detected.

The third scenario concerns case where the output of a CFAR detector misses the detection altogether. This could be the case if a one dimensional CFAR operates along the range dimension where the gradient in the presence of range smear is very flat locally, leading to a missed detection.

The first of the aforementioned cases can be simulated by simply imposing small range errors on the measurements. That is, the target’s position is always known to a fair degree of accuracy. Examples of these measurements are shown in Figure 3.3.

The middle case is simulated with a time varying bandwidth. To do this properly requires in depth knowledge of the statistical properties of FM radio bandwidth. As a ‘make-do’ effort, and using insight from the work by Griffiths and Baker [34], a simulated FM radio station playing mostly reggae and rock music (chosen for their higher bandwidths) could be given a normally distributed bandwidth with a mean of 80 kHz (in between rock and reggae) and a standard deviation of 2 kHz. This large standard deviation is an attempt to cater for the large changes in bandwidth.

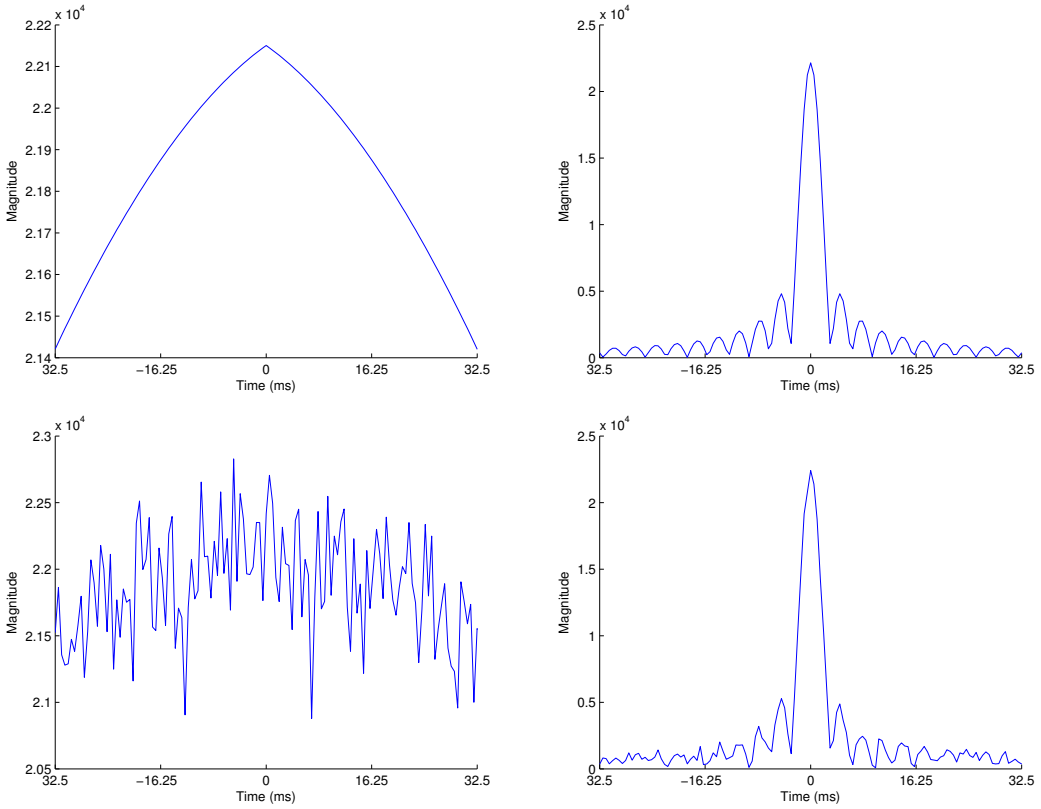

Figure 3.2: The matched filter response for signals with (clockwise from top left): 200 Hz bandwidth, no noise; 20 kHz bandwidth, no noise; 20 kHz bandwidth with noise; 200 Hz bandwidth with noise.

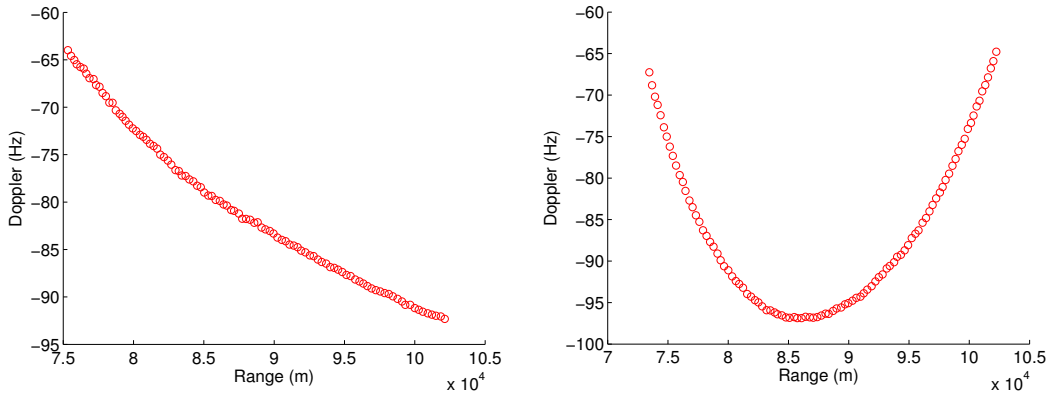

Figure 3.3: Commensal radar measurements for linear (left) and turning (right) targets where centroiding is effective in locating the targets peak.

The target can then be thought to be anywhere within the range resolution cell that is produced by this instantaneous bandwidth. This can be done by choosing a normally distributed point from within this range cell with a mean corresponding to the center of the cell. Examples of measurements of these types are shown in Figure 3.4.

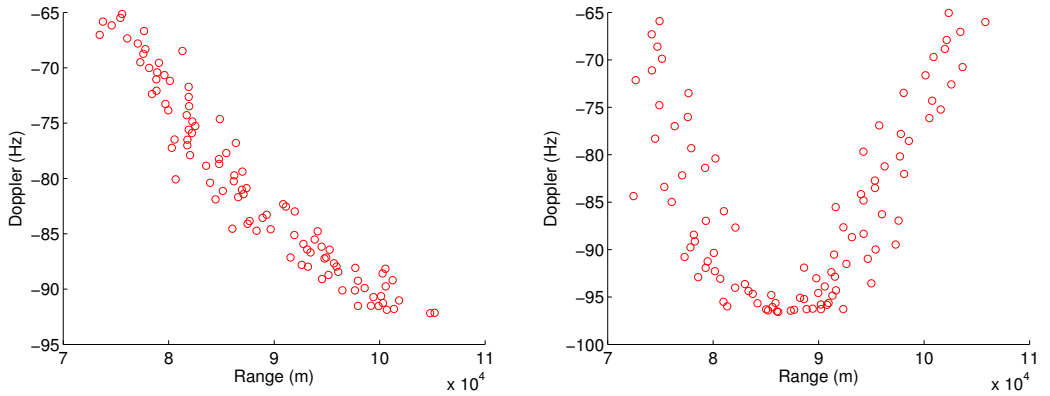

Figure 3.4: Commensal radar measurements for linear (left) and turning (right) targets with ineffective centroiding.

The last case can be simulated by decreasing the probability of detection. This will then result in 'gaps' in the measurements which will have to be dealt with the the filters' coast functions.

Having dealt with the parameters specific to the commensal radar, clutter needs to be considered. This is done (as in the literature [24, 15]) by injecting false detections into the environment according to a 'clutter density', β_{fa}. This represents the number of clutter points per volume per scan. This clutter density is related to the probability of false alarm p_{fa} by $p_{fa} = \beta_{fa}/V$, where V is the search volume - although in our two dimensional case, V refers to the area.

The number of false detections (n_{fa}) at each scan is Poisson distributed:

$$P_{fa}(n_{fa}) = \frac{(\beta_{fa}V)^{n_{fa}} exp(-\beta_{fa}V)}{n_{fa}!} \tag{3.3}$$

These false detections are then injected uniformly into the ARD plot, spanning -150 to 150 Hz on the Doppler axis and 0 to 200 km on the range axis. This deviates somewhat from the true nature of false detections. Stronger clutter returns are experienced at zero Doppler and smaller ranges from all the stationary clutter. Suppression techniques such as ECA or conjugate gradient cancellation remove this stationary clutter, in some cases leaving the zero Doppler region completely free of clutter. Also, there are likely to be more clutter returns at shorter bistatic ranges, as the returns are stronger. While the simulated clutter is more uniform than real clutter might be, the filters can still be tested by these gating and track association challenges.

In the real world, a target's bistatic RCS fluctuates with time [36]. These fluctuations can be severe, sometimes to the point where the target disappears altogether. This leads to missed detections. These missed detections can be accommodated for in the simulations by setting a probability of detection P_d variable. For each sample, a uniformly distributed random variable is compared to $(1-P_d)$ to determine whether or not the target is detected.

3.2 The tracking filters

The observations of the simulated target made by the simulated commensal radar can now be tracked and so the three filters described in Chapter 2 are implemented in MATLAB.

The Kalman filter uses the stabilised form of the covariance matrix that was presented in Chapter 2 while the RGN filter uses the more straightforward approach. Setting the forgetting factor of the RGN filter to unity results in the Kalman filter. Thus when $\lambda = 1$, we are effectively comparing the two methods of calculating the covariance matrix. Both the Kalman and RGN filters make use of only the 0th and 1st derivatives.

The polynomial filter is implemented as the composite polynomial filter of second degree. This means that a second derivative of the range and Doppler is calculated and used for tracking. However, a method is required to avoid quick settling. This is a phenomenon where the filter forms 'confident' estimates immediately after initialisation causing large errors in the initial stages. Quick settling is mitigated by initialising the EMP as a 0th degree polynomial, then switching to 1st degree and finally to second degree. These transitions are fixed at $k = 5$ and $k = 8$ for the 1st degree and 2nd degree transitions respectively. The transition to the 2nd degree FMP then happens at the appropriate time (according to θ). Figure 3.5 shows the three filters tracking linear and turning targets in a clutter free scenario.

The next step is to add clutter, the addition of which requires that the tracking filters be enhanced somewhat. Gating functionality must be added so that data association can be performed as well as coast functionality to allow the filters to cope with missed detections.

The Kalman and RGN filters make use of ellipsoidal gates. The innovation $\tilde{y}_n$, which is the difference between the filtered state estimate and the measurement, is computed and used in the equation:

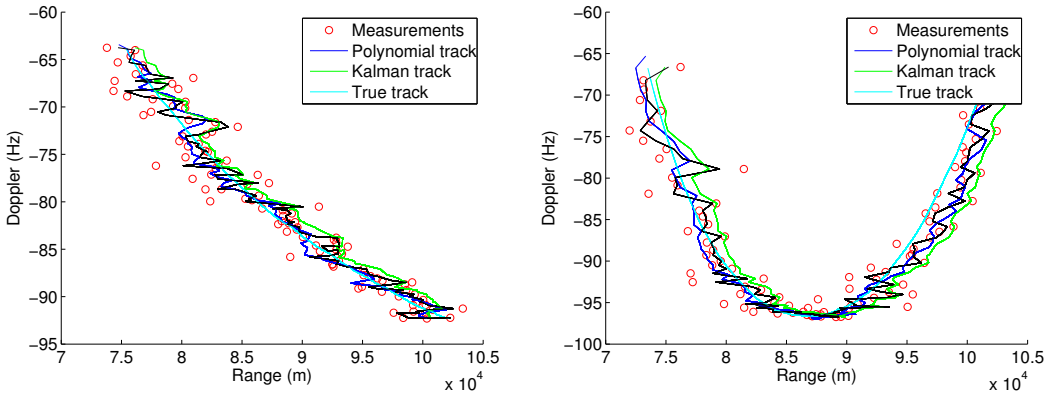

Figure 3.5: The polynomial, Kalaman and recursive Gauss-Newton filters tracking a linear (left) and turning (right) target in range-Doppler space with no clutter.

$$\tilde{y}_n S_{n|n}^{-1} \tilde{y}_n^T \leq G \tag{3.4}$$

where $S_{n|n}$ is the innovation covariance matrix and G is the size of the gate. Thus the innovation is weighted by the inverse of its covariance and compared to a predefined gate size.

The polynomial filter makes use of a similar gating technique. For the 0th and 1st degree polynomials, however, a rectangular gate [15] is used. Once the transition to the 2nd degree polynomial is made, the same elliptical gate in Equation 3.4 is used.

Data association on the measurements falling within the gate is performed using the nearest neighbour technique [14]. The measurement with the shortest statistical distance given by:

$$y_{NN} = \min_i (y_i^2 - \hat{x}_k^2)^{\frac{1}{2}} \tag{3.5}$$

is assigned to the track. $i = 1, 2, .., N_g$ are the measurements which fall within the track's gate for a given CPI k. While methods such as multiple hypothesis tracking [37, 24] and joint probabilistic data association [38] are known to perform

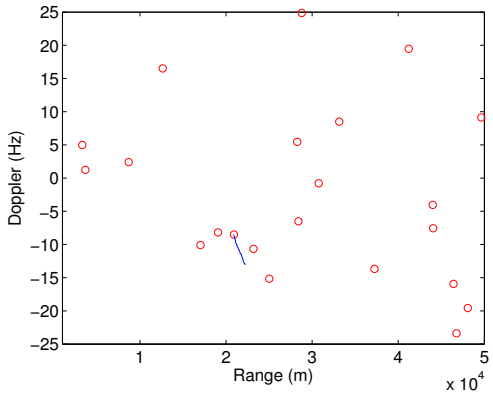

Figure 3.6: The Kalman filter tracking a target's bistatic range and bistatic Doppler over time (blue line) in the presence of false detections (red 'o's).

better, the nearest neighbour technique is favoured here because of its simplicity.

In addition to data association, decisions need to be made as to whether or not a track should be confirmed. This is done using the M-out-of-N track confirmation logic. Blackman discusses various other methods in his book [14], however the M-out-of-N method is chosen for its simplicity. This track confirmation logic, described succinctly by its name, requires M updates out of N scans (or CPIs in this case) for a track to be confirmed.

In the event of a missed detection, the filters are coasted using the relevant transition matrix. Figure 3.6 shows the Kalman filter tracking a target's trajectory in the presence of clutter.

3.3 Conclusions

The simulation environment used to test the tracking filters was described in this chapter. Two target motion models which are most relevant to passive tracking of commercial airliners are considered.

These targets are observed by a commensal radar. False detections as well as missed detections are included in the commensal radar simulations and can be

adjusted by the probability of detection P_d and clutter density β_{fa} variables.

The tracking filters described in Chapter 2 can then be used to track the targets in the range-Doppler space. However, gating and data association techniques as well as coasting functionality first needed to be incorporated into the filters. This was to enable the filters to track the targets' trajectories despite the clutter and missed detections.

The simulation environment built in this chapter demonstrates that the filters are able to track the simulated targets. The next step is to assess the performance of the filters and is the focus of the next chapter. Chapter 5 then tests the filters on real data.

Chapter 4

Evaluating the filters

The aim of this chapter is to evaluate the performance of the tracking filters. Performance in this context is multi-faceted and includes metrics such as computational load, tracking errors and data association statistics.

Chapter 2 highlighted various strengths and weaknesses of the three filters under consideration. This chapter gives us the opportunity to investigate these attributes. As stated in Chapter 2, the Kalman filter acts as the 'gold standard'.

The RGN filter, with its forgetting factor, is expected to cope with the large errors in the range measurements at the expense of the smoothness of its track. Various values of λ will be investigated. The polynomial filter is expected to produce smooth plots, perhaps at the expense of greater tracking errors and possible divergence. Its computational load is expected to be greater. Also pointed out in Chapter 2 was that the polynomial filter's data association statistics might be better.

Thus an investigation into these performance metrics is necessary to either confirm, or contradict, the assertions made in the literature. We begin with computational load.

4.1 Computational load

A crucial aspect when evaluating the performance of the filters is computational load. While tracking a single target is a fairly trivial procedure, initiating tracks for each detection (both true and false) increases the computational load considerably. Thus, increasing the clutter density will result in an increase in the computational load of the filters. Even a small improvement in computational efficiency of an individual filter will go a long way in improving the computational load of the tracking system.

The computational load of each filter is evaluated firstly by counting the number of arithmetic multiplications and additions. Further insight is then gained by timing the initialisation and update functions of the filters.

4.1.1 Counting operations

The computational load of the filters is compared by evaluating the number of operations required for the initialisation of each filter as well as an update. Tracks are initialised for each detection. Depending on whether new measurements fall within their respective gates, these tracks are either updated or coasted to the next CPI. Tracks that coast three times consecutively or fail to meet the M-out-of-N logic are deleted. This means that computational efficiency is just as important in the track initialisation phase as it is in the track update phase.

As a result, the number of additions, multiplications and matrix inversions carried out by each of the filters in these phases is considered. As an example, the pseudo-code for the Kalman filter update as well as the polynomial filter update is presented below.

```
y    = pos_obs;
K       = P_pred*H'*inv(S);
x_hat   = x_pred + K*(y-H*x_pred);
P       = (eye(4) - K*H)*P_pred*(eye(4) - K*H)' + K*R*K';
```

```
x_pred = PHI_t*x_hat;
P_pred = PHI_p*P*PHI_p'+ Q;
S      = H*P_pred*H' + R;
S_inv  = inv(S);
```

Noting that `x_hat` is of dimension 4x1 while `P_pred`, `PHI_t` and `PHI_p` are 4x4 matrices, the Kalman filter update code above appears to be quite expensive in terms of computational load when compared to the polynomial filter code below. Here, `ek`, `z2`, etc. are 2x1 vectors. Considering that multiplying a mxn matrix by a nxp matrix requires mnp multiplications and additions, this makes the polynomial filter in this instance seem highly attractive.

```
ek = yk - z0;
z2 = z2 + alpha_f*ek;
z1 = z1 + 2*z2 + beta_f*ek;
z0 = z0 + z1 - z2 + gamma_f*ek;
P = P_f2;
S = H*P*H' + R;
Sinv = inv(S);
```

Further, this particular polynomial track update code is for the 2nd degree FMP filter, where the weights α, β and γ, as well as the filter covariance matrix are all fixed (depending only on θ). They can thus be precomputed and stored in memory, further reducing the computational load over the EMP whose weights are time dependant.

The tallies for the initialisation of each filter is presented in Table 4.1 while the filter update tallies are in Table 4.2.

In looking at the tables, the polynomial filter comes across as the clear winner in terms of computational load. The recursive Gauss-Newton filter is more computationally efficient in the update procedure. This is because of the way in which the covariance matrix is computed. The Kalman filter uses a more stable

Table 4.1: Number of arithmetic operations per filter initialisation.

Filter	Multiplications	Additions	Inverses
Kalman	240	260	1
Polynomial	30	21	1
Recursive Gauss-Newton	257	260	1

Table 4.2: Number of arithmetic operations per filter update.

Filter	Multiplications	Additions	Inverses
Kalman	512	618	1
0th degree EMP	26	33	1
1st degree EMP	16	21	1
2nd degree EMP	39	31	1
2nd degree FMP	6	12	1
Recursive Gauss-Newton	434	442	1

method of computing the covariance matrix, while the recursive Gauss-Newton filter uses the simpler method. This is not a requirement, either method can be used for either filter.

The results in the tables, however, only consider multiplications, additions and inverses. The polynomial filter implemented as a composite polynomial filter requires a series of decision logic. Depending on the number of samples in the track, a different degree of the polynomial filter is implemented. This is done to improve initialisation and filter performance. The effects of these decisions is evaluated in Section 4.1.2.

4.1.2 Timing results

The above section estimates computational load by counting the number of additions, multiplications and inverses performed by each filter for track initialisation and update. However, further insight is gained by timing the initialisation and updates of tracks.

This is done by using the 'tic-toc' commands. The initialise and update function

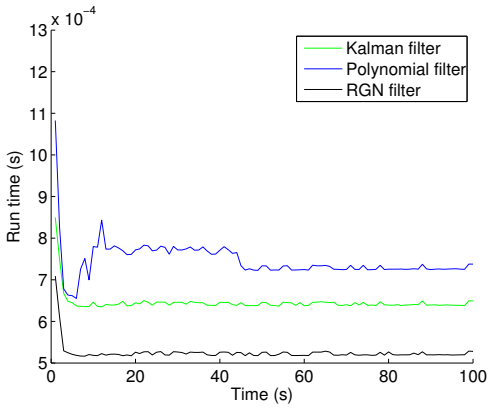

Figure 4.1: Processing time of each filter.

calls are timed from the main source code as a target is tracked for 100 s. The results are shown in Figure 4.1 where the update time for a given scan is plotted against the corresponding scan time. The results shown are the average values for 1000 runs.

As anticipated from the results in Section 4.1.1, the recursive Gauss-Newton filter is faster than the Kalman filter as a result of the different methods for calculating the covariance matrix. Interestingly, the difference in initialisation times between the Kalman and recursive Gauss-Newton filters (which should be exactly the same) corresponds to the number of properties defined in the filter class. Defining more properties has a significant impact on the initialisation time of the filter.

The polynomial filter is the slowest. This is in contrast to what was expected from the results in Section 4.1.1, but confirms the assertions made by the literature. What was ignored in the above section, however, was the overhead of the composite polynomial filter. There are four stages in each update, the expanding memory polynomial of degree zero, one and two, as well as the fading memory polynomial of degree two. This is done as the expanding memory filter is self initialising. The stages of degree (i.e. from zero to one to two) prevents quick settling where the filter incorrectly estimates the parameters initially, resulting in large divergences in the first few scans. The switch to the fading memory polynomial filter is made as this filter is adaptive making it better suited to tracking

manoeuvring targets.

As a result, for each update, the filter has to determine which of the four variants of the polynomial filter to apply by evaluating k. This is implemented with a chain of `if-elseif` statements. This process is clearly quite cumbersome and so ultimately the polynomial filter is not as computationally efficient as was hoped.

Changes from the one filter to the next, which occur at $k = 5$, $k = 8$ and $k = 43$ (for a fading parameter $\theta = 0.9$) are evident in the polynomial's timing results. Despite the decision overhead, the improvement in computational efficiency in moving from expanding memory to fading memory is evident.

The reader should note, however, that while a small effort was made to optimise the initialisation and update methods for the three filters, this was by no means a comprehensive effort. Thus, there is still room for improvement.

4.2 Tracking performance

This section discusses the performance of the filters from the target tracking point of view. Both the range and Doppler errors are evaluated for each of the filters as well as some data association statistics.

Range and Doppler errors give an indication as to which of the filters track the target's true trajectory through range-Doppler space most effectively. Plotting these errors over a number of scans will give an indication as to divergence and maximum error. A low standard deviation of errors might indicate that the track is quite smooth. The opposite does not necessarily apply, however. In these simulations, the Kalman filter is considered to be the 'gold standard' while the RGN and polynomial filters are compared to it. The filters' forgetting factor and fading parameter are changed in order to investigate their effect on this aspect of tracking performance.

All the filters use the same nearest neighbour data association technique. However, in the case of the recursive Gauss-Newton and Kalman filters, the slightly

different covariance matrices (because of the damping factor), might lead to slightly different data association techniques. More interesting though, is the polynomial filter with its cumbersome initialisation. Thus data association statistics, particularly true track confirmation, provides useful insight for filter selection.

4.2.1 Filter and radar constants

This section briefly describes the values that are held constant for the remaining sections on tracking performance. The important simulation parameters are summarized in Table 4.3.

Table 4.3: Simulation parameters

Parameter	Value
Transmit center frequency	89 MHz
Mean bandwidth	80 kHz
Standard deviation of bandwidth	2 kHz
Bistatic baseline	20 km
Target acceleration noise	0.1 m/s

The key parameters and initial values for the filters include the process noise covariance matrix, based on Howland's work [7]:

$$Q = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.02 & 0 \\ 0 & 0 & 0 & 0.05 \end{pmatrix} \tag{4.1}$$

The measurement noise covariance matrix:

$$R = \begin{pmatrix} \sigma_R^2 & 0 \\ 0 & \sigma_d^2 \end{pmatrix} \tag{4.2}$$

where $\sigma_R = 6.7$ m and $\sigma_d = 0.1$ Hz. These values were chosen by a tuning process

based on the measurement errors imposed by the changing range resolution.

The polynomial filter's covariance matrix is calculated from the length of the polynomial k for the case of the expanding memory polynomial filter or from the fading parameter θ for the case of the fading memory polynomial filter. The Kalman and recursive Gauss-Newton filters on the other hand require their covariance matrices to be initialised. The Kalman filter's covariance matrix initialisation follows Howland [7]:

$$P_{0|0} = \begin{pmatrix} 5 & 0 & 0 & 0 \\ 0 & 0.04 & 0 & 0 \\ 0 & 0 & 0.0225 & 0 \\ 0 & 0 & 0 & 0.1 \end{pmatrix} \tag{4.3}$$

The recursive Gauss-Newton filter's covariance matrix is initialised using values that were arrived at after a tuning process:

$$W_{0|0} = \begin{pmatrix} \sigma_R^2 & 0 & 0 & 0 \\ 0 & \sigma_d^2 & 0 & 0 \\ 0 & 0 & 1600 & 0 \\ 0 & 0 & 0 & 1600 \end{pmatrix} \tag{4.4}$$

In setting the RGN filter's forgetting factor to one, one arrives at the Kalman filter. Thus, using different covariance matrix initialisations allows us to investigate their effect on filter performance. This will be evaluated in the sections to come.

4.2.2 Range and Doppler errors

The range and Doppler RMS errors for the three different filters are examined here for a 100 s target track. The RMS range and Doppler errors obtained from 1000 Monte Carlo runs are plotted against time to give an indication as to whether or not the filters converge and how quickly this convergence happens.

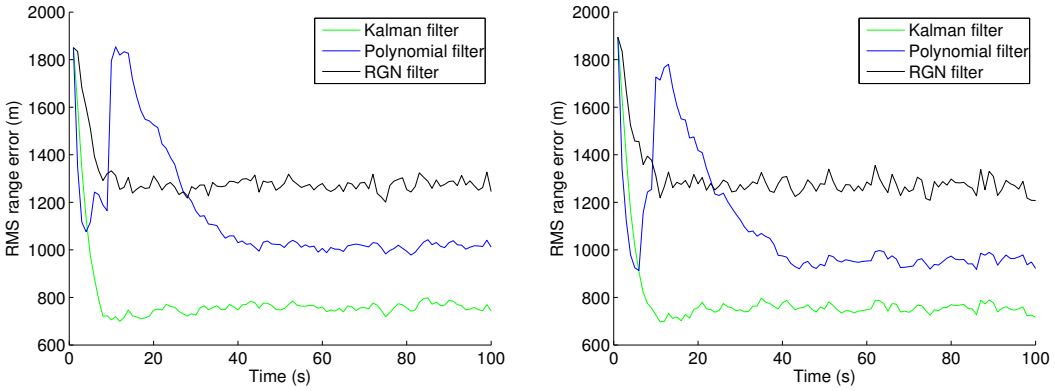

Figure 4.2: RMS range errors for each filter plotted against time for 1000 Monte Carlo runs. Left: linear target. Right: Turning target.

We start by looking at the range errors.

Figure 4.2 shows the range errors for the three filters tracking a linear target as well as a turning target. The RGN filter uses the default forgetting factor of $\lambda = 0.8$ while the polynomial filter's fading parameter is set to its default of $\theta = 0.9$. The Kalman filter, for both the linear and turning target, settles to an range error of just under 800 m after about 10 s. The RGN filter (again for both target models) takes about the same time to settle at a range error of about 1300 m. The polynomial filter experiences an increase in range error after transitioning from a first degree to second degree expanding memory polynomial filter. A steady state range error of about 1000 m (about 950 m for the turning target) is reached after about 40 s, just before the transition to the fading memory polynomial filter.

The results in Figure 4.3 investigate the effect that the forgetting factor and fading parameters have on the RGN and polynomial filters respectively. The forgetting factors used are 0.6, 0.7, 0.9 and 1. The fading parameter values considered are 0.7, 0.8, 0.95 and 0.99. Results are obtained for the linear target and are compared to the results obtained for the default values of $\lambda = 0.8$ and $\theta = 0.9$ in Figure 4.2. The polynomial filter is discussed first.

Low values of θ result in a shorter memory. This leads to the filter effectively

favouring new measurements and disregarding the measurement history. The resulting track follows the noisy measurements closely and as a result the RMS range errors for the polynomial filter in the top left plot of Figure 4.3 are larger than in Figure 4.2. Another consequence of the shorter memory length is that the filter switches to the fading memory polynomial sooner. The switch to fading memory happens at $k = 4.36/(1 - \theta)$. This is evident from the top left plot in Figure 4.3 where the filter settles to a RMS error of about 2000 m just after 10 s and then switches to the FMP after 15 s, meaning that the second degree EMP phase of the filter is effectively bypassed.

The remaining plots in Figure 4.2 show that increasing the fading parameter improves the RMS range error for the polynomial filter at the expense of settling time. Larger values of θ mean that the switch to the fading memory polynomial happens later. Also, it would seem that the switch from first degree to second degree expanding memory polynomial (at 9 s) results in a large increase in RMS range error. This error is reduced slowly until the the fading memory polynomial takes over and the RMS error stops dropping and stabilises.

The RGN filter responds similarly to increases in its forgetting factor λ. Like the polynomial filter's fading parameter, the forgetting factor adjusts the filter's memory length. Increasing λ from 0.6 results in the RMS range error decreasing from about 1700 m in the top left plot of Figure 4.3 down to about 800 m in the bottom right plot when $\lambda = 1$. Here the RGN filter achieves the same range error as the Kalman filter, though taking slightly longer to settle. This discrepancy in settling time comes from the different methods of initialising and updating the filter covariance matrices.

We now backtrack and turn our attention to the RMS Doppler errors shown in Figure 4.4. Unlike the range errors, there is a clear difference between the linear and turning targets for the polynomial filter. The Doppler error settles at a value almost ten times greater for the turning target. The Kalman and RGN filters are both very close and are not affected by the different target model.

Figure 4.5 plots the RMS Doppler errors for different values of λ and θ. These results show that, for the case of the polynomial filter, Doppler is tracked better

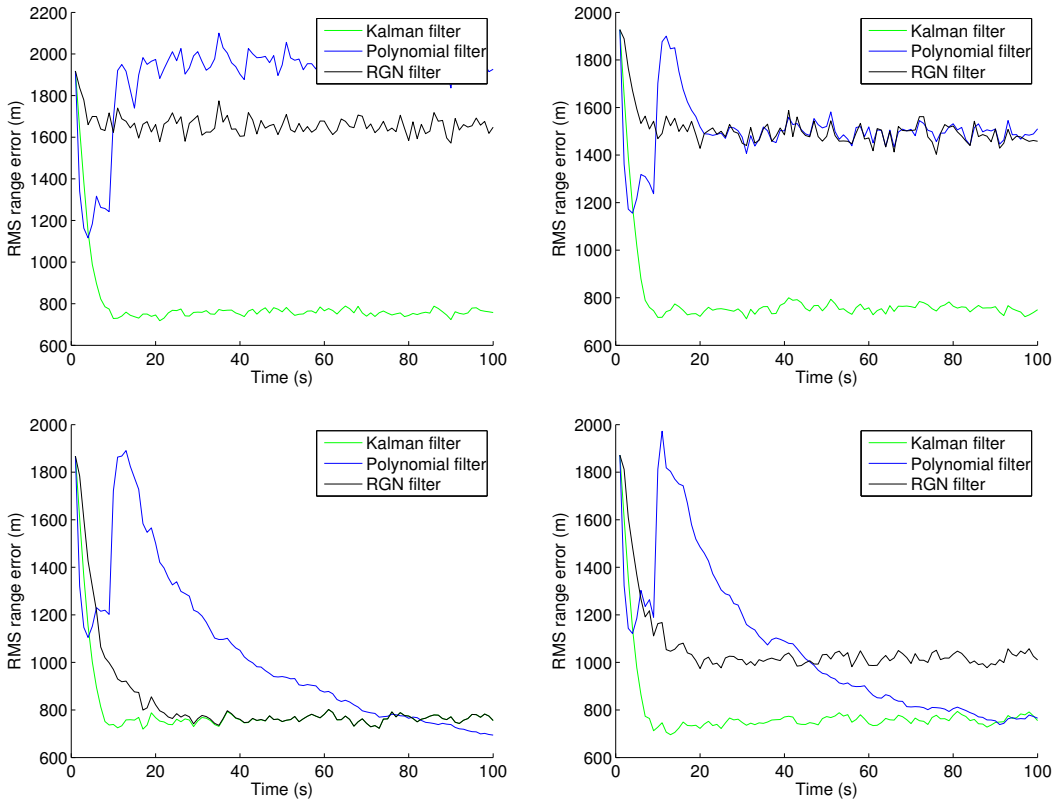

Figure 4.3: RMS range errors for each filter plotted against time for 1000 Monte Carlo runs. Clockwise from top left: $\lambda = 0.6$ and $\theta = 0.7$, $\lambda = 0.7$ and $\theta = 0.8$, $\lambda = 0.9$ and $\theta = 0.95$, $\lambda = 1$ and $\theta = 0.99$.

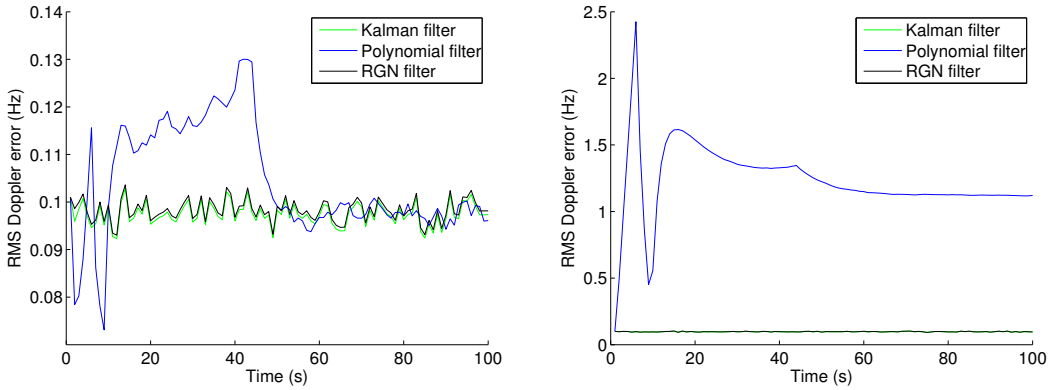

Figure 4.4: RMS Doppler errors for each filter plotted against time for 1000 Monte Carlo runs. Left: linear target. Right: Turning target.

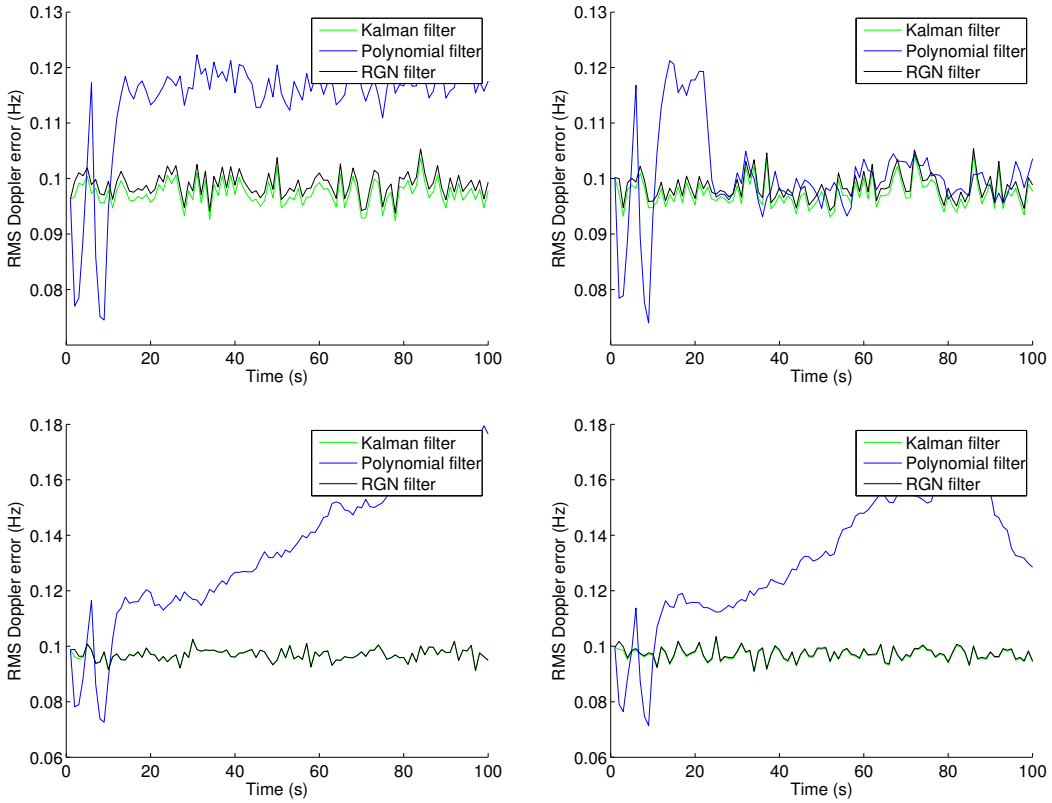

Figure 4.5: RMS Doppler errors for each filter plotted against time for 1000 Monte Carlo runs. Clockwise from top left: $\lambda = 0.6$ and $\theta = 0.7$, $\lambda = 0.7$ and $\theta = 0.8$, $\lambda = 0.9$ and $\theta = 0.95$, $\lambda = 1$ and $\theta = 0.99$.

with lower values of θ. The errors fluctuate during the initialisation phase up until the second order EMP is reached. From here, the error grows until the FMP takes over. Low values of θ do not result in the same Doppler error as the Kalman and RGN filters being reached while high values wait too long for the transition to FMP and as a result the error grows.

The RGN filter's RMS Doppler errors follow those of the Kalman filter very closely, being only slightly worse for lower values of the forgetting factor. The errors get closer as the value of λ increases and are the same for $\lambda = 1$.

Tables 4.4 and 4.5 show the aggregate variance of the range errors as well as aggregate maximum range error for the 1000 Monte Carlo runs. The Kalman

filter achieves the lowest variance and lowest maximum range error. The variance and maximum range errors decrease with increasing λ and θ for the RGN and polynomial filters respectively, although the RGN filter achieves values closer to that of the Kalman filter for high λ.

Similar results for the Doppler measurements appear in the appendices and show that the Kalman and RGN filters achieve similar variances and maximum Doppler errors, with higher values of λ resulting in slightly better performance for the RGN filter than for the Kalman filter. The polynomial filter, on the other hand, has higher variances and larger maximum Doppler errors.

Table 4.4: Variance of the range errors for the three different filters with different values of λ and θ.

Kalman	λ	RGN	θ	Polynomial
	0.6	1.004e+06	0.7	1.335e+06
	0.7	8.289e+05	0.8	8.199e+05
2.471e+05	0.8	6.140e+05	0.9	5.266e+05
	0.9	4.265e+05	0.95	4.700e+05
	1	2.867+05	0.99	4.686e+05

Table 4.5: Aggregate of the maximum range errors in a given scan for the three different filters with different values of λ and θ.

Kalman	λ	RGN	θ	Polynomial
	0.6	4.574e+03	0.7	5.256e+03
	0.7	4.179e+03	0.8	4.093e+03
2.277e+03	0.8	3.596e+03	0.9	3.381e+03
	0.9	3.094e+03	0.95	3.317e+03
	1	2.758e+03	0.99	3.316e+03

4.2.3 Data association

Data association refers to the logic required to determine which measurement belongs to which track. Probabilistic data association (PDA) and joint probabilistic data association (JPDA), are some common methods implemented in

conventional tracking problems [15]. Nadjiasngar et al [24] have even developed PDA methods for the polynomial filter, however, nearest neighbour data association is used for all three filters in this project and is discussed in Chapter 3.

Although the same techniques are used, the different covariance matrices of the filters will lead to different data association performances. The Kalman and RGN filters have slightly different covariance matrices which are used in the computation of the gating function and will thus have an effect on the whether or not the target measurements fall within the gate. Thus true track confirmation, false track confirmation and true track deletion are all metrics of interest. The polynomial filter, as a completely different filter, has multiple stages of initialisation and a different gating technique and so is also expected to perform differently from the other two filters.

Three metrics, true track confirmation, false track confirmation and true track deletion are considered. The three filters are compared against each other for different values for the fading parameter and forgetting factor. Table 4.6 summarises the important simulation parameters that are used.

Table 4.6: Data association simulation parameters.

Parameter	Value
Probability of detection (P_d)	0.7
Number of clutter points per area (β_{fa})	8.3e-6
Gate size (Kalman and RGN)	100
Gate size polynomial (0th and 1st degree)	5 Hz, 5 km
Gate size polynomial (2nd degree)	100
M-out-of-N logic	3-out-of-4

The large gate sizes used could result in many false tracks being confirmed. However, they were selected after a tuning process to accommodate the large variations in the range measurements.

Specific simulation environments were built for each of the metrics. True track confirmation was assessed by generating a target for four CPIs in an environment free of clutter. This simulation was run 1000 times, counting the number of times the track was confirmed by the N-out-of-M track confirmation logic. Dividing

this number by 1000 gives a probability of true track confirmation (P_{ttc}).

The propensity of each filter to falsely confirm tracks was assessed by counting the number of number of false tracks confirmed in 1000 opportunities. This was done by running the filters for four CPIs (enough to allow track confirmation) in a target-free clutter environment and then counting the number of false tracks confirmed. This was repeated 1000 times, the tallies for each run being adding to the false track confirmation counter.

True track deletion was evaluated by counting the number of times the target track (which persists for 100 s) is prematurely deleted in 1000 runs. While there is no clutter, missed detections do occur. Dividing this tally by 1000 gives a probability of true track deletion (P_{ttd}).

Tables 4.7, 4.8 and 4.9 show the results. The polynomial filter was not affected by changing the fading parameter. This is expected for the track confirmation tests, as the earliest the transition to the FMP occurs is after $k = 15$, but not for track deletion.

The Kalman filter has a very low probability of true track confirmation. Comparing this value to the RGN when $\lambda = 1$ shows that it is most likely as a result of filter covariance initialisation. Decreasing λ results in increasing P_{ttc}. The polynomial achieves the highest P_{ttc}.

Table 4.7: Probability of true track confirmation for the three different filters with different values of λ. The probability of detection is 0.7.

Kalman	λ	RGN	Polynomial
	0.6	0.652	
	0.7	0.641	
0.201	0.8	0.616	0.671
	0.9	0.586	
	1	0.565	

The number of false tracks confirmed, shown in Table 4.8, is low for both the polynomial and Kalman filers. The RGN filter, however, confirms a large number of false tracks. The number decreases with increasing λ but remains extraordi-

narily high even when $\lambda = 1$. Again, this difference is as a result of the filter covariance matrix.

Table 4.8: The number of false tracks confirmed for the three different filters with different values of λ.

Kalman	λ	RGN	Polynomial
	0.6	1237	
	0.7	836	
23	0.8	590	11
	0.9	450	
	1	351	

The probability of true track deletion evaluates how easily a track is deleted as a result of missed detections and measurements falling outside the gate. As a result, increasing P_d results in a decrease in P_{ttd}. The results in Table 4.9 show the Kalman filter as having a very high P_{ttd}. This result is closely related to the P_{ttc} in Table 4.7 as they both depend on measurements falling within the gate. The polynomial filter has the lowest P_{ttd} while increasing the RGN filter's forgetting factor increases the P_{ttd} from a starting value only slightly above the polynomial.

Table 4.9: Probability of true track deletion with $P_d = 0.7$ for the three different filters with different values of λ.

Kalman	λ	RGN	Polynomial
	0.6	0.462	
	0.7	0.467	
0.891	0.8	0.501	0.435
	0.9	0.501	
	1	0.532	

4.3 Conclusions

In this chapter, the tracking filters under discussion are tested against various performance metrics. The metrics considered are computational load, RMS

tracking errors, variance of the tracking errors, maximum tracking errors as well as true track and false track confirmation and true track deletion.

The metrics considered give an all round indication of each filter's suitability to range-Doppler tracking. As was expected from the literature in Chapter 2, the polynomial filter is computationally more burdensome than the Kalman and RGN filters. This is in spite of the relatively simple update equations and is caused by the overhead associated with the composite implementation. Reducing the degree of the polynomial filter would improve the computational load and is worth investigating.

In addition to this, the initialisation of the filters can be improved by carefully considering the variables that are defined. It was shown that this had a significant effect on the execution time of the filter initialisations. Further efforts in this regard would improve the computational load of all the filters in question.

Moving on to the tracking errors, the RMS range errors showed no response to a different target transition model. The errors obtained by all the filters were very similar for both the linear and turning target models. The same can not be said for the RMS Doppler errors. The polynomial filter's performance was worse for the turning target.

In general, the Kalman filter tracked range and Doppler better than the other filters. The polynomial filter requires a compromise between good range tracking and good Doppler tracking. That compromise most likely lies between $\theta = 0.8$ and $\theta = 0.9$. The RGN filter improves in both range and Doppler tracking with increasing values of λ. The fact that in some cases it does not meet the performance of the Kalman filter points to the importance that the filter and measurement covariance matrices play. Thus careful consideration needs to given in the selection of these parameters.

The polynomial filter performed the best in the data association experiments. It achieved the best probability of true track confirmation, the lowest probability of false track confirmation and confirmed the fewest false tracks. The gate sizes for the RGN and Kalman filter were very large yet the Kalman filters true track

confirmation probability and number of false tracks confirmed was low. This suggests poor covariance matrix selection. The RGN filter, with different filter and measurement covariance matrices (by choice and not theory) performs far better. Thus, it becomes clear that there is also a trade-off between tracking accuracy and data association performance for the Kalman/RGN filters, determined, in part by the selection of their covariance matrices.

While these conclusions are interesting and demonstrate a trade off between tracking accuracy and data association performance, it remains to be seen how the filters perform on real commensal radar data. These experiments are performed in the next chapter.

Chapter 5

Application to real data

Chapter 4 presents interesting results regarding the filters' performance on simulated data. However the validity of these results is contingent on the accuracy and life-likeness of the simulation environment. Thus, before any real conclusions can be drawn on the filters, they need to be applied to real data.

This chapter makes use of recorded FM band commensal radar data. The three tracking filters are applied to the data to give a more accurate indication as to how well the filters can track targets in range and Doppler. Simulations are, at best, very simplistic assumptions of the real world and can easily overlook aspects of the real world that have significant effects.

The challenge involved in real data tracking is estimating the parameters such as measurement noise covariance matrices as well as initial values of filter covariance matrices. Appropriate gate values must also be selected.

These aspects, along with the effects of the fading parameter and forgetting factor, will be investigated in this chapter so that the targets can be successfully tracked.

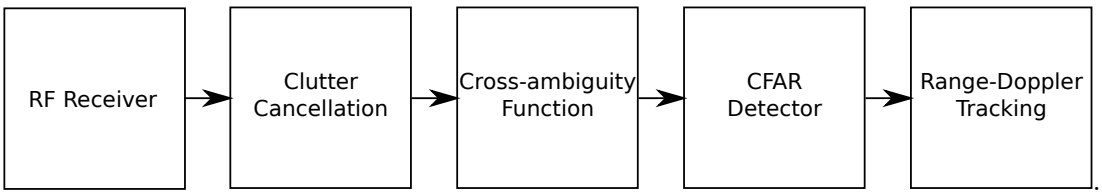

Figure 5.1: The processing chain of the commensal radar.

5.1 Integration

Running the filters on real commensal radar data requires some level of integration into the RRSG's 'commensal radar'. Figure 5.1 shows the steps performed by the commensal radar's data processor. The clutter cancellation, using the conjugate gradient method, removes most of the clutter. After this, the cross-ambiguity function is computed by cross-correlating the surveillance channel with multiple Doppler-shifted copies of the reference channel. These cross-ambiguity functions are then displayed by *ARDView* as ARD plots. ARDView is software which forms part of a PhD thesis [39] and is an integral component of the commensal radar [10, 8]. In addition to plotting the cross-ambiguity functions, ARDView implements a CFAR detector to detect targets (Figure 5.2). Once the targets have been detected, range-Doppler tracking can be performed.

While not in permanent operation, regular measurement campaigns involving portable receivers mean that there is an abundance of archived commensal radar data which can be analysed in ARDView. A variety of CFAR detection algorithms are catered for and can be performed along either the range or Doppler dimension. For these experiments, a greatest-of CFAR detector along the Doppler dimension was found to perform well and is used throughout the experiments. The CFAR detection data is then streamed to MATLAB for tracking.

The MATLAB code plots the detections on a bistatic range versus bistatic velocity scale. Bistatic velocity and bistatic Doppler frequency are related by the wavelength. Bistatic velocity is favoured here over bistatic Doppler frequency because it is more meaningful.

Due to the fact that centroiding functionality has not yet been built into ARD-

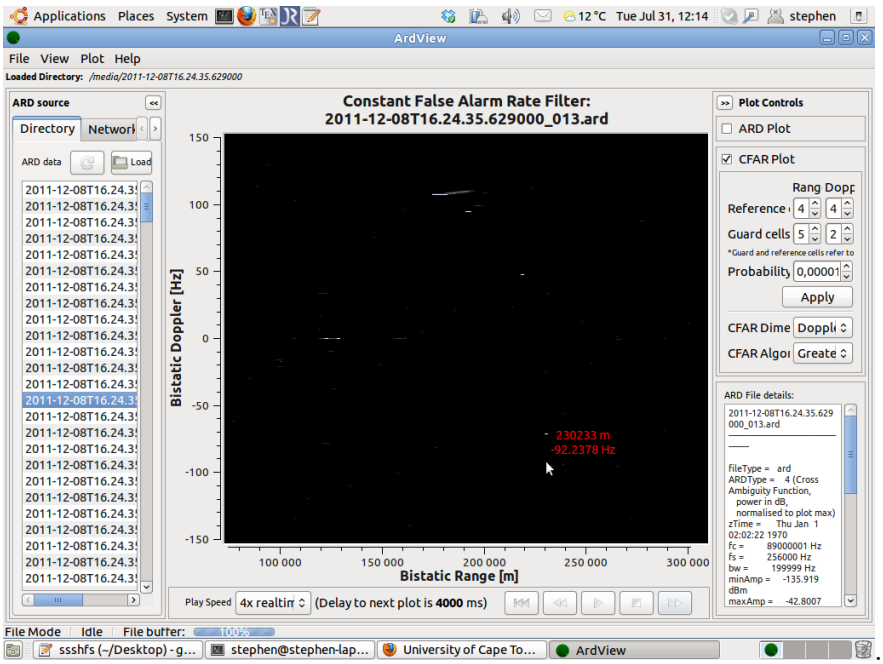

Figure 5.2: Screenshot of *ARDView* which is used to provide the filters with detection data.

View, a single target often appears as up to ten closely spaced detections with the same bistatic velocity and closely spaced in range. This will be clear in the plots that follow in the next section and obviously results in multiple tracks being formed for a single target. As the target progresses along its trajectory, these tracks are either maintained or deleted depending on the measurements available. This is unideal and so a simple centroid function was written to resolve the multiple detections of a target into a single detection.

The function operates by searching through the detection matrix for all the measurements with the same bistatic velocity. The bistatic ranges of these variables are averaged to produce a single aggregate bistatic range, which together with the bistatic velocity measurement, forms the new centroided measurement. Of course, clutter points sharing the same bistatic velocity but with significantly different range points could skew the data, however the low clutter density and good bistatic velocity resolution makes this quite unlikely. The function is not perfect, but targets are reduced to two detections at worst, which is acceptable

for this application.

5.2 Target of interest

The RRSG has various datasets of commensal radar measurements resulting from various measurement campaigns making use of several receiver sights. Targets come and go in these datasets quite frequently, though rarely exist for more than 30 CPIs.

The dataset focused on in the experiments to follow was recorded on 8 December 2011 using Constansiaberg's Five FM transmissions at 89 MHz. The receiver site was Malmsbury. The dataset specifies a best case range resolution of 1171 m and a Doppler resolution of 0.25 Hz.

At the very beginning of this dataset, a target appears at a bistatic range of about 200 km and a Doppler frequency of 115 Hz (the CFAR display shows Doppler and not velocity). The target persists for about 30 CPIs and closes in range while its Doppler frequency remains fairly constant. This target is used to compare the filters and is henceforth referred to as the 'target of interest'. The 'phosphor trace' of the target of interest is clearly visible in Figure 5.3. Other targets do appear later in the dataset, but do not persist for very long.

The target of interest is missed (not detected by the CFAR) only once during its persistence. Range smear is confined to about 10 km. When the CFAR'd data is passed to MATLAB, a range smeared target appears as several detections closely spaced in range and sharing the same Doppler.

5.3 Results

We begin our experiments with the Kalman filter. The filter is initialised in the exact same manner as in Chapter 4 and all parameters are kept the same.

The filter fails to track the target of interest. In addition to this, false tracks

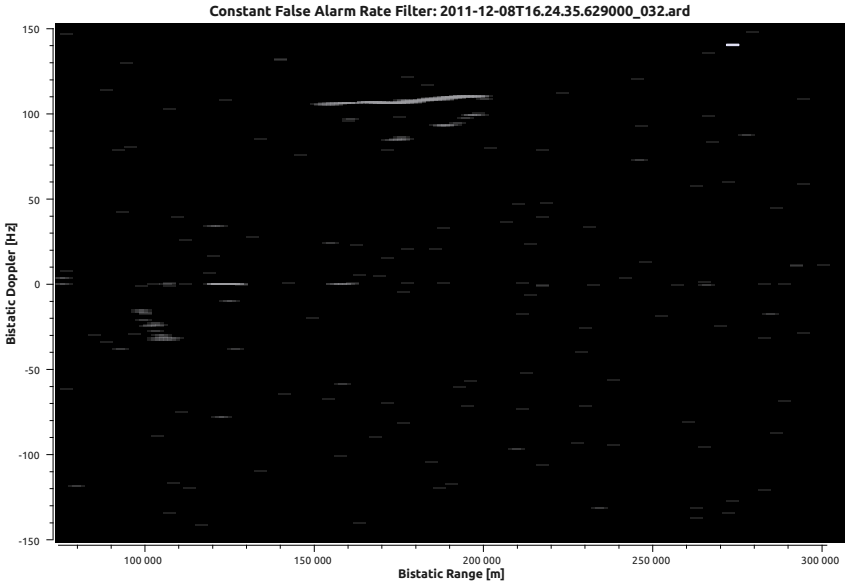

Figure 5.3: The 'phosphor trace' of the target of interest can be seen in the CFAR output starting at about 200 km and 115 Hz.

run wild across the bistatic range-velocity plot, with detections having large differences between bistatic velocity being associated with one another. This behaviour, which is shown in Figure 5.4, is clearly as a result of the very large gate ($G = 100$).

To minimise the number of false detections, the gate size is reduced in stages and the results are observed. Values below ten begin to produce an acceptably low number of false tracks. Decreasing the gate all the way down to $G = 2$ all but eradicates the false tracks. The true targets, however, are still not tracked.

This problem is addressed by tweaking the measurement noise covariance matrix. The estimation of the variance of the range measurement errors used in the simulations in Chapter 4 ($\sigma_R^2 = 45$) worked suitably well. However, this value is very low given the nature of the measurements and thus likely the cause of no true track confirmations.

Keeping $G = 2$ then, σ_R^2 is increased systematically. A much higher value of $\sigma_R^2 = 100e3$ resulted in the target of interest's track being confirmed after 14

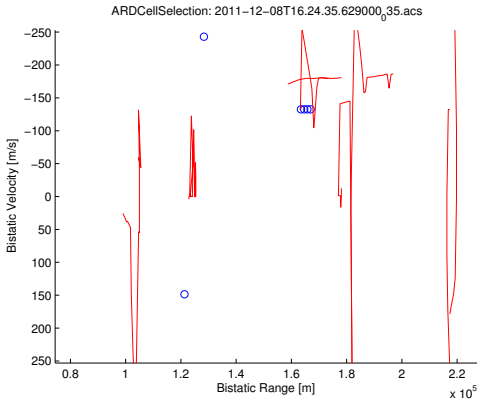

Figure 5.4: Tracking filter output showing poor tracking performance for Kalman filter with large gate. The red lines are false tracks and the blue circles are the detections from the CFAR detector.

CPIs but is lost shortly afterwards. Having achieved true track confirmation, the gate is increased to $G = 4$, and the track is confirmed after 4 CPIs. This track is then lost and subsequently reacquired several times during the target's persistence, but very few false tracks occur. The confirmed track can be seen in Figure 5.5.

Increasing the gate to $G = 6$ sees further improvements, with the track being confirmed after 3 CPIs - the first opportunity for track confirmation. The target is tracked continuously for about 25 CPIs after which an incorrect measurement assignment results in track divergence. This divergence is shown in Figure 5.6 and is as a result of the larger gate allowing detections which are fairly distant in the bistatic velocity dimension to be associated. Clearly, a larger gate improves the probability of track confirmation and track maintenance at the expense of more false tracks. However, with this gate value, the false tracks that do appear are soon deleted and do not run wild as was illustrated in Figure 5.4.

The range error variance and gate size arrived at in the above experiments are taken to be somewhat optimal - balancing true track initiation probability against false track confirmation and true track deletion probabilities. Thus, it is these values ($\sigma_R^2 = 100e3$ and $G = 6$) that are used in the following experiments with the RGN filter.

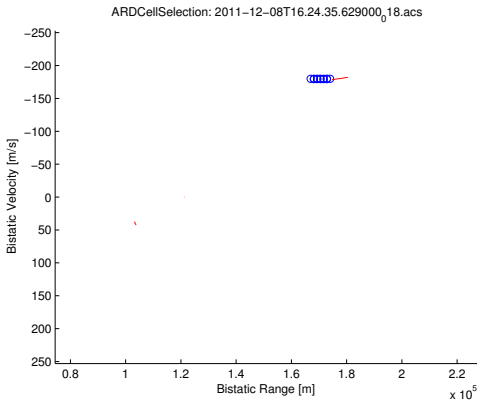

Figure 5.5: Deleting and reacquiring the target results in this short target track.

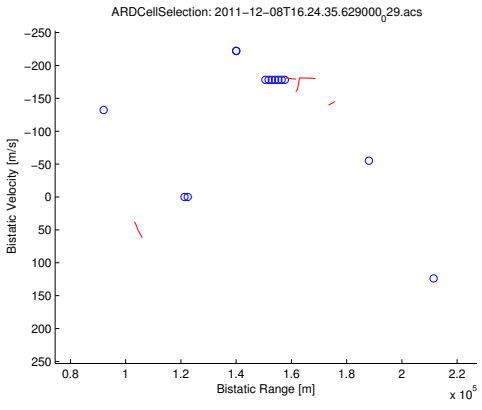

Figure 5.6: Tracking filter output showing the wrong measurement being assigned to the filter resulting in true track being lost.

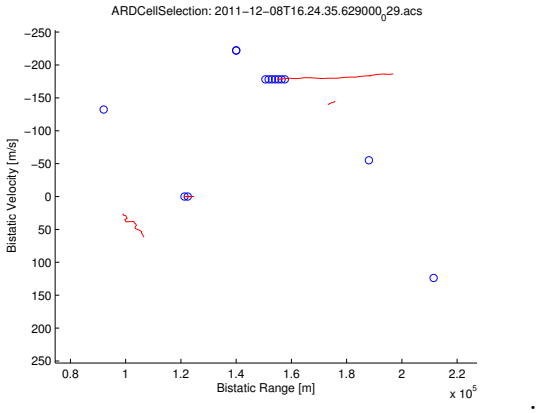

Figure 5.7: The RGN filter tracking the target of interest for the duration of its persistence with $\lambda = 0.8$.

Starting with $\lambda = 1$, the target of interest is confirmed after 3 CPIs. The filter goes on to behave similarly to the Kalman filter above, with the track being deleted and reacquired several times during the target's persistence. The same measurement assignment error occurs, resulting in the track diverging. The number of false tracks confirmed is similar.

Decreasing λ to 0.8 sees the track still being confirmed after 3 CPIs. The target is tracked better however, with no track deletions occurring. The target is tracked for the duration of its persistence. This complete track is shown in Figure 5.7. Targets which appear later on in the dataset are also tracked very well. Figure 5.8 shows the phosphor trace of these targets in the CFAR display as well as the target tracks.

Further decreasing λ to 0.6 results in the tracking filter's performance degrading. More false tracks are formed which span the entire velocity axis. The false tracks persist for many CPIs, sometimes tracking true targets as well.

In considering the polynomial filter, all initialisations and parameters are kept the same as in Chapter 4. This involves an initial gate of 2 m/s for velocity measurements and 5000 m for range measurements. After transitioning to the 2nd degree expanding memory polynomial when $k = 8$, elliptical gating is employed with a gate size $G = 100$.

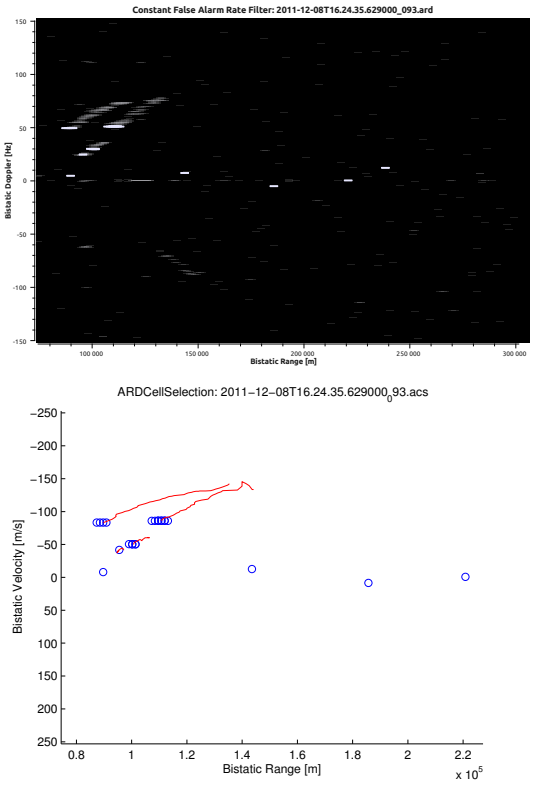

Figure 5.8: Three targets appear later on in the data set (top). The three targets being tracked by the RGN filter with $\lambda = 0.8$ (bottom).

Starting with a fading parameter of $\theta = 0.99$, the track is confirmed after 3 CPIs. The target is successfully tracked for the duration of its persistence. After the target disappears, the track is appropriately deleted. The tracking performance is similar to that of the Kalman filter. While the target of interest is successfully tracked, other targets which appear later in the dataset are not always confirmed and tracked.

In an attempt to improve true track confirmation, the rectangular gate size is increased to 5 m/s and 15 km, then to 10 m/s and 20 km. Larger gates do result in more of the other targets being tracked and while false tracks appear slightly more often, they are still kept to a minimum. This is an expected trade off with increasing the gate sizes.

As performance is still not satisfactory, the initial rectangular gate is done away with and replaced by an elliptical gate, taking into account filter covariance and measurement noise covariance matrices. Again, performance improves slightly (in terms of track confirmation), but tracking performance remains on a par with the Kalman filter.

Due to the fact that the tracks never persist for much more than 10 CPIs ($k \leq 10$), the transition to the FMP never occurs. Even for low values of θ, such as $\theta = 0.6$, the transition only happens at $k = 11$.

5.4 Conclusions

This chapter detailed the application of the three tracking filters to real commensal radar data.

The first conclusion that can be made is that the simulated and real data are very different. Applying the exact same filters to the simulated and real data results in very different tracking performance. More false tracks were formed by the Kalman and RGN filters and the Kalman failed to successfully initialise the target of interest.

Tweaking the measurement error covariance matrix and gate parameters improved the performance of all the filters. The Kalman was able to track the target of interest competently enough, but was outshone by the RGN filter with $\lambda = 0.8$. In this configuration the target was tracked from beginning to end. More false tracks occurred with lower values of λ, but occurrences were acceptable for $\lambda = 0.8$. As expected from Chapter 4's results, the polynomial filter confirmed the fewest false tracks, but surprisingly, was not as competent at confirming true tracks. This somewhat strange behaviour of the polynomial filter could be due to the measurement noise covariance matrix and can be investigated further.

In addition to this, it would be interesting to investigate the performance of other data association methods. As the problems experienced here (track deletion, false track confirmation and track initiation) are essentially data association problems, it would certainly be worthwhile exploring multiple hypothesis tracking and joint probabilistic data association techniques.

Unfortunately, as truth data was unavailable, tracking errors were not obtainable. However, given the successful tracking shown (particularly by the RGN filter), it would be fair to assume these errors to be reasonable. However, individual filters cannot be directly compared.

Chapter 6

Conclusions and recommendations

The objective of this project was to evaluate several tracking filters assigned to the task of range-Doppler space tracking. This is intended to aid the identification of targets in ARD plots and involved using simulated measurements and then real commensal radar measurements. The Kalman, recursive Gauss-Newton and polynomial filters were considered.

A simulation environment was built in MATLAB and used to evaluate the filters based on several performance metrics. The performance metrics considered were computational load, tracking errors and data association statistics. The next step was to apply the tracking filters to real commensal radar data.

The conclusions drawn from the above steps are presented in the next section and are followed by recommendations that can be made for future work.

6.1 Conclusions

The three filters are derived using different methods. The Kalman and RGN filters, which take the probabilistic and statistical approaches respectively, end

up as two very similar filters, differing only by the forgetting factor of the RGN filter. The polynomial filter uses the least squares method like the RGN filter, but by fitting a m degree polynomial to the observation data instead.

Out of these three filters, the RGN filter was found to perform the best, having a low computational load, good tracking errors and good data association statistics. Of the three filters applied to the commensal radar data, it performed the best, tracking all the targets with few false tracks being formed.

Detailed conclusions for each of the filters are presented below and are followed by recommendations for future work.

6.1.1 Kalman filter

The ubiquitous Kalman filter, derived from probabilistic methods and widely studied, was found by Howland to work well for range-Doppler tracking applications. It is favoured over the many other filters for its computational efficiency.

The filter performed well in the target tracking simulations by quickly settling to reasonably low tracking errors. While the polynomial filter showed the potential to improve on the Kalman filter's results, convergence was slow.

However, the Kalman filter's data association statistics were poor. The probability of true track confirmation was low, while many false tracks were confirmed. True track deletion was also poor. This was, however, shown to come from the filter covariance initialisation. This was demonstrated by the RGN filter with a forgetting factor of unity (which had a different filter covariance matrix initialisation). Thus, the importance of proper parameter selection was demonstrated and the Kalman filter's performance in this regard can easily be improved on.

Applying the filter to real data required some tuning of the parameters, notably the measurement noise covariance matrix as well as the gate. This tuning led to reasonable tracking performance, although the target of interest was lost and reacquired from time to time.

On the whole, the Kalman filter lived up to its reputation and performed well in all regards.

6.1.2 Polynomial filter

The literature suggested that the polynomial filter might perform well from the data association point of view, but at greater computational cost.

Computationally, the polynomial was indeed found to be more expensive. While the actual update equations are quite simple, the decision logic associated with the composite polynomial filter makes it slightly more expensive than the Kalman and RGN filters.

The literature's conjecture was further confirmed by the data association simulations where the polynomial filter performed the best out of the three tracking filters. The filter confirmed the fewest false tracks, had the highest probability of true track confirmation and the lowest probability of false track confirmation.

When applied to the real commensal radar data however, the polynomial filter was outshone by the RGN filter. While the polynomial filter still confirmed the fewest false tracks, and was not susceptible to false tracks spanning the Doppler/velocity axes, it did not confirm true tracks as effectively as was suggested by the simulations. This behaviour points to gating techniques, track confirmation logic and data association techniques. The filter's fading parameter did not come into play as the track seldom lasted long enough for the transition to occur.

In terms of tracking errors, the polynomial filter was found to be sensitive to changes in its fading parameter. A compromise was found to exist between accurate range tracking and accurate Doppler tracking. Trends showed that the polynomial's range errors decreased to values lower than those of the other filters. However, this convergence took much longer than the other filters. This is likely an effect of the polynomial's tedious initialisation procedure and susceptibility to quick settling. Transitions from one polynomial degree to another were observed

to cause large disturbances in tracking errors.

While the polynomial filter performed well in all the simulations, with excellent data association performance at the cost of slightly greater computational load and greater errors, its application to real data showed slightly disappointing performance.

6.1.3 RGN filter

The RGN filter is derived from statistical methods and introduces a forgetting factor to exponentially diminish the significance of older measurements. The filter arrived at is identical to the Kalman filter, differing only be the forgetting factor which is applied to the covariance matrix.

As expected, the computational load of the RGN filter is very similar to that of the Kalman filter and is not as high as that of the polynomial filter. In fact, with filter covariance computation used, the RGN filter was more efficient than the Kalman filter (although the same filter covariance computation can be used for the Kalman filter too).

The modifications made by the forgetting factor to the filter's covariance matrix were noticeable in the data association simulations, with higher forgetting factors decreasing the probability of true track confirmation while increasing the number of false tracks confirmed.

These observations were echoed when the tracking filter was applied to real data. A compromise of $\lambda = 0.8$ was found to yield performance far superior to the other filters. The RGN filter confirmed most of the target tracks in the dataset and tracked the target of interest the best out of the three filters.

The tracking errors obtained from the simulations showed a trend of improving tracking performance with increasing values of forgetting factor. As expected, a forgetting factor of unity lead to Kalman like performance for both range and Doppler tracking.

However, different filter covariance matrix initialisations and filter covariance update equations were observed to have significant effects on filter performance. A trade off was observed between tracking accuracy and data association performance. Methods leading to lower tracking errors resulted in poorer data association statistics and vice versa. The same observation applies to the forgetting factor.

As tracking targets in real data is what we are interested in, the RGN filter certainly appears to be the tracking filter of choice for this application, although future work might show otherwise.

6.2 Recommendations

Based on the above conclusions, the following recommendations for future work are made:

- For all three filters, further analysis can be carried out on the effects of the filter, measurement noise and process noise covariance matrices. These matrices play an important role in filter performance as well as the data association performance.
- Multiple hypothesis tracking and probabilistic data association methods are worth investigating. While the nearest neighbour technique was adequate, further improvements in tracking performance could be obtained by exploring other approaches.
- While the polynomial filter's performance was not the best, further experimentation could improve its performance. The filter struggled to confirm true tracks, a problem which could be explored by relaxing the M-out-of-N requirements (say to 3-out-of-5, for example). A completely different track confirmation method could also be explored.
- In addition to this, it would be worthwhile implementing a 1st degree polynomial filter. The second degree filter implemented here results in

longer delays before the transition to the FMP. Given the fleeting nature of many of the targets in the commensal radar data, this hampers the filter somewhat as the FMP stage is never reached.

- In order to obtain a better comparison between the filters, a consistent set of performance metrics, assessing smoothness amongst other things, must be developed.

Appendix A

Polynomial equations

For convenience, this appendix contains the polynomial equations from Morrison's book [17] that are used as a part of the composite polynomial filter.

A.1 Expanding memory polynomial

Firstly, the expanding memory polynomial's update and covariance equations are presented.

A.1.1 Update equations

0th degree:

$$(z_0^*)_{n+1,n} = (z_0^*)_{n,n-1} + \frac{1}{(n+1)}\epsilon_n \tag{A.1}$$

1st degree:

$$(z_1^*)_{n+1,n} = (z_1^*)_{n,n-1} + \frac{6}{(n+2)(n+1)}\epsilon_n \tag{A.2}$$

$$(z_0^*)_{n+1,n} = (z_0^*)_{n,n-1} + (z_1^*)_{n+1,n} + \frac{2(2n+1)}{(n+2)(n+1)}\epsilon_n \tag{A.3}$$

2nd degree:

$$(z_2^*)_{n+1,n} = (z_2^*)_{n,n-1} + \frac{30}{(n+3)(n+2)(n+1)}\epsilon_n \tag{A.4}$$

$$(z_1^*)_{n+1,n} = (z_1^*)_{n,n-1} + 2(z_2^*)_{n+1,n} + \frac{18(2n+1)}{(n+3)(n+2)(n+1)}\epsilon_n \tag{A.5}$$

$$(z_0^*)_{n+1,n} = (z_0^*)_{n,n-1} + (z_1^*)_{n+1,n} - (z_2^*)_{n+1,n} + \frac{3(3n^2+3n+2)}{(n+3)(n+2)(n+1)}\epsilon_n \tag{A.6}$$

A.1.2 Covariance matrix

The covariance matrix for the expanding memory polynomial filter can be calculated from:

$$S_{n+1,n}^* = \sigma_v^2 Q(n+1)Q(n+1)^T \tag{A.7}$$

where σ_v^2 is the variance of the observation errors and the elements of $Q(n+1)$ for $i, j = 0, 1, ..., m$:

$$[Q(n+1)]_{i,j} = \frac{1)}{i!}\frac{d^i}{dr^i}\phi_j(r)|_{r=n+1} \tag{A.8}$$

A.2 Fading memory polynomial

The fading memory polynomial update and covariance equations are presented here.

A.2.1 Update equations

0th degree:

$$(z_0^*)_{n+1,n} = (z_0^*)_{n,n-1} + (1-\theta)\epsilon_n \tag{A.9}$$

1st degree:

$$(z_1^*)_{n+1,n} = (z_1^*)_{n,n-1} + (1-\theta)^2\epsilon_n \tag{A.10}$$

$$(z_0^*)_{n+1,n} = (z_0^*)_{n,n-1} + (z_1^*)_{n+1,n} + (1-\theta^2)\epsilon_n \tag{A.11}$$

2nd degree:

$$(z_2^*)_{n+1,n} = (z_2^*)_{n,n-1} + \frac{1}{2}(1-\theta)^3\epsilon_n \tag{A.12}$$

$$(z_1^*)_{n+1,n} = (z_1^*)_{n,n-1} + 2(z_2^*)_{n+1,n} + \frac{3}{2}(1-\theta)^2(1+\theta)\epsilon_n \tag{A.13}$$

$$(z_0^*)_{n+1,n} = (z_0^*)_{n,n-1} + (z_1^*)_{n+1,n} - (z_2^*)_{n+1,n} + \frac{3}{2}(1-\theta^3)\epsilon_n \tag{A.14}$$

A.2.2 Covariance matrix

The covariance matrices for the fading memory polynomial are calculated as:

$$S_{n+1,n}^* = \sigma_v^2 F(r) A F(r)^T \tag{A.15}$$

where σ_v^2 is the variance of the observation errors (which are necessarily equal) and:

$$[F(r)]_{i,j} \equiv \frac{(-1)^i}{i!}\frac{d^i}{dr^i}p_j(r) \tag{A.16}$$

and the elements of A for $\lambda, \mu = 1, 2, .., m$ are:

$$a_{\lambda,\mu} = \binom{\lambda + \mu}{\lambda} \frac{1 - \theta}{(1 + \theta)^{\lambda+\mu+1}} \tag{A.17}$$

Appendix B

Additional results

The Doppler error variances and maximum errors, which are referred to in Chapter 4, are presented here.

Table B.1: Variance of the Doppler errors for the three different filters with different values of λ and θ.

Kalman	λ	RGN	θ	Polynomial
	0.6	3.571e-03	0.7	4.515e-03
	0.7	3.527e-03	0.8	3.365e-03
3.475e-03	0.8	3.533e-03	0.9	3.667e-03
	0.9	3.472e-03	0.95	5.936e-03
	1	3.416e-03	0.99	6.349e-03

Table B.2: Aggregate of the maximum Doppler errors in a given scan for the three different filters with different values of λ and θ.

Kalman	λ	RGN	θ	Polynomial
	0.6	2.721e-01	0.7	3.037e-01
	0.7	2.703e-01	0.8	2.546e-01
2.695e-01	0.8	2.717e-01	0.9	2.597e-01
	0.9	2.712e-01	0.95	3.137e-01
	1	2.652e-01	0.99	3.225e-01

Appendix C

EBE Faculty: Assessment of Ethics in Research Projects

Any person planning to undertake research in the Faculty of Engineering and the Built Environment at the University of Cape Town is required to complete this form before collecting or analysing data. When completed it should be submitted to the supervisor (where applicable) and from there to the Head of Department. If any of the questions below have been answered YES, and the applicant is NOT a fourth year student, the Head should forward this form for approval by the Faculty EIR committee: submit to Ms Zulpha Geyer (Zulpha.Geyer@uct.ac.za; Chem Eng Building, Ph 021 650 4791).Students must include a copy of the completed form with the final year project when it is submitted for examination.

Name of Principal Researcher/Student: Stephen Middleton

Department: ELECTRICAL ENGINEERING

If a Student: YES **Degree:** M.Eng (Radar) **Supervisor:** Michael Inggs

If a Research Contract indicate source of funding/sponsorship: NA

Research Project Title: Tracking in the Range-Doppler Space

Overview of ethics issues in your research project:

Question 1: Is there a possibility that your research could cause harm to a third party (i.e. a person not involved in your project)?	NO
Question 2: Is your research making use of human subjects as sources of data?	NO
Question 3: Does your research involve the participation of or provision of services to communities?	NO
Question 4: If your research is sponsored, is there any potential for conflicts of interest?	NO

If you have answered YES to any of the above questions, please append a copy of your research proposal, as well as any interview schedules or questionnaires (Addendum 1)vand please complete further addenda as appropriate.

I hereby undertake to carry out my research in such a way that:

- there is no apparent legal objection to the nature or the method of research; and
- the research will not compromise staff or students or the other responsibilities of the University;
- the stated objective will be achieved, and the findings will have a high degree of validity;
- limitations and alternative interpretations will be considered;
- the findings could be subject to peer review and publicly available; and
- I will comply with the conventions of copyright and avoid any practice that would constitute plagiarism.

Signed by:

	Full name and signature	**Date**
Principal Researcher/Student:	Stephen Middleton	31 August 2012

This application is approved by:

Supervisor (if applicable):	Michael Inggs	31 August 2012
HOD (or delegated nominee):	Janine Buxey	31 August 2012

Bibliography

[1] N. J. Willis and H. D. Griffiths, eds., *Advances in Bistatic Radar*. SciTech, 2007.

[2] D. Tan, H. Sun, Y. Lu, M. Lesturgie, and H. Chan, "Passive radar using Global System for Mobile communication signal: theory, implementation and measurements," *Radar, Sonar and Navigation, IEE Proceedings* -, vol. 152, pp. 116 – 123, june 2005.

[3] H. Griffiths and N. Long, "Television-based bistatic radar," *Communications, Radar and Signal Processing, IEE Proceedings F*, vol. 133, pp. 649 –657, december 1986.

[4] F. Canini, A. Di Lallo, L. Timmoneri, and D. Vigilante, "Use of Digital-Television terrestrial (DTV) signals for passive radars," in *Radar Symposium (IRS), 2010 11th International*, pp. 1 –4, june 2010.

[5] X. He, T. Zeng, and M. Cherniakov, "Signal detectability in SS-BSAR with GNSS non-cooperative transmitter," *Radar, Sonar and Navigation, IEE Proceedings* -, vol. 152, pp. 124 – 132, june 2005.

[6] P. Falcone, F. Colone, A. Macera, and P. Lombardo, "Localization and tracking of moving targets with WiFi-based passive radar," in *Radar Conference (RADAR), 2012 IEEE*, pp. 0705 –0709, may 2012.

[7] P. Howland, D. Maksimiuk, and G. Reitsma, "FM radio based bistatic radar," *Radar, Sonar and Navigation, IEE Proceedings* -, vol. 152, pp. 107 – 115, june 2005.

[8] C. Tong and M. R. Inggs, "Report on the Commensal Radar work at UCT during 2011," Progress report rrsg7372, University of Cape Town, Private Bag Rondebosch 7701, South Africa, Mar. 2012.

[9] H. Griffiths, C. Baker, H. Ghaleb, R. Ramakrishnan, and E. Willman, "Measurement and analysis of ambiguity functions of off-air signals for passive coherent location," *Electronics Letters*, vol. 39, pp. 1005 – 1007, june 2003.

[10] C. Tong, M. Inggs, and G. Lange, "Processing design of a networked passive coherent location system," pp. 692 –697, may 2011.

[11] R. Nadjiasngar, S. Middleton, and M. Inggs, "Doppler-only tracking with the recursive Gauss-Newton filter," *Accepted for publication*, 2012.

[12] Y.-C. Xiao, P. Wei, and T. Yuan, "Observability and Performance Analysis of Bi/Multi-Static Doppler-Only Radar," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 46, pp. 1654 –1667, oct. 2010.

[13] K. Becker, "Three-dimensional target motion analysis using angle and frequency measurements," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 41, pp. 284 – 301, jan. 2005.

[14] S. Blackman, *Muliple-Target Tracking with Radar Applications.* Artech House, 1986.

[15] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems.* Artech House, 1999.

[16] A. H. Jazwinski, *Stochastic processes and filtering theory.* Academic Press, 1970.

[17] N. Morrison, *Introduction to Sequential Smoothing and Prediction.* McGraw-Hill Book Company, 1969.

[18] M. Richards, J. Scheer, and W. Holm, eds., *Principles of modern radar: basic principles.* SciTech, 2010.

[19] N. Morrison, *Tracking Filter Engineering The Gauss-Newton and Polynomial Filters*. The IET, 2011.

[20] N. Morrison, R. Lord, and M. Inggs, "The Gauss-Newton algorithm in passive aircraft tracking using doppler and bearings," pp. 1 –5, oct. 2007.

[21] R. Nadjiasngar and M. Inggs, "The Recursive Gauss-Newton Filter," *ArXiv e-prints*, Oct. 2011.

[22] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. pp. 431–441, 1963.

[23] R. Nadjiasngar and M. Inggs, "Gauss-Newton Filtering incorporating Levenberg-Marquardt Methods for Radar Tracking," *ArXiv e-prints*, Oct. 2011.

[24] R. Nadjiasngar, M. Inggs, Y. Paichard, and N. Morrison, "A new probabilistic data association filter based on composite expanding and fading memory polynomial filters," in *Radar Conference (RADAR), 2011 IEEE*, pp. 152 –156, may 2011.

[25] V. Aidala, "Kalman Filter Behavior in Bearings-Only Tracking Applications," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. AES-15, pp. 29 –39, jan. 1979.

[26] Y. Chan and S. Rudnicki, "Bearings-only and Doppler-bearing tracking using instrumental variables," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 28, pp. 1076 –1083, oct 1992.

[27] P. Howland, "Passive Tracking Of Airborne Targets Using Only Doppler And Doa Information," pp. 37 –39, may 1995.

[28] E. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pp. 153 –158, 2000.

[29] T. Schei, "A finite-difference approach to linearization in nonlinear estimation algorithms," in *American Control Conference, 1995. Proceedings of the*, vol. 1, pp. 114 –118 vol.1, jun 1995.

[30] C. Wu and C. Han, "Strong tracking finite-difference extended Kalman filtering for ballistic target tracking," in *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*, pp. 1540 –1544, dec. 2007.

[31] N. Morrison, R. Lord, and M. Inggs, "The Gauss-Newton algorithm applied to track-while-scan radar," pp. 1 –5, oct. 2007.

[32] P. Reyneke, N. Morrison, D. Kourie, and C. de Ridder, "Smoothing irregular data using polynomial filters," in *ELMAR, 2010 PROCEEDINGS*, pp. 393 –397, sept. 2010.

[33] X. Rong Li and V. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 39, pp. 1333 – 1364, oct. 2003.

[34] H. Griffiths and C. Baker, "Measurement and analysis of ambiguity functions of passive radar transmissions," in *Radar Conference, 2005 IEEE International*, pp. 321 – 325, may 2005.

[35] R. Deming, J. Schindler, and L. Perlovsky, "Multi-Target/Multi-Sensor Tracking using Only Range and Doppler Measurements," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 45, pp. 593 –611, april 2009.

[36] J. Pisane, S. Azarian, M. Lesturgie, and J. Verly, "Automatic real-time collection of RCS of airplanes in a real bistatic low-frequency configuration using a software defined passive radar based on illuminators of opportunity," in *Radar Conference (RADAR), 2012 IEEE*, pp. 0950 –0955, may 2012.

[37] R. Singer, R. Sea, and R. Housewright, "Derivation and Evaluation of Improved Tracking Filters for Use in Dense Multi-Target Environments," *IEEE Transactions on Information Theory*, vol. IT-20, pp. 423–432, 1974.

[38] Y. Bar-Shalom and E. Tse, "Tracking in a Cluttered Environment with Probabilistic Data Association," *Automatica*, vol. 11, pp. 451–460, Sept. 1975.

[39] C. Tong, "PhD Proposal," Mar. 2012.