Target Tracking using Multisensor Data Fusion for an Unmanned Aerial Vehicle Sense and Avoid System

by

Shuaib Omar

BSc (Eng) Electrical and Computer Engineering

A thesis submitted to the Department of Electrical Engineering,

University of Cape Town, in fulfilment of the requirements

for the degree of

Master of Science

at the

UNIVERSITY OF CAPE TOWN

Supervisor:

Dr. Simon Winberg

June 2012

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the degree of Master of Science in Engineering in the University of Cape Town. It has not been submitted before for any degree or examination in any other university.

Signature of Author: . . . . . .

Cape Town

29 May 2012

Abstract

The last decade has seen a proliferation of unmanned aerial vehicles (UAV) put to use in a myriad of applications. However, the true potential of UAVs lies in its timely adoption to civil airspace. For this to occur, UAVs need to exhibit increased levels of autonomy. The principal impediment in achieving this autonomy is the UAV's capability to sense and avoid. The aim of this dissertation is to develop an airborne threat tracking system using multisensor data fusion for a UAV sense and avoid system, termed the threat tracking unit (TTU).

This thesis entails an extensive study of relevant literature in the fields of target tracking, estimation and multisensor data fusion. Following this, a simulation environment was developed to test and qualify various iterations of the TTU.

Initial TTU designs entailed the design of single sensor tracking filters in order set a baseline with which the performance of multisensor tracking filters can be compared against. Subsequent to this, multisensor fusion architectures were investigated drawing on the shortcomings of their single sensor counter-parts.

A centralised fusion architecture (FA1) demonstrated superior performance in both position and velocity tracking. However, this architecture possesses a single point of failure in flight-critical situations. Comparatively, a distributed fusion architecture (FA2) displayed adequate performance. This architecture, nevertheless, is characterised by increased robustness and lower computational load.

Finally, in order to further reduce computational complexity, a full dynamic feedback fusion architecture (FA3) was developed. Although FA3 displayed reduced tracking accuracy, feasible results were still obtained.

For Firdows, my dearest wife.

Acknowledgements

I would like to express my deepest gratitude to **The All Mighty** for granting me the strength to complete this massive task.

Thank you Dr. Simon Winberg for your constant support and supervision throughout this journey.

I am grateful to my parents and siblings for always encouraging me and wishing the best for me.

I acknowledge Tellumat for providing me with the financial aid to undertake this study. I would especially like to thank Chris Williams, Simon Norval and Johann Gerber. My gratitude goes to Eugene Julie for always being keen to bounce ideas off and provide me with valuable information.

Amir Patel, my studies and research would never have been the same without you working by my side. It has surely been an adventure. In reply to you I shall say, “Cheers!”

I would also like to thank all my friends.

Last, but most importantly, I am especially indebted to my wife, Firdows Omar. Without her this dissertation would not exist; there were times when I lost my way and hope and she pulled me back on the straight path. For that I am truly grateful. Thank you for your unwavering support and patience.

Nomenclature

ANEES	Average Normalised Estimation Error Squared
ATC	Air Traffic Control
BOT	Bearings-only Tracking
BRF	Body Reference Frame
CA	Collision Avoidance
DCM	Direction Cosines Matrix
EKF	Extended Kalman Filter
EO	Electro-Optic
FA	Fusion Architecture
FAA	Federal Aviation Administration
FC	Fusion Centre
FLIR	Forward Looking Infra-Red
FOR	Field of Regard
FOV	Field of View
IFR	Instrument Flight Rules
IRST	Infra-Red Search and Track
KF	Kalman Filter
LFM	Linear Frequency Modulation
LSE	Least Squares Estimation
MCS	Monte Carlo Simulations
MSDF	Multisensor Data Fusion
NED	North-East-Down
NMAC	Near Mid Air Collision
PF	Particle Filter
PLKF	Pseudo Linear Kalman Filter

RMSE	Root Mean Squared Error
SAA	Sense and Avoid
SAS	Sense and Avoid System
SS	Self Separation
TTU	Threat Tracking Unit
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
VFR	Visual Flight Rules

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

This dissertation aims to design an airborne tracking filter using multisensor data fusion. This filter is to be integrated into a tracking subsystem, termed the *threat tracking unit*, which forms part of a larger UAV sense and avoid system (SAS).

The remainder of this chapter concerns the research focuses, the research objectives and motivations for these.

1.1 Background

The ability to deploy unmanned aerial vehicles (UAVs) in a myriad of life-critical missions has seen its presence rapidly increase over the last decade [**1**]. Some of these UAV-suited applications are border patrol, battlefield surveillance (depicted in Figure 1.1), etc. Traditionally, UAV mission success would be dependent on some form of human involvement. Whether it is in direct remote-control (man-in-the-loop) [**2**] or assessment of data gathered; required human resources scale with mission criticality and number of UAVs available in said mission. Consider, for example, a military application where multiple UAVs are providing battlefield surveillance. This would require numerous operators to make inferences about the vast amounts of data arriving at the ground station. Therefore, in order to fully reap the advantages provided by UAVs, a need is to mitigate human involvement; gearing UAVs toward full autonomy.

Figure 1.1: Predator UAV providing surveillance for aircraft carrier at sea [**3**].

The principal impediment in achieving this full autonomy (and mitigation of human resources) is the UAV's capability to sense and avoid. This means that a UAV needs to gather information about its immediate surroundings (sense) and, based on this information, evaluate whether a potential threat [**2**] exists and determine the necessary action to be performed (avoid). This is analogous to the 'see and avoid' process used by pilots [**4**], when deciding whether an aircraft manoeuvre is required to maintain separation.

The possible applications of a fully autonomous UAV are vast. Therefore, the future sees a movement from the operation of UAVs in segregated airspace [**5**] to civil airspace; where UAVs collocate with other manned aircraft. This civil airspace integration presents many technical and operational challenges. The most important of which, again, is the UAV's sense and avoid (SAA) capabilities. Other integration challenges include: proper cohabitation with cooperative sensors such Air Traffic Control (ATC), TCAS and ADSB;

autonomous flight of UAVs must adhere to regulations under Visual Flight Rules (VFR) and Instrument Flight Rules (IFR); as well as other safety and security issues.

A UAV's sense and avoid capabilities can be divided into two high-level subsystems [**2**], as its name implies: sensory and avoidance; where the sensory component is responsible for threat detection and tracking. This high-level view is illustrated in Figure 1.2. The threats in this case, which to avoid, are other aircraft and obstacles in the UAVs field of view (FOV).

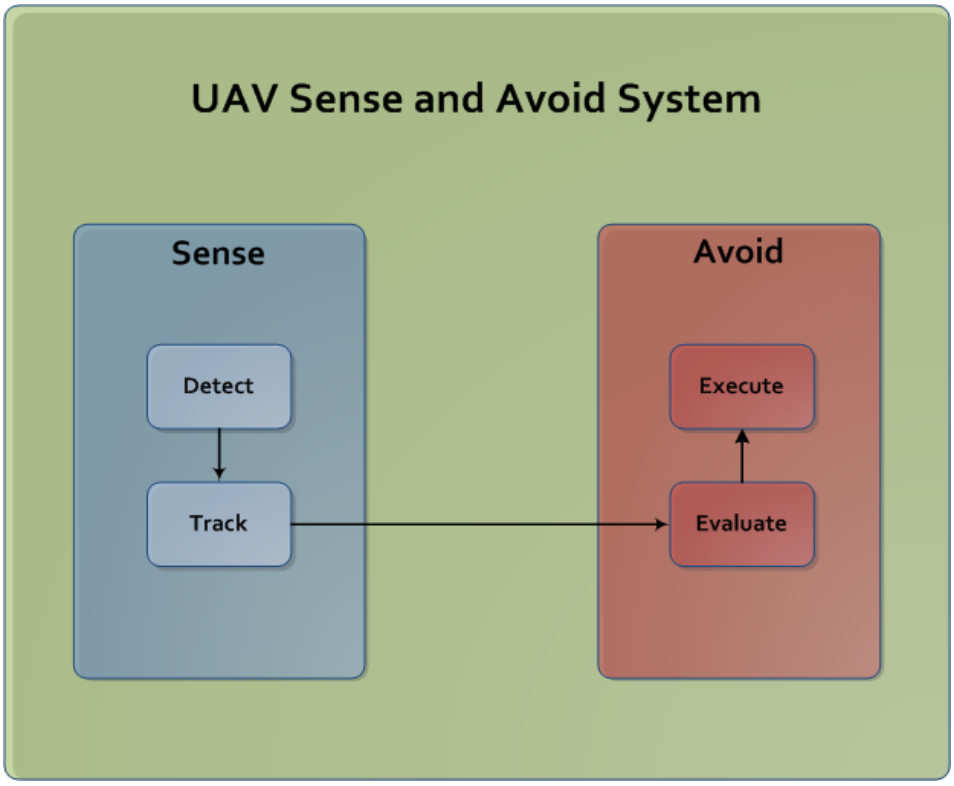

Figure 1.2: High-level view of a sense and avoid system.

1.2 Standards for a Sense and Avoid System

As an official set of standards for sense and avoid (SAA) systems did not exist, the Federal Aviation Administration (FAA) recognized the fact that this would become of paramount importance for the future of UAVs. They addressed this need, in 2008, by holding a series of workshops where experts

in collision avoidance (CA), airborne surveillance and UAV operation convened. There they discussed and documented the issues that become apparent when replacing the "see and avoid" ability of a pilot with that of the sense and avoid capability of a UAV. The workshops resulted in a clearly defined set of capabilities which a SAS should have in order to reliably, safely and autonomously avoid obstacles.

An SAA must exhibit an equivalent level of safety as that of manned aircraft. This must hold true in controlled[1] and uncontrolled airspace. The implication of this is that the SAS needs to communicate with ATC providers and/or other cooperative systems such as Traffic Alert and Collision Avoidance System (TCAS) II, Automatic Dependent Surveillance-Broadcast (ADS-B), etc. A cooperative aircraft is one that has a transponder[2] aboard and operating [**2**]. When collision avoidance is performed by an aircraft (manned or unmanned) that makes use of ATC management or one or more of the aforementioned systems, then that is termed cooperative collision avoidance.

TCAS II is a transponder based system which transmits its ownship[3] position to other aircraft. It operates independently from ATC and also analyses the projected flight path of approaching aircraft and issues ‘Resolution Advisories’ to the pilot to resolve potential mid-air collisions [**6**]. TCAS is required internationally to be fitted in aircraft with more than 30 seats or weighing more than 15000 kg and is usually found on general aviation aircraft, but it only issues avoidance manoeuvres in the vertical plane.

In contrast to TCAS, ADS-B sees a move away from surveillance radar to a Global Navigation Satellite System (GNSS). An aircraft fitted with ADS-B is able to derive its precise position from the GNSS constellation, via a GNSS receiver [**7**]. This position is packaged along with other aircraft information such as speed, heading, altitude and flight number, and simultaneously transmitted to other ADS-B capable receivers as well as satellite

[1] Controlled airspace is that in which air traffic control (ATC) service is provided to flights.
[2] A transponder provides an electronic means of aircraft identification.
[2] A transponder provides an electronic means of aircraft identification.
[3] The term “ownship” refers to “our own arcraft”.

communications transceivers. The latter forwards the information on to ATC centres. Figure 1.3 illustrates this process.

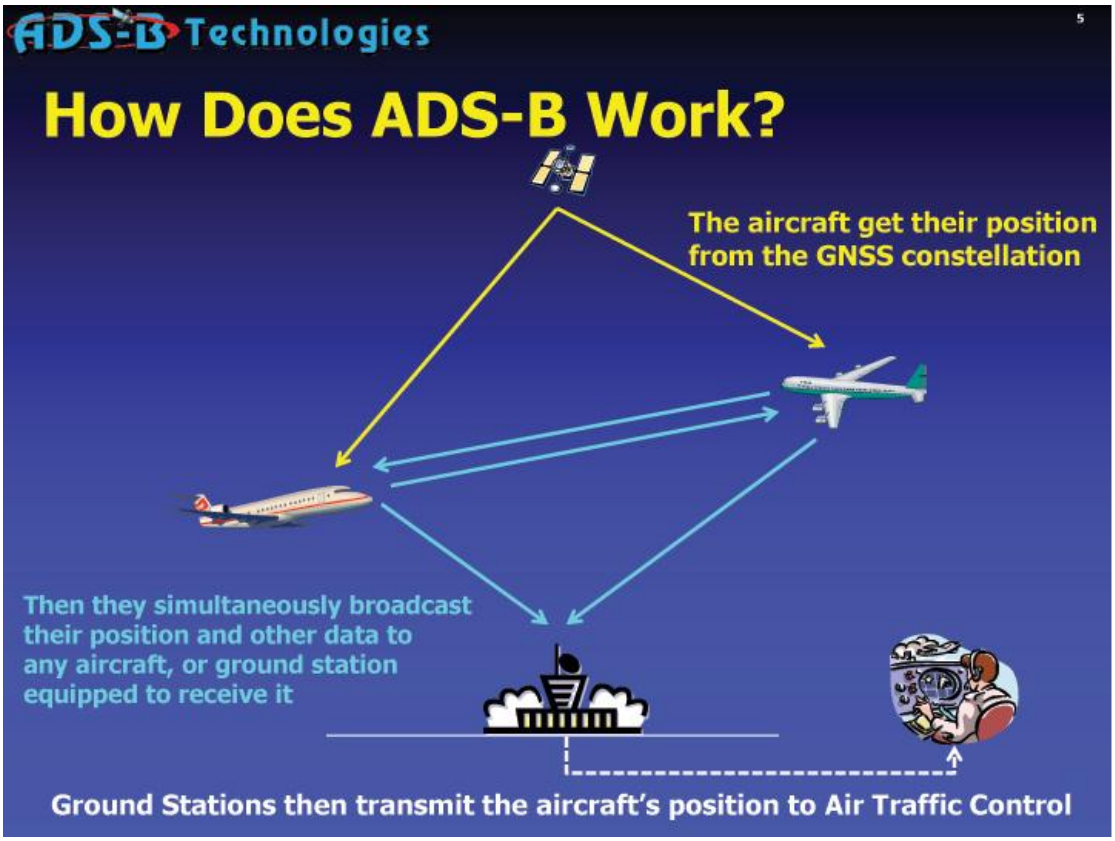

Figure 1.3: ADS-B in operation [**7**].

The FAA defines SAA as the capability of an Unmanned Aerial System[4] (UAS) to remain well clear from and avoid collisions with other airborne traffic; this consequently results in the two main functions of a SAS: Self Separation (SS) and CA (see Figure 1.4). SS is the ability of the UAV to constantly remain well clear from other aircraft. It involves commanding small bank angles, whereas CA involves drastic last minute evasive manoeuvres. SS can be thought of as maintaining a distance so that a CA manoeuvre will not become necessary.

[4] An Unmanned Aerial System is comprised of three components: a UAV; a wireless communications data link; and a ground station.

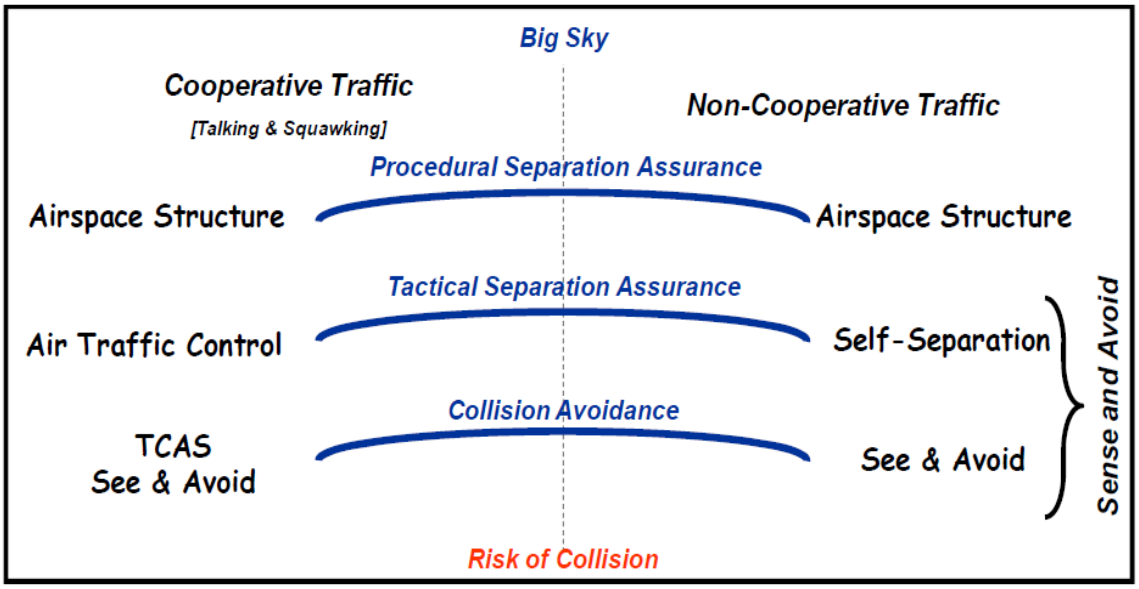

Figure 1.4: Layered relationship between cooperative and non-cooperative means of avoiding collisions [**8**]. CA, at the bottom, has the highest risk of collision as it is the UAV's last line of defence.

With reference to the aforementioned, a SAS is comprised of eight sub-functions:

i. **Detect** – Determine presence of potential hazards.

ii. **Track** – Estimated position and velocity (state) of a single intruder based on one or more surveillance reports.

iii. **Evaluate –** Assess collision risk based on intruder and UAV states.

iv. **Prioritize –** Determine which intruder tracks have met a collision risk threshold.

v. **Declare –** Decide that action is needed.

vi. **Determine Action –** Decide on what action is required.

vii. **Command –** Communicate determined action.

viii. **Execute –** Respond to the commanded action.

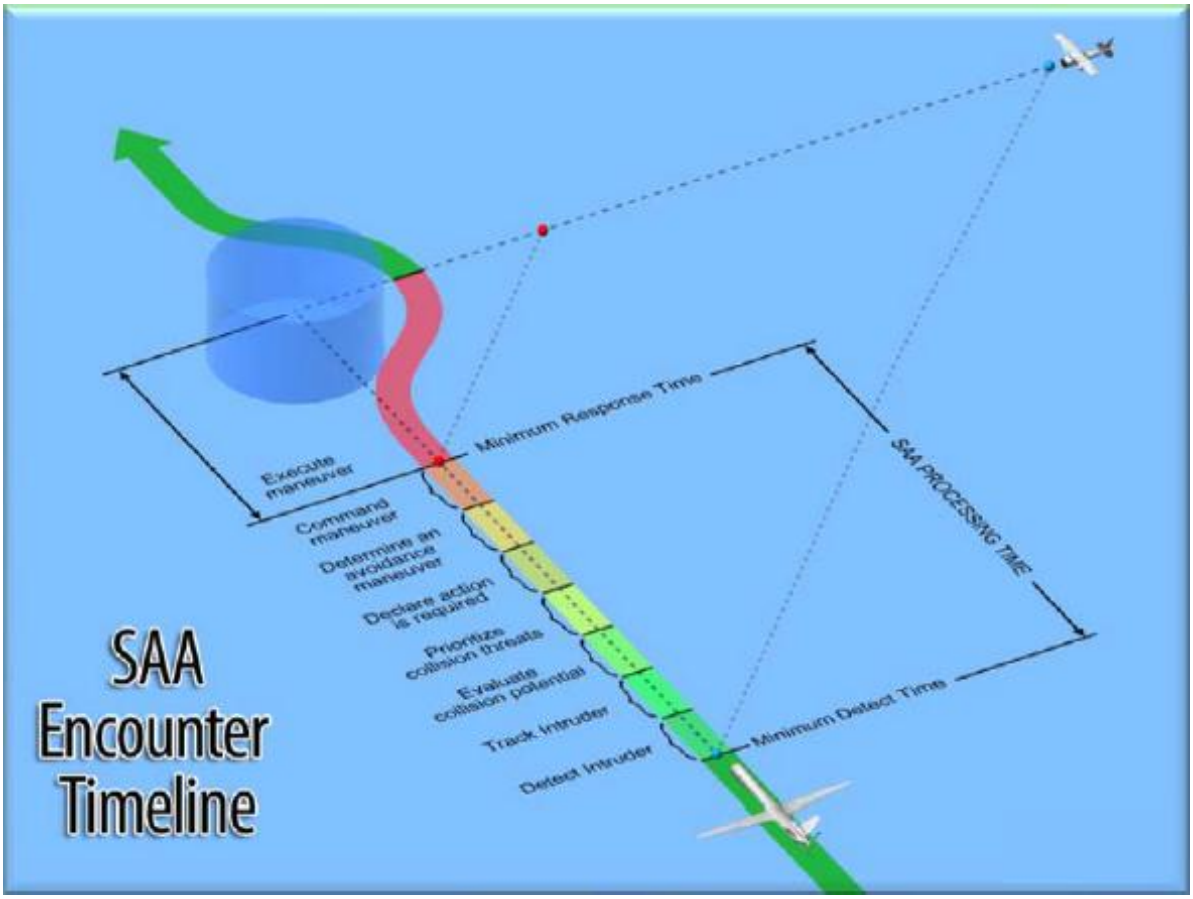

Figure 1.5: The eight SAA sub-functions.

Figure 1.5 shows the sub-functions on an encounter timeline. It brings to light the order of functions to be performed by a SAS from the moment of threat detection. Moreover, it shows the importance of limits (thresholds) in an encounter; here described as Minimum Detect Time and Minimum Response Time. These limits indicate that in order for a UAV to successfully evade a threat, headed on a collision course, there exists some fixed time that detection needs to occur. Furthermore, that the entire computational process of detection, tracking and evaluation as well as mechanical actuation of the determined avoidance manoeuvre needs to take place within a specified time limit[5] (SAA Processing Time); this takes into account physical limitations and provides the UAV enough time with which to evade the target. Consider the following scenario; if any of the sub-functions take longer than their specified time (defined in the figure), the SAA Processing Time will be pushed

[5] This last moment of action varies with different targets and is also dependent on ownship parameters.

back after the Minimum Response Time. The consequence of this, as one would assume, could lead to catastrophic results as the UAV will not be able to perform its avoidance manoeuvre in time to successfully evade the threat.

According to United States National Airspace System (NAS) regulations [**2**] a pilot must avoid collisions by remaining "well clear" from other aircraft. Translating this obscure term into an explicit specification for a SAS proved problematic for the team of experts; the term suggests that a pilot should use his subjective assessment when performing an evasive manoeuvre. Functionality that is clearly not able to be part of a sense and avoid system. The solution to this problem was to define a safety volume for the UAV as well as dynamic thresholds dependent on each aircraft being tracked in the UAV's FOV.

The volumes and thresholds that were defined are as follows:

- **Collision Volume**. A cylindrical volume around the UAV that travels with it and has a horizontal radius of 500 feet and a vertical height of 200 feet[6]. If an aircraft penetrates this volume then that is considered a Near Mid Air Collision (NMAC). An NMAC is potentially very dangerous, even if the two aircraft do not collide, as the UAV can be caught in the wake turbulence of the other aircraft and lose control.
- **Collision Avoidance Threshold**. Although illustrated in Figure 1.6 as a boundary, this is in actual fact a dynamic length of time that varies with for example, a threat's closure-rate (speed), distance and airspace class. It is the time from the moment a threat penetrates the Collision Avoidance Threshold to the moment of the two aircraft's closest point of approach (CPA) along their flight trajectories. The threshold is determined so that the UAV has enough time, once the threshold has been breached, to keep the threat out of its Collision Volume by performing an evasive manoeuvre.

[6] That is 100 feet above and below the UAV.

- **Self Separation Threshold**. Similarly to the Collision Avoidance Threshold, the Self Separation Threshold is also a dynamic length of time that is affected by similar variables mentioned above. It is the time from the moment a threat penetrates the Self Separation Threshold to the moment of CPA along their flight trajectories. This threshold ensures that the UAV remains well clear from the threat aircraft, by performing a SS rather than a CA evasive manoeuvre. The term well clear is thus defined as a threat aircraft being outside of the Collision Avoidance Threshold.

The SS and CA thresholds, therefore, have the following effects, respectively:

- If a threat aircraft penetrates the SS Threshold, the UAV would perform an evasive SS manoeuvre (commanded by the SAS of course). By the definition of the threshold it allows the UAV enough time to perform this evasive manoeuvre while keeping well clear from the threat aircraft (outside of the Collision Avoidance Threshold).
- If a threat aircraft penetrates the CA Threshold, the UAV would perform an evasive CA manoeuvre. By the definition of the threshold it allows the UAV enough time to perform this last-minute evasive manoeuvre while preventing an NMAC (keeping the threat aircraft outside of the Collision Volume).

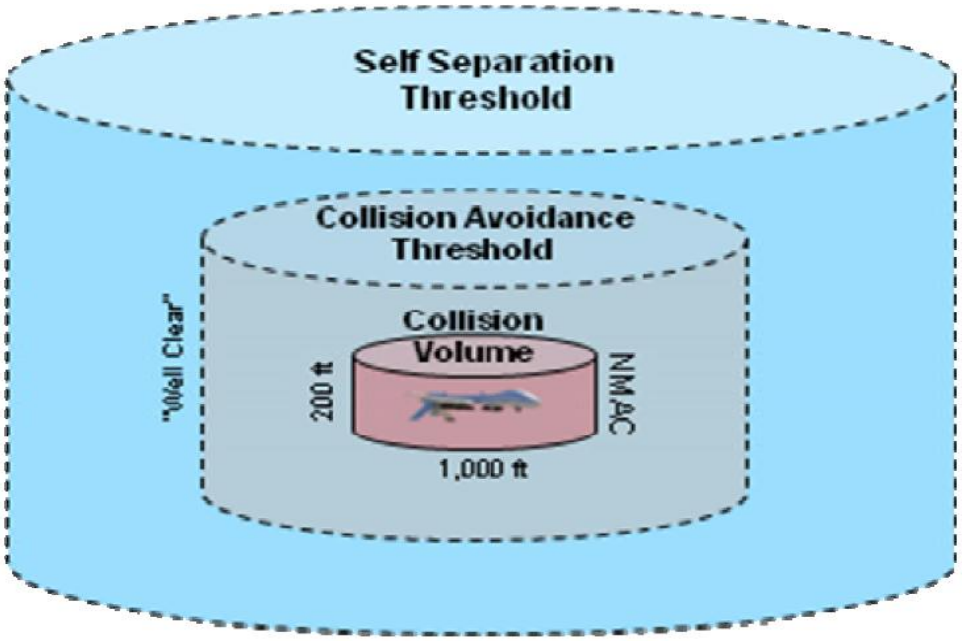

Figure 1.6: Visual representation of the Collision Volume as well as the SS and CA Thresholds [**2**]. A UAV is said to be well clear of an intruder aircraft if the intruder is outside of the CA Threshold.

A prerequisite for a SAS to make any inferences about its surrounds, is the presence of one or more sensors on-board the UAV platform. They provide the SAS with the capability to detect aerial threats and other obstacles in the UAV's path. Sensors commonly found on a UAV include electro-optic (EO) cameras, forward looking infra-red (FLIR)/infra-red search and track (IRST) sensors, and radars. Sensors on-board a UAV, which feed a SAS, must have a field of regard (FOR) of ± 110° in azimuth (relative to longitudinal axis out of the nose of the UAV) and ± 15° in elevation [**9**]. This is to ensure that sufficient coverage is provided to the SAS for reliable operation, especially during manoeuvres.

Table 1.1: Field of Regard requirements for an on-board UAV sensor used by a sense and avoid system.

Direction	**Sensor Field of Regard**	
Azimuth	-110°	+110°
Elevation	-15°	15°

A sensor's FOV is the volume in which it can detect targets while it is fixed in a single orientation. Whereas FOR is the overall FOV when the sensor occupies the extreme mechanical positions that is available to it (for example in a mechanically steered radar). The implication of this is that the FOV cannot be greater than the FOR. To explain this practically, let us consider a 2-Dimenional case where a sensor's FOV subtends a 30° arc. In the sensor's rest position, the longitudinal axis out of the centre of the lens is at 0° (see Figure 1.7). In this orientation, the FOV is ±15° (FOV1). Other than the rest position, there are another two orientations available: where the axis is centred at -15° and +15°. At -15° the FOV is -30° - 0° and at +15° the FOV is 0° - 30° (FOV2 and FOV3 respectively). Therefore the FOR for this scenario is ± 30°.

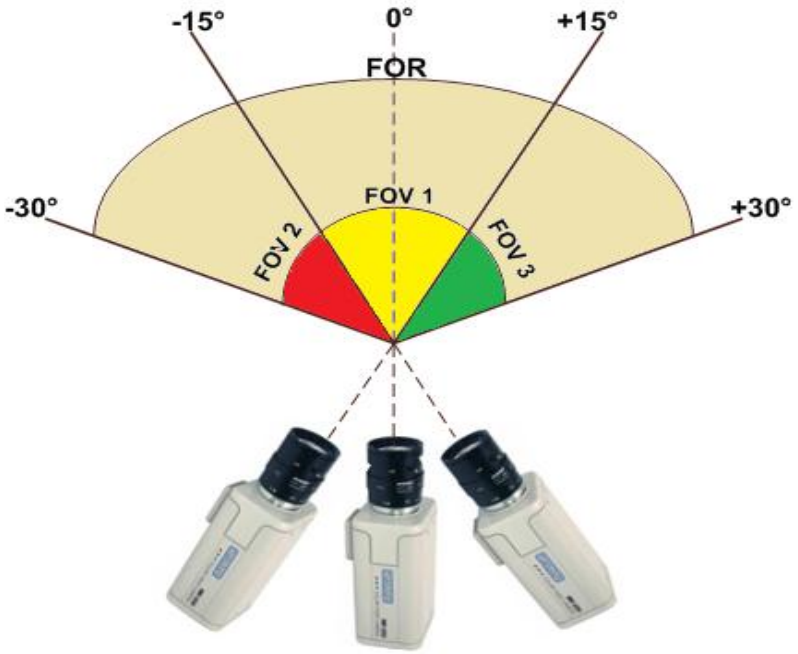

Figure 1.7: The FOR and FOV of a camera sensor with three allowable mechanical positions. Note FOV2 and FOV3 share -15° to 0° and 0° to 15° with FOV1, respectively.

In summary, the specifications (and rules and regulations) presented in this section have been developed so that a SAS which adheres to them will allow the smooth[7] integration of UAVs into civil airspace. Moreover, it will realise the

[7] Relatively smooth would probably be a better description as the words smooth and integration are, in practice, not usually married, as engineers would know all too well.

aviation community's dream of regular, fully autonomous UAV missions in civil airspace with risks to system safety properly mitigated while preserving existing air traffic flows [**8**].

1.3 Problem Description

As mentioned at the start of this dissertation, a robust and fully autonomous SAS is crucial for the future of UAVs. This SAS has the potential to become a reality because of the major advances actively being made in sensor technologies as well as computing power. Historically, the capabilities and functionality of systems were inherently limited by the technology on which they were built. This is no longer a limiting factor and researchers can begin to wholly utilise available resources.

Thus, through technological advancements, it is feasible for a SAS to make inferences on large amounts of data gathered from multiple sensors, allowing more accurate decisions to be made and the boundaries of UAVs to be pushed farther. The fusion of this multisensor data can be greatly beneficial to manned aircraft as well; where a single image is constructed from constituent sensor reports [**10**] (see Figure 1.8). This single, more meaningful "master sensor" report is what is presented to the pilot, opposed to multiple sensor reports, allowing him/her to focus more on the flying of the aircraft and less on complex sensor systems.

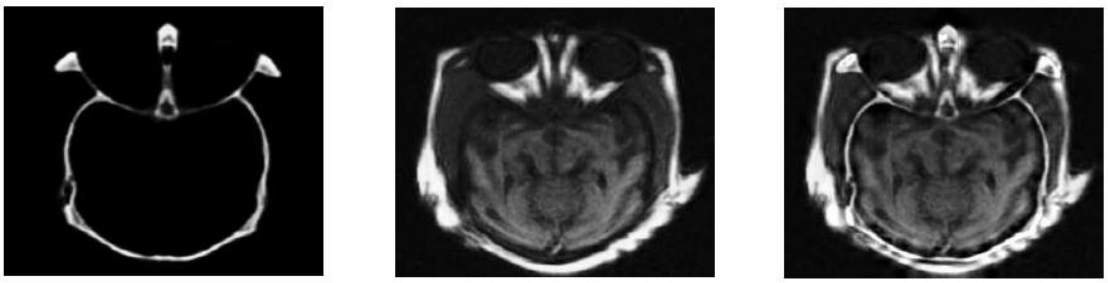

Figure 1.8: Fusion of multiple sensor images. From left to right: a CT image; an MRI image; and the, more informative, fused image [**10**].

The raison d'être of this dissertation is to present and discuss findings from the design and implementation of a threat tracking unit (here forth known as the TTU). In this project a UAV is equipped with both a passive EO camera as well as an active Doppler radar.

Kopp [**11**] is of the belief that sensor fusion techniques are ‘at an early phase in their revolutionary life cycle’. Later in his article he expresses his view: ‘It [sensor fusion] is another technology where the best is yet to come’. At the end of this journey, having analysed and solved the problems highlighted above, there would be considerable contributions to be made toward the inevitable realisation of a fully autonomous UAV SAS. Therefore, this research adds value to current research in a number of ways:

- By critically reviewing current trends in the chosen domain.
- Build on this current research and identify areas that are lacking.
- Address these gaps by critically examining and providing empirical solutions.

1.4 Scope and Objectives

The FAA’s SAA Workshop [**2**] sets the scene for this dissertation. Figure 1.9 is a diagrammatic representation of the functionality required within a SAS. In this context, however, this thesis is only concerned with the development of the tracking functionality. Therefore this dissertation addresses the sub-function of **Track** and is illustrated by the blocked region in Figure 1.9.

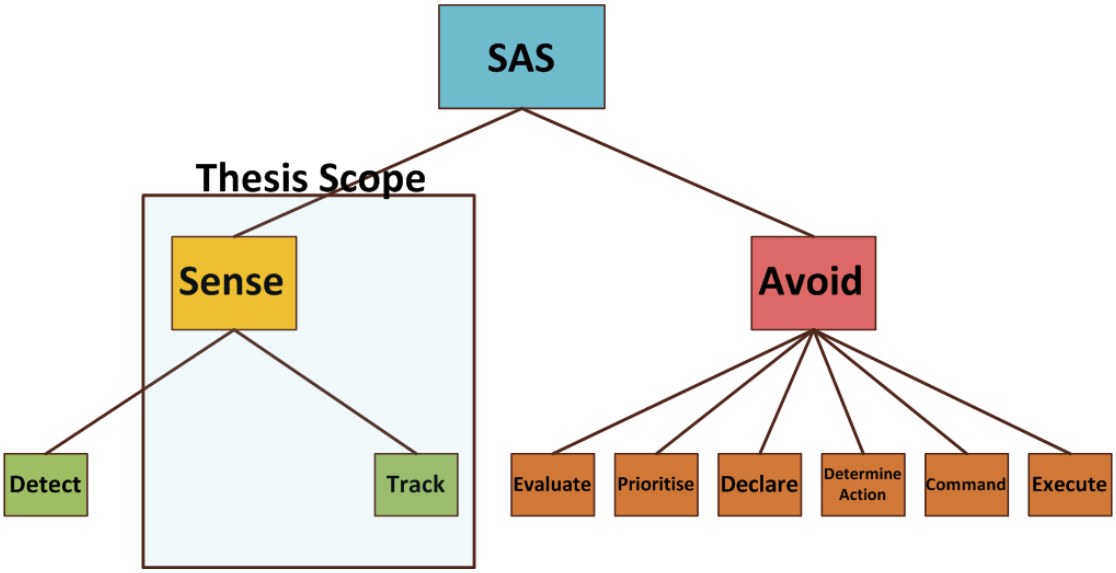

Figure 1.9: Thesis scope.

In this project a Doppler radar and an EO camera are available in order to perform target tracking. Thus, the overall aim of this research is to design an airborne tracking filter using multisensor data fusion. This leads to the following individual thesis objectives, with which to achieve this research aim:

1. Clarify the definition of multisensor data fusion.
2. To motivate the use the multiple, redundant sensors in a SAS.
3. As multisensor data fusion for target tracking is a non-trivial task, to provide a brief theoretical background on requisite tools and techniques in estimation and tracking.
4. Review current research on multisensor data fusion for target tracking.
5. Design and implement an airborne target tracking algorithm using multisensor data fusion.
6. Present performance metrics with which to evaluate the accuracy and performance of algorithms.
7. Demonstrate that the fusion filter more accurately tracks a target than a single sensor tracking filter.
8. Derive and discuss conclusions from the results acquired during algorithm development and knowledge gained from the literature study.

Through the completion of these objectives, the aim is to arrive at a suitable airborne tracking fusion architecture for a UAV SAS; of which the *Threat Tracking Unit* block, illustrated in Figure 1.10, will be composed.

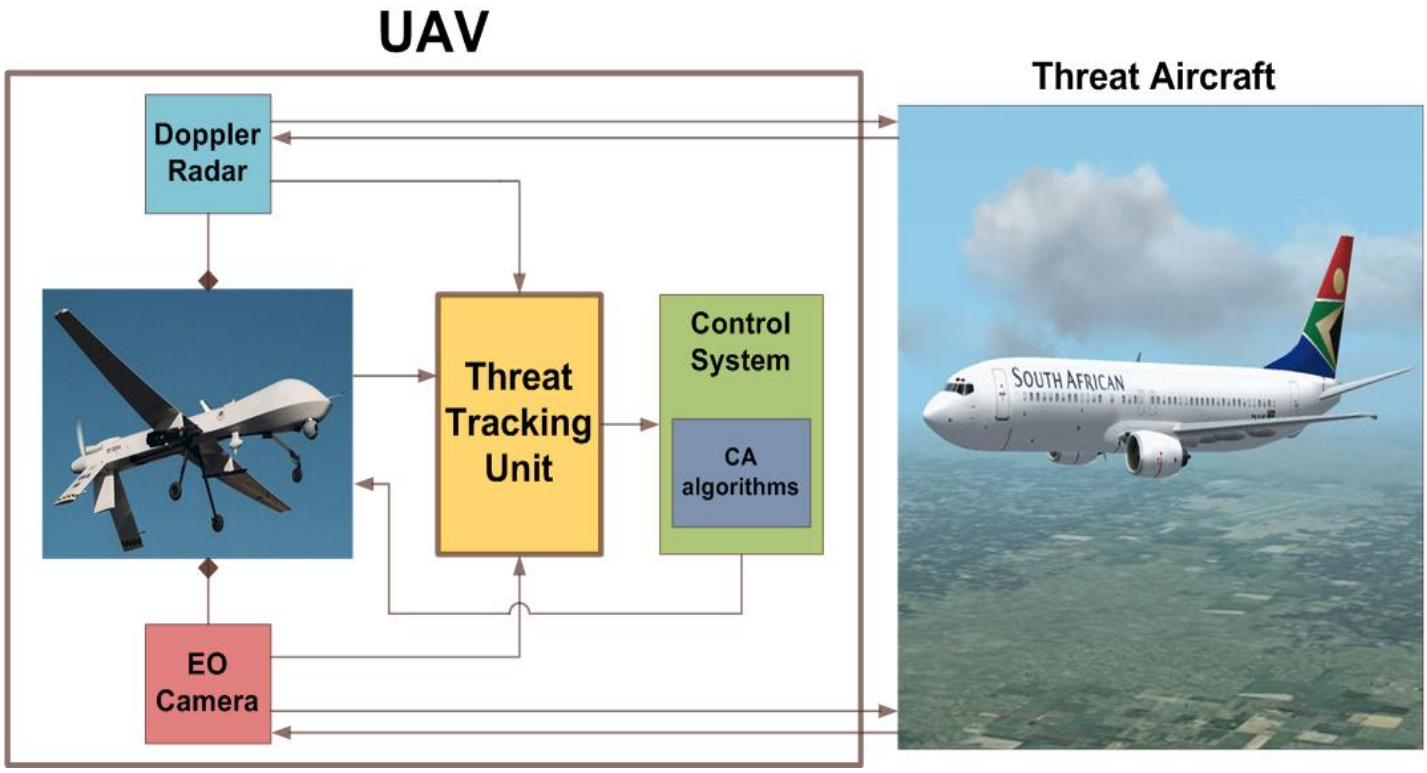

Figure 1.10: System overview. This thesis is concerned with the development of the TTU block. It is assumed that IMU and Actuator blocks exist inside the UAV picture.

Objective 1 to *objective 4* form the core of the Literature Review in Chapter 2. In order to achieve *objective 5*, the assumptions made during this study are:

- Two sensors are on-board the UAV platform: an EO camera and a Doppler radar.
- The EO camera provides its measurements to the SAS at a faster rate than the Doppler radar.
- A vision subsystem exists that detects the presence of other airborne threats and converts the detected image pixels to body-referenced target azimuth and elevation (using a pin-hole camera model).

- A radar subsystem exists that performs the necessary radar signal processing and converts the raw radar data to body-referenced target range, azimuth, elevation and range rate.
- Both sensors are fixed to the UAV centre of gravity.
- The UAV is modelled as a point mass travelling with constant velocity.
- The dynamics of threat aircraft trajectories obey a constant velocity model.
- Only one threat needs to be tracked at a time.
- There exists an Avoid subsystem whose inputs are the outputs from the tracker designed in this dissertation. The Avoid subsystem encompasses all the other SAA capabilities not in the scope of this thesis.
- Algorithms will not be tested on deployment hardware, but in simulation only. This is as a result of various risks present during flight tests.

1.5 Thesis Outline

Chapter 2 provides an extensive review of relevant literature pertaining to multisensor data fusion, estimation and target tracking. This chapter will form the foundation upon which the TTU algorithms will be designed.

Chapter 3 describes the design methodology adopted in this thesis.

Chapter 4 details the design of the simulation environment. This environment will be utilised extensively for the evaluation of the TTU algorithms.

Chapter 5 describes the detailed design of the TTU system. This entails the development of multiple airborne tracking filter architectures.

Chapter 6 consists of detailed comparisons of the various fusion architectures for the TTU system. These are discussed thoroughly and interpretations are made.

Chapter 7 concludes the thesis by unifying all the information obtained in the previous chapters.

Chapter 2: Literature Review

This literature review will clarify the term multisensor data fusion; motivate the need for multiple, redundant sensors and the advantages thereof; present and critically review present-day tracking filters[8]; and, when deemed necessary, provide a theoretical overview of mathematical tools and techniques. The literature review will draw to a close with a summary of topics covered and present the motivation for the algorithmic development to come.

The study within this review of literature focuses on objectives 1 to 4 as set out in the introductory chapter of this dissertation:

1. Clarify the definition of multisensor data fusion.
2. To motivate the use the multiple, redundant sensors in a SAS.
3. As multisensor data fusion for target tracking is a non-trivial task, to provide a brief theoretical background on requisite tools and techniques in estimation and tracking.
4. Review current research on multisensor data fusion for target tracking.
5. Design and implement an airborne target tracking algorithm using multisensor data fusion.
6. Present performance metrics with which to evaluate the accuracy and performance of algorithms.
7. Demonstrate that the fusion filter more accurately tracks a target than a single sensor tracking filter.
8. Derive and discuss conclusions from the results acquired during algorithm development and knowledge gained from the literature study.

Figure 2.1 illustrates how thesis objectives were translated into fields of study in the literature review. The reason for adopting this approach was to keep the literature review focused and to only discuss issues germane to satisfy the objectives of this thesis.

[8] More emphasis is placed on the utilization of traditional tracking filters in multisensor tracking filters.

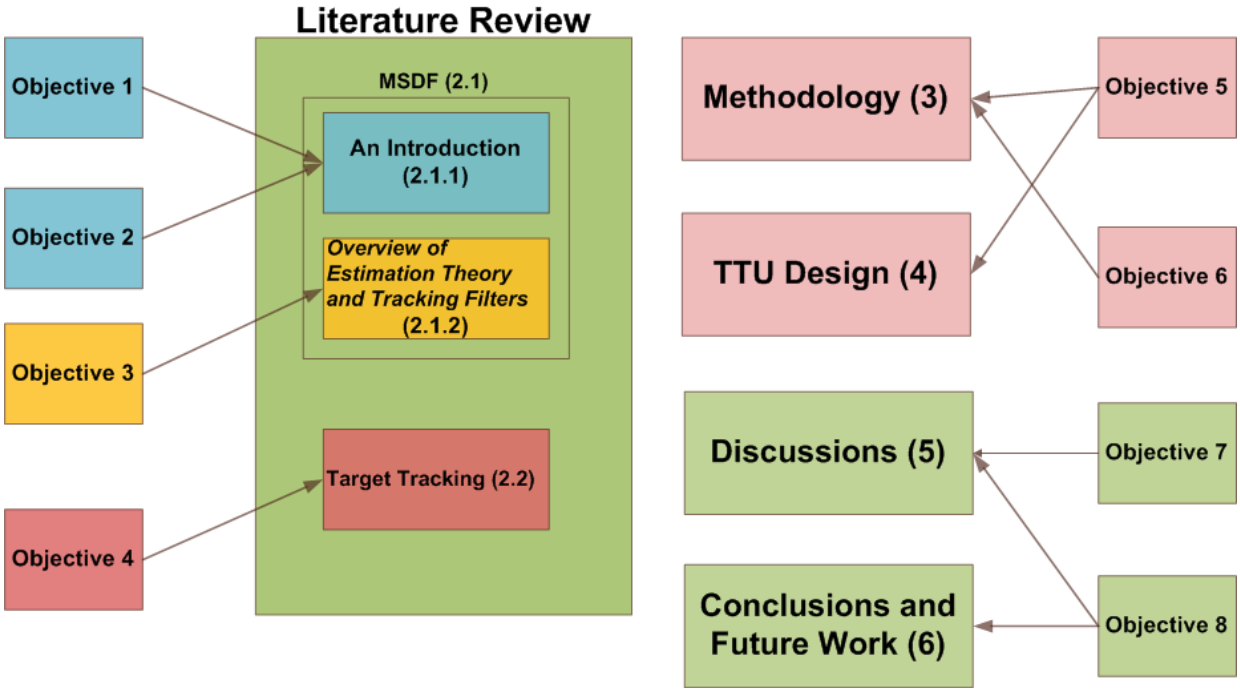

Figure 2.1: Mapping from thesis objectives to fields of study and, consequently, headings in the Literature Review. On the right, the mappings of other objectives are illustrated for the sake of completeness.

At the end of this literature review it is hoped that a thorough understanding of the literature and pertinent issues to target tracking are demonstrated. Furthermore, that the reader has a better understanding of these areas and that a clear justification for the algorithm development has emerged.

2.1 Multisensor Data Fusion

2.1.1 An Introduction

Multisensor data fusion (MSDF) seeks to combine data from multiple sensors to provide more meaningful information than from a single sensor [**4**]. This is analogous to the cognitive process evident in humans; we make decisions and draw conclusions based on a number of sensory inputs. MSDF may even provide information that is not possible with a single sensor. Moreover, MSDF allows for the detection, isolation and recovery of sensor failures [**12**] [**13**].

It is widely considered that Waltz and Llinas published the seminal text on MSDF in 1990; in which they explain that '[MSDF] involves significant integration of a number of research disciplines' [**14**]. Furthermore, data fusion spans a variety of applications. Some of these well suited applications are air-to-air and surface-to-air defence [**15**] [**16**], maritime surveillance [**17**] [**18**] and remote sensing [**19**] [**20**]. In [**21**] an airborne surveillance and tracking system is proposed which utilises MSDF; this system stands out from other designs, employing data fusion, as it consists of a particularly complex sensor suite, viz. a radar, an Identification Friend or Foe (IFF) system, an Electronic Support Measures (ESM) system, a Spotlight Synthetic Aperture Radar (SSAR) and a FLIR sensor. An example of an airborne platform with a similar sensor suite is illustrated in Figure 2.2. In order to understand the performance gains possible from utilising multiple, redundant sensors in a system, such as in [**21**], let us first consider the advantages of individual sensors.

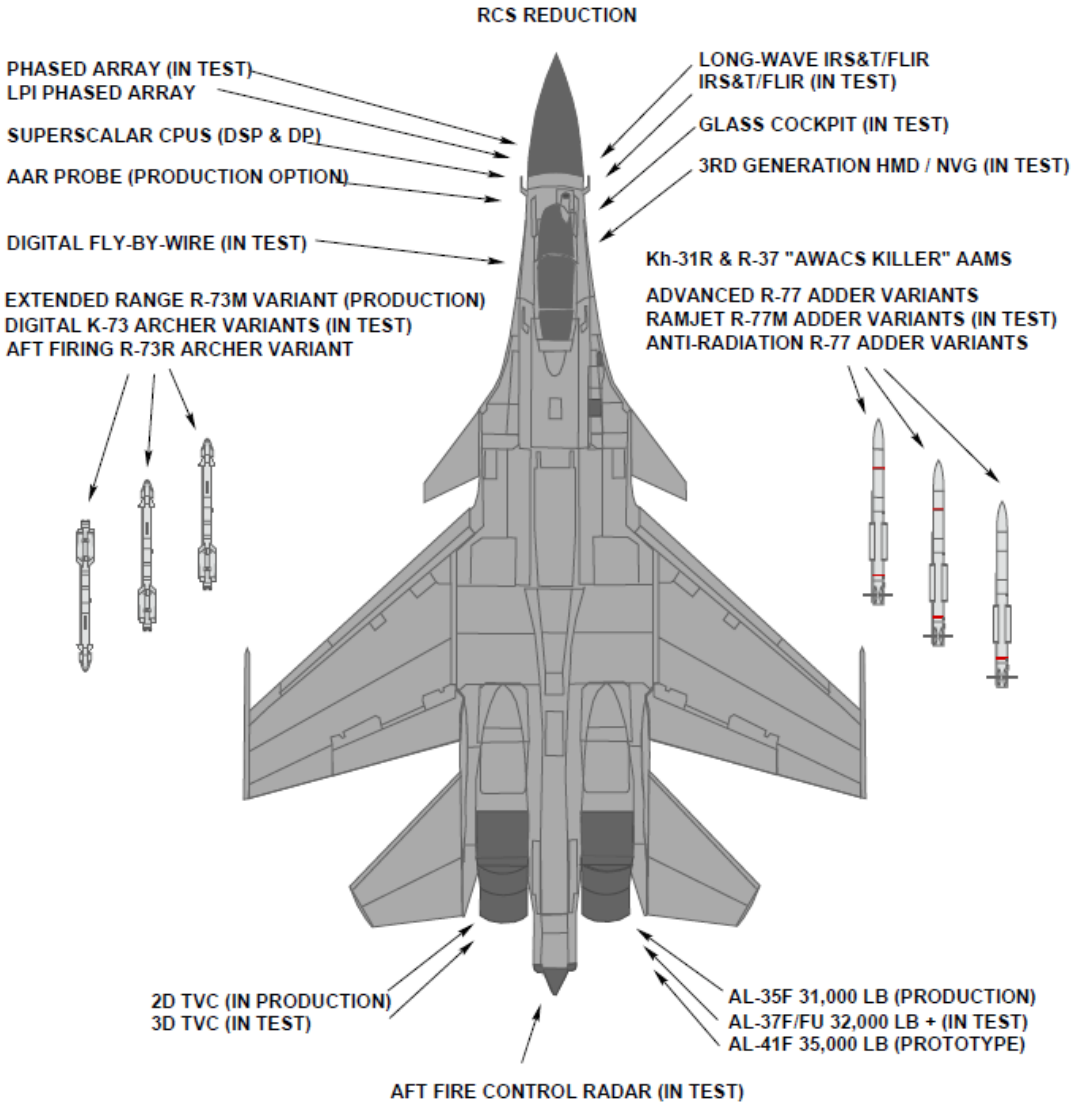

Figure 2.2: Fighter aircraft with radar and IR sensors. MSDF is the enabling technology to fully reap the benefits of having multiple, redundant sensors as part of a system [**11**].

For the purpose of this explanation (and to motivate our premise), let's review a radar and an EO sensor. By analysing their individual advantages and disadvantages, we would be in good stead in understanding the implications of their presence in a MSDF tracking system:

1. Radar is an active sensor[9], whereas an EO sensor is a passive one.
2. An EO sensor is heavily dependent on atmospheric effects, including fog, rain and haze. [**22**]. This is especially relevant as the average

[9] An active sensor is one that emits electromagnetic radiation.

global annual cloud cover is circa. 61 per cent [**23**] (see Figure 2.3). Radar, on the other hand, is an all-day, all-weather sensor.

3. Sensors have inherent limitations on their sampling frequencies. An EO camera has a data rate that is faster than that of radar.
4. An EO sensor has greater angular resolution than radar. Radar, generally, has a greater effective range in which it can detect targets.
5. A radar measurement has a higher dimensionality than that of an EO sensor.

The effects of the pros and cons of each sensor, listed above, would have the following effects on a MSDF system:

1. The EO sensor would be able to detect targets but the radar would not during a fade in the radar return as a result of radar cross section scintillation [**24**], or other electronic countermeasures (ECM). This implies that when one subsystem stops operating that the overall tracking system can continue to be operational, thus making the tracking system more robust.
2. The tracking system would have all-day, all-weather operation. Weather, including cloud cover and rain, is a critical factor when selecting sensors.
3. An increased overall data rate results.
4. The system utilises the more accurate measurements from either sensor, improving overall system accuracy.
5. The addition of radar measurements leads to state observability [**25**].

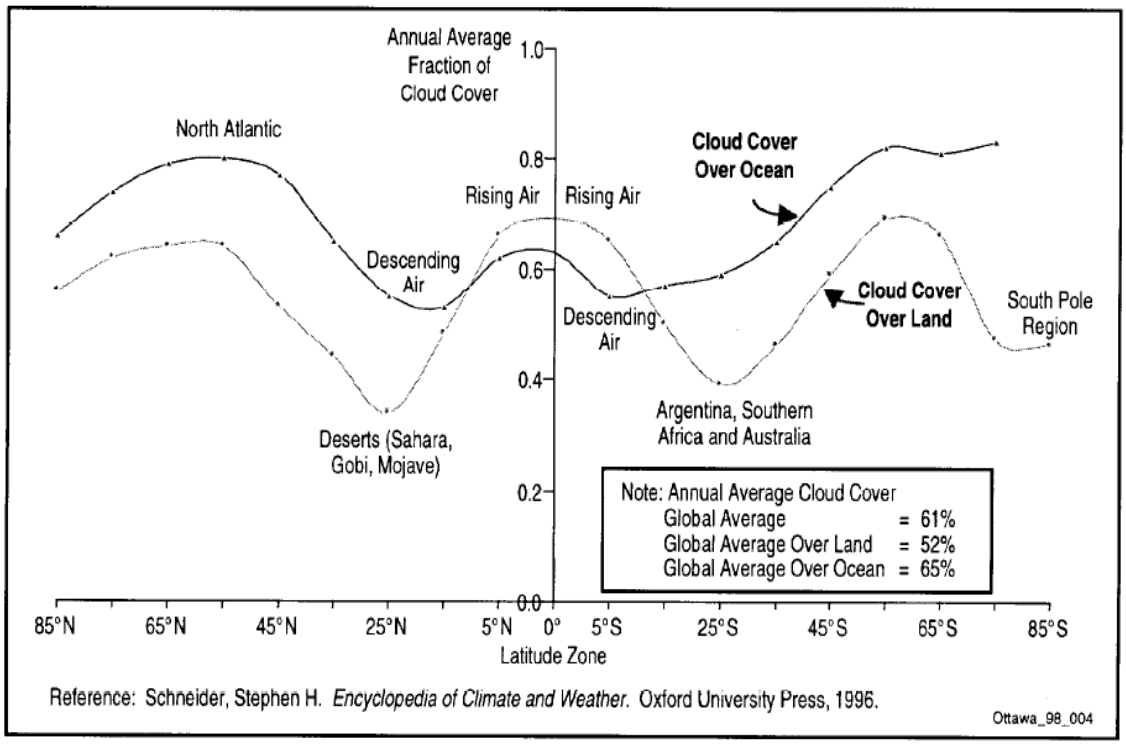

Figure 2.3: Cloud cover around the globe [**23**].

A further motivation for the need of MSDF is that, in our technologically-driven times, ECM systems are becoming more sophisticated and, therefore, the future will see more of the onus placed on passive sensors and algorithms. Moreover, advances in stealth technology, over the last two decades, have resulted in significant reductions in radar and IR signatures [**11**].

MSDF is the enabling technology with which the best attributes of both sensors, reviewed above, can be harvested. Therefore, in light of the above, the advantages of MSDF can be summarised in Table 2.1 below, adapted from [**4**].

Table 2.1: Advantages of MSDF.

Category of Advantage	**Advantage**	**Operational Advantage**
Improved system reliability	Multiple sensor systems have an inherent redundancy.	Graceful degradations.
Extended spatial coverage	That is, one sensor can look where another cannot.	Improved probability of detection.
Extended temporal coverage.	One sensor can detect when another cannot.	Robust system. E.g. less vulnerable to ECMs.

It is worth noting that, despite the advantages listed above, Hall and Steinberg [**26**] warn that the following precepts should be kept in mind when implementing a MSDF system:

- There is no substitute for a good sensor.
- Data fusion cannot correct errors in processing of individual sensor data.
- MSDF can result in poor performance if incorrect information about sensor performance is used.
- There is no such thing as a perfect data fusion algorithm.

There are three types of fusion architectures: centralised; distributed and hybrid. The architecture depends on the data that arrives at the fusion centre (FC) i.e. raw sensor data implies a centralised architecture (see Figure 2.4); processed data or state estimates implies a distributed architecture (see Figure 2.5); and a combination of data results in a hybrid architecture. Hybrid architectures are computationally intensive and not suitable for practical implementations. Therefore, they are not considered in the literature.

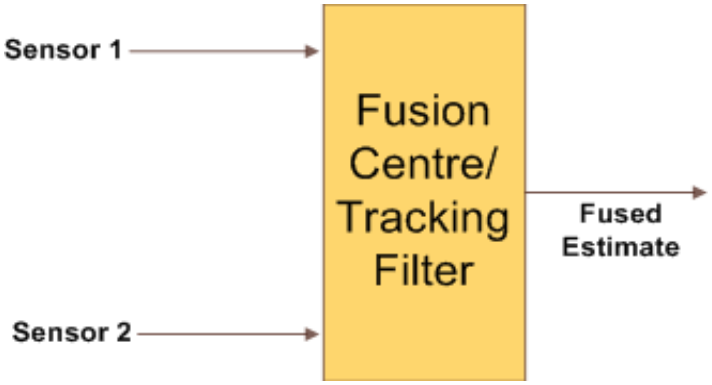

Figure 2.4: Centralised fusion architecture.

Centralised architectures imply raw sensor data arrives at the FC. Here there are three methods with which data can be fused in the FC:

- Measurement fusion method; based on minimising the mean square error [**27**].
- Parallel fusion [**28**].
- Sequential fusion.

Assume two sensors i and j with measurement equations z_k^i and z_k^j, respectively. When making use of the measurement fusion method, the fused measurement z_k^F at time instant k is then:

$$z_k^F = z_k^i + R_k^i\left(R_k^i + R_k^j\right)^{-1}\left(z_k^j - z_k^i\right). \tag{2.1}$$

Where R_k^i and R_k^j are the covariance matrices of z_k^i and z_k^j, respectively. The fused covariance matrix is determined by:

$$R_k^F = \left({R_k^i}^{-1} + {R_k^j}^{-1}\right)^{-1}. \tag{2.2}$$

These fused measurements are then tracked in order to produce an estimate of a target's state vector.

The second method of performing centralised fusion is parallel fusion and is the most intuitive one of the three as multiple sensors can be visualised as one. It involves combining all the observation vectors (from multiple sensors) and measurement matrices into larger composite ones. The composite observation becomes [**28**]:

$$z_k^F = \begin{bmatrix} {z_k^i}^T & {z_k^j}^T \end{bmatrix}^T. \tag{2.3}$$

With a measurement matrix:

$$H_k^F = \begin{bmatrix} {H_k^i}^T & {H_k^j}^T \end{bmatrix}^T. \tag{2.4}$$

And composite measurement covariance:

$$R_k^F = diag\left(\begin{bmatrix} R_k^i & R_k^j \end{bmatrix}\right). \tag{2.5}$$

These new composite matrices affect only the *corrector* stage of the Kalman Filter (KF) equations; and are used as such to track the state vector of a target. In principle, parallel fusion can accommodate any number of sensors. However, the computational resources required quickly multiply as sensors are added to the system. This point is evident by analysing the matrix dimensions of the composite matrices formed. Consider a state vector of dimension n, x observation vectors of dimension m^x with measurement matrices (H) of dimension $m^x \times n$. The composite observation vector will then have dimension $m = \sum_{i=1}^{x} m^x$ and the composite measurement matrix will

have dimension $m \times n$. As an example, if $n = 6, x = 2\ and\ m^x = 3$ then the dimension of composite H is 6×6. This increase in matrix dimensions is a challenge as H and R are involved in the computationally-expensive matrix inverse operation each time-step. Therefore, one can see that as sensors are added to the system the viability of this approach lessens.

The other method of performing centralised fusion is sequential fusion. Here each sensor's observation vector is sequentially used to update the state estimate. That implies that the KF *corrector* stage is repeated for as many times as there are sensors in the system; at each iteration using the state and covariance estimates produced from utilising the previous sensor's observation vector. Once all observation vectors have been processed the KF continues its normal filter operation. Sequential fusion is also very computationally expensive as it requires the computation of a new gain matrix and prediction every time-step for every observation from every sensor in the system.

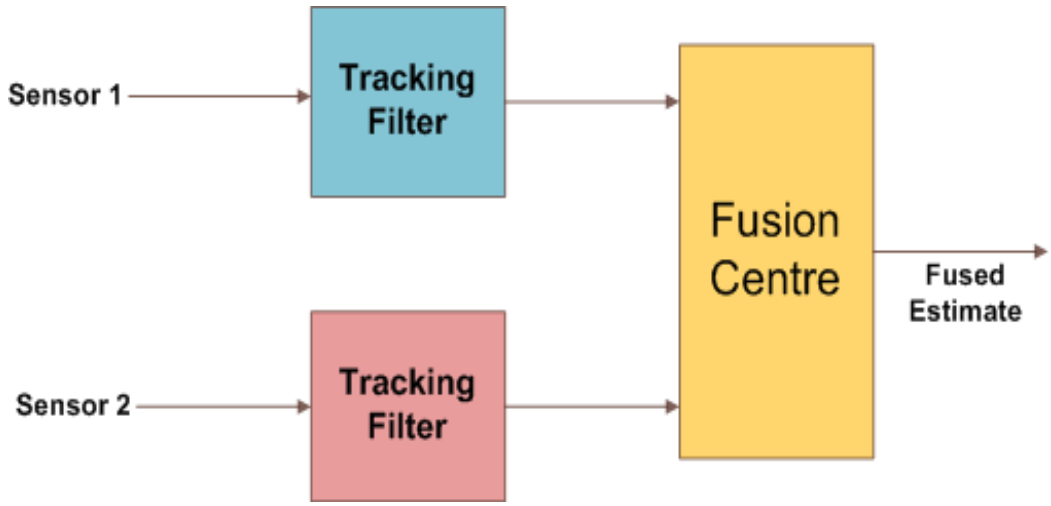

Figure 2.5: Distributed fusion architecture.

Distributed architectures, as its name suggests, are distributed in nature. There exists an independent tracking filter for each sensor in a system, illustrated in Figure 2.5 above. Each tracking filter produces its own local state

estimate and these state estimates are fused using the state vector method of fusion (an example of a distributed fusion architecture is illustrated in Figure 2.6). The fused covariance and state estimate are evaluated as follows [**29**]:

$$P_{k|k}^{F} = \left({P_{k|k}^{i}}^{-1} + {P_{k|k}^{j}}^{-1} \right)^{-1} \tag{2.6}$$

and

$$\hat{X}_{k|k}^{F} = P_{k|k}^{F} \left({P_{k|k}^{i}}^{-1} \hat{X}_{k|k}^{i} + {P_{k|k}^{j}}^{-1} \hat{X}_{k|k}^{j} \right). \tag{2.7}$$

Where $\hat{X}_{k|k}^{i}$, $\hat{X}_{k|k}^{j}$ are the posterior state estimates produced from each local tracking filter and $P_{k|k}^{i}$, $P_{k|k}^{j}$ are their covariance matrices.

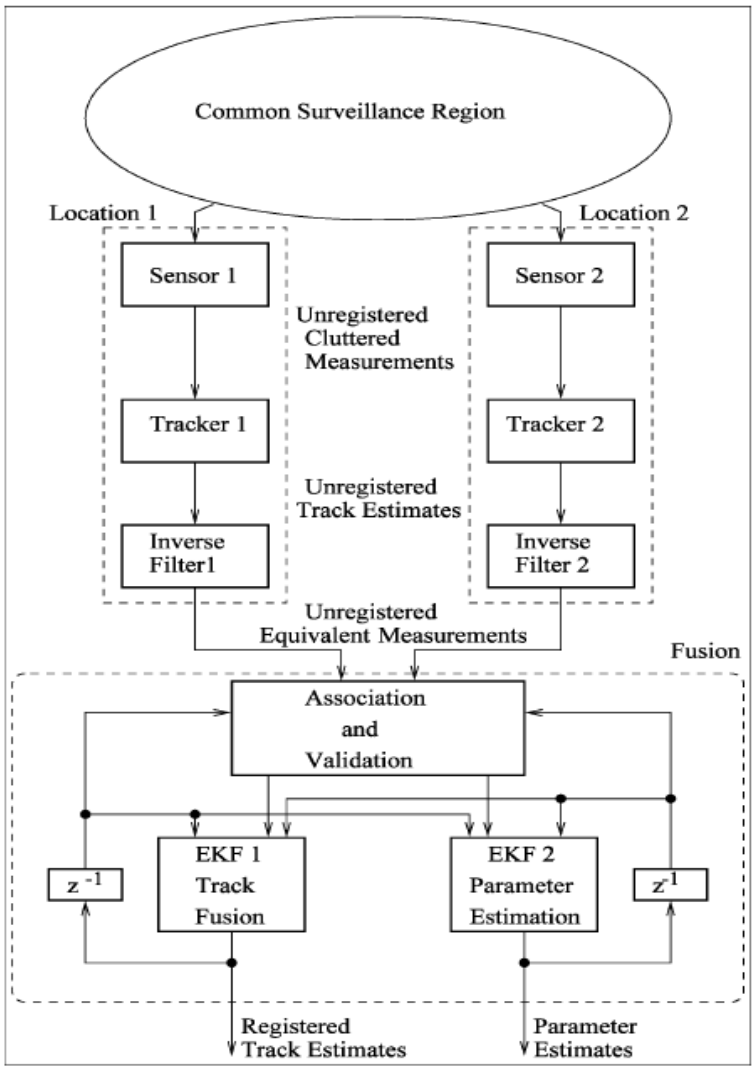

Figure 2.6: An example distributed fusion architecture application.

A common assumption made in distributed architectures is that local filter estimates are uncorrelated [**29**] [**30**] [**31**]. Multiple fusion architecture are developed in [**30**] to alleviate this problem. The reason for the aforementioned assumption is that when accounted for, there is a significant increase in filter complexity. The implications of a common process model used by each local filter are twofold: it first and foremost makes it feasible to fuse the individual state estimates; and second, it causes the individual state estimates to be correlated [**32**].

Equations (2.6) and (2.7) hold only under the aforementioned assumption. If local estimates are correlated, then this correlation needs to be accounted for in the filter in order to fuse the local estimates. This is accomplished by calculating the cross-covariance $P^{ij} = P^{ji^T}$ between the two state estimates and then using this to fuse them. The predicted cross-covariance is [**32**]:

$$P^{ij}_{k|k-1} = \Phi P^{ij}_{k-1|k-1}\Phi^{'} + GQG'. \tag{2.8}$$

And the estimate cross-covariance is:

$$P^{ij}_{k|k} = \left[I - K^i_k H^i_k\right] P^{ij}_{k|k-1} \left[I - K^j_k H^j_k\right]^T. \tag{2.9}$$

Equations (2.12) and (2.13) provide a recursive means of calculating the cross-covariance between $\hat{X}^i_{k|k}$ and $\hat{X}^j_{k|k}$. The fused estimate and covariance matrix are determined by the following equations:

$$\begin{aligned} \hat{X}^F_{k|k} = \hat{X}^i_{k|k} - \Big(P^i_{k|k} & \\ - P^{ij}_{k|k}\Big)\Big(P^i_{k|k} + P^j_{k|k} - P^{ij}_{k|k} & \\ - P^{ij\ T}_{k|k}\Big)^{-1}\left(\hat{X}^j_{k|k} - \hat{X}^i_{k|k}\right) & \end{aligned} \tag{2.10}$$

and

$$P_{k|k}^{F} = P_{k|k}^{i} - \left(P_{k|k}^{i} - P_{k|k}^{ij}\right)\left(P_{k|k}^{i} + P_{k|k}^{j} - P_{k|k}^{ij} - P_{k|k}^{ij\ T}\right)^{-1}\left(P_{k|k}^{i} - P_{k|k}^{i\ T}\right). \quad (2.11)$$

The reader should note that, according to the assumptions and objectives laid out in Chapter 1, the scope of this thesis is limited to single target tracking. As such, data association techniques are not included here. An excellent survey on data association for MSDF applications can be found in [**4**].

In summation of this introduction to MSDF, the choice of fusion architecture is dependent on the application at hand as well as the hardware and software limitations imposed on the system. Distributed fusion is a much less mature area of research compared to centralised fusion, which has seen much research [**33**]. Distributed architectures require less computational resources [**34**] and exhibits higher survivability[10], but centralised architectures (through measurement fusion) result in more accurate tracking filters [**35**]. Furthermore, measurement fusion is the most widely used method when performing centralised fusion. An important consideration when selecting fusion methods is matrix dimensions; centralised fusion methods require that observation vectors from all the sensors, present in the system, have the same dimension and hence their measurement matrices. Whereas in distributed fusion, the state vector from all independent tracking filters are required to have the same dimensions.

[10] The entire tracking system which employs a centralised fusion architecture has a single point of failure – the FC. The reader is reminded that in a centralised architecture the FC and tracking filter are one and the same.

2.1.2 Overview of Estimation Theory and Tracking Filters

Tracking using MSDF is a non-trivial task, which spans a number of engineering disciplines. In what follows, this subsection seeks to provide a background on mathematical techniques used in estimation and the realisation of tracking filters. The work presented in this section is used later in this thesis.

State estimation has its roots in least squares estimation (LSE). Invented in 1795 by Gauss to obtain the best estimate of orbital parameters for asteroids [**4**], LSE estimates the state vector using a batch processing technique where multiple measurements (over multiple sensor returns) are stored and then processed simultaneously [**36**]. This LSE method can be extended into a nonlinear case as well as a recursive form. The recursive LSE approach provides an updated state estimate after each sensor return, i.e. it does not store successive measurements. A least squares estimator is similar to a Kalman Filter in that they both minimise the expected mean squared error between the actual and estimated states; they are also dissimilar in that the LSE assumes deterministic state dynamics. As the reader may have deduced already, the recursive form of LSE has led to the innovation of the ubiquitous KF.

The KF is an extension to the least squares estimator; it incorporates process noise which allows for the modelling of random target motion. It is a linear estimator that, as long as the target dynamics and measurement noise are accurately modelled and the system is Gaussian, minimises the mean squared error. The KF provides a convenient measure of estimation accuracy through its state covariance matrix [**25**]. Furthermore, the KF has a number of useful features pertinent to MSDF applications [**4**]:

- A wide variety of measurement models (observation equation) and process/plant models (state equation) can be modelled.
- Quantitative evaluation of the contribution of each sensor to overall estimation accuracy.

- It is robust and fairly simple to implement.

In what follows, the KF algorithm is described and its equations are presented; the reader is directed to [**37**] for the detailed mathematical derivations of the KF equations and discussions of its applications. Consider a discrete-time target dynamics process which is modelled in the discrete-time Markov form [**38**]:

$$X_{k+1} = \Phi X_k + \Gamma w_k + u_{k+1|k}. \tag{2.12}$$

Where X_k is the $n - dimensional$ target state vector to be estimated. w_k is the zero-mean, white, Gaussian process noise[11] with covariance Q. Φ is the known state transition matrix, and $u_{k+1|k}$ is the known deterministic input.

Equation (2.12) describes the target dynamics in terms of a Markov process[12] represented by the state equation.

Sensor measurements arrive as linear combination of the state variables:

$$z_k = HX_k + v_k. \tag{2.13}$$

Where z_k is the $m - dimensional$ observation vector. v_k is the zero-mean, white, Gaussian measurement noise with covariance R. H is the $m \times n$ measurement matrix.

[11] An example of process noise when modelling a target is its random accelerations.
[12] A Markov process is a process in which its statistical representation in the future is determined solely by the present state, i.e. independent of the past.

Given Equations (2.12) and (2.13) the KF *predictor* and *corrector* stages are [**25**]:

a) Corrector

$$K_k = P_{k|k-1}H'\left(HP_{k|k-1}H' + R\right)^{-1} \tag{2.14}$$

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k\left(z_k - H\hat{X}_{k|k-1}\right) \tag{2.15}$$

$$P_{k|k} = \left(I - K_k H\right)P_{k|k-1} \tag{2.16}$$

b) Predictor

$$\hat{X}_{k+1|k} = \Phi\hat{X}_{k|k} + u_{k+1|k} \tag{2.17}$$

$$P_{k+1|k} = \Phi P_{k|k}\Phi' + Q \tag{2.18}$$

Here K_k is the Kalman gain and I is the Identity Matrix. For future reference, the dimensions of the KF matrices are summarized in Table 2.2 below.

The *predictor* stage (also known as the time update) defined by Equations (2.17) and (2.18) serves to propagate in time the estimated state vector $\hat{X}_{k|k}$ and covariance $P_{k|k}$. The *corrector* stage (also known as the measurement update) defined by Equations (2.14) - (2.16), provides feedback into the system by incorporating the current measurement z_k into the *a priori* state estimate $\hat{X}_{k|k-1}$ to yield an improved *a posteriori* state estimate $\hat{X}_{k|k}$.

In (2.15) the factor $\left(z_k - H\hat{X}_{k|k-1}\right)$ is known as the innovation (or residual) vector with a corresponding innovation covariance matrix defined by the factor $(HP_{k|k-1}H' + R)$. The innovation is an important quantity as it provides insight into the performance of the filter by giving us a measure of how well the filter is performing.

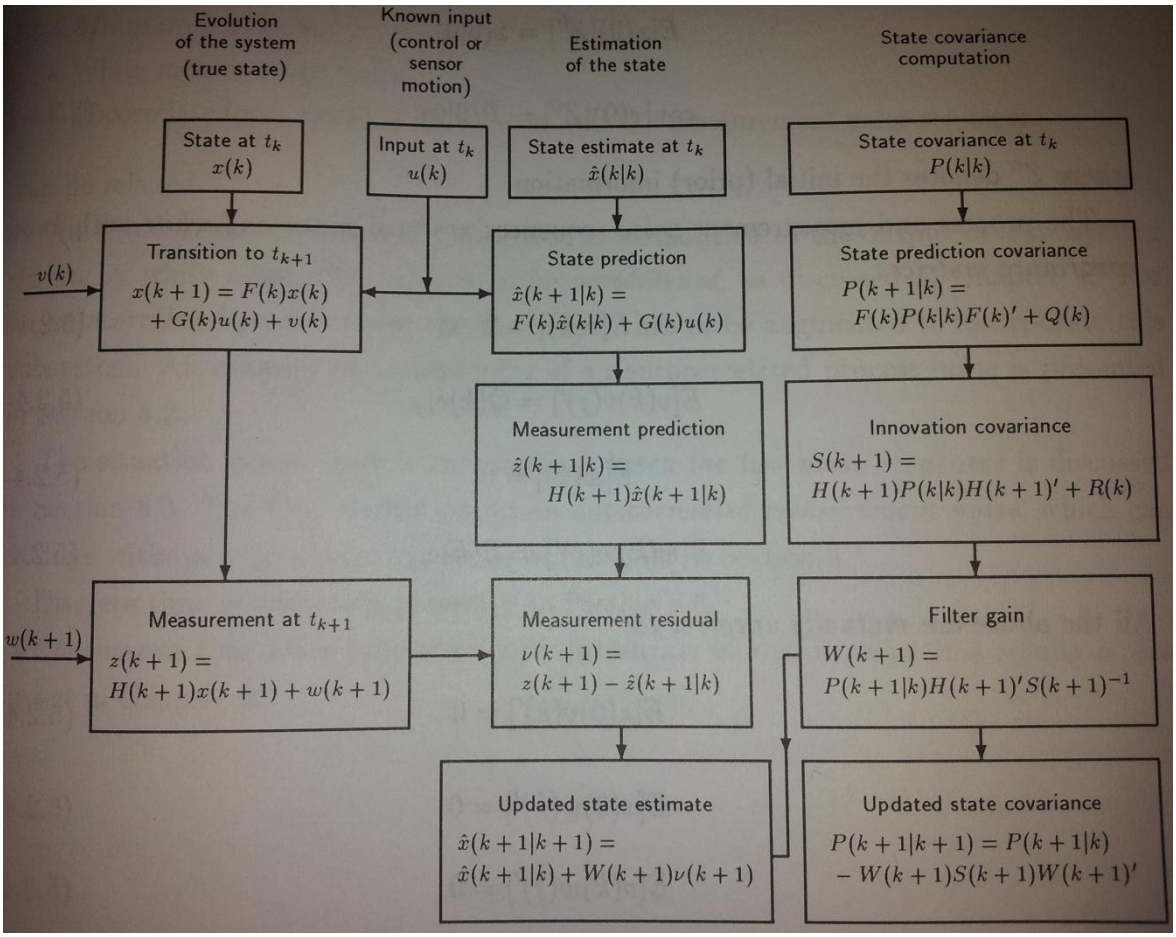

Figure 2.7: Single step Kalman filter overview obtained from page 219 of [**39**].

The Kalman gain in (2.14) tells us that as the measurement covariance R tends to zero, the real measurement is trusted more and, conversely, the predicted measurement $H\hat{X}_{k|k-1}$ is trusted less. This is in direct contrast to the way the *a priori* state covariance $P_{k|k-1}$ affects the innovation covariance; as $P_{k|k-1}$ tends to zero the real measurement is trusted less and, conversely, the predicted measurement is trusted more. Therefore the Kalman gain

dynamically weights either the real or predicted measurement when determining the posterior estimate, dependent on both the state and measurement covariance matrices and hence the innovation covariance. A graphical interpretation of the KF algorithm is displayed in Figure 2.7.

Table 2.2: KF matrix dimensions.

Matrix		**Dimensionality**
Name	**Designator**	
State vector	X	$n \times 1$
State covariance	P	$n \times n$
Process noise covariance	Q	$q \times q$
Transition	Φ	$n \times n$
Noise	Γ	$n \times q$
Observation vector	z	$m \times 1$
Measurement	H	$m \times n$
Measurement noise covariance	R	$m \times m$

Before delving into the realm of nonlinear filters, the system designer has other options to handle nonlinear systems. One solution is the Pseudo Linear Kalman Filter (PLKF). Another option is a new filter developed in [**40**] (based on the KF); it accommodates nonlinear systems yet it does not require any form of linearization process. The PLKF transforms a nonlinear measurement equation into a more convenient linear form [**25**], negating the need for a nonlinear filter. Consider a sensor measurement as that in Equation (2.13),

but the measurement matrix H is replaced with $h(X)$ as the measurement equation is nonlinear. Through algebraic manipulation, the pseudo measurement is expressed as:

$$z_s = H\big(h(X)\big)X = 0. \tag{2.19}$$

Where $H\big(h(X)\big) = H_s(z)$.

Then the standard KF equations (Equations (2.14) - (2.18)) are used to estimate the state vector with H_s replacing H, except for two notable differences:

i. The residual in Equation (2.15) is replaced by the factor:

$$\big(0 - H_s(z)\hat{X}_{k|k-1}\big). \tag{2.20}$$

ii. R_s replaces R, with:

$$R_s = gRg'. \tag{2.21}$$

Where g is found by taking the partial derivateive of the pseudo measurement vector with respect to the actual measurement vector:

$$g = \frac{\partial z_s}{\partial z}. \tag{2.22}$$

When state dynamics or measurement models are no longer Gaussian in nature, standard KF equations do not hold true. That is the target dynamics or the measurement matrix is nonlinear. This implies that the transition matrix or the measurement matrix would be a nonlinear combination of the state variables. Under these conditions the Extended Kalman Filter (EKF) becomes

applicable. The EKF is a nonlinear extension of its linear counterpart: the KF; which is apparent from Equations (2.23) - (2.25). Historically speaking, the practical implementation of EKFs was limited by hardware platforms of the era. This is asserted by the fact that the EKF is significantly more computationally expensive than its linear equivalent. The *predictor* and *corrector* stages of the EKF are:

a) Corrector

$$K_k = P_{k|k-1} H_X(k)' \left(H_X(k) P_{k|k-1} H_X(k)' + R \right)^{-1} \quad (2.23)$$

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k \left(z_k - h(\hat{X}_{k|k-1}) \right) \quad (2.24)$$

$$P_{k|k} = \left(I - K_k H_X(k) \right) P_{k|k-1} \quad (2.25)$$

b) Predictor

$$\hat{X}_{k+1|k} = \Phi \hat{X}_{k|k} + u_{k+1|k} \quad (2.26)$$

$$P_{k+1|k} = \Phi P_{k|k} \Phi' + Q \quad (2.27)$$

It is evident that the *predictor* stages for the KF and the EKF are the same, but *corrector* stages differ. In the KF case the observation vector was a linear relation of the state variables, hence the measurement matrix was linear in nature. In this scenario the observation vector is a nonlinear function of the state vector, i.e. $h(X)$. According to the assumptions around which the EKF are built, the error dynamics can be accurately characterised by a linearized first-order Taylor series expansion and disregarding the second and higher-

order terms [**41**] [**42**]. Therefore, by performing a Taylor series expansion of this $h(X)$ and evaluating this about the predicted estimate, we arrive at the linearized measurement matrix [**25**]:

$$H_X\,(k) = \left.\frac{\partial h(X)}{\partial X}\right|_{X=\hat{X}_{k|k-1}} . \tag{2.28}$$

In the preceding EKF example, it was assumed that the system to be estimated consisted of a nonlinear measurement process only. If the target dynamics were, instead, nonlinear, then only the *predictor* stage of the EKF would change. In this scenario the transition matrix is a nonlinear function of the state variables. The resulting *predictor* equations would be determined by the linearization of the transition matrix.

Through the linearization of the observation, or state models, errors are introduced into the system and, to quote Denham and Pines [**43**], 'invalidates all properties of optimality and convergence of the KF'. Furthermore the computational processing of the EKF equations requires significantly more resources; this is evident from the juxtaposition of measurement matrices of the KF and EKF. In the KF the measurement matrix is constant, whereas in the EKF the linearized measurement matrix is evaluated at every time epoch. Furthermore, the predicted measurement calculation first involves evaluation and then matrix multiplication.

For highly nonlinear systems, there are other more complex filters that can be used. These filters seek to alleviate the inherent errors present in the EKF due to the linearization process, at the expense of even more computing resources. Some of these filters include used in tracking applications are: Iterated EKF, Second order EKF [**44**], Unscented KF (UKF) [**44**] [**45**] and Particle Filter [**46**]. Interacting Multiple Model (IMM) Filters are typically used to track manoeuvring targets. This is achieved through the use of multiple KF models, one for each target manoeuvre stage. Then the individual state

estimates and covariance matrices (from each KF) are combined through the use of a Markov model for the transition between manoeuvre stages [**39**].

The reader should note that in recent years it has been proven that the use of Equation (2.25) should be avoided [**47**]. According to Li and Jilkov [**47**], the reason for this is twofold: first, it invites horrible numerical problems; and second, it is theoretically valid only when the Kalman gain K is truly optimal (which, in practice, is rarely the case). The new equation is given as:

$$P_{k|k} = \big(I - K_k\ H_X(k)\big)P_{k|k-1}\big(I - K_k\ H_X(k)\big) + K_k\ RK_k'. \tag{2.29}$$

In summary, this subsection provided a clear definition of MSDF, conveyed some of its applications and motivated the need for multiple, redundant sensors. Following this, the reader was equipped with estimation and KF theory required in order perform MSDF and realise the filters developed in Chapter 5. The above satisfies thesis objectives 3 and 4 laid out in section 1.4.

2.2 Target Tracking

Tracking is the process of determining a target's quantities of interest (state vector), including target position and velocity, under the influence of noisy sensor measurements. As outlined in Section 2.1.2, when a target's dynamic and measurement processes are assumed Gaussian then a linear filter, for example the Kalman filter, can be used to estimate the state vector of interest. Blackman and Popoli [**25**] state that, in this case, the "estimation of the mean target state and the associated covariance matrix is all that is required to define the probability density function (PDF) associated with the target position in state space". If, on the other hand, this Gaussian assumption is not accurate, such as when the target dynamics or measurement process is nonlinear, then a nonlinear filter becomes necessary. Blackman and Popoli

[**25**] go further where they declare that, in this case, it is better to propagate the PDF directly in target state space.

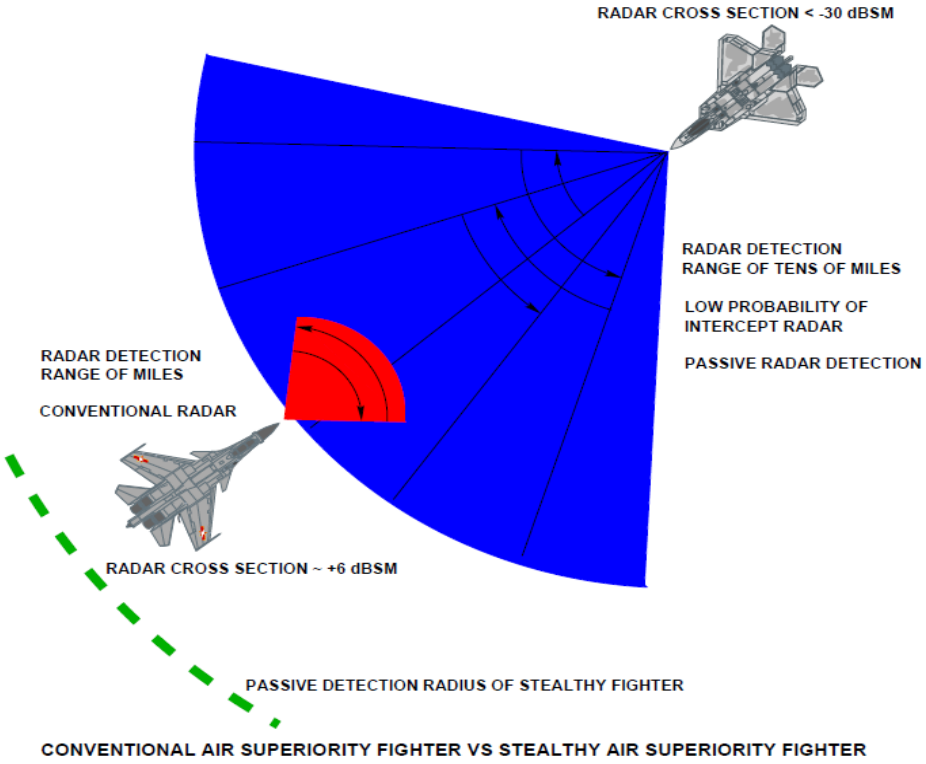

Figure 2.8: Airborne radar in action [**11**].

One of the first choices faced by the system designer when designing a tracking filter[13] is selecting the coordinate system in which to track. Under the assumptions laid out in section 1.4 of this thesis, this literature study is constrained, for the most part, by the presence of these two sensors as part of the tracking system. As such, in order to track a target, one could use one of the following sensor configurations:

- Bearings-only (camera)
- Doppler radar
- Multiple sensors.

13 A tracking filter is so called as it **tracks** random variables in the state vector by **filtering** out noise from multiple sources. Moreover, the process of finding the best estimate from noisy data amounts to filtering out the noise.

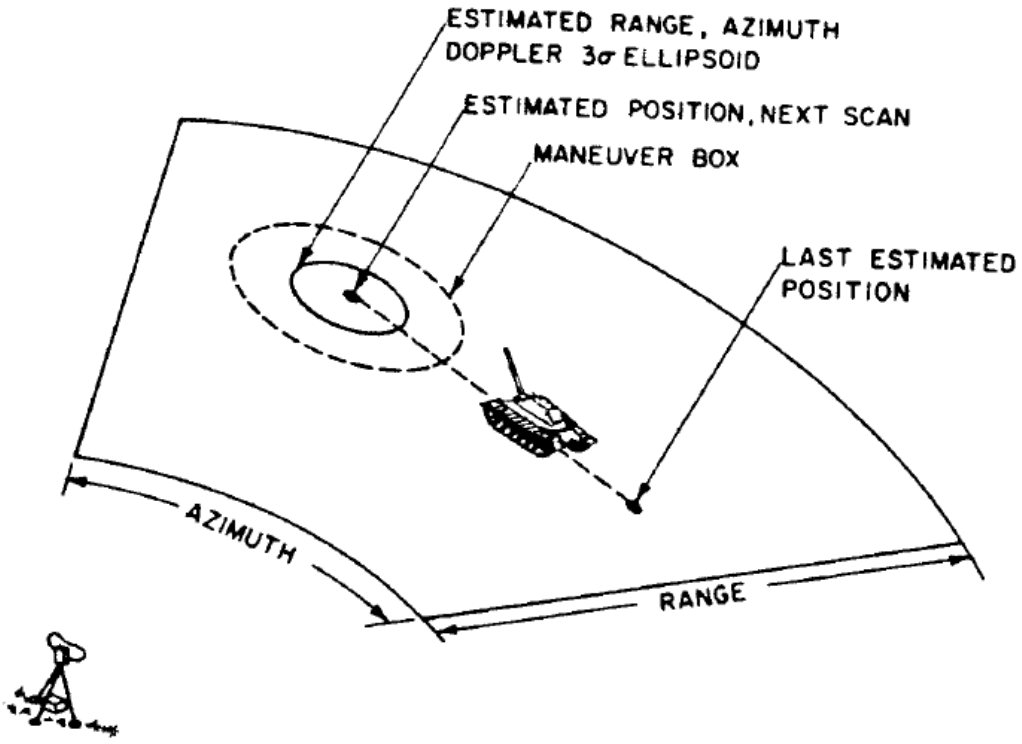

Figure 2.9: Ground-based radar target tracking [**36**].

In light of the above, this section will proceed in a logical fashion. It will provide an overview of tracking with each sensor leading up to, ultimately, tracking with multiple, redundant sensors.

2.2.1 Selecting a Tracking Coordinate System

All tracking systems require a frame of reference in which to track targets. The choice of which is not usually a straightforward task. It must be one that is meticulously thought out in order to make the most informed decision. Much like the selection of a fusion architecture (described in Section 2.1), the selection of a coordinate system in which to perform tracking is heavily dependent on the application at hand [**25**]. In the end of this section, it will be clear that the choice of tracking coordinate system plays a significant role in tracker performance.

The North-East-Down (NED) system is the best choice if multiple targets are to be tracked as a fixed inertial frame is necessary. Blackman [**24**] asserts this by stating that it is preferable to use a non-rotating or inertial coordinate system so that multiple target tracks can be processed with respect to the same fixed reference. An advantage of tracking in Body Reference Frame BRF (see Figure 2.10) is that it negates the need for a measurement conversion to the NED system. Furthermore, for a single target, the BRF system makes sense. However, Fasano [**48**] highlights the major disadvantage of tracking in BRF when he says, “relative motion in BRF includes attitude dynamics, which makes it more difficult to track, unlike its projection in NED, which only depends on relative position dynamics”.

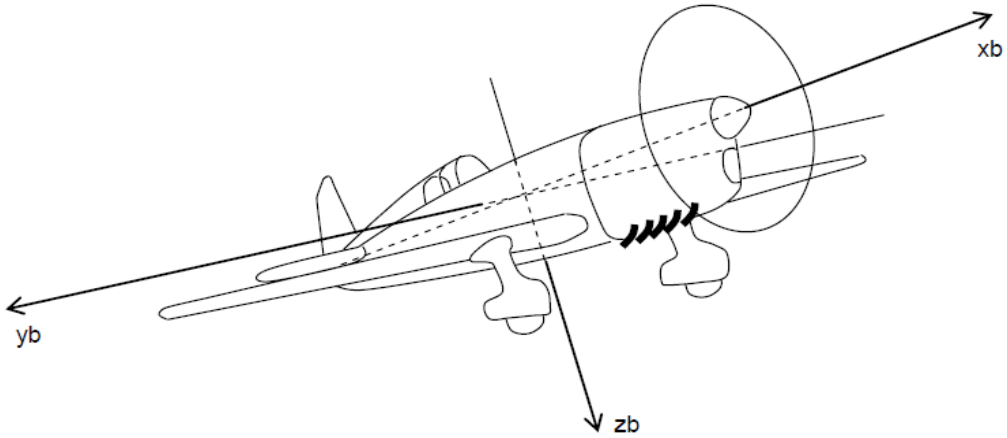

Figure 2.10: Body axis coordinate system. Also known as the body reference frame (BRF) [**49**].

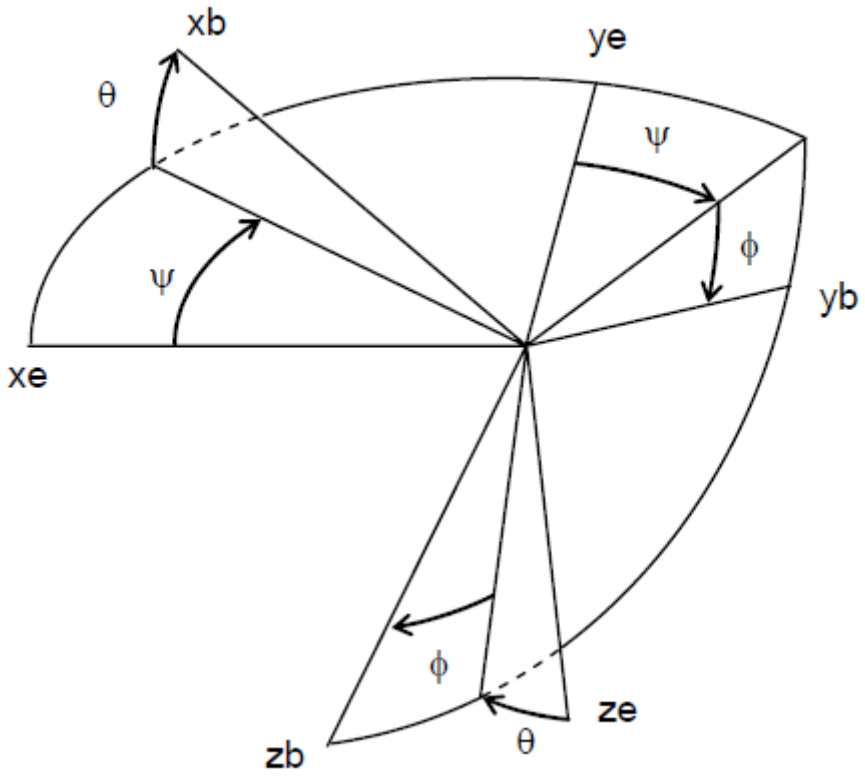

Figure 2.11: Relationship between BRF and a fixed earth frame, e.g. the NED system. [**49**]

Tracking is generally performed in either the geocentric North-East-Down reference frame or platform-centric Body Reference Frame (BRF)[14]. BRF has its origin at the aircraft centre of gravity and axes along longitudinal, lateral and vertical aircraft axes. The NED frame has the same origin as the BRF and its axes are determined by the north direction and down direction pointing to the centre of the Earth. The east axis is the direction perpendicular to the north and down axes. It is worth noting that the NED system is not strictly inertial for a moving platform. The reason for this is that platform axes are slowly changing their attitude (orientation in inertial space) as the platform moves over the Earth's surface. Blackman [**25**] affirms that these rotations are significant only when near the North Pole and, therefore, the NED system is considered an inertial one for moving platforms. As an example of tracking in

[14] Tracking in BRF is often referred to as tracking in "body" or "body coordinates".

BRF, Pearson [**50**] presents a single target tracking system which tracks a target's position relative to an airborne sensor's pointing axes.

Four centuries ago Rene Descartes watched a fly walk across his ceiling and wondered how to capture its position [**51**]. This ultimately resulted in the birth of the Cartesian coordinate system. This is one of the coordinate systems with which to perform tracking; the other being spherical. The choice of Cartesian or spherical space for this purpose each come with their own set of caveats. For example, if a Cartesian frame is chosen then the measurement equation of the filter is nonlinear[15]. This is in contrast to the choice of a spherical frame where the target dynamic model is nonlinear, i.e. the state transition is a nonlinear function of the state variables. This nonlinearity, in the dynamics, results in the introduction of pseudo accelerations in the angular components (even for a constant velocity target model) [**52**]. However, for distant targets these pseudo accelerations are small [**53**]. Another example is when a near constant velocity model [**54**] is expressed is spherical space it is significantly more complex than its projection in Cartesian space [**47**].

It is considered that systems that make use of multiple, geographically separated sensors must use Cartesian coordinates [**25**]. In a technical report by Zollo and Ristic [**53**], they prove that tracking in Cartesian space is more accurate during non-manoeuvring periods of the target trajectory; whereas tracking in spherical coordinates is, conversely, more accurate during manoeuvring periods.

When the Cartesian space is considered, tracking is commonly performed by a linear approximation of the measurement by taking the partial derivatives with respect to the target state and feeding this into the *corrector* stage of an EKF (see Section 2.1.2). Duan et al. [**55**] propose a method which reduces the nonlinearity of the measurement by forming a pseudo measurement which is the product of the range and range rate. To steer away from a nonlinear filter,

[15] This is true in the case of airborne tracking.

an alternative is to convert the spherical sensor measurements to Cartesian space and then de-bias them [**53**] [**48**]. This allows the use of the linear KF for tracking.

2.2.2 Passive Bearings-Only Tracking

Monocular vision-based tracking has seen extensive research in the last decade [**56**] [**57**], especially in the field of UAV navigation. The reason for this proliferation is the increase in digital imagery technology as well as camera's light-weight nature makes them (visible or infra-red) suitable payloads for UAVs. Tracking with angles only to estimate a target's state vector is termed bearings-only tracking (BOT). As stated in Chapter 1, a passive sensor provides only angular measurements of a target, i.e. target azimuth and elevation. Therefore, the problem of BOT is a nonlinear one. As a result, EKFs [**58**] [**59**], UKFs [**56**] and PFs [**60**] are usually employed as the tracking filter.

A tracking system utilising only a passive angle-only sensor is not generally able to estimate the position of a target[16] [**25**]. Despite this, it is possible to determine target states using multiple measurements (not in a single sensor observation) at different time instants. The problem geometry of classical BOT is illustrated in Figure 2.12. A further problem exists (see Figure 2.13) when a target's motion is purely radial, i.e. along the ownship line-of-sight (LOS). Here the target is moving, but the measured angle θ stays constant. A means of mitigating this effect is for the ownship (observer) to perform manoeuvres in order to take measurements when the target's motion is not radial [**61**]. This is only possible if the observer is manoeuvrable and is not generally considered an optimal solution.

[16] In most cases single camera bearings-only tracking filters would diverge.

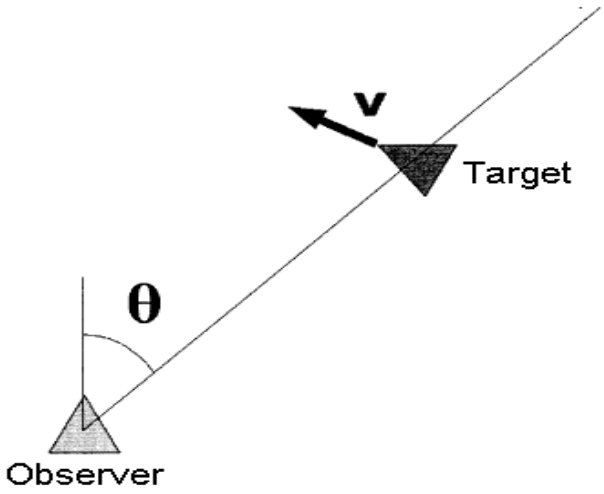

Figure 2.12: Bearings-only problem geometry [**62**].

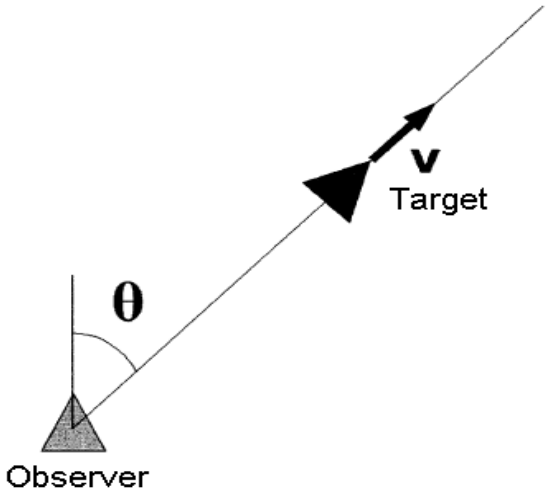

Figure 2.13: Target motion along observer LOS [**62**]. Angle theta remains constant in this scenario.

Notwithstanding this general consensus on bearings-only tracking, Bethke [**62**] proposes a solution to estimate the target state in a single sensor observation. This is achievable through the use of multiple, cooperative UAVs each equipped with its own EO camera. In order to determine the position and velocity of the target, multiple observations from multiple UAVs are

simultaneously combined (see Figure 2.14). This results in an unambiguous location of the target, regardless of target motion, found through the intersection of multiple rays from each observer to the target. Once the position is found, a linear KF filter is used to track the target's position and velocity. This solution, although novel and exciting, is not very practical. Yes, Bethke [**62**] presents acceptable results; but it is rare that so many resources would be available for a single mission. A less costly SAA solution would be more lucrative for industry.

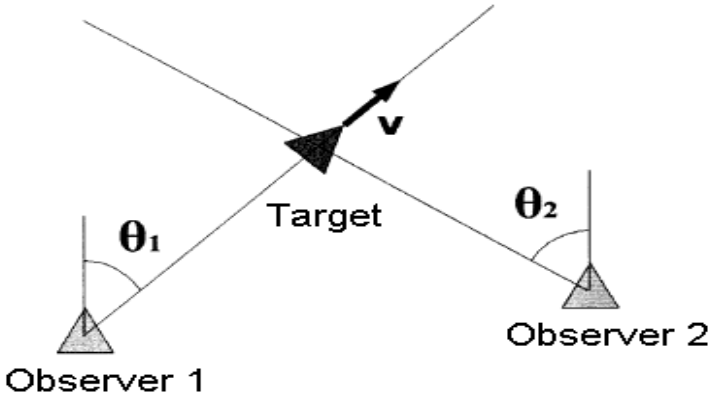

Figure 2.14: Using multiple UAVs to determine target position [**62**].

The following method describes an alternative to target tracking employing angles only measurements. It is technically not considered BOT, yet it does make use of angular measurements only. Watanabe [**63**] presents a tracking filter whose observation vectors are composed of detected pixels in the image plane.

In his system the existence of an image processing subsystem is assumed which detects potential threats in an image. Furthermore, the camera is assumed to be fixed to the centre of gravity of the UAV. Moreover, a pinhole[17]

17 For an excellent read on camera models, homographies and other image processing techniques, the reader is directed toward [**86**].

camera model is assumed as that in Figure 2.15, with the camera's X_C axis aligned with the principal (or optical) axis. Therefore the image plane is found at:

$$X_C = f. \tag{2.30}$$

Where f is the focal length of the camera. The origin of the camera frame is called the optical centre. The principal axis is the line originating from the optical centre, perpendicular to the image plane; and the point where they cut is the principal point.

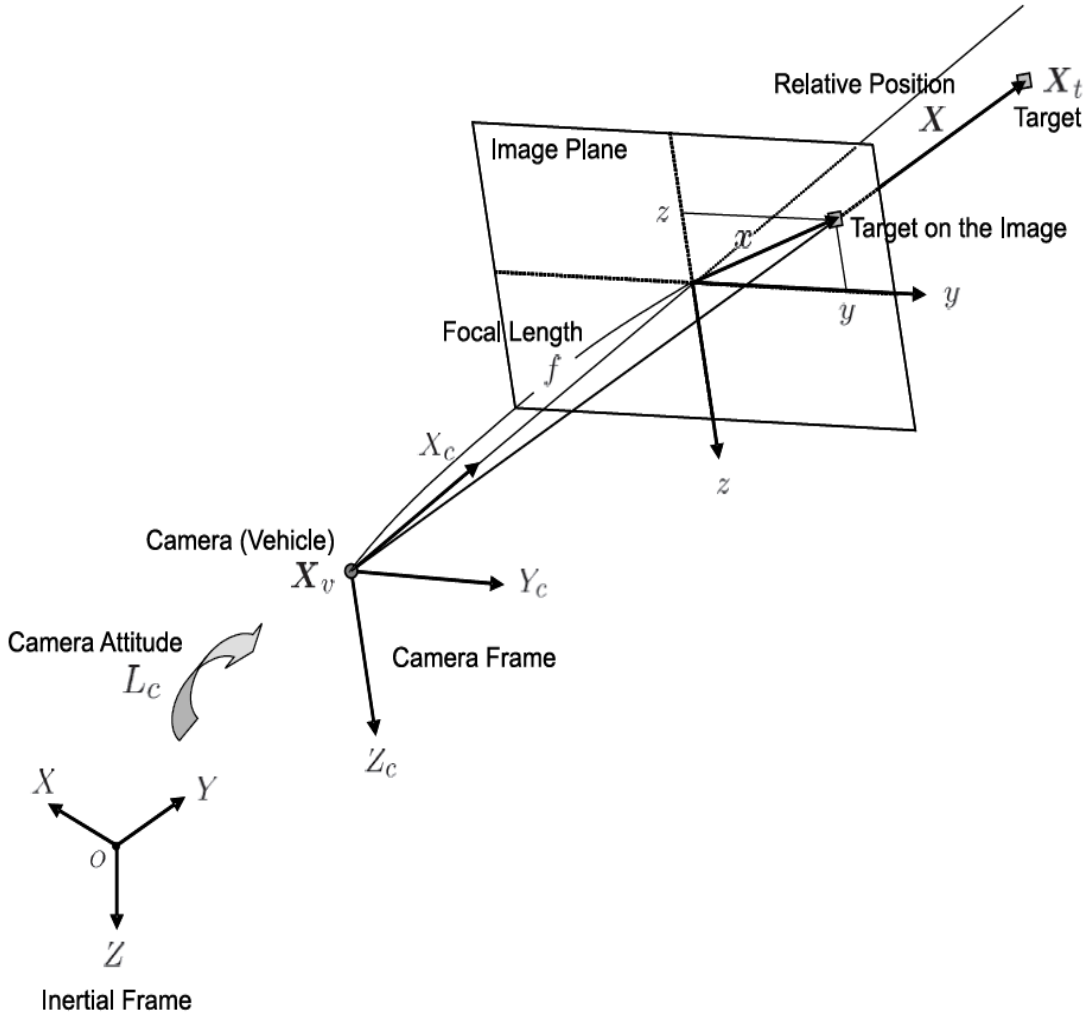

Figure 2.15: Pinhole camera model [**63**].

As a result of similar triangles (see Figure 2.16), the target position in the image at time step k is:

$$x_k = \begin{bmatrix} y_k \\ z_k \end{bmatrix} = \frac{f}{X_C^k} \begin{bmatrix} Y_C^k \\ Z_C^k \end{bmatrix}. \tag{2.31}$$

Following this, a transformation is performed so that estimation of the state vector may proceed in the inertial frame through the use of an EKF.

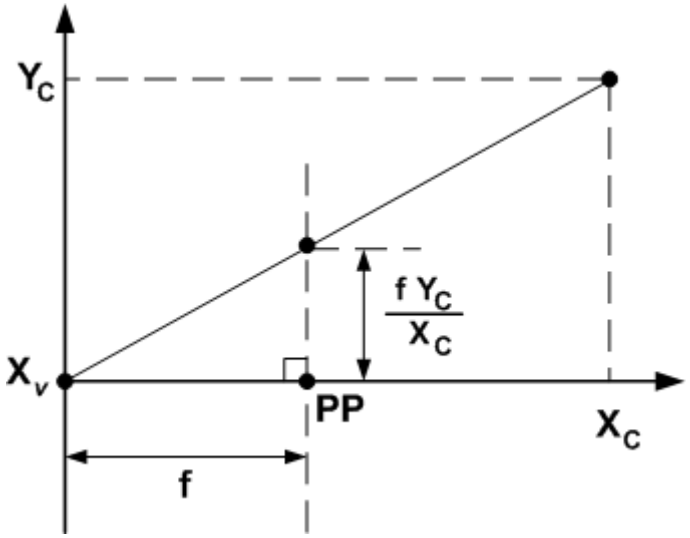

Figure 2.16: Projection of problem geometry in 2D.

Gupta [**41**] follow a similar approach as Watanabe [**63**], but performs tracking in the spherical coordinate system. Their reasons for this choice are similar to those laid out in Section 2.2.1. They state that the measurement equation is highly nonlinear in Cartesian space. By utilizing the EKF, the *corrector* stage results in a very complex Jacobian, which they cannot afford.

2.2.3 Tracking with Doppler Information

When an active Doppler radar is utilized in a tracking system, range rate (Doppler) information can be extracted from analyzing the frequency variation in the reflected signal (see Figure 2.17). One would assume that the inclusion of range rate measurements would result in better estimates produced by a

tracker, especially the velocity components. This is asserted by the results presented in [**55**], which indicate that tracking accuracy can be greatly improved with the use of Doppler information. This increased accuracy comes at the expense of increased filter complexity.

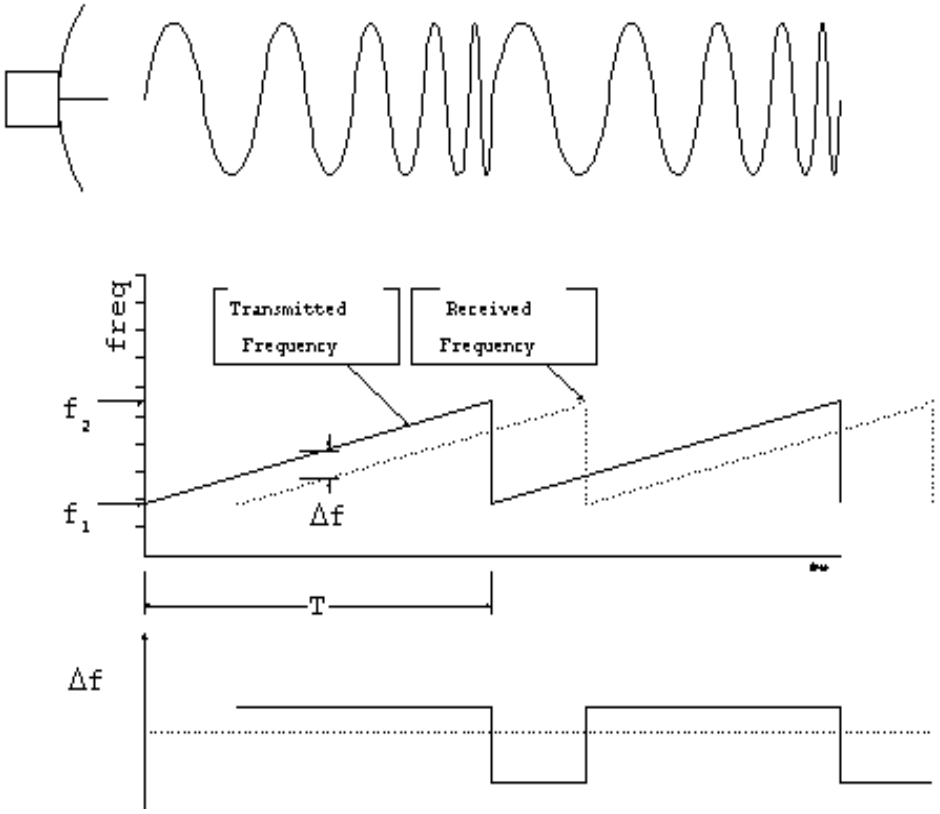

Figure 2.17: Frequency modulated continuous wave (FMCW) signal transmitted by a radar. The range rate is determined by the frequency difference in transmitted and reflected signals.

Notwithstanding the performance gain resulting from the use of Doppler information, range rate measurements are not always used in tracking filters. This is due to the measurements being highly nonlinear in Cartesian space. Moreover, large biases may be introduced in the posterior estimate [**64**].

Tracking with range rate measurements is further complicated in that measurement errors in range and range rate are correlated for certain radar waveforms [**65**]; with upsweep chirp Linear Frequency Modulation (LFM) waveforms there exists a negative correlation and with downsweep chirp LFM

waveforms there exists a positive correlation. The result of this can be seen in the radar measurement covariance matrix R with elements appearing off the main diagonal.

As a result of the above, there is not much focus in the literature on tracking with range rate measurements (especially so with regard to fusion – more on that in the subsequent section). Although complex, there are a number of methods in order to solve this problem. Duan et al. [**44**] describe that there are two ways of achieving this end:

- Tracking in mixed coordinates [**65**] [**44**] [**42**] [**66**].
- Tracking in Cartesian coordinates [**38**] [**55**] [**67**] [**68**].

Tracking in mixed coordinates implies that sensor measurements are used **directly** in the filter; that is the target's state vector and process noise are in Cartesian space while the sensor measurement and its measurement noise are in the sensor frame. This requires the linearization of the observation vector about the predicted estimate as described in subsection 2.1.2. Therefore, tracking is based on nonlinear filters such as an EKF. The disadvantage of this method is that the linearization process may introduce large errors in the mean and covariance of the posterior estimate. This equates to suboptimal filter performance and could result in the divergence of the tracking filter. Duan [**55**] state that tracking with mixed coordinates using an EKF usually results in poor tracking accuracy.

Bellotto and Hu [**69**] affirm the aforementioned point in [**55**] and state that, in general, better performance is achieved through the use of a UKF. The UKF predicts a state estimate by transforming an entire probability distribution. The difference between an EKF and a UKF is that the linearization process of the EKF is replaced with an unscented transformation. The unscented transformation, with carefully chosen weighted points[18], captures the mean

[18] These weighted points are also called sigma points.

and covariance of the probability distributions [**69**]. An alternative method of transforming probability distributions is to sample the distribution and then transform the samples separately; if the samples are drawn at random, then this approach is called Particle Filtering [**70**]. In contrast to PFs, UKFs make use of a smaller number of points and therefore make them less computationally intensive and, hence, more suitable to real-time applications [**69**].

A common assumption made when tracking with mixed coordinates and the availability of Doppler measurements is that measurement errors are uncorrelated. However, recent research proves that in reality this is not always the case [**65**] [**44**] [**66**]. As stated earlier, when tracking with Doppler information there exists a correlation in measurement errors between range and range rate. In an upsweep chirp waveform the correlation coefficient ρ^{corr} is positive valued and implies negative correlation; whereas in downsweep chirp, ρ^{corr} is negative in value and implies a positive correlation [**66**]. The general consensus in the literature is that when a negative correlation exists and is accounted for in the tracking filter, the result is increased tracking system performance; furthermore when ρ^{corr} is around -0.9, significant reduction is evident in all the state estimation errors [**65**] [**44**] [**66**]. Bar-Shalom [**66**] extrapolates these findings by proving that tracking performance under positive correlation is worse than that when there is no correlation. However, Yuan et al. [**65**] build on the work presented in [**66**] and prove, via simulations, that tracking performance is not always worse in the positive correlation case as opposed to the no correlation case. In what follows, the assumption is that a radar exists and provides the tracking system with measurements of target range, azimuth, elevation and range rate denoted by ρ_k, θ_k, φ_k and $\dot{\rho}_k$, respectively.

When measurements errors in range and range rate are correlated the resulting measurement covariance is:

$$R = \begin{bmatrix} \sigma_\rho^2 & 0 & 0 & \rho^{corr}\sigma_\rho\ \sigma_{\dot{\rho}} \\ 0 & \sigma_\theta^2 & 0 & 0 \\ 0 & 0 & \sigma_\varphi^2 & 0 \\ \rho^{corr}\sigma_\rho\ \sigma_{\dot{\rho}} & 0 & 0 & \sigma_{\dot{\rho}}^2 \end{bmatrix}. \quad (2.32)$$

In order to decorrelate the measurement errors, let [**44**]:

$$L = -\left[\rho^{corr}\sigma_\rho\ \sigma_{\dot{\rho}}\ \right] \times \left(\left[\sigma_\rho \quad \sigma_\theta \quad \sigma_\varphi\ \right]\right)^{-1} \quad (2.33)$$
$$= \begin{bmatrix} -\rho^{corr}\sigma_{\dot{\rho}} \big/ \sigma_\rho & 0 & 0 \end{bmatrix}.$$

Then:

$$B = \begin{bmatrix} I_3 & 0_{3\times 1} \\ L & 1 \end{bmatrix}. \quad (2.34)$$

Where I_3 is the 3×3 identity matrix and $0_{3\times 1}$ is a 3×1 vector of zeros.

By first performing Cholesky factorization [**39**] and then pre-multiplying Equation (2.34) on both sides of the observation vector, a new pseudo measurement results (while the range, azimuth and elevation remain unchanged):

$$\varepsilon_k = h_\varepsilon(X_k) + v_k^\varepsilon \quad (2.35)$$

$$\varepsilon_k = \frac{-\rho^{corr}\sigma_{\dot{\rho}}\rho_k}{\sigma_\rho} + \dot{\rho}_k \quad (2.36)$$

$$h_{\varepsilon}(X_k) = -\rho^{corr}\frac{\sigma_{\dot{\rho}}}{\sigma_{\rho}}\sqrt{x_k^2 + y_k^2 + z_k^2} + \frac{x_k\dot{x}_k + y_k\dot{y}_k + z_k\dot{z}_k}{\sqrt{x_k^2 + y_k^2 + z_k^2}} \tag{2.37}$$

The measurement noise of the new pseudo measurement is now uncorrelated with all other measurement noises and is still a zero-mean Gaussian white noise sequence with variance:

$$\sigma_{\varepsilon}{}^2 = \left(1 - \rho^{corr\,2}\right)\sigma_{\dot{\rho}}^2. \tag{2.38}$$

In contrast to mixed coordinate tracking, tracking in Cartesian-only space necessitates a conversion of sensor measurements from the sensor frame[19], in which they originate, to Cartesian space. This conversion is the basis of all Cartesian-only tracking. The conversion of measurements present a number of flaws [**44**]: converted measurement errors are state dependent; and, converted measurement sequences are not white anymore.

A means of mitigating this effect and consequently a method of tracking in Cartesian space with range rate measurements, is that performed by Duan et al. [**55**]. Their work follows on from that of [**67**]. Suchomski [**67**] presents a method of tracking in Cartesian space with a linear filter when a sensor is providing nonlinear measurements. This method extends that of prior research [**38**], which only operated in 2-dimensions, and realises a tracking filter that is based on debiased consistent converted measurements. Duan et al. [**55**] modify the former by including range rate measurements. Furthermore, through the formation of a pseudo measurement η_k that is the product of the range and range rate measurements:

[19] Sensor measurements are generally either described in polar or spherical coordinates.

$$\eta_k = \rho_k \dot{\rho}_k = x_k \dot{x}_k + y_k \dot{y}_k + z_k \dot{z}_k. \tag{2.39}$$

The presence of this pseudo measurement reduces the nonlinearity of the measurement equation with respect to the state vector. The first step that needs to be performed is the conversion of radar measurements to Cartesian space:

$$x'_k = \rho_k \cos\theta_k \cos\varphi_k \tag{2.40}$$

$$y'_k = \rho_k \sin\theta_k \cos\varphi_k \tag{2.41}$$

$$z'_k = -\rho_k \sin\varphi_k \tag{2.42}$$

Following this, these converted measurements are debiased as follows:

$$\begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} = \begin{bmatrix} x'_k \left[1 - \left(e^{-\sigma_\theta^2} e^{-\sigma_\varphi^2} - e^{-\frac{\sigma_\theta^2}{2}} e^{-\frac{\sigma_\varphi^2}{2}}\right)\right] \\ y'_k \left[1 - \left(e^{-\sigma_\theta^2} e^{-\sigma_\varphi^2} - e^{-\frac{\sigma_\theta^2}{2}} e^{-\frac{\sigma_\varphi^2}{2}}\right)\right] \\ z'_k \left[1 - \left(e^{-\sigma_\varphi^2} - e^{-\frac{\sigma_\theta^2}{2}}\right)\right] \end{bmatrix}. \tag{2.43}$$

Where σ_θ and σ_φ are the measurement errors in azimuth and elevation, respectively.

The debiased consistent converted measurement vector $[x_k \quad y_k \quad z_k \quad \eta_k]^T$ has a covariance matrix R_k [**67**]:

$$R_k = \begin{bmatrix} & R_a & & \sigma_{x\eta} \\ & & & \sigma_{y\eta} \\ & & & \sigma_{z\eta} \\ \sigma_{x\eta} & \sigma_{y\eta} & \sigma_{z\eta} & {\sigma_\eta}^2 \end{bmatrix}. \quad (2.44)$$

Where R_a is defined as:

$$R_a = \begin{bmatrix} R_a^{xx} & R_a^{xy} & R_a^{xz} \\ R_a^{xy} & R_a^{yy} & R_a^{yz} \\ R_a^{xz} & R_a^{yz} & R_a^{zz} \end{bmatrix}. \quad (2.45)$$

Appendix A lists the full measurement covariance matrix and related equations when using the debiased consistent converted measurements KF method.

Another method of tracking in Cartesian coordinates is that presented in [**68**]. Wang et al. [**68**] propose a method of sequentially processing the range and range rate measurements. This correlates with the findings of Blackman and Popoli [**25**] as they state that the inherent nonlinearity in the system can be reduced by first processing the least nonlinear measurement first.

Zollo and Ristic present another tracking filter in [**53**]; this time tracking a manoeuvring target with spherical coordinates. The filter displays satisfactory tracking performance and is one that is found in a number of operational systems [**71**]. The system comprises three linear KFs; one for range/range rate, and three for the direction cosines[20] (DCM) in the north, east and down directions. The state vector of the range/range rate filter is:

[20] The pointing angle of sensors as well as aircraft attitude is generally described with direction cosines relative to the NED axis.

$$X_{\rho} = \begin{bmatrix} \rho \\ \dot{\rho} \\ \ddot{\rho} \end{bmatrix}. \tag{2.46}$$

And the state vector for the north filter is:

$$X_N = \begin{bmatrix} \Lambda_N \\ \dot{\rho}_N \\ \ddot{\rho}_N \end{bmatrix}. \tag{2.47}$$

Another two state vectors exist (not shown here) for the east and down directions. Λ_N, Λ_E and Λ_D are the direction cosines in the north east and down directions respectively. Components of target position in the north, east and down directions can be found as follows:

$$\rho_N = \rho\Lambda_N \tag{2.48}$$

$$\rho_E = \rho\Lambda_E \tag{2.49}$$

$$\rho_D = \rho\Lambda_D \tag{2.50}$$

The reader is urged to consult [**53**] for the remainder of the mathematics required for this scenario. Furthermore the derivation of the target dynamics used in [**53**] can be found in [**24**]. Once the direction cosines are known target angular position with respect to the NED system can be represented by *stabilised* azimuth η and *stabilised* elevation ε:

$$\eta = \tan^{-1}\left(\frac{\Lambda_E}{\rho\Lambda_N}\right) \tag{2.51}$$

and

$$\varepsilon = -\sin^{-1}\Lambda_D. \tag{2.52}$$

In summary, most of the literature reviewed above track targets in Cartesian space as it is generally considered that it yields more accurate filters but is

more computationally expensive [**53**]. However, these filters do not track in inertial NED, which would be most convenient for airborne platforms. Furthermore, it is assumed that the sensor, providing measurements, is stationary. Blackman and Popoli [**25**] claim that the use of acceleration states in the state vector is required, for satisfactory performance if a velocity measurement, such as range rate, is available. However, the literature described above, as well as the designs presented later in this thesis prove that that is not always the case. Moreover, tracking in mixed coordinates is considered to yield poor accuracy, but through fusion of multiple sensor's data we prove the contrary.

2.2.4 Tracking in the Presence of Multiple Sensors

Tracking a target while utilising measurements from multiple, redundant sensors is a fairly novel technique. There is definitely less literature on the matter compared to that of the proliferation on radar tracking. More specifically, there has been little discussion in the literature about fusing data from a Doppler radar and a passive angle-only sensor for tracking.

The presence of multiple sensors in a tracking system results in registration error [**38**]; a fundamental issue when considering the implementation of a MSDF system, yet no literature could be found which addresses the problem of time registration. All the literature that has been investigated makes the assumption that all sensors present in the system have the same sampling frequency.

Two types of registration error exist, namely spatial and temporal. Space registration addresses the fact that multiple sensors provide measurements with respect to their own sensor frame. The solution is found through pre-processing of the sensor data by performing the necessary coordinate transformations and conversions. This ensures that each sensor's

measurements are with respect to a single frame of reference. Time registration addresses the fact that measurement rates, from multiple independent sensors, are asynchronous, i.e. they have different sampling periods. In our case an EO camera provides measurements of:

$$z_k^C = [\theta_k^C \quad \varphi_k^C]^T. \tag{2.53}$$

At a faster rate than the radar provides its measurements:

$$z_k^R = [\rho_k^R \quad \theta_k^R \quad \varphi_k^R \quad \dot{\rho}_k^R]^T. \tag{2.54}$$

To solve this problem and to synchronise the measurements, data compression is performed. Data compression combines a number of observations (measurements) into a single observation that is a weighted average of the component observations; where more recent observations are weighted more heavily [**25**]. The combined measurement (z_{DC}^C) is found using a least squares curve fit to the N[21] observations:

$$z_{DC}^C = \frac{1}{N}\sum_{i=1}^{N} z_i^C + \frac{12a}{(N-1)(N+1)}\sum_{i=1}^{N}(i - \frac{N+1}{2})z_i^C. \tag{2.55}$$

Assuming the EO camera has a measurement covariance matrix R^C, the composite camera measurement covariance matrix R_{DC}^C is evaluated as:

$$R_{DC}^C = \frac{1}{N}(1 + 12a^2)R^C. \tag{2.56}$$

[21] N is the ratio $\Delta T^R/\Delta T^C$.

The factor a in Equation (2.55) and (2.56) determines whether older or more recent measurements should be more heavily weighted. It lies in the region of $(0, 0.5)$ where a larger value more heavily weights recent observations.

When designing a MSDF system the estimation engineer needs to decide on the fusion architecture to make use of, namely centralised, distributed or hybrid. Centralised fusion combines multiple sensor observations (raw sensor data), whereas distributed fusion combines multiple tracking filters' states estimates. When a distributed architecture is employed, the main issues that need to be addressed are:

- Architectural structure. How sensor nodes are connected and how they share information.
- Tracking filters employed.
- Fusion algorithms. The way in which data is fused from each node.

These issues, consequently, are what differentiate researchers' MSDF systems.

Tracking with multiple sensors is then a problem involving the proper organisation and use of the tools and techniques discussed in Sections 2.1 and 2.2 thus far. There are a multitude of architectures and combinations of sensors found in MSDF literature. A subset of these includes:

- Distributed fusion of a Doppler radar and an IR sensor [**29**] [**72**].
- Fusion of radar and LIDAR [**73**].
- Fusion of radar, IR and LIDAR [**74**].
- Fusion of laser range finder (LRF) and EO camera [**69**].
- Fusion of radar and EO sensors [**70**].
- Fusion of EO camera, LIDAR and inertial measurement unit (IMU) sensors [**75**].

In [**29**], Qingchao and Wenfei propose a MSDF system that comprises two distributed local tracking filters: one EKF for the radar subsystem and one

EKF for the IR sensor subsystem. A pseudo measurement is formed by augmenting the radar and IR observation vectors; the range rate component is decoupled from the radar and concatenated to the IR observation vector. The advantage in creating this pseudo measurement is that measurement errors in range and range rate are not correlated anymore. Moreover, the increased dimensionality of the IR observation vector increases state observability. The fusion architecture is illustrated in Figure 2.18. The globally fused estimate and covariance is fed back to the IR subsystem only; termed partial feedback. The validity of this technique is affirmed by Zhu et al. in [**76**], where they state that by performing feedback, the accuracy of the local filter is improved. According to Qingchao and Wenfei [**29**] their reason for performing partial over full feedback is that, in so doing, the whole tracking system is not reliant on the fusion centre.

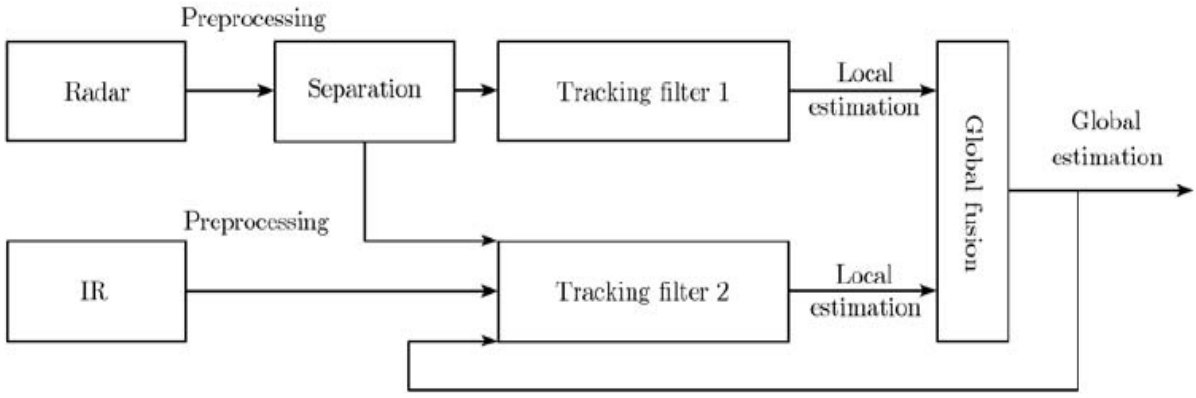

Figure 2.18: Qingchao and Wenfei proposed fusion architecture [**29**].

Once the local filters (Tracking filter 1 and Tracking filter 2) each produce their own local estimate, they are fused using Equations (2.6) and (2.7). Furthermore, two other variations of the architecture in Figure 2.18 are implemented: one variant where no range rate measurements are available and no separation occurs; and one variant with range rate measurements but no separation (radar observation vector used as is). They have ascertained, through simulations, that the variant using range rate measurements and no separation results in an error decrease of 21.68% compared to the no range

rate variant. Furthermore, their proposed architecture (Figure 2.18) sees a further 3.75% error reduction. However, this design does not take into account that radar and IR sensors operate at different frequencies. In order to acount for this a technique such data compression should be performed. Furthermore, it is assumed that the platform containing the sensors is stationary.

In [**72**] a similar design is presented as in [**29**], but with a few differences. Firstly, the radar does not measure relative radial velocity. Furthermore, instead of an EKF for the IR local filter, the PLKF method (see Section 2.1.2) is used to derive the filter. Local estimates are fused in a similar fashion to that of [**29**]. Following fusion, feedback is performed by setting the global estimate as the local estimates for the next iteration of the filter. However, in contrast to [**29**], here the information distribution coefficient β^i is introduced which proportionally feeds the global estimate back. Performance is compared to that of centralised fusion with radar and IR sensors as well as radar and IR only tracking. The results prove that the proposed distributed approach not only outperforms individual sensor tracking, but tracking performance is comparable to that of centralised fusion. Furthermore, the distributed architecture reduces the system computational load and exhibits good real-time performance.

Blanc et al. [**73**] also proposes a distributed MSDF system that consists of a LIDAR and a radar with EKFs as each local filter. However, they do not make the assumption as in [**29**] and [**72**] that local state estimates are uncorrelated. As a result Equations (2.8) to (2.11) are used in order to fuse the local estimates. Furthermore, Blanc et al. [**73**] presents a method of fusing local state estimates using a PF. Their PF fusion is based on: first, calculation of particle weights from different sensor measurements; and second, validation of the fused estimate by the local estimates. They conclude that the PF method and EKF (i.e. standard distributed fusion) exhibit similar performance, however the EKF approach is much less computationally expensive.

A centralised fusion architecture comprising of radar, IR and EO sensors is developed by Fasano in [**48**]. The proposed architecture (see Figure 2.19) has the radar as the primary sensor and IR and EO as aiding sensors; this is because only the radar is used to update a target's track status.

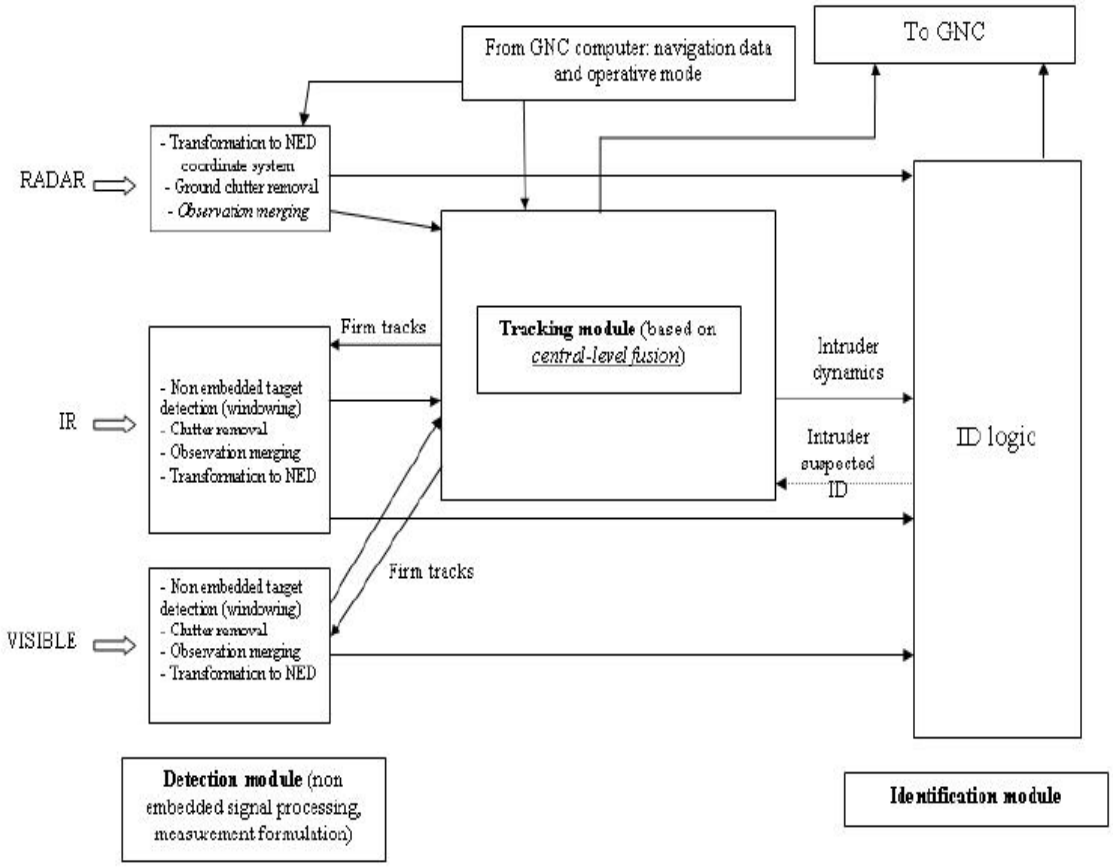

Figure 2.19: Sensor fusion logical architecture implemented in [**48**]. Centralised fusion consisting of radar, IR and EO sensors.

The EO camera uses information from the tracker in order to create a valid search window; the target's predicted estimate is utilised in order to define a region of interest in the image where valid target detection may occur. Once a target is detected, the necessary coordinate transformations are performed and the detected target position is transferred to the FC. This algorithm is illustrated in Figure 2.20. Finally centralised fusion is performed with

measurements from radar, EO and IR sensors in the FC in order to track a target.

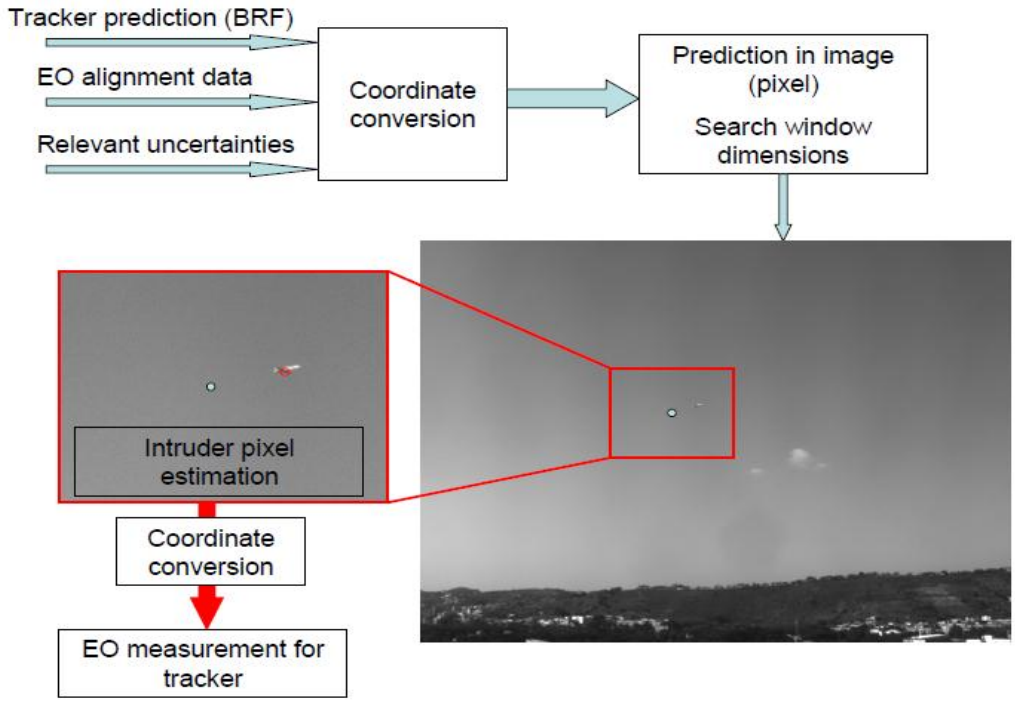

Figure 2.20: EO camera target detection algorithm [**48**].

In [**77**], Leibowicz et al. realise a system that consists of radar and ESM sensors. Their motive for fusion is not in search of improved filter accuracy, but rather to avoid alerting the enemy of one's presence. This is achieved by limiting the radar's activity at a specified time while allowing target tracking to continue by utilising only the passive sensor.

2.3 Summary

The review of literature stressed the need to have in place a tracking system that is accurate and robust enough for a UAV SAS, which would be a component in enabling its timely adoption into civil airspace. This vision will only be possible through the appropriate use of MSDF. Thus, it is evident from this literature review that target tracking using MSDF is by no means a trivial task. It is one that encompasses numerous techniques from various

engineering disciplines [**14**]. In summation, when designing a MSDF system the following needs to be addressed:

- Choice of tracking coordinate system.
- Choice of fusion architecture.
- Fusion algorithm selection.

In order to perform these tasks, a sound understanding of MSDF techniques and estimation fundamentals are required. First a definition of MSDF was produced, highlighting its similarity to the cognitive process evident in humans. The use of multiple, redundant sensors have many practical and operational advantages and these were discussed through the aid of an example. These advantages are: imperviousness to ECM; all-weather, all-day operation; improved overall tracking performance; and extended spatial and temporal coverage. That being said, Hall and Steinberg [**26**] warn that MSDF is not magic and that considerable effort is required to get the best performance out of a MSDF system.

Based on the provided review there is a need for research on target tracking with Doppler information in MSDF systems. Moreover, an even greater need for research on fusion architectures for a tracking subsystem that forms part of a UAV SAS; research that will be key for UAV's adoption in civil airspace.

To address this need, empirical research will be implemented. Specifically, such research will attempt to ascertain the most appropriate fusion architecture for a UAV SAS. As such multiple fusion architectures will be developed as well as radar and camera only trackers in order to bring to light the performance gains from performing MSDF. The next stage of this research will detail the methodology for the design of the fusion architectures, including details on the research strategy adopted, presentation of metrics with which to assess the designs and the development of the simulation environment.

Chapter 3: Methodology

The development of a tracking algorithm using MSDF is no easy task. Moreover, the development of an airborne one is considerably more complex. The focus of this chapter is to describe the strategy employed in this thesis in order to realise an airborne tracking filter. Sound engineering principles are adhered to and details of testing the proposed designs are discussed. This entails presenting performance metrics with which system performance can be ultimately measured. It also provides motivation for the components designed in Chapter 5..

3.1 Overview of Research Strategy

Chapter 2 identified a number of gaps in existing research: there is a lack of fusion literature which includes the use of a Doppler radar; the amount of literature on airborne tracking is limited; and, MSDF is a fairly new field of research which requires active research and development for its maturity.

The research strategy employed in this thesis is akin to the model based design (MBD) approach applied to research. A benefit of MBD is that it allows for the test and verification of designs early on in its life-cycle [**78**], when the cost of change is minimal; therefore, it has seen extensive use in the automotive and aerospace industries [**78**]. The MBD methodology is outlined below [**79**] and illustrated in Figure 3.1:

- **Research**
  The first step in this approach is to perform a study of relevant literature.

- **Requirements Definition**
  Based on the results and findings from the research phase, system requirements are defined.

- **Modelling the Process/Plant**

  The process under investigation as well as the environment with which it interacts is modelled.

- **Algorithm Design**

  This phase is responsible for the design and implementation of algorithms, using the information from the process model and its environment, which satisfy the outputs from the Requirements Definition phase.

- **Simulation**

  Simulations are performed which test the algorithms in the modelled environment. Furthermore system evaluation is performed by defining performance metrics and assessing system performance based on them. At the end of the Simulation phase, the Algorithm Design phase can be performed again in order to increase system performance.

- **Deployment**

  The last phase is the deployment on to target hardware. This involves integration with other algorithms or subsystems and, finally, placement on an embedded platform.

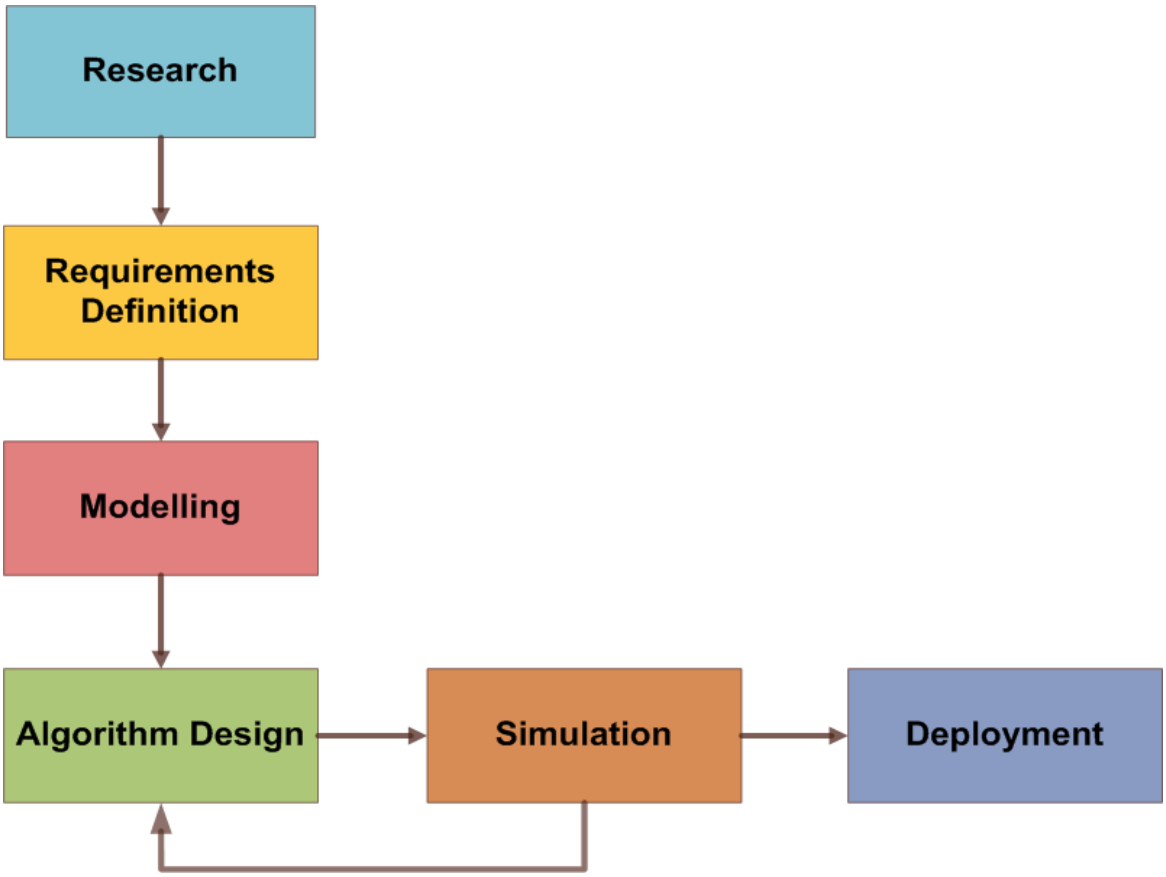

Figure 3.1: MBD approach.

The reason why this approach was adopted is twofold: first, it requires the modelling of the simulation environment early on, a very important aspect when dealing with dynamic systems. And second, in Chapter 5: multiple fusion architectures and filters are developed, which are assessed to determine the most suitable one for an airborne platform; this strategy works well for this iterative develop-simulate-test process. In what follows, how this MBD approach was applied to this thesis is detailed.

3.2 Research and Requirements Definition

A background study was performed in Chapter 1 which highlighted the need for the TTU. The problem description then justified this reasoning. Following this an exhaustive literature survey was performed in Chapter 2 which

reviewed current literature and various approaches in order to arrive at a solution to the problem at hand. The aforementioned points as well as the requirements from both [**2**] and [**9**], led to the definition of the thesis objectives detailed in section 1.4.

3.3 Modelling and Simulation

The modelling of the simulation environment is a crucial part in the design of the system. The reason for this is that the system developed is placed in this environment where it undergoes rigorous testing and verification. The plant is the scenario of airborne threat detection, from when a sensor on-board the UAV detects the presence of a threat until it is no longer in its FOV (i.e. the tracking period). This entails the mathematical modelling of both the UAV and threat dynamics in the inertial NED system. Furthermore sensors need to be modelled; however, sensor measurements appear in the BRF (as we assume that the sensor is at the centre of gravity of the UAV). Moreover sensor measurements are of relative[22] dynamics. This measurement process as well as the mathematical modelling of UAV and threat dynamic models is addressed in section, which describes the simulation environment.

3.4 Algorithm Design

In order to design a suitable airborne multisensor tracking filter a sound foundation of tracking and filter performance, under various conditions, is required. As such, the approach taken to arrive at the final fusion architecture starts with the development of a fairly simple filter (with no sensor data fusion) on which incremental improvements and modifications are made. This allows verification and performance of each filter, which has been developed, to be analysed at every incremental stage. Moreover, the filters produced at each incremental stage become the building blocks for the final fusion architecture.

[22] Relative dynamics is defined as the vector difference between threat and ownship (UAV) dynamics.

The design of the tracking filters and fusion architectures form the core of Chapter 5. Furthermore each of the designed filters shall track relative dynamics in the inertial NED system.

3.5 Testing the Threat Tracking Unit and Performance Metrics

This section details how each filter developed in Chapter 5 is to be properly tested. First the results from a single flight scenario are presented. This allows visual inspection to be performed to determine whether a filter is performing as expected. To determine by how much, Monte Carlo simulations are performed and a number of performance metrics presented, the result of which will be discussed on a quantitative level to assess their significance in Chapter 6. The following tests and metrics are presented in this section:

1. Visual evaluation.
2. Filter Accuracy Test
    - Root mean squared error (RMSE) in position.
    - Root mean squared error (RMSE) in velocity.
3. Filter Credibility Test
    - Average normalised estimation error squared (ANEES).
4. Tracking filter consistency test. This entails:
    - Innovation consistency test.
    - Innovation autocorrelation test.

To clarify the notions of filter consistency and efficiency, the following definitions are presented: a **consistent** filter is one that provides an increasingly accurate state estimate over time; and an **efficient** filter is one that minimises the MSE [**37**].

3.5.1 Visual Evaluation

Filter performance is firstly visually evaluated by presenting the estimation error of position and velocity components in NED. Intuitively, the closer these values are to zero, the more accurate the filter is. Furthermore, filter state covariance is illustrated which provides a measure of estimate uncertainty. Along the diagonal of the state covariance matrix is the principal uncertainties in each of the state vector elements. At each time-step with the arrival of a new measurement and the performance of the *corrector* stage, the state covariance is expected to decrease, and eventually reach steady-state.

3.5.2 Filter Accuracy

By performing large-scale Monte Carlo simulations (MCS), we have the ability to gain great insight into filter performance. The first metric determined during MCS is the RMSE in position and can be described as:

$$RMSE_k^{pos} = \sqrt{\frac{1}{N^{MC}} \sum_{i=1}^{N^{MC}} \begin{pmatrix} \left(N_k^i - \widehat{N}_{k|k}^i\right)^2 + \\ \left(E_k^i - \widehat{E}_{k|k}^i\right)^2 + \\ \left(D_k^i - \widehat{D}_{k|k}^i\right)^2 \end{pmatrix}}. \tag{3.1}$$

Where N^{MC} is the number of Monte Carlo runs; the vectors $[N_k^i \quad E_k^i \quad D_k^i]$ and $\left[\widehat{N}_{k|k}^i \quad \widehat{E}_{k|k}^i \quad \widehat{D}_{k|k}^i\right]$ are the target's true position and position estimate in inertial NED, respectively. Similarly the velocity RMSE is:

$$RMSE_k^{vel} = \sqrt{\frac{1}{N^{MC}} \sum_{i=1}^{N^{MC}} \begin{pmatrix} \left(\dot{N}_k^i - \dot{\widehat{N}}_{k|k}^i\right)^2 + \\ \left(\dot{E}_k^i - \dot{\widehat{E}}_{k|k}^i\right)^2 + \\ \left(\dot{D}_k^i - \dot{\widehat{D}}_{k|k}^i\right)^2 \end{pmatrix}}. \tag{3.2}$$

The RMSE in position and velocity gives a physical measure of the accuracy of the tracking filter.

3.5.3 Filter Credibility

As stated in Chapter 2, the filter estimated state covariance $P_{k|k}$ provides a self-assessment of how well a filter is performing. However, this self-assessment is not always reliable due to a number of reasons. The level of reliability (or how true) the estimated covariance is, is termed filter credibility. A means of determining credibility is through the calculation of the ANEES.

The ANEES is determined by the following:

$$ANEES_k = \frac{1}{N^{MC}\mathrm{n}} \sum_{i=1}^{N^{MC}} \left(X_{k|k}^i - \hat{X}_{k|k}^i\right)' {P_{k|k}^i}^{-1} \left(X_{k|k}^i - \hat{X}_{k|k}^i\right). \tag{3.3}$$

Where n is the dimension of X and $\left(X_{k|k}^i - \hat{X}_{k|k}^i\right)$ is the estimation error.

If the estimation error[23] is close to the estimated covariance, the ANEES will be approximately 1 and the filter is credible [**80**]. Moreover, the closer the ANEES is to 1, the more credible the filter. If the filter error covariance estimate is larger than the estimation error then the filter is said to be *pessimistic,* alternatively the filter would be deemed *optimistic*. This can be visualized by plotting the ANEES along with its 95% confidence interval. When the ANEES is above the 95% interval the filter is optimistic and when below, pessimistic.

3.5.4 Filter Consistency

The metrics defined above, viz. RMSE and ANEES require knowledge of **true** threat states. However, in contrast to simulation, practical real-world tracking systems are not aware of true data – there is no way that this data can be made available to the system. Therefore, evaluation of system performance in

[23] Estimation error is defined as the difference between the true state vector and its estimate (produced by the filter).

these pragmatic scenarios requires innovation analysis. In section 2.1.2 it was stated that the innovation/residual is an important factor in the KF and provides a measure of how well the filter is performing; the following two tests (presented in [**81**]) prove this:

- Innovation consistency test.
- Innovation autocorrelation[24] test (test for whiteness).

If the filter is performing correctly then the innovation sequence is zero mean and white with covariance S_k. Therefore, if the innovation sequence is consistent and white then the filter is consistent.

3.5.4.1 Innovation Consistency Test

The consistency test determines if the innovation is consistent. This test involves determining whether the innovation sequence is bounded by $\pm 2\sigma$, where:

$$\sigma = \sqrt{S_k}. \tag{3.4}$$

If approximately 95% of the values of the values lie within the $\pm 2\sigma$ bound, then the innovation sequence is consistent.

3.5.4.2 Innovation Autocorrelation Test

The autocorrelation test tests the innovation sequence for whiteness. Assume that the innovation sequence is denoted by ν_k, then by performing MCS the autocorrelation is:

[24] From statistical theory, a white noise sequence is independent and identically distributed and, therefore, has zero autocorrelation [**39**].

$$r(\tau) = \frac{1}{N^{MC}} \sum_{k=0}^{N^{MC}-\tau-1} \nu_k{}^T \nu_{k+\tau}. \tag{3.5}$$

Equation (3.5) is then normalised by dividing $r(\tau)$ by $r(0)$ – this yields the normalised autocorrelation of the innovation. In order to prove whiteness the normalised autocorrelation must be randomly distributed about zero. To prove that the normalised autocorrelation is in fact random we first need to compute its $\pm 2\sigma$ bound. For large enough N^{MC} it can be assumed that $r(\tau)$ is zero mean with covariance $1/N$. Then the standard deviation is given by:

$$\sigma = \sqrt{1/N^{MC}}. \tag{3.6}$$

Now it can be verified whether the normalised autocorrelation has 95% of its values in the $\pm 2\sigma$ bound; if it does then the innovation sequence ν_k is white.

If the innovation sequence was proved consistent and white from the above two tests, then the tracking filter is consistent.

Now that a platform for the evaluation of proposed designs has been established, thesis objective 6 has been satisfied; and the design of the TTU can commence.

Chapter 4: Simulation Environment

Modelling of the plant and the simulation environment needs to occur before the system can be designed and tested, as stated in section 3.2. Moreover, the TTU requires a simulation environment in which it can be tested to be developed. Therefore, this chapter presents the modelling of the simulation environment. This entails the mathematical modelling of the UAV and threat dynamic models as well as their generation in inertial space. Sensor measurements are generated in BRF and this process is explained. Moreover, using the above information, the flight scenario is described.

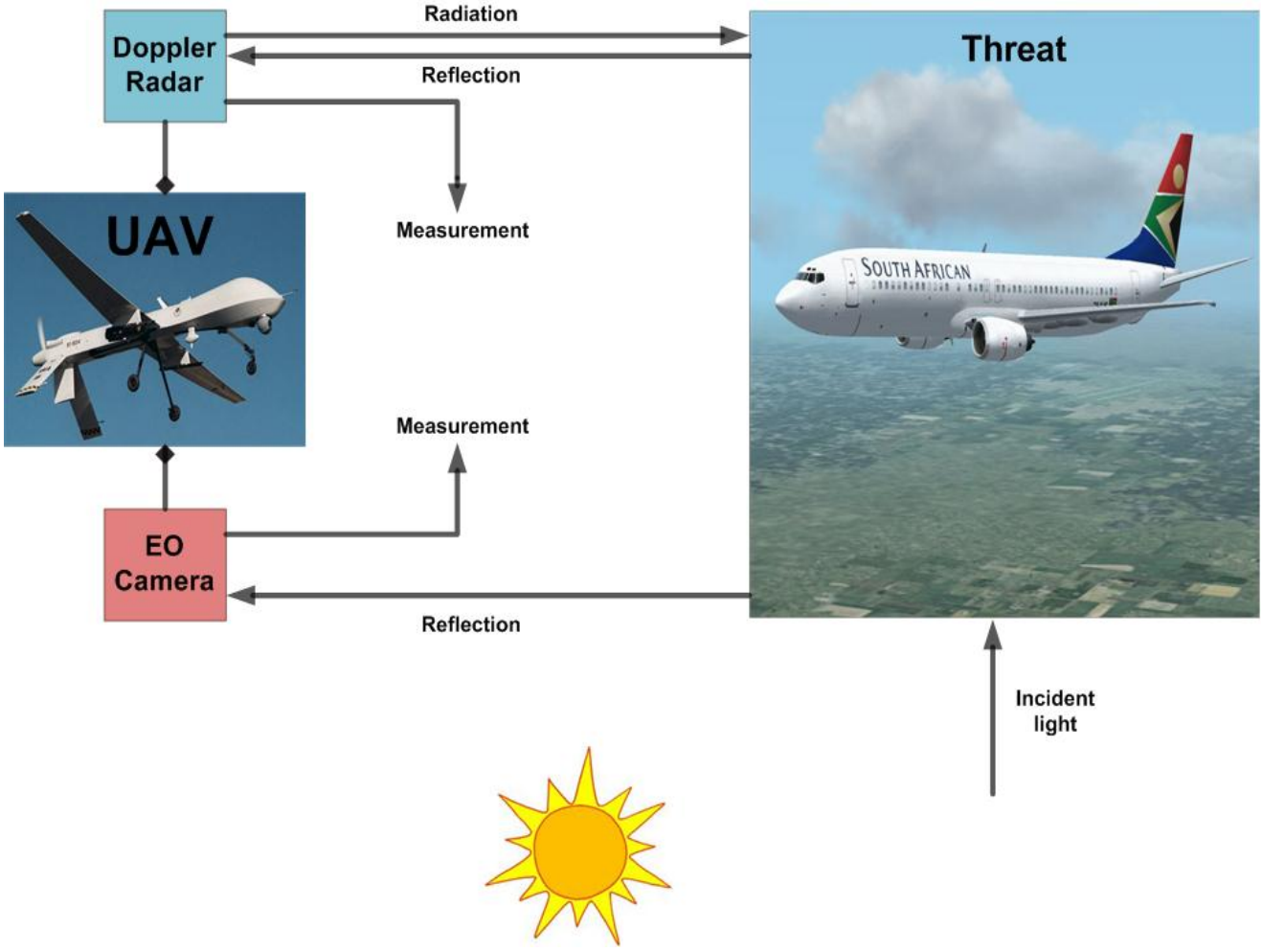

Figure 4.1: Simulation environment to be modelled.

4.1 States of Interest

The proper modelling of the simulation environment is critical to the success of the TTU design (see Chapter 3). First mathematical models for threat and UAV are presented. This information is used to generate threat and ownship trajectories in inertial space. Then the process of modelling Doppler radar and EO camera sensor measurements in inertial space is explained. Finally, the flight scenario used in the simulations is presented.

For the design of the TTU, the quantities of interest are the threat's position and velocity. The velocity components are of importance as the control system performing CA requires accurate threat velocity estimates [**79**] (see Figure 1.10). However, sensor **relative** measurements arrive at the TTU in the BRF with spherical coordinates.

The relationship between UAV, threat and relative motion is defined as:

$$X^{rel} = X^{threat} - X^{UAV} \tag{4.1}$$

and

$$V^{rel} = V^{threat} - V^{UAV}. \tag{4.2}$$

Where X and Y are position and velocity vectors respectively.

At this juncture it is possible to either select the BRF or NED frame for tracking. The simplest solution is to track relative motion in BRF, extract the threat's contribution (using Equations (4.1) and (4.2)) and then transform this to NED. However, things are not always simple; the literature review (in Chapter 2) revealed that relative motion in BRF depends on attitude (angular rates), unlike its projection in NED [**48**]. Therefore our objective is to track target motion in NED.

As stated before the measurements received from sensors are of relative motion in BRF. Therefore, we need to transform our sensor measurements to the NED frame. Once that transformation has been performed we have inertial **relative** measurements. The implication of using this inertial relative observation vector in the filter is that relative motion is tracked. Based on this notion, the state vector is chosen as:

$$X_k^{rel} = \left[N_k^{rel}\ \dot{N}_k^{rel}\ E_k^{rel}\ \dot{E}_k^{rel}\ D_k^{rel}\ \dot{D}_k^{rel}\ \right]^T. \tag{4.3}$$

Where N, E, D and $\dot{N}$, $\dot{E}$, $\dot{D}$ are position and velocity respectively at time instant k in the inertial NED system.

4.2 UAV and Threat Models

The first issue that needs to be addressed (referring to Figure 4.1), is the dynamic motions of the UAV and threat. The modelling of aircraft motion in state space is a very important aspect of any tracking system. The most commonly used models are the Singer acceleration model [**48**] and the near constant velocity model [**54**]. The assumption made in Chapter 1 stated that the UAV and threat are modelled as 3D point masses, travelling with constant velocity. However, in reality, while travelling with constant velocity small accelerations affect the velocity. Therefore, the near constant velocity model was chosen to model both UAV and threat dynamics. This model adds white noise with small effects to the velocity, which accounts for unpredictable modelling errors [**54**] (small random accelerations).

The model presented in [**54**] presents a 2D model which has been extended to 3D. This extension adds perturbations to a platform's Down direction. The 3D near constant velocity model is presented in what follows.

A platform's dynamics is modelled in the discrete-time Markov form:

$$X_{k+1} = \Phi X_k + \Gamma w_k. \tag{4.4}$$

X_k is the platform's state vector. w_k is the zero-mean, white, Gaussian process noise (random accelerations) with covariance Q. Φ is the state transition matrix, given as:

$$\Phi = \begin{bmatrix} 1 & \Delta T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.5}$$

Where ΔT is the sampling period.

According to the near constant velocity model, the process noise is:

$$w_k = [w_k^N \; w_k^E \; w_k^D]. \tag{4.6}$$

In [**54**] it is stated that w_k^N and w_k^E are the noisy accelerations along the N and E axes respectively, while w_k^D is the noisy velocity along the D axis. However, we wish to add noisy accelerations to the D axis as well. Therefore, the resulting process noise covariance matrix is:

$$Q = \begin{bmatrix} \Delta T^4/4 & \Delta T^3/2 & \Delta T^2/2 \\ \Delta T^3/2 & \Delta T^2/2 & \Delta T \\ \Delta T^2/2 & \Delta T & 1 \end{bmatrix}. \tag{4.7}$$

And:

$$\Gamma = \begin{bmatrix} \Delta T^2/2 & 0 & 0 \\ \Delta T & 0 & 0 \\ 0 & \Delta T^2/2 & 0 \\ 0 & \Delta T & 0 \\ 0 & 0 & \Delta T^2/2 \\ 0 & 0 & \Delta T \end{bmatrix}. \tag{4.8}$$

In order to transform position or velocity from BRF to NED, the UAV's attitude is first required. Euler 3-2-1 angles (Φ – roll, Θ - pitch, Ψ – yaw; see Figure 4.2) have been chosen to represent attitude as it is an intuitive method of describing attitude.

It is worth noting that the Euler 3-2-1 system suffers from a singularity as the pitch of a platform approaches $\pm 90°$. This phenomenon would cause a tracking filter to become unstable. Despite this fact, Euler angles were chosen to describe the platform's attitude as pitching to $\pm 90°$ is a highly unlikely event in normal flight.

Once the UAV attitude has been described with Euler angles [**81**], the transformation or direction cosine matrix (DCM) can be found which relates the two frames of reference.

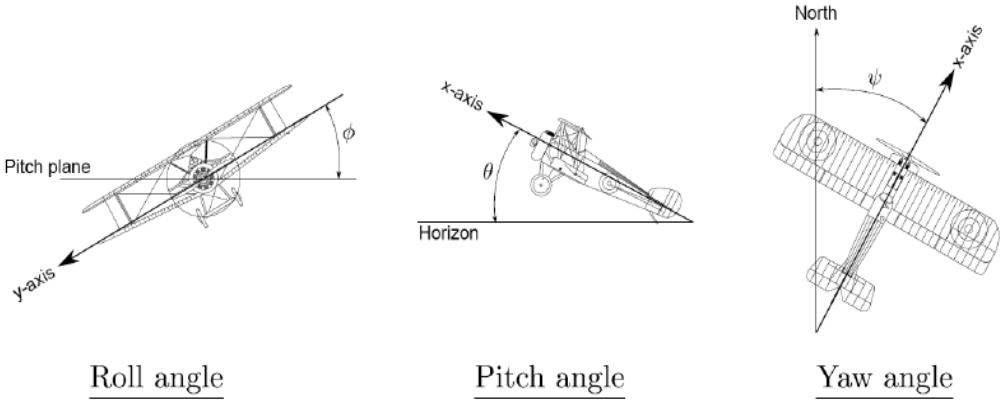

Figure 4.2: Euler angles illustration [**79**].

The DCM derived in [**82**] is:

$$DCM = \begin{bmatrix} C_\Psi C_\Theta & C_\Psi S_\Theta S_\Phi - S_\Psi C_\Phi & C_\Psi S_\Theta C_\Phi + S_\Psi S_\Phi \\ S_\Psi C_\Theta & S_\Psi S_\Theta S_\Phi + C_\Psi C_\Phi & S_\Psi S_\Theta C_\Phi - C_\Psi S_\Phi \\ -S_\Theta & C_\Theta S_\Phi & C_\Theta C_\Phi \end{bmatrix}. \quad (4.9)$$

Where $C_{\mathrm{x}} = \cos x$ and $S_{\mathrm{x}} = \sin x$

The DCM is used as follows:

$$X^{NED} = DCM \times X^{BRF}. \quad (4.10)$$

Where X^{BRF} is a 3×1 vector with respect to the BRF and X^{NED} is its projection in the inertial NED system.

4.3 Flight Scenario

Tracking of the threat is to be performed with respect to the NED system. Before the TTU can perform this function, the simulation environment needs to be set up. Furthermore, it needs to provide the TTU with all the required information in order to perform the tracking. In this context, the initial data required is illustrated in Figure 4.3. Therefore the following procedures need to be performed:

- Threat and UAV trajectory generation in NED.
- Determination of the relative dynamics in NED.
- UAV attitude calculation.
- Projection of the relative dynamics in BRF.
- Generation of radar and camera measurements.

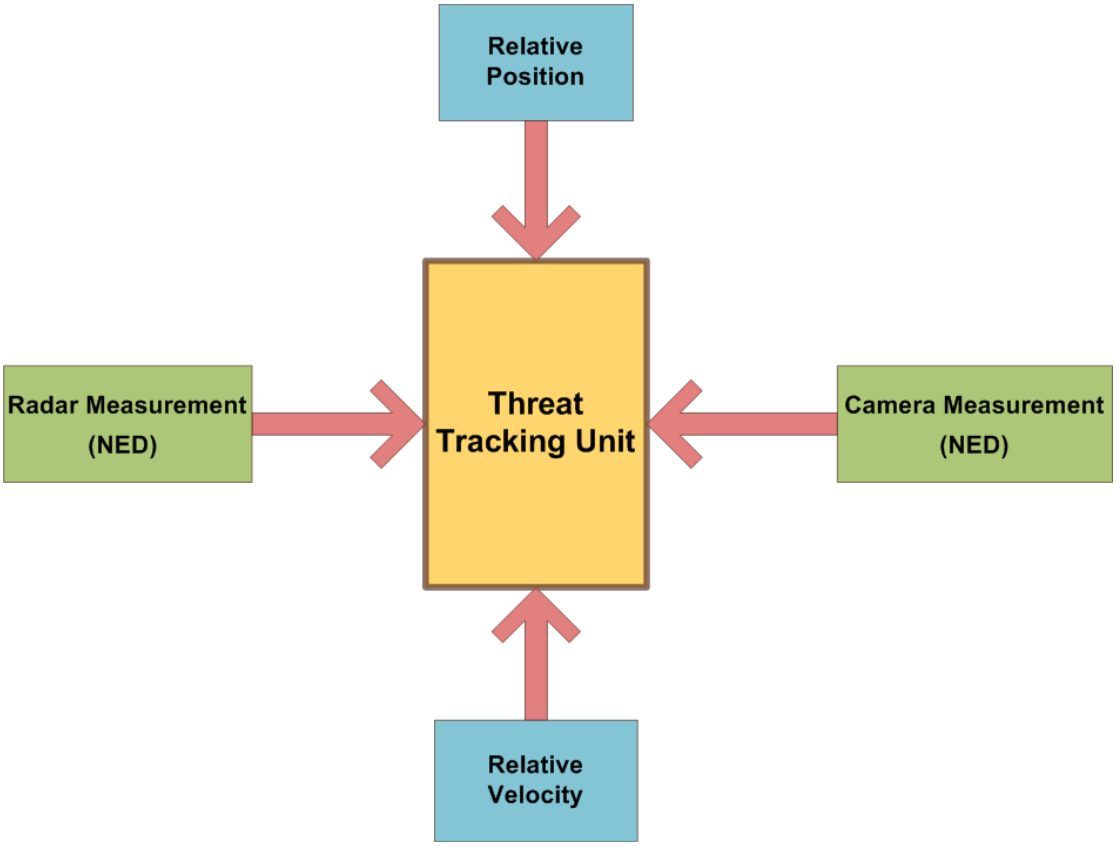

Figure 4.3: Inputs to the TTU.

For threat and UAV trajectory generation in NED we start by defining initial positions and velocities for both UAV and threat. That is defining two state vectors of Equation (4.3). The state vectors are then evolved in the state space with Equation (4.5) using the near constant velocity models. This involves generating the random vector w_k from a normal distribution with zero mean and covariance Q. For the tests that follow, the initial state vectors are presented in the table below.

Table 4.1: UAV and threat initial states.

State	Position $(\boldsymbol{N}, \boldsymbol{E}, \boldsymbol{D})$	Velocity $(\dot{\boldsymbol{N}}, \dot{\boldsymbol{E}}, \dot{\boldsymbol{D}})$
$\boldsymbol{X_k^{UAV}}$	$[0\ m,\ 0\ m,\ 0\ m]$	$[50\ m/s,\ 0\ m/s,\ 0\ m/s]$
$\boldsymbol{X_k^{threat}}$	$[30000\ m,\ 0\ m,\ 0\ m]$	$[-100\ m/s,\ 0\ m/s,\ 0\ m/s]$

The reader should note that the simulations performed in this thesis assumes VFR flight conditions as it has been proved that most NMACs (see Chapter 1) occur under these conditions. As a result the maximum speed of threats is limited to $150\ m/s$ [**79**].

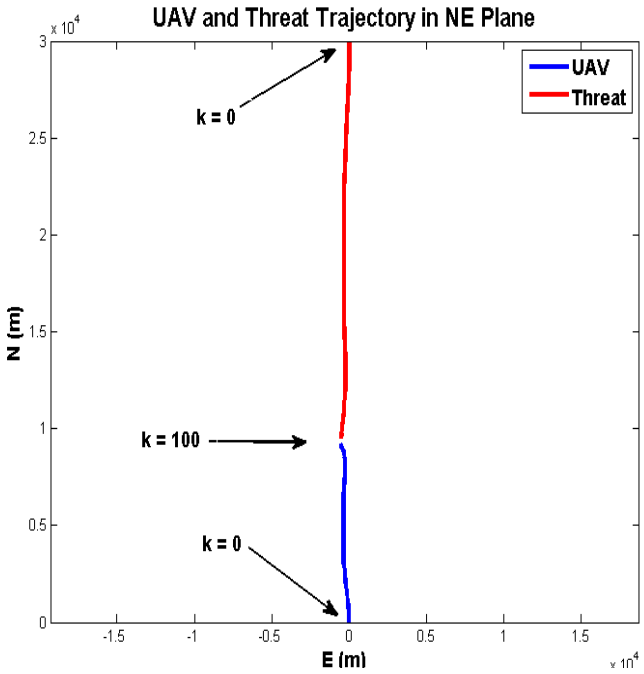

Figure 4.4: UAV and Threat flight path shown in the NE plane.

The trajectories illustrated in Figure 4.4 are the result of evolving the initial states listed in Table 4.1. The UAV starts at the origin of the NED system, at $k = 0$, flying due north; at the same time-step the threat is located $30\ km$ north of the UAV, flying due south. There is no altitude separation between the two platforms and is considered a head-on collision scenario. The simulation is run until $k = 100$. Note that the effect of the process noise (near constant velocity model) is quite evident when inspecting both platforms' trajectories.

The next step is to find the relative position and relative velocity using the trajectories generated above and the relationships in Equations (4.1) and (4.2) respectively. Following this, the UAV attitude is determined and used to

transform the relative position and velocity to the BRF using Equation (4.10). The attitude is determined as follows:

$$\Psi = \tan^{-1}\left(\frac{\dot{E}}{\dot{N}}\right) \tag{4.11}$$

and

$$\Theta = -\tan^{-1}\left(\frac{\dot{D}}{\sqrt{\dot{N}^2 + \dot{E}^2}}\right). \tag{4.12}$$

With relative motion now modelled in the BRF, sensor measurements are formed by first sampling the true relative trajectory (according to sensor data rate); followed by a conversion to spherical space as that is system in which the sensor operates. Noise is then added akin to the method of determining the process noise, but with covariance R. The sensor measurements are now relative motion in spherical coordinates with respect to BRF; exactly how a sensor reports measurements. However, we require the sensor measurements in relative spherical with respect to the NED system. In order to transform the radar position measurements ($\rho_k^R, \theta_k^R, \varphi_k^R$) to NED we could use the DCM, however this would first require a conversion to Cartesian coordinates which would introduce more errors into the measurements. The radar range rate ($\dot{\rho}_k^R$) does not require a transformation as it is the same in NED as it is in its measure BRF.

In order to transform the camera measurements (θ_k^C, φ_k^C) to NED we cannot use the method described above. The conversion to Cartesian coordinates requires the range to the threat; of which the camera has no knowledge. Therefore another method is proposed that addresses the camera measurement deficiency and the potential injection of errors into the radar measurement. It allows the direct transformation from BRF to NED. First the direction cosines are found using sensor measurements:

$$\Lambda_1 = \cos\theta_k^{BRF} \cos\varphi_k^{BRF} \tag{4.13}$$

$$\Lambda_2 = \sin\theta_k^{BRF} \cos\varphi_k^{BRF} \tag{4.14}$$

$$\Lambda_3 = -\sin\varphi_k^{BRF} \tag{4.15}$$

Then the direction cosines along the N, E and D axes are determined with:

$$\begin{bmatrix} \Lambda_N \\ \Lambda_E \\ \Lambda_D \end{bmatrix} = DCM \begin{bmatrix} \Lambda_1 \\ \Lambda_2 \\ \Lambda_3 \end{bmatrix}. \tag{4.16}$$

Now the stabilised azimuth and elevation angles can be determined by the method described by Zollo and Ristic [**53**] in Chapter 2 using Equations (2.48) to (2.50). Similarly, for the radar observation vector, the relative range with respect to the NED system is determined using Equations (2.51) and (2.52). In summation, the entire process of initialising required parameters is illustrated in Figure 4.5.

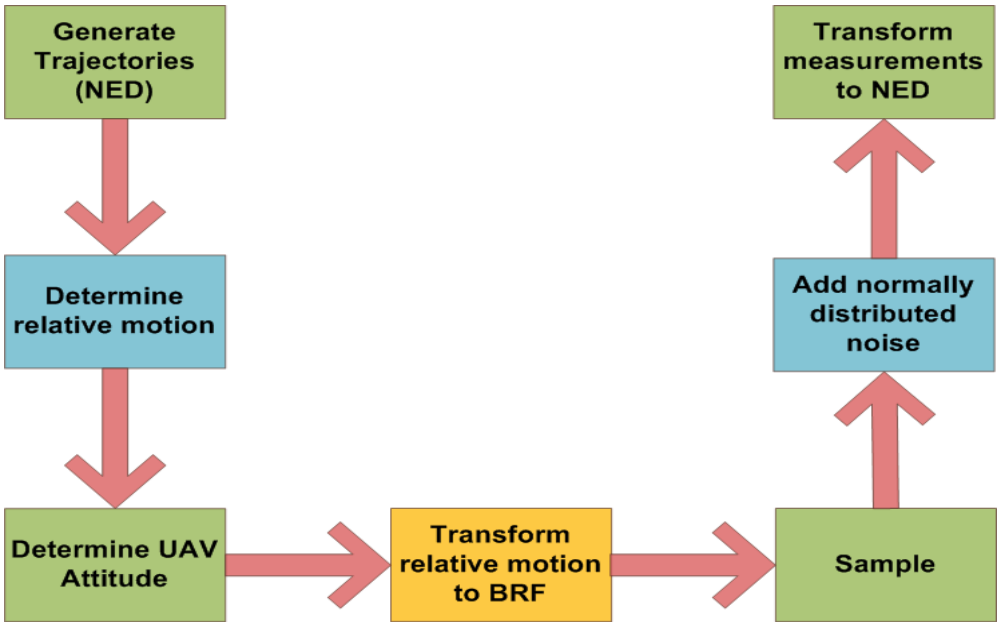

Figure 4.5: Initial simulation setup procedure.

A cross check with Figure 4.3 confirms that all components in generating the flight scenario have been addressed. Now that the TTU has all its required inputs, it needs to produce required outputs, viz. threat position and velocity estimates in the inertial NED system. In order to achieve this, multiple fusion architectures with their corresponding tracking filters are designed in Chapter 5 from which the best is selected for the final architecture.

Chapter 5: TTU Design

5.1 Introduction

This chapter describes the design of the TTU using MSDF. Thesis *objective 5* is to “Design and implement an airborne target tracking algorithm using multisensor data fusion”. However, before developing the fusion architectures using multiple redundant sensors, single sensor tracking filters are developed to illustrate their accuracies and set a baseline with which the filters resulting from MSDF can be evaluated against. Furthermore, the single sensor filters that are developed form the building blocks from which the fusion architectures are made up. Thus, this chapter begins by first developing EO camera and radar only tracking filters. Two cases for the radar only tracking filter are discussed: ranging radar and Doppler radar. In order to illustrate the filter complexity versus performance gain trade-off, nonlinear as well as linear ranging radar tracking filters are developed. As stated in Chapter 2, the complication in tracking with Doppler information is that measurement errors in range and rate are correlated [**65**]. However, a common approach is to disregard this fact. Therefore, two Doppler radar filters are developed: one which assumes measurement errors in range and range rate are not correlated; and the other assumes this correlation exists and is accounted for in the filter.

Following this, multiple fusion architectures and their respective tracking filters are designed. In order to evaluate the performance of the fusion architectures, Monte Carlo simulations are performed. The reason for multiple fusion architectures being developed is so that quantitative evaluation can occur and the best fusion architecture for airborne tracking using MSDF can be exposed; which will form the *Threat Tracking Unit* block in the system overview (Figure 1.10). This chapter ensures the accomplishment of thesis *objective 5*. Furthermore, the results from this chapter will see either the proving or disproving of the hypotheses set out in Chapter 3.

After a filter has been developed, it is integrated into the simulation environment described in Chapter 4 and then tested as set out in the Methodology. For each filter, first the results from a single filter run of k time-steps are presented. Thereafter Monte Carlo simulations are performed. The results of 500 independent Monte Carlo runs (N^{MC}) over k steps are then presented, illustrating the performance metrics detailed in section 3.5.

Table 5.1 lists all the tracking filters that are developed in this chapter and the estimators which they consist of.

Table 5.1: Designed tracking filters and the estimators on which they are built.

Design	Radar Local Filter	Camera Local Filter	Fusion Centre Filter
EO camera filter		EKF	
Nonlinear range radar filter	EKF		
Linear range radar filter	KF		
Uncorrelated Doppler radar filter	EKF		
Correlated Doppler radar filter	EKF		
FA1 (centralised)			EKF
FA2 (distributed)	EKF	EKF	
FA3 (distributed)	KF	EKF	

5.2 Common Filter Parameters

This section presents information that is required in the design of all the filters discussed in this chapter. In what follows, the superscripts R, C and F denote that the component belongs to either the radar, EO camera or fused subsystem, respectively.

For each filter the relative motion between UAV and threat is modelled in the discrete time Markov form:

$$X_{k+1}^{R} = \Phi X_{k}^{R} + \Gamma w_{k}, \tag{5.1}$$

and

$$X_{k+1}^{C} = \Phi X_{k}^{C} + \Gamma w_{k}. \tag{5.2}$$

Where the state vectors are:

$$X_{k}^{R} = X_{k}^{C} = X_{k}^{F} = \left[N_{k}^{rel}\ \dot{N}_{k}^{rel}\ E_{k}^{rel}\ \dot{E}_{k}^{rel}\ D_{k}^{rel}\ \dot{D}_{k}^{rel}\ \right]^{T}. \tag{5.3}$$

The state vectors contain the components of relative motion with respect to the inertial NED frame of reference.

Then the threat's position and velocity in NED are determined by the sum of the estimated state vector in Equation (5.3) and the UAV's current position and velocity – possibly provided by an IMU.

The state transition and noise matrices are given as:

$$\Phi = \begin{bmatrix} 1 & \Delta T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\Gamma = \begin{bmatrix} \Delta T^2/2 & 0 & 0 \\ \Delta T & 0 & 0 \\ 0 & \Delta T^2/2 & 0 \\ 0 & \Delta T & 0 \\ 0 & 0 & \Delta T^2/2 \\ 0 & 0 & \Delta T \end{bmatrix}.$$

With $\Delta T^R = 2$ and $\Delta T^C = 1/2$. That is, the radar provides a new measurement every $2\ s$ and the EO camera every $500\ ms$. Therefore, the camera provides measurements to the TTU four times faster than the radar does. This translates to radar and camera measurement frequencies of $0.5\ Hz$ and $2\ Hz$, respectively. Note that cameras generally provide measurements at higher frequencies than this, however this value was chosen to account for the image processing time required to successfully detect a target.

The acceleration disturbances for both UAV and threat in the north, east and down directions are set to:

$$w_k = [0.05\ m/s^2\ 0.05\ m/s^2\ 0.05\ m/s^2]. \quad (5.4)$$

As all of the filters are performing airborne tracking, state estimates of relative motion are produced. Thus, the final step, in order to extract the states of interest (of the threat), is to add the UAV's position and velocity in NED to the state vector estimate. This then yields the estimate of only the threat position and velocity in the inertial NED system.

The relative range for the flight scenario under review is illustrated in Figure 5.1.

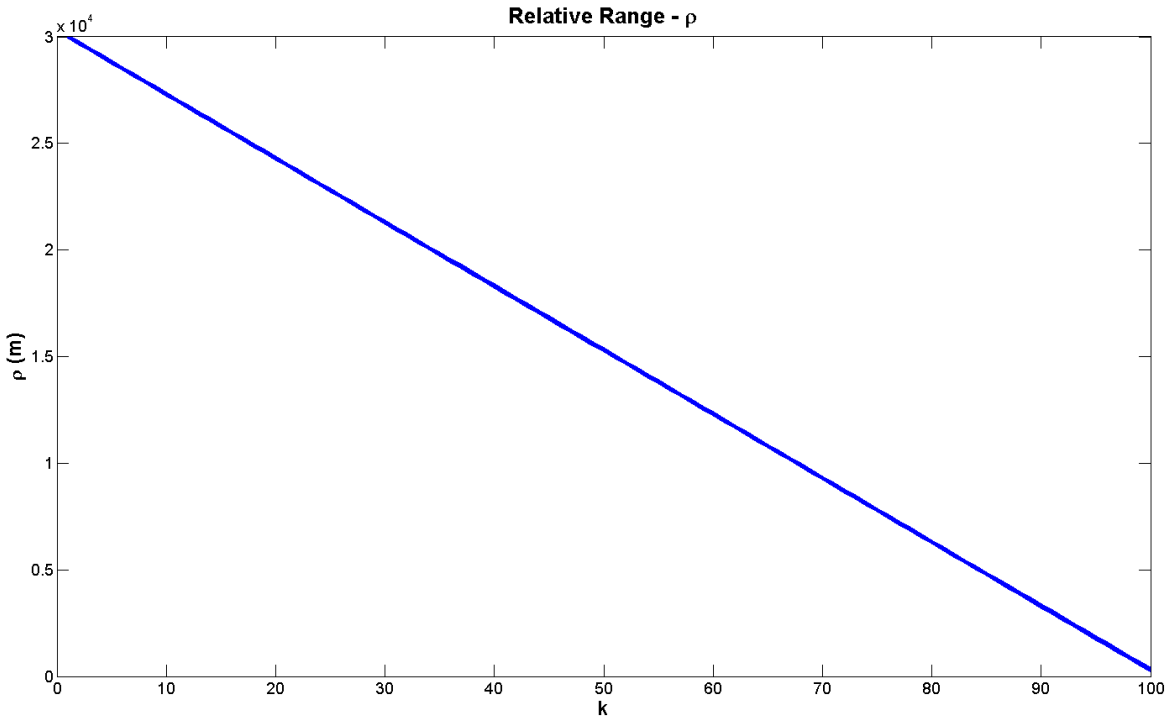

Figure 5.1: Relative range of flight scenario.

5.3 EO Camera Tracking Filter

In our progressive design structure, leading up to the development of tracking filters using multiple redundant sensors, the first tracking filter developed is that of the EO camera. As the relative motion dynamics has been described in Cartesian coordinates, tracking is performed with mixed coordinates; the filter uses the camera's measurements directly (raw). Therefore a nonlinear EKF is selected as the tracking filter.

With the EO camera measuring only target relative azimuth and elevation, its observation vector is:

$$z_k^C = [\theta_k^C \quad \varphi_k^C]^T + v_k^C. \tag{5.5}$$

Where v_k^C is the zero-mean, white, Gaussian camera measurement noise with covariance matrix R^C:

$$R^C = \begin{bmatrix} \sigma_\theta^{R^2} & 0 \\ 0 & \sigma_\varphi^{R^2} \end{bmatrix}. \tag{5.6}$$

Where σ_θ and σ_φ are the measurement errors in azimuth and elevation respectively; whose values are shown below.

Table 5.2: EO camera measurement errors.

Sensor	Measurement Error	Value
EO Camera	σ_θ^C	$0.02\ radians$
	σ_φ^C	$0.02\ radians$

The measurement errors tabulated in Table 5.2 imply camera measurement errors in azimuth and elevation of 1.15°. These small angular measurement errors are commonly found in EO cameras [**48**]. It will be evident in the following sections that the camera angular errors are a great deal smaller than those of the radar's. These accurate measurements can be confirmed by plotting the true relative azimuth and elevation against the camera measured values (see Figure 5.2 and Figure 5.3).

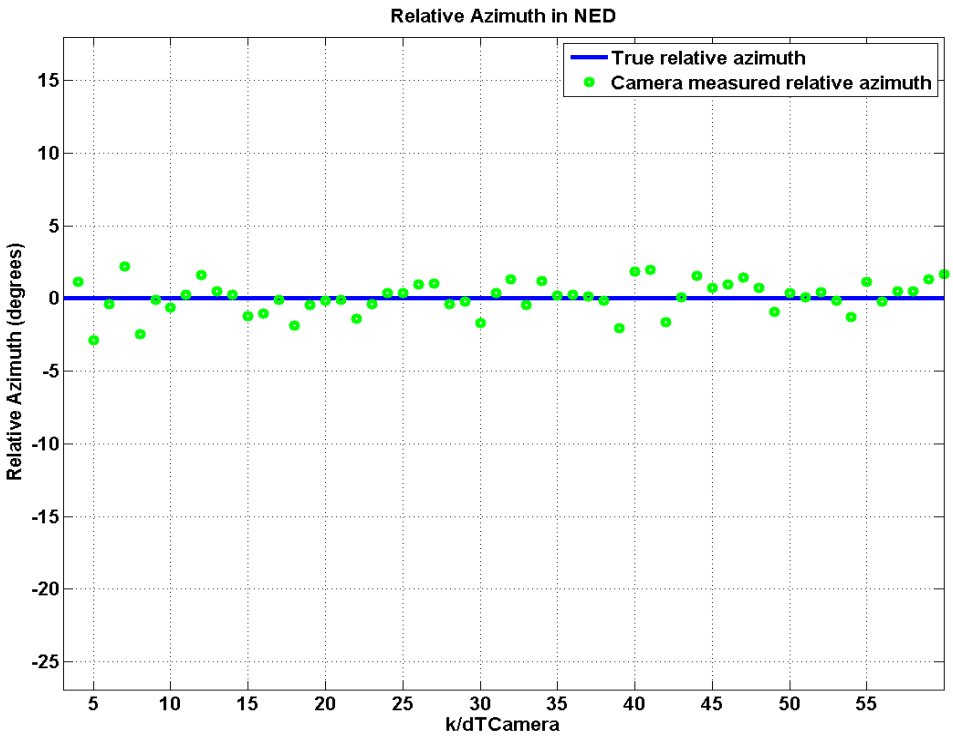

Figure 5.2: Relative azimuth in NED - truth and camera measured values.

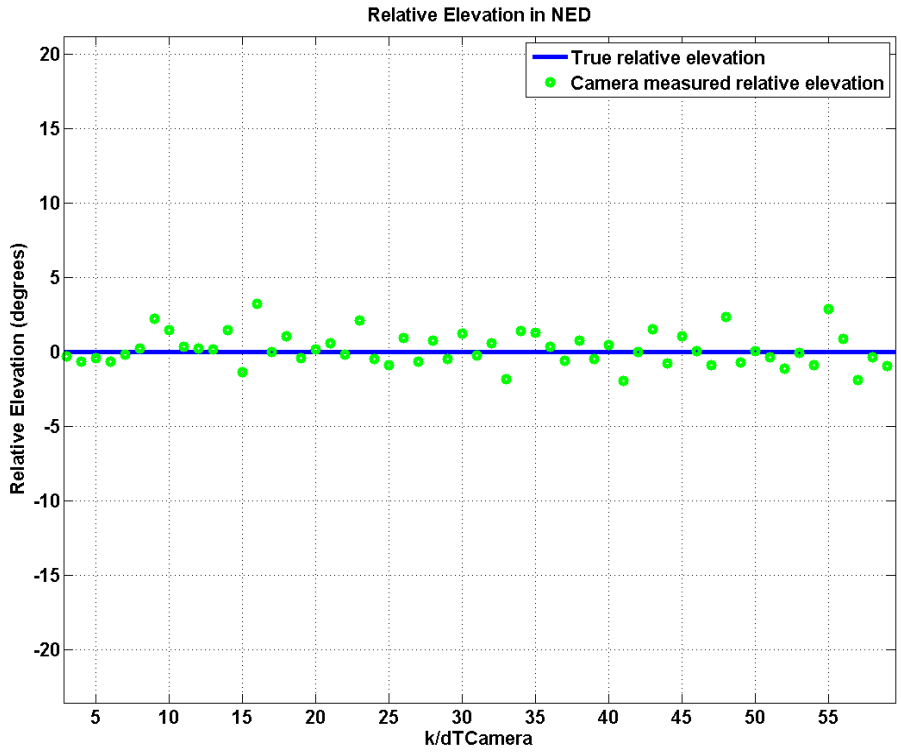

Figure 5.3: Relative elevation in NED - truth and camera measured values.

The first solution is tracking in mixed coordinates. This implies the camera's measurements must be modelled in the sensor frame as the platform's dynamics have been modelled in the Cartesian space.

Figure 5.4: Bearings-only tracking with an EKF.

Standard EKF equations are used in order to track the target. This involves first *prediction* as that in Equations (2.26) and (2.27), and *correction* performed by Equations (2.23) to (2.25).

In order to perform the update phase, the linearized measurement matrix is determined using a first-order Taylor series approximation around the current state estimate as follows:

$$h^C(X) = z_k^C = \begin{bmatrix} \tan^{-1}\left(\frac{E}{N}\right) \\ -\sin^{-1}\left(\frac{D}{\sqrt{N^2+E^2+D^2}}\right) \end{bmatrix}, \tag{5.7}$$

$$H_X^C(k) = \left.\frac{\partial h^C(X)}{\partial X}\right|_{X=\hat{X}_{k|k-1}^C}. \tag{5.8}$$

See Appendix A1 for the complete linearized measurement matrix.

Now the standard EKF equations are used to track the threat. Using the linearized measurement matrix the innovation covariance is calculated and hence the Kalman gain using Equation (2.23). Then the innovation and hence the state estimate is determined by Equation (2.24). Following this the state estimate covariance matrix is produced from Equation (2.25). The results from this filter are illustrated in the following section.

The reader should note that the EO camera was assumed to be able to detect the target at all times during the simulation; that is, at all target ranges. The reason for exaggerating the camera's detection range is twofold: first, so that the radar and camera filters can be easily compared against one another; and second, to make the positive effects of fusion more apparent. The camera's detection range is considerably smaller than the radar's, and, had it been taken into account, the full benefits of fusion would only be visible in the last few iterations of the simulation.

5.3.1 Results

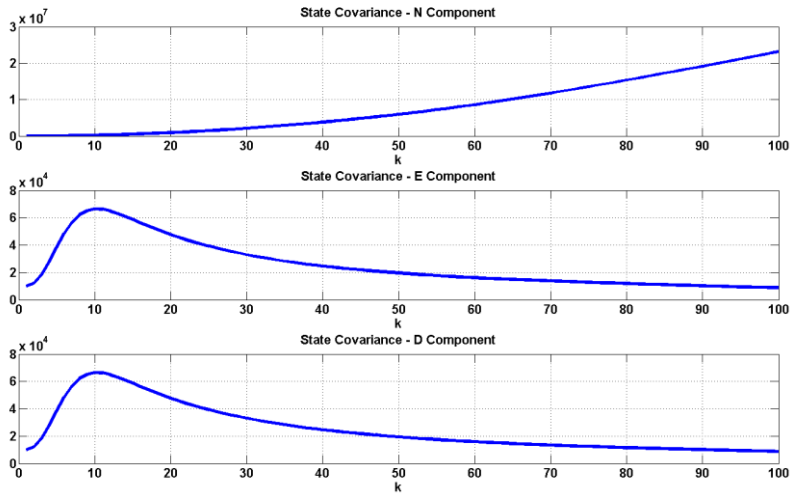

Figure 5.5: Camera filter - estimated state error covariance of N, E and D components.

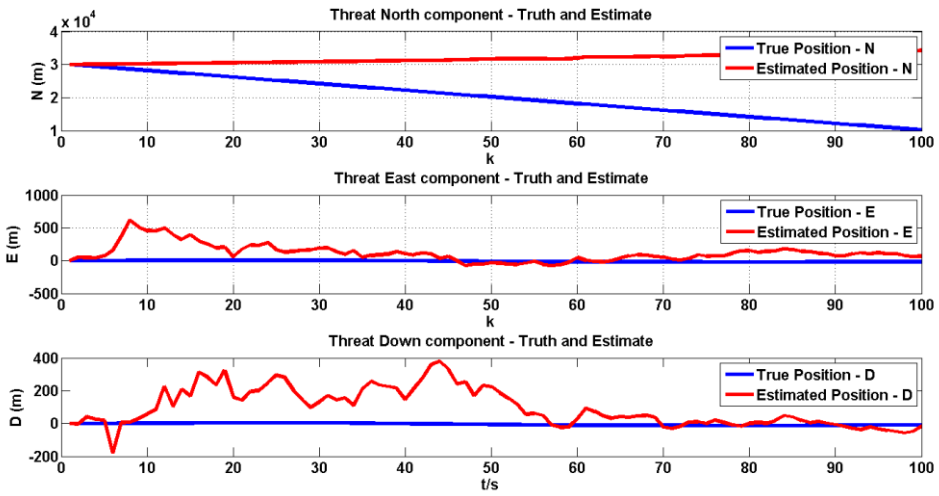

Figure 5.6: Camera filter - True and estimated threat position in N, E and D directions.

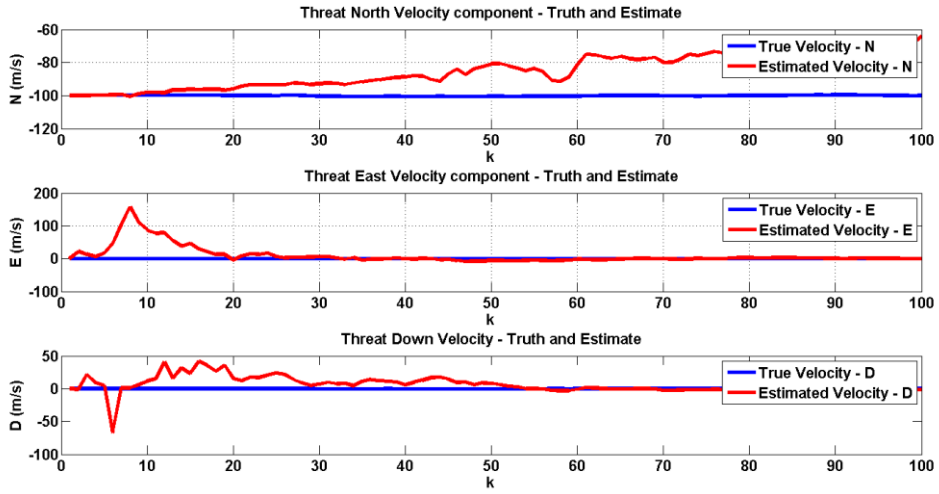

Figure 5.7: Camera filter - True and estimated threat velocity in N, E and D directions.

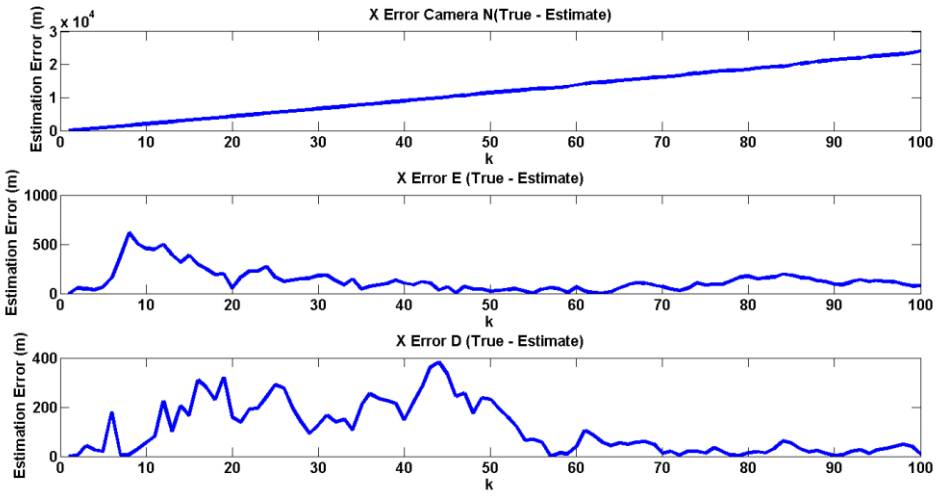

Figure 5.8: Camera filter - Estimation error in N, E and D directions.

5.3.2 Monte Carlo Simulations

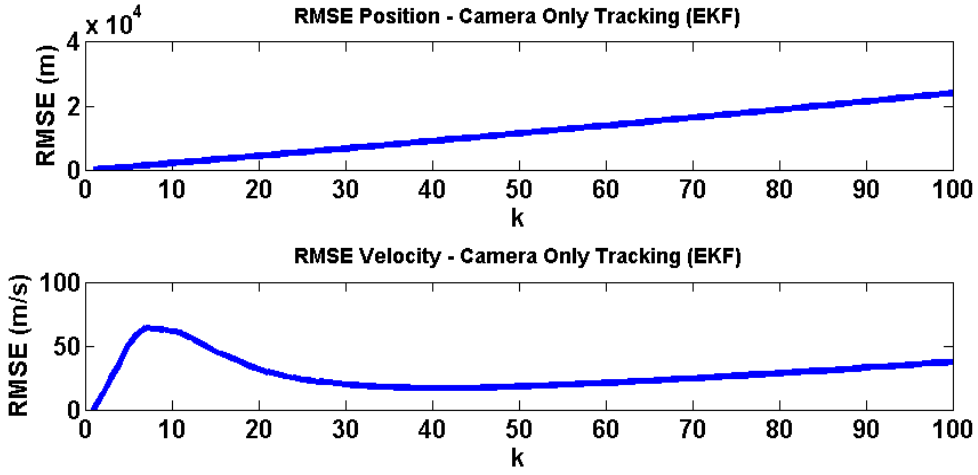

Figure 5.9: Camera only filter RMSE in position and velocity.

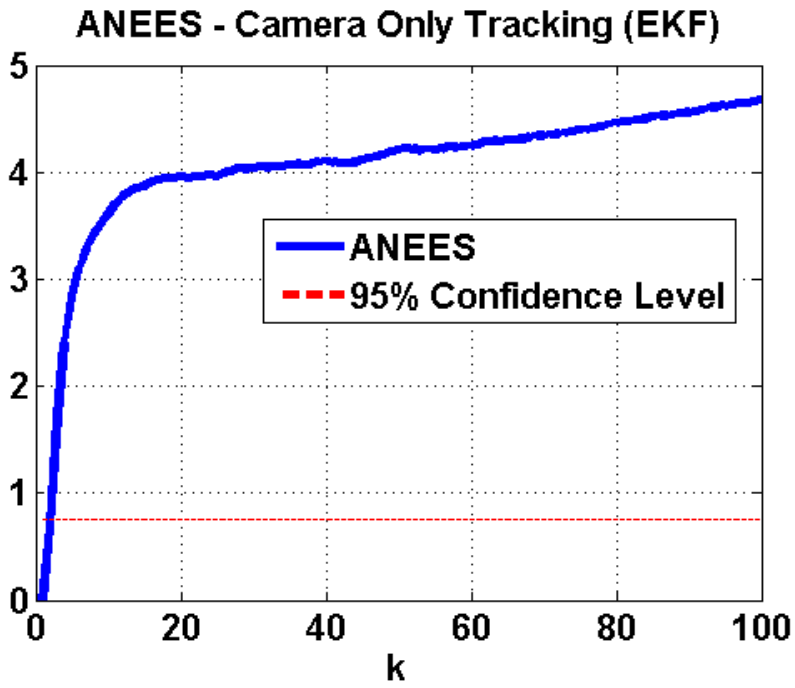

Figure 5.10: Camera only filter ANEES.

5.3.3 Summary

In summary, it is evident that the EO camera filter diverges. The results show that the threat position and velocity in the east and down directions are adequately tracked, but neither in the north direction. Therefore the EO camera tracking filter cannot suitably track the threat using angle-only information.

5.4 Ranging Radar Tracking Filter

For a ranging radar-only tracker we develop two filters. The first filter uses radar measurements exactly as they arrive from the radar, therefore necessitating the use of a nonlinear EKF. The second design utilises a linear filter to produce the state estimate; radar measurements are converted to Cartesian space and then debiased. Furthermore, the radar measurement noise matrix is projected from the sensor frame (BRF) to the Cartesian system.

The radar measures relative range, azimuth and elevation. Therefore the corresponding observation vector is:

$$z_k^R = [\rho_k^R \quad \theta_k^R \quad \varphi_k^R]^T + v_k^R. \tag{5.9}$$

Where v_k^R is the zero-mean, white, Gaussian radar measurement noise with covariance matrix R^R:

$$R^R = \begin{bmatrix} \sigma_\rho^{R^2} & 0 & 0 \\ 0 & \sigma_\theta^{R^2} & 0 \\ 0 & 0 & \sigma_\varphi^{R^2} \end{bmatrix}. \tag{5.10}$$

Where σ_ρ, σ_θ and σ_φ are the measurement errors in range, azimuth and elevation respectively; whose values are shown below.

Table 5.3: Radar measurement errors.

Sensor	Measurement Error	Value
Doppler Radar	σ_ρ^R	$200\ m$
	σ_θ^R	$0.3\ radians$
	σ_φ^R	$0.3\ radians$

The measurement errors above are exaggerated and, in practical radars, would be significantly smaller. However, these values were chosen to illustrate the effects of MSDF when one sensor's measurements are very accurate and the other's not. The accurate sensor's measurements are that of the EO camera's listed in Table 5.2.

The effects of the radar's measurement errors are displayed in the following figures which contain the relative true and radar measured range, azimuth and elevation in NED respectively.

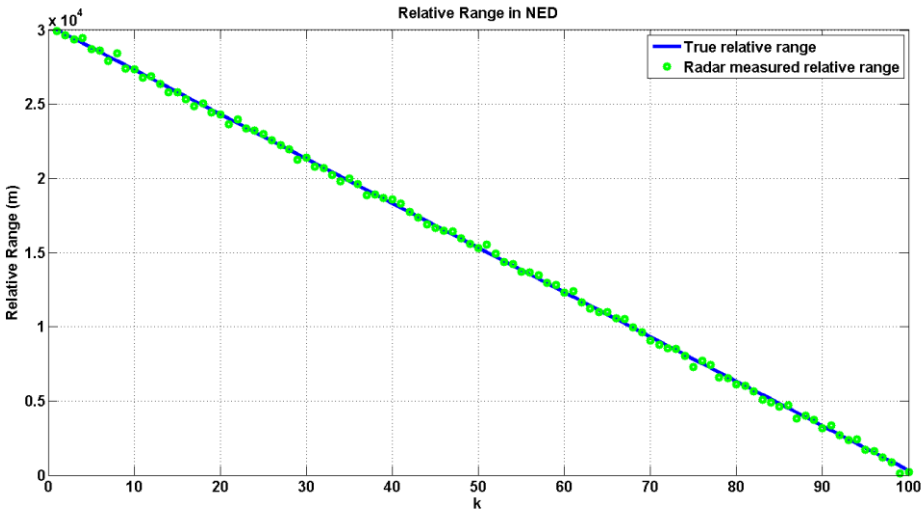

Figure 5.11: Relative range in NED - truth and radar measured values.

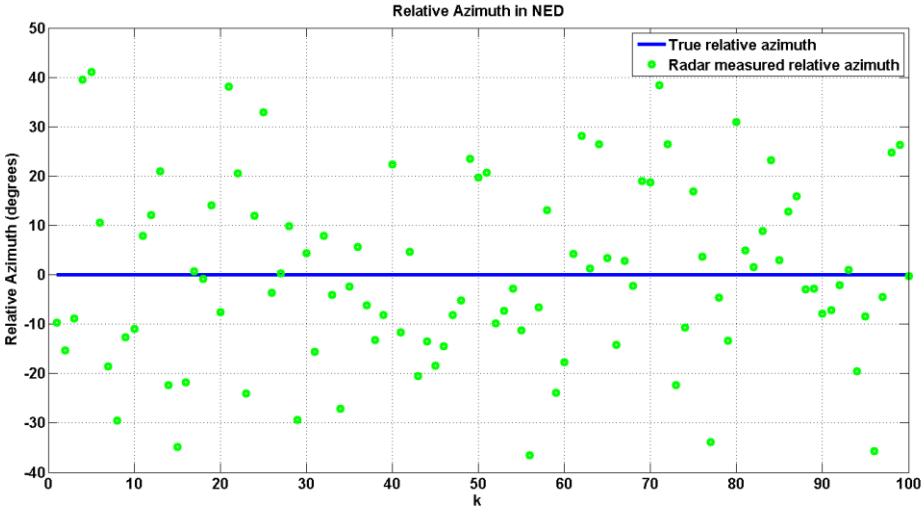

Figure 5.12: Relative azimuth in NED - truth and radar measured values.

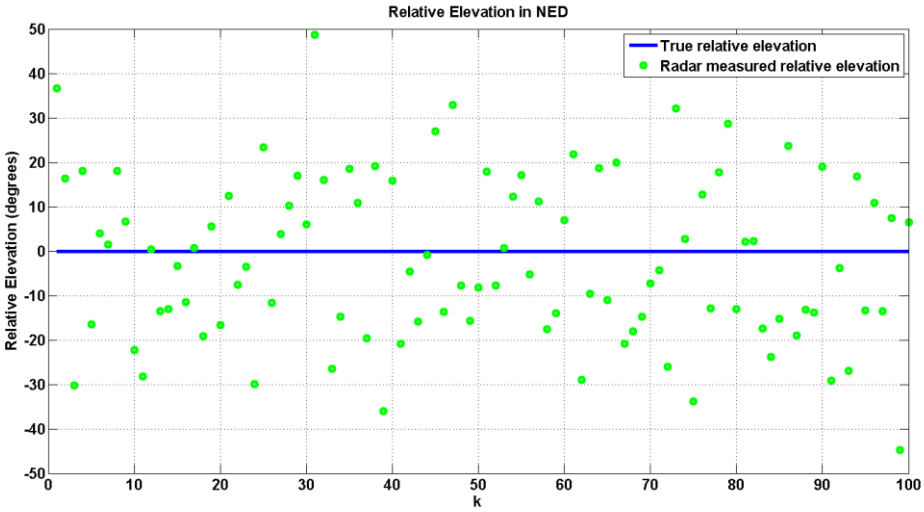

Figure 5.13: Relative elevation in NED - truth and radar measured values.

5.4.1 Nonlinear Approach

The first solution, much like our bearings-only tracker, involves tracking in mixed coordinates. As the radar's measurements are nonlinear combinations of the state variables, a nonlinear filter is required in order to perform tracking. We have selected the EKF in order to achieve this.

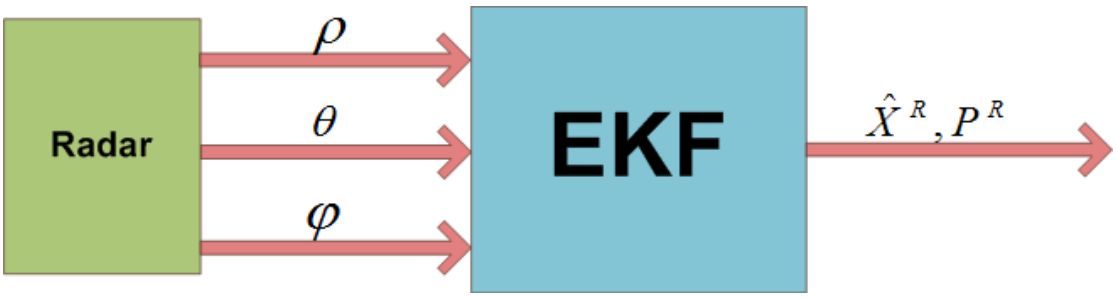

Figure 5.14: Nonlinear ranging radar tracking filter.

The linearized radar measurement matrix is found by calculating the Jacobian of the observation vector with respect to the state variables:

$$h^R(X) = z_k^R = \begin{bmatrix} \sqrt{N^2 + E^2 + D^2} \\ \tan^{-1}\left(\frac{E}{N}\right) \\ -\sin^{-1}\left(\frac{D}{\sqrt{N^2 + E^2 + D^2}}\right) \end{bmatrix}, \tag{5.11}$$

$$H_X^R(k) = \left.\frac{\partial h^R(X)}{\partial X}\right|_{X=\hat{X}_{k|k-1}^R}. \tag{5.12}$$

See Appendix A2 for the complete linearized measurement matrix.

Now we are able to produce state estimates using the standard EKF equations. The results of this filter are shown below.

5.4.1.1 Results

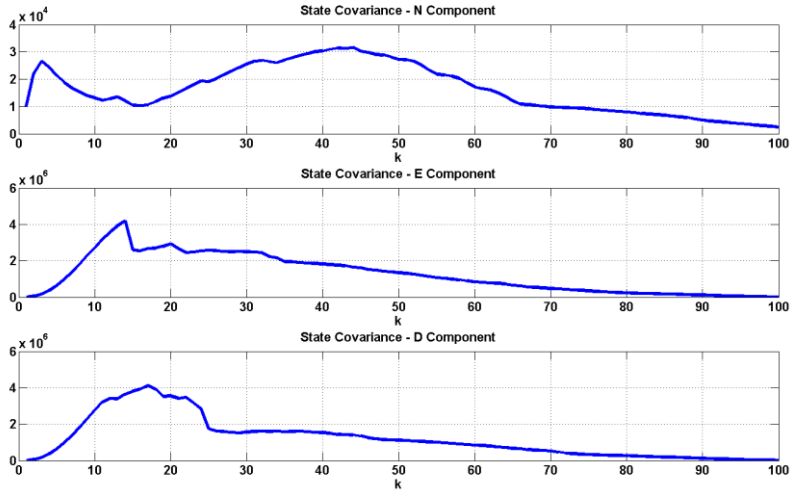

Figure 5.15: Radar nonlinear filter (EKF) – Estimated state error covariance of N, E and D position components.

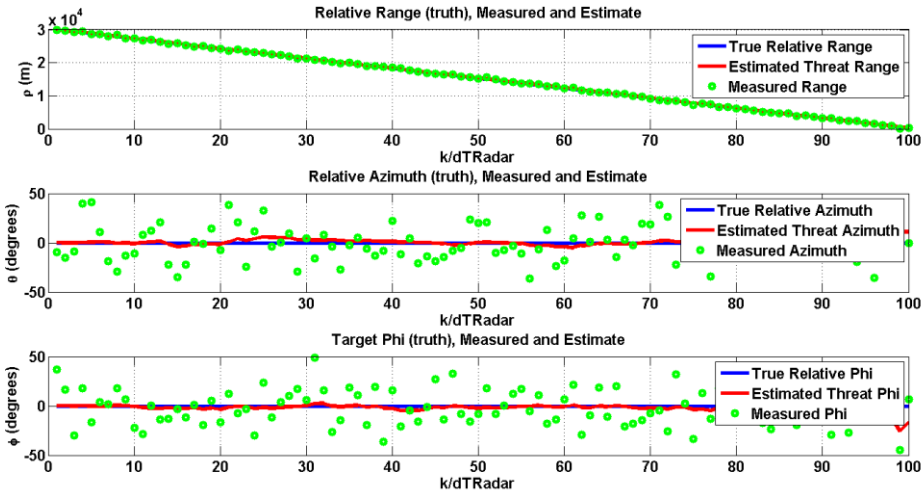

Figure 5.16: Radar nonlinear filter (EKF) - True, estimated and measured relative motion range, azimuth and elevation in the BRF.

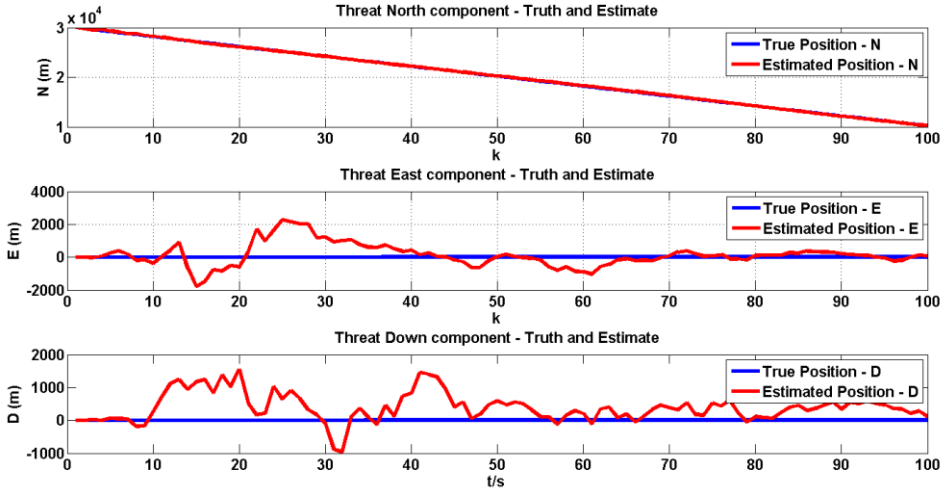

Figure 5.17: Radar nonlinear filter (EKF) – True and estimated threat position in N, E and D directions.

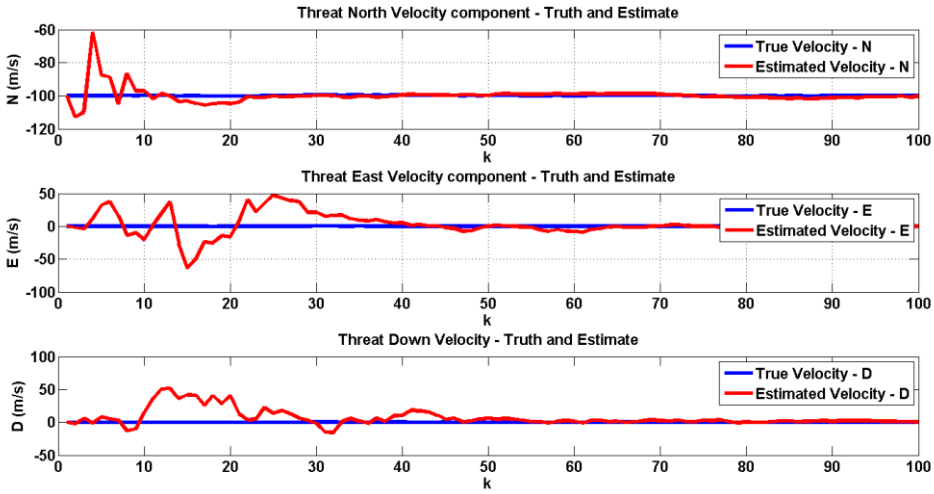

Figure 5.18: Radar nonlinear filter (EKF) – True and estimated threat velocity in N, E and D directions.

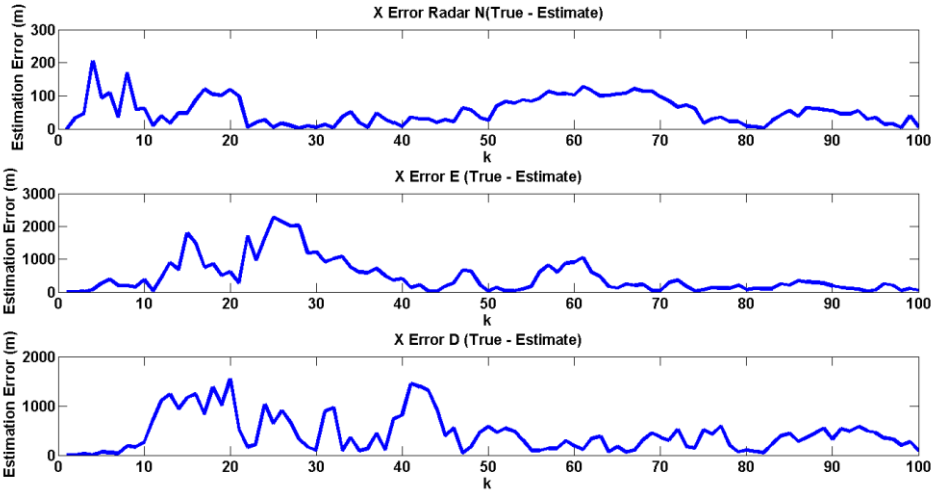

Figure 5.19: Radar nonlinear filter (EKF) – Position estimation error in N, E and D directions in a single flight scenario.

5.4.1.2 Monte Carlo Simulations

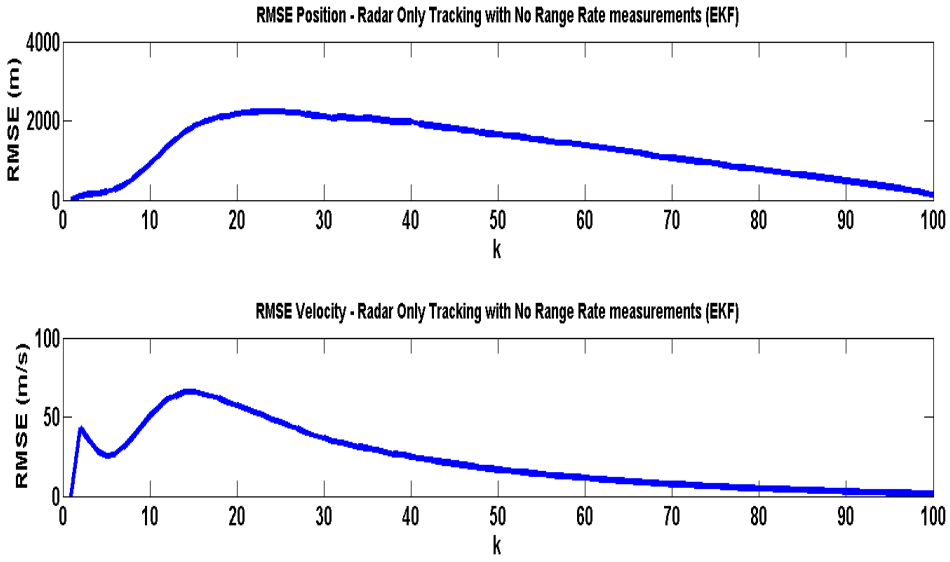

Figure 5.20: Radar nonlinear filter with no range rate measurements - RMSE in position and velocity.

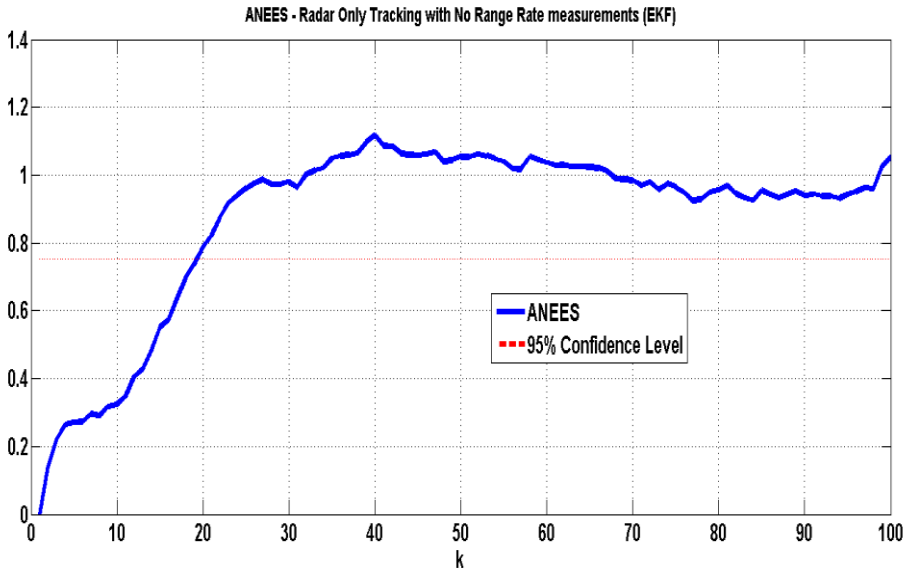

Figure 5.21: Radar filter with no range rate measurements - ANEES.

5.4.2 Linear Approach

The second filter developed, in contrast to all the filters presented thus far, is one that performs tracking in Cartesian coordinates only. First measurements are converted to Cartesian coordinates and then debiased. Concurrently the radar measurement matrix is determined in Cartesian space. These inputs feed a linear KF which utilise Equations (2.14) to (2.18) to produce estimates of relative position and velocity. The tracking filter is illustrated below.

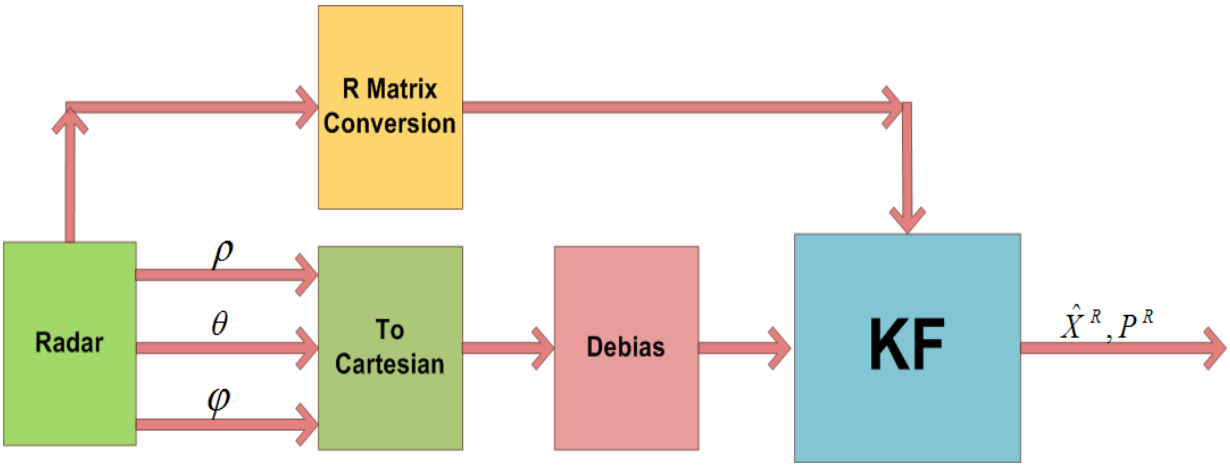

Figure 5.22: Second range radar filter developed. A linear solution which involves measurement conversion to Cartesian coordinates.

Radar relative measurements originate in spherical coordinates with respect to the BRF. However, in section 4.3 they were transformed to the NED system. Therefore the first step is to convert Equation (5.9) to Cartesian coordinates using the following equations:

$$N_k' = \rho_k \cos\theta_k \cos\varphi_k \tag{5.13}$$

$$E_k' = \rho_k \sin\theta_k \cos\varphi_k \tag{5.14}$$

$$D_k' = -\rho_k \sin\varphi_k \tag{5.15}$$

It was shown in Chapter 2 that converting measurements to Cartesian coordinates and using this in a tracking filter, leads to biased and inconsistent state estimates. Therefore, before tracking, the measurements need to be debiased. If measurements of range, azimuth and elevation are zero-mean, Gaussian and no correlations exist in measurement errors, then debiasing of the converted measurements is performed as follows [**53**]:

$$\begin{bmatrix} N_k \\ E_k \\ D_k \end{bmatrix} = \begin{bmatrix} N_k' \left[1 - \left(e^{-\sigma_\theta^2} e^{-\sigma_\varphi^2} - e^{-\frac{\sigma_\theta^2}{2}} e^{-\frac{\sigma_\varphi^2}{2}}\right)\right] \\ E_k' \left[1 - \left(e^{-\sigma_\theta^2} e^{-\sigma_\varphi^2} - e^{-\frac{\sigma_\theta^2}{2}} e^{-\frac{\sigma_\varphi^2}{2}}\right)\right] \\ D_k' \left[1 - \left(e^{-\sigma_\varphi^2} - e^{-\frac{\sigma_\theta^2}{2}}\right)\right] \end{bmatrix}. \tag{5.16}$$

Now that we have the converted observation from Equation (5.16) we need to transform R^R (the measurement covariance matrix in the radar local spherical frame) to the Cartesian frame as well.

By finding the Jacobian of Equations (5.13) to (5.15) with respect to the state vector in Equation (5.3), we arrive at:

$$A = \begin{bmatrix} \cos\theta_k^R \cos\varphi_k^R & -\rho_k^R \cos\theta_k^R \sin\varphi_k^R & -\rho_k^R \sin\theta_k^R \cos\varphi_k^R \\ \sin\theta_k^R \cos\varphi_k^R & -\rho_k^R \sin\theta_k^R \sin\varphi_k^R & \rho_k^R \cos\theta_k^R \cos\varphi_k^R \\ \sin\varphi_k^R & \cos\varphi_k^R & 0 \end{bmatrix}. \tag{5.17}$$

Since the transformation A from radar coordinates to Cartesian is a linear one, we can make the following statement about R^R in Cartesian space:

$$R^{RConv} = AR^R A^T. \tag{5.18}$$

Using the above information the standard KF equations are used to produce state estimates and its associated estimated state covariance. The results from this approach are illustrated below.

5.4.2.1 Results

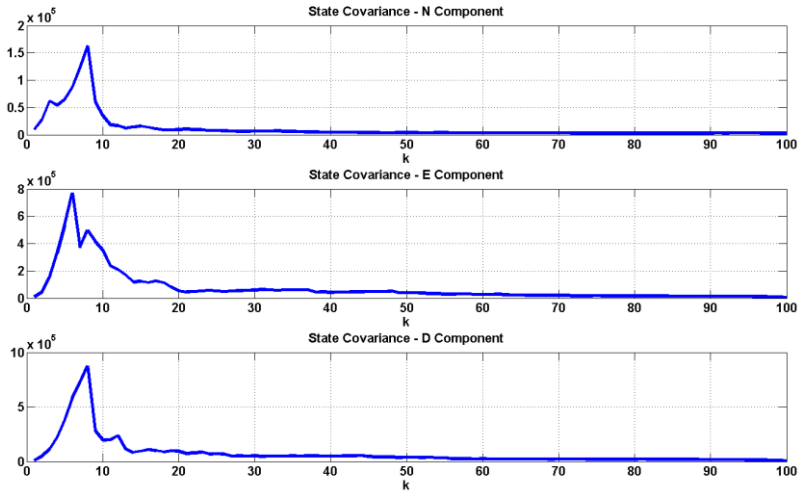

Figure 5.23: Radar linear KF tracking in Cartesian coordinates – estimated state error covariance.

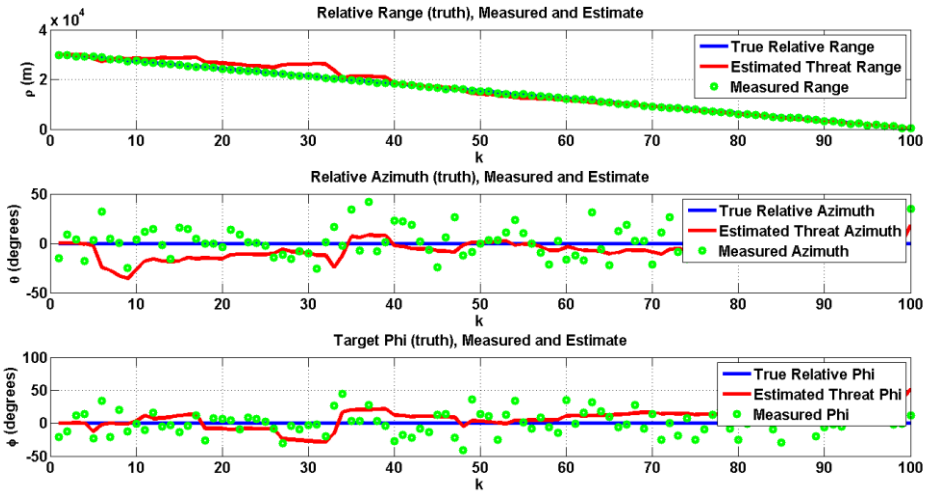

Figure 5.24: Radar linear KF tracking in Cartesian coordinates – True, estimated and measured relative range, azimuth and elevation in BRF.

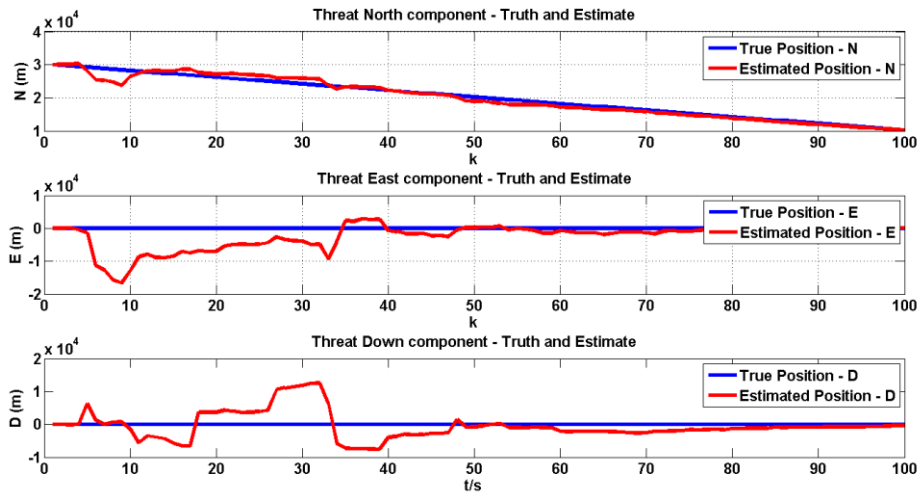

Figure 5.25: Radar linear KF tracking in Cartesian coordinates – True and estimated position in N, E and D directions.

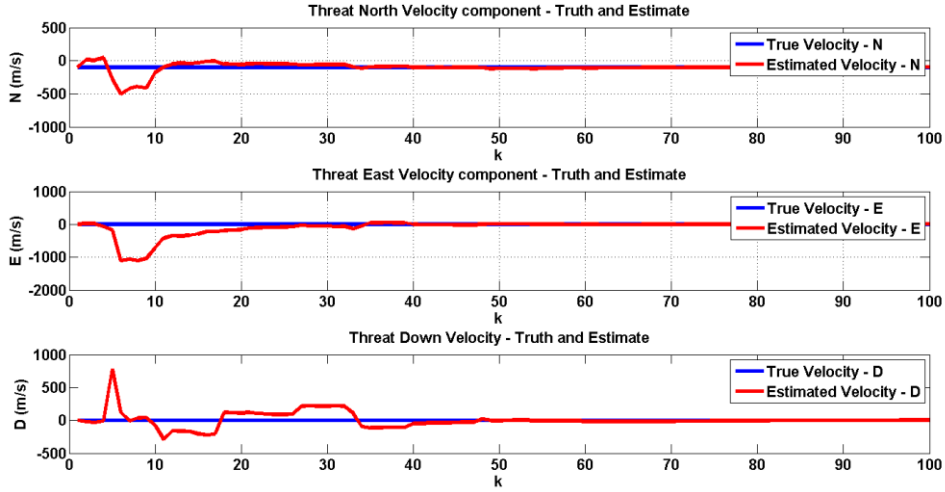

Figure 5.26: Radar linear KF tracking in Cartesian coordinates – True and estimated velocity in N, E and D directions.

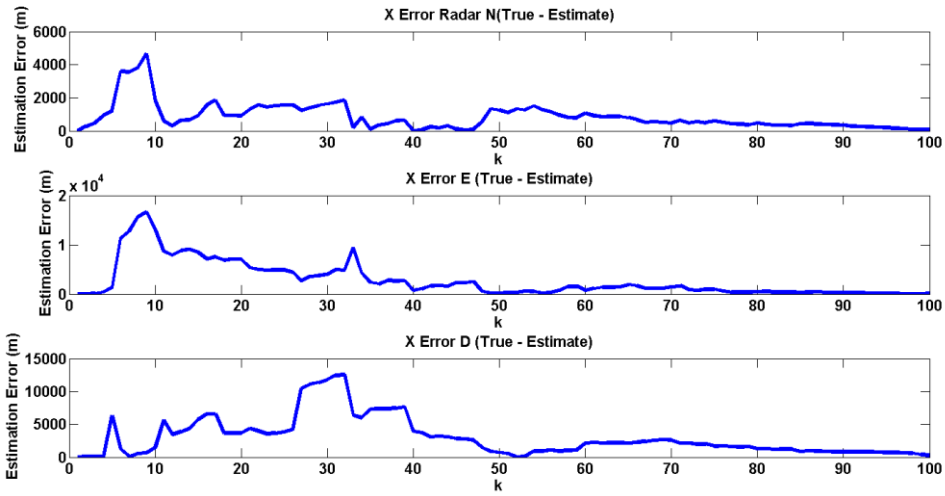

Figure 5.27: Radar linear KF tracking in Cartesian coordinates – Estimation error in N, E and D directions in a single flight scenario.

To illustrate the importance of the measurement covariance conversion to Cartesian space of the KF, performed in Equations (5.19) and (5.20), the NED threat position components are compared to the estimated values when no conversion is performed. The result is illustrated below.

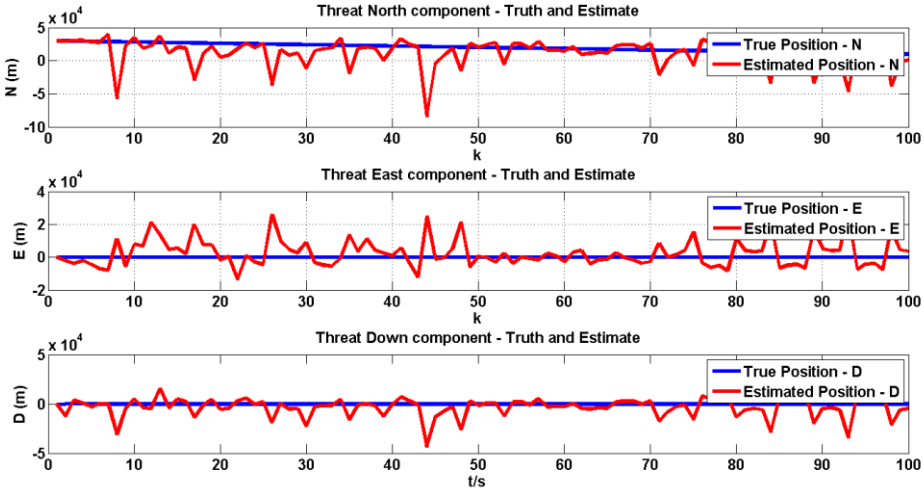

Figure 5.28: Radar linear filter (KF) tracking with Cartesian coordinates - True and estimated threat position with no measurement covariance matrix transformation.

5.4.2.2 Monte Carlo Simulations

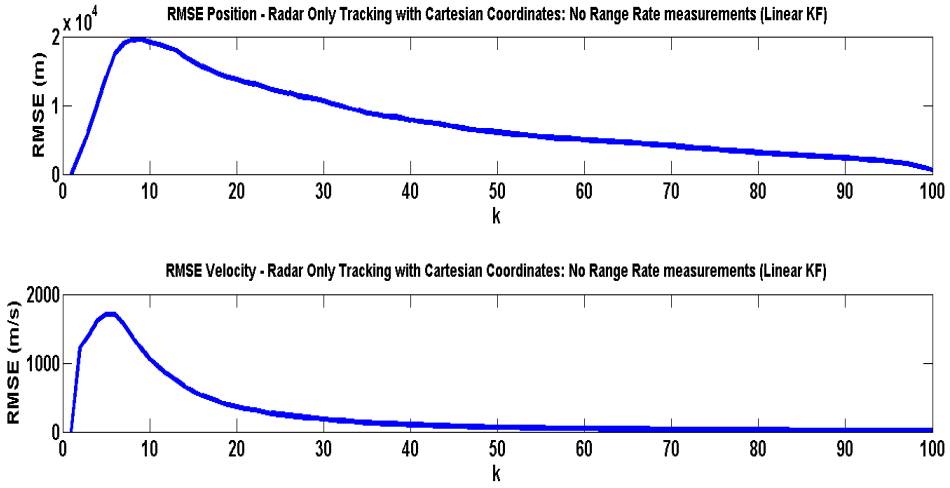

Figure 5.29: Radar linear filter tracking in Cartesian coordinates with no range rate measurements - RMSE in position and velocity.

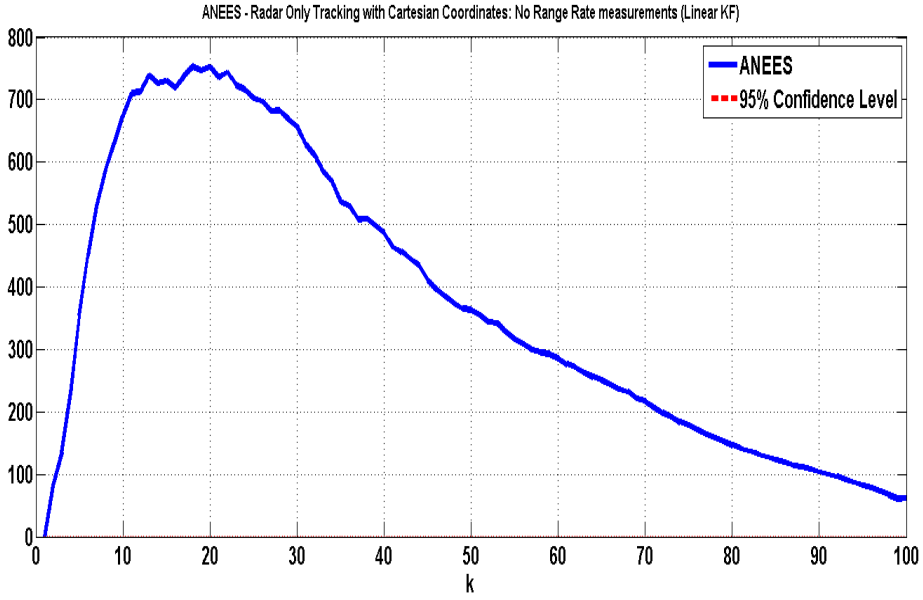

Figure 5.30: Radar linear filter tracking in Cartesian coordinates with no range rate measurements - ANEES.

5.4.3 Summary

In summary, both the process noise and measurement noises are set to high values in which the filters are required to track. It was shown that both the

nonlinear and linear range radar tracking filters are able to track the threat under these conditions. Threat position and velocity have been adequately tracked once the filter's steady state has been reached. However, the linear filter takes significantly longer to converge.

5.5 Doppler Radar Tracking Filter

Two filters with which to perform Doppler radar airborne tracking using range rate measurements are developed. The filters satisfy two cases according to the filter's knowledge of the existence of correlation between measurement errors in range and range rate:

- Case 1: the filter is unaware of the correlation between measurement errors in range and range rate. That is $\rho^{corr} = 0$. Furthermore, correlation is unaccounted for in the filter.
- Case 2: $\rho^{corr} \neq 0$ and the filter is aware and has accounted for this correlation.

It was discovered in Chapter 2 that in reality there exists a correlation in measurement errors between range and range rate. However, many designs in the literature make the assumption of Case 1, i.e. that the correlation does not exist. The reason for this is that when the correlation is accounted for the filter complexity increases. In what follows we present two Doppler radar tracking filters: first, a filter assuming ideal conditions, addressing Case 1; and second, a filter that accounts for the inherent correlation in measurement errors of range and range rate (Case 2). The reason for developing these two filters is to bring to light any performance differences resulting when correlation is accounted for, compared to when it is not.

Under Case 1 an EKF is used as the tracking filter allowing us to make use of the radar measurements directly. Therefore, tracking is performed in mixed coordinates. The second filter developed (Case 2) is an extension to the first with $\rho^{corr} = -0.9$ which implies negative correlation modelling an upsweep

LFM waveform. It has been proven in [**66**] that when ρ^{corr} is around -0.9 to significantly reduce state estimate errors, as opposed to positive correlation. Decorrelation is performed akin to the method presented in [**44**], described in subsection 2.2.3 of the literature review.

The radar measures relative range, azimuth, elevation and range rate. Therefore the corresponding observation vectors are:

$$z_k^R = [\rho_k^R \quad \theta_k^R \quad \varphi_k^R \quad \dot{\rho}_k^R]^T + v_k^R. \tag{5.19}$$

Where v_k^R is the zero-mean, white, Gaussian radar measurement noise with covariance matrix R^R:

$$R^R = \begin{bmatrix} \sigma_\rho^{R^2} & 0 & 0 & 0 \\ 0 & \sigma_\theta^{R^2} & 0 & 0 \\ 0 & 0 & \sigma_\varphi^{R^2} & 0 \\ 0 & 0 & 0 & \sigma_{\dot{\rho}}^{R^2} \end{bmatrix}. \tag{5.20}$$

However, as stated in Chapter 2, measurement errors in range and range rate are correlated. Therefore, in contrast to Equation (5.20), the measurement covariance matrix becomes:

$$R^R = \begin{bmatrix} \sigma_\rho^{R^2} & 0 & 0 & \rho^{corr}\sigma_\rho^R\sigma_{\dot{\rho}}^R \\ 0 & \sigma_\theta^{R^2} & 0 & 0 \\ 0 & 0 & \sigma_\varphi^{R^2} & 0 \\ \rho^{corr}\sigma_\rho^R\sigma_{\dot{\rho}}^R & 0 & 0 & \sigma_{\dot{\rho}}^{R^2} \end{bmatrix}. \tag{5.21}$$

Where the range and range-rate measurement errors are correlated with ρ^{corr}. And σ_ρ, σ_θ, σ_φ and $\sigma_{\dot{\rho}}$ are the measurement errors in range, azimuth, elevation and range rate respectively; whose values are shown below.

Table 5.4: Doppler radar measurement errors.

Sensor	Measurement Error	Value
Doppler Radar	σ_ρ^R	$200\ m$
	σ_θ^R	$0.3\ radians$
	σ_φ^R	$0.3\ radians$
	$\sigma_{\dot{\rho}}^R$	$50\ m/s$

The effect of the range rate measurement error is displayed in the following figures which contain the relative true and Doppler radar measured range rate with respect to the NED system.

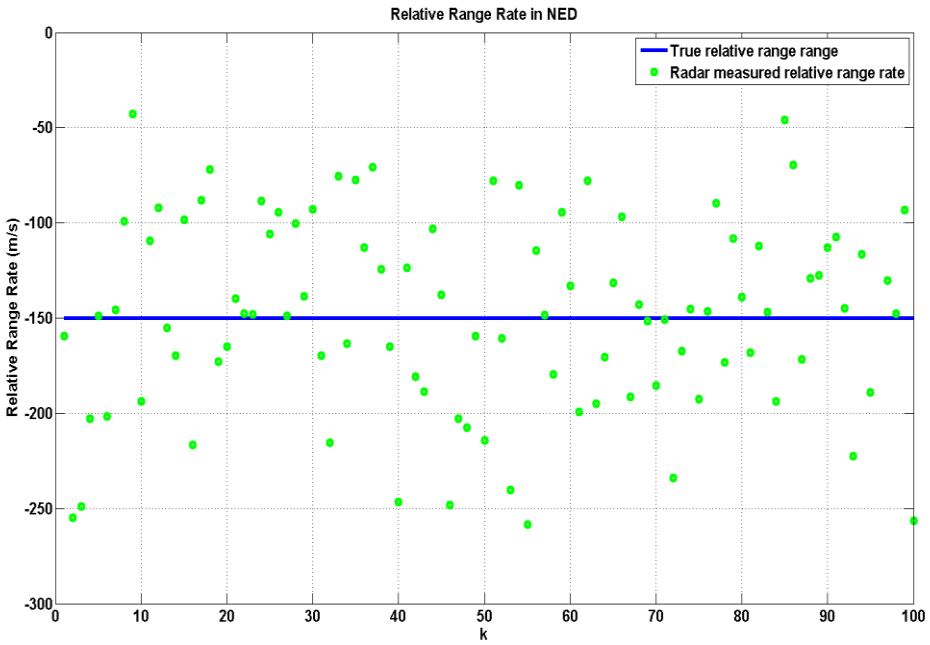

Figure 5.31: Relative range rate in NED - truth and radar measured values.

5.5.1 EKF Under Ideal Conditions

The first Doppler radar tracking filter developed is one that assumes correlation in measurement error do not exist. Range rate measurements are highly nonlinear in Cartesian space and therefore, an EKF is chosen as the tracking filter.

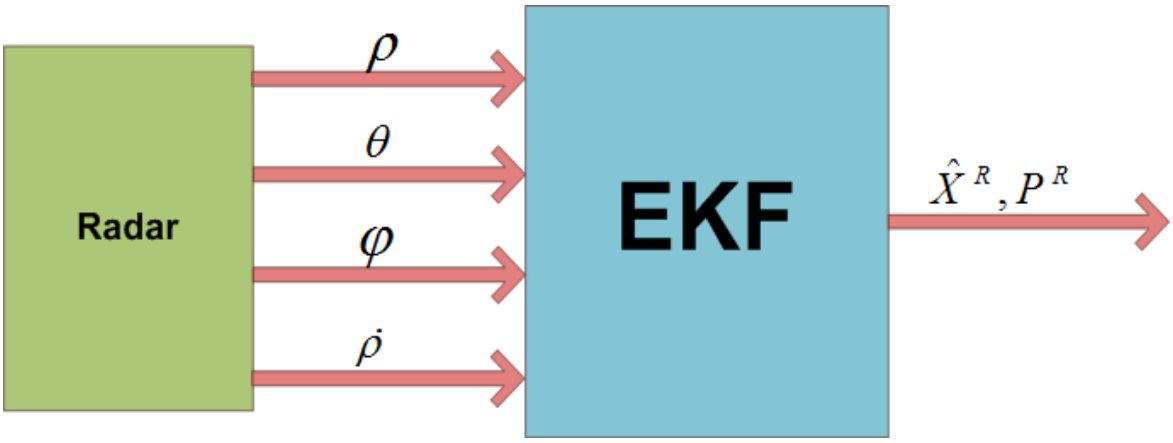

Figure 5.32: EKF with range rate measurements, assuming no correlation.

The filter design is the same as that of the ranging radar described in subsection 5.4.1 above. However, the inclusion of range rate information results in a new observation vector and linearized measurement matrix:

$$h^R(X) = z_k^R = \begin{bmatrix} \sqrt{N^2 + E^2 + D^2} \\ \tan^{-1}\left(\frac{E}{N}\right) \\ -\sin^{-1}\left(\frac{D}{\sqrt{N^2 + E^2 + D^2}}\right) \\ \frac{N\dot{N} + E\dot{E} + D\dot{D}}{\sqrt{N^2 + E^2 + D^2}} \end{bmatrix}, \tag{5.22}$$

$$H_X^R(k) = \left.\frac{\partial h^R(X)}{\partial X}\right|_{X=\hat{X}_{k|k-1}^R}. \tag{5.23}$$

See Appendix A3 for the complete linearized measurement matrix.

As before, at this point the standard EKF equations produce state and covariance estimates. The results of this filter with $\rho^{corr} = 0$ are shown below.

5.5.1.1 Results

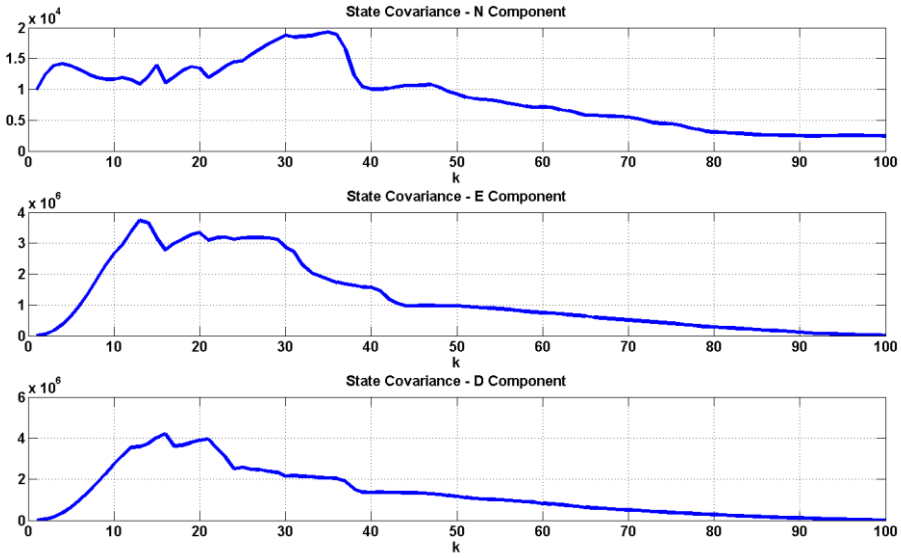

Figure 5.33: Radar EKF with range rate measurements – estimated state error covariance.

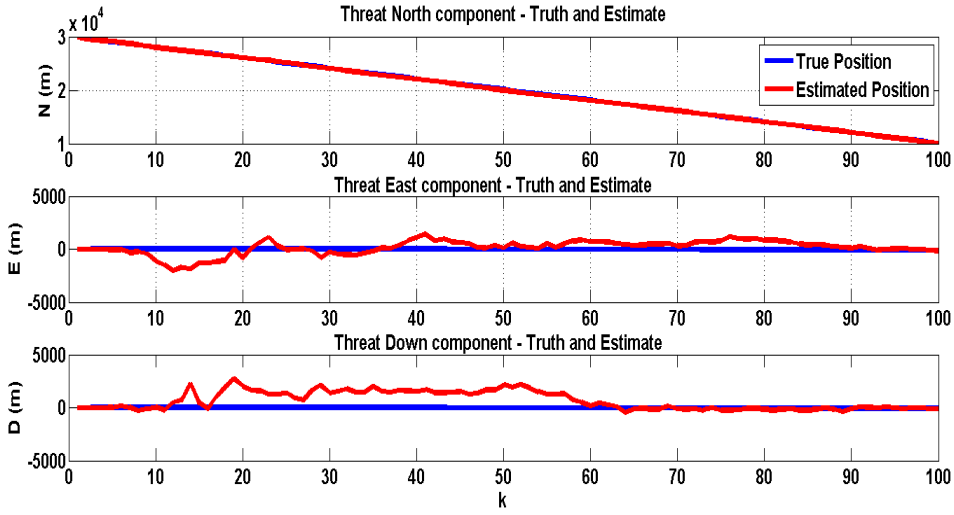

Figure 5.34: Radar EKF with range rate measurements – true and estimated threat position in N, E and D directions.

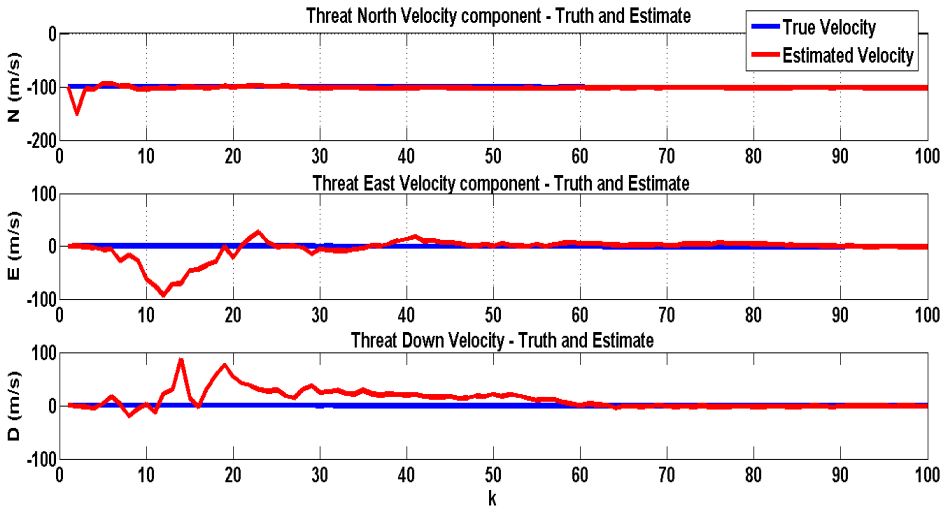

Figure 5.35: Radar EKF with range rate measurements – true and estimated threat velocity in N, E and D directions.

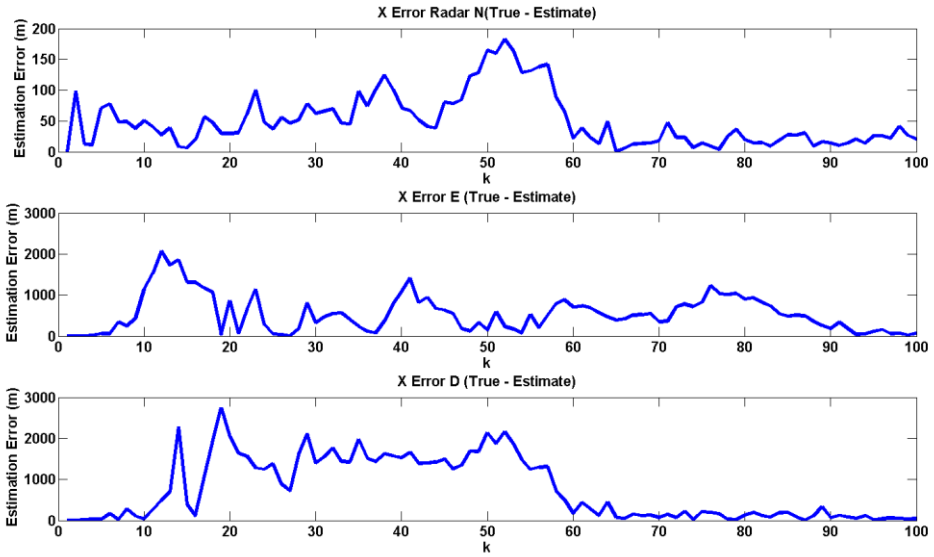

Figure 5.36: Radar EKF with range rate measurements – estimation error in N, E and D directions in a single flight scenario.

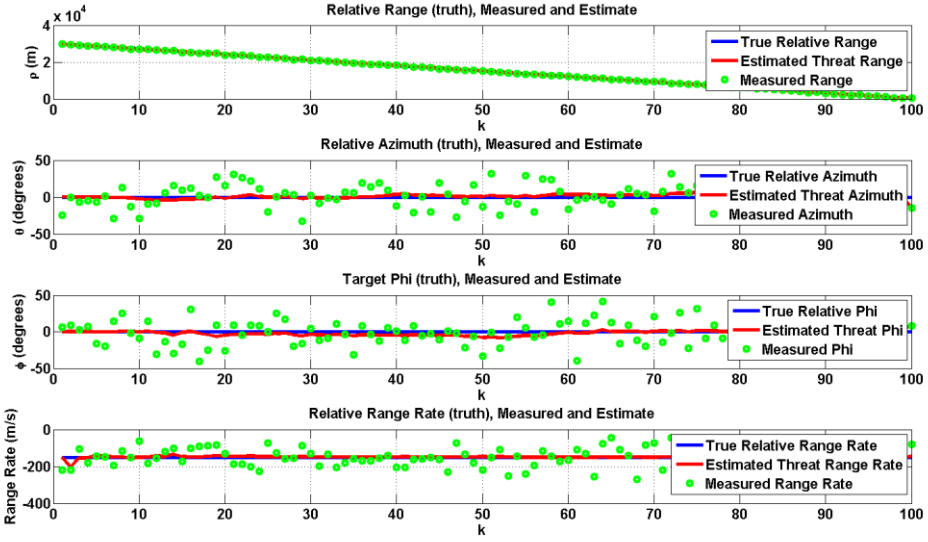

Figure 5.37: Radar EKF with range rate measurements – true, estimated and measured relative range, azimuth, elevation and range rate with respect to the BRF.

5.5.1.2 Monte Carlo Simulations

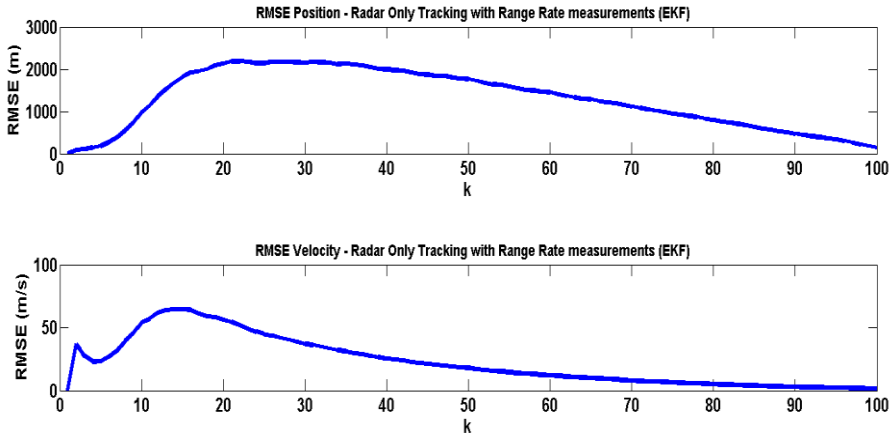

Figure 5.38: Radar EKF with range rate measurements – RMSE in position and velocity.

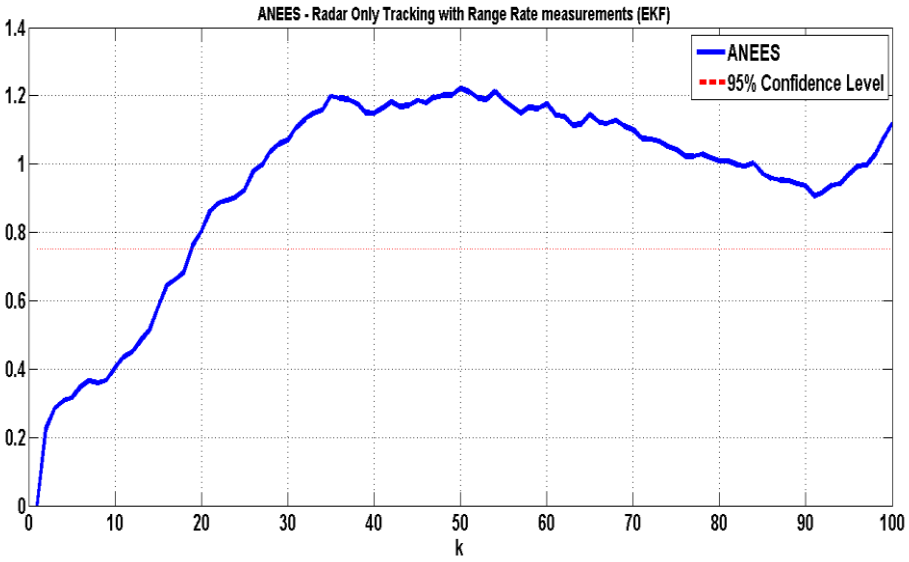

Figure 5.39: Radar EKF with range rate measurements - ANEES.

5.5.2 EKF with Correlated Measurement Errors

In Chapter 2 methods were presented for decorrelation and it was discovered that these filters, which account for the correlation, are more computationally intensive. However, there was no evidence which compared the performance between the two. Therefore a filter which accounted for the correlation was developed.

This filter assumes that measurement errors in range and range rate are correlated with $\rho^{corr} = -0.9$. In order to address this issue, the filter is designed to decorrelate the errors through the formation of a pseudo measurement [**55**], as discussed in the literature review (see section 2.2.3). This method is similar to the debiased consistent converted measurements method [**55**] also discussed in the literature review, however tracking is performed in Cartesian coordinates only. The method proposed by Duan et al. [**55**] was preferred as the conversion to Cartesian coordinates introduces further measurement errors.

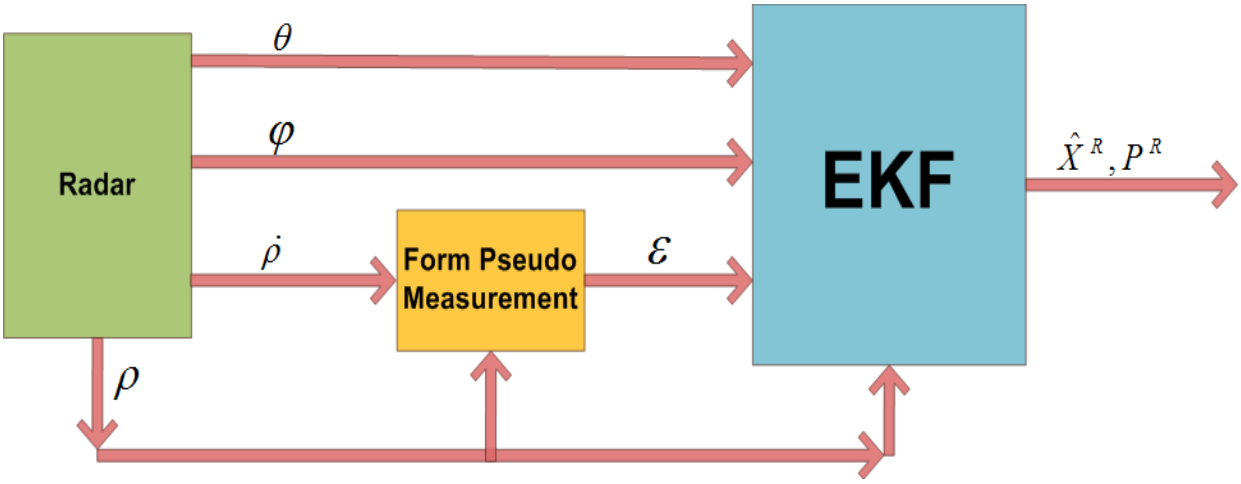

Figure 5.40: EKF with range rate measurements, assuming correlation exists. The EKF takes the range, azimuth, elevation and pseudo measurements as inputs. The pseudo measurement is formed using the range and range rate measurements.

The pseudo measurement formed is:

$$\varepsilon_k = \frac{-\rho^{corr}\sigma_{\dot{\rho}}\rho_k}{\sigma_\rho} + \dot{\rho}_k,$$

with covariance:

$$\sigma_\varepsilon{}^2 = \left(1 - \rho^{corr\,2}\right)\sigma_{\dot{\rho}}^2.$$

The new radar observation vector is now:

$$z_k^R = [\rho_k^R \quad \theta_k^R \quad \varphi_k^R \quad \varepsilon_k^R]^T + v_k^R. \tag{5.24}$$

Then the linearized measurement matrix is the same as in 5.5.1, and through the use of the EKF equations yield state and covariance estimates. The results of the correlated filter are presented below.

5.5.2.1 Results

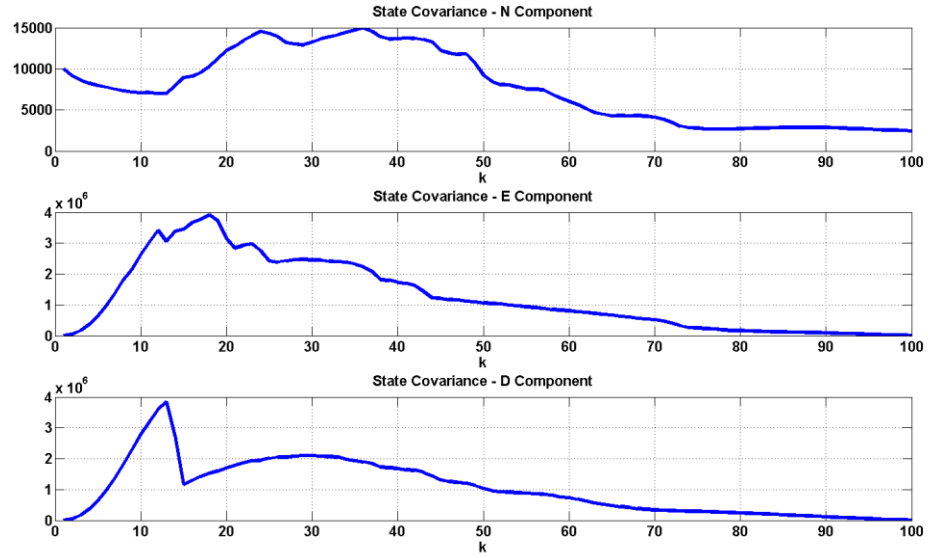

Figure 5.41: Radar EKF with range rate measurements and correlation – estimated state error covariance.

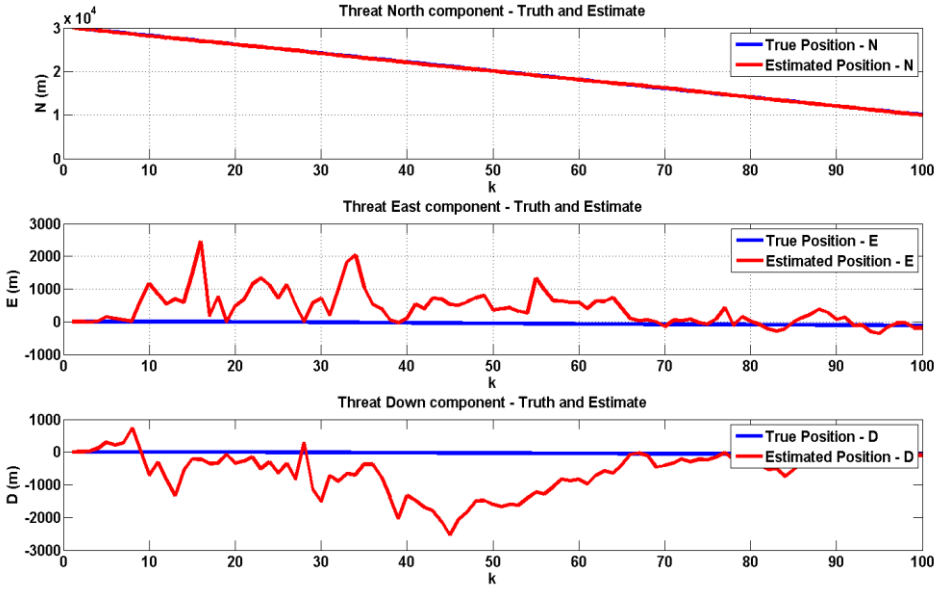

Figure 5.42: Radar EKF with range rate measurements and correlation – true and estimated threat position in N, E and D directions.

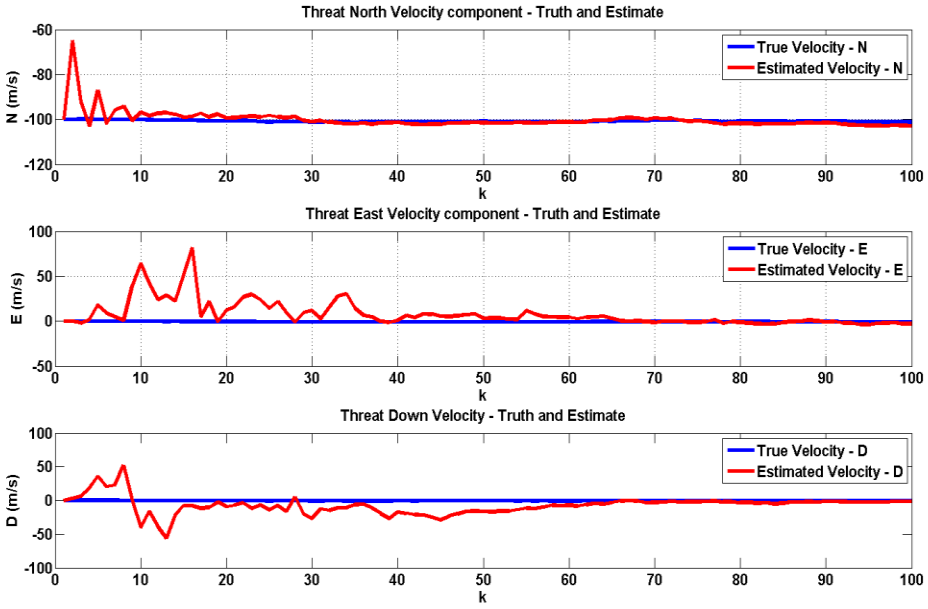

Figure 5.43: Radar EKF with range rate measurements and correlation – true and estimated threat velocity in N, E and D directions.

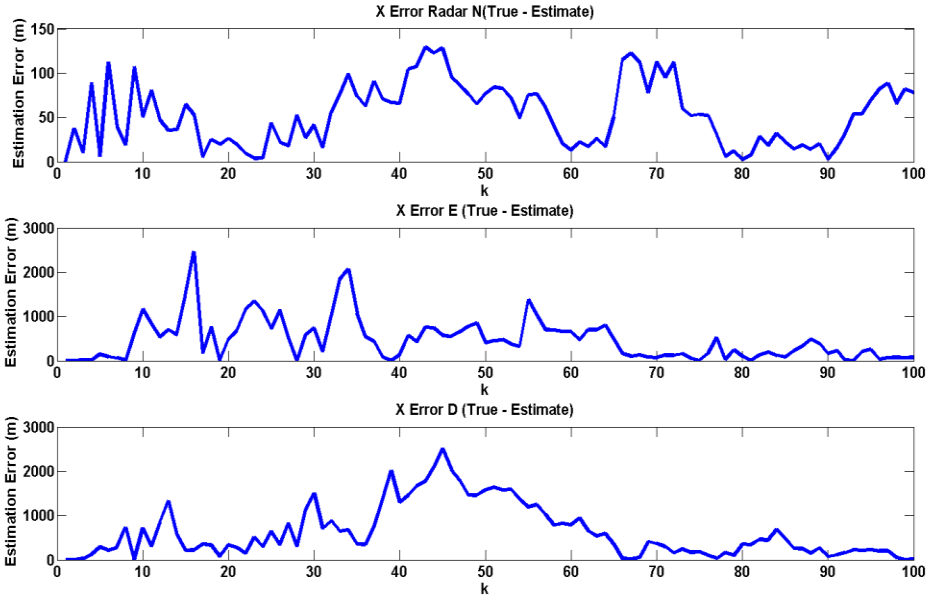

Figure 5.44: Radar EKF with range rate measurements and correlation – estimation error in N, E and D directions in a single flight scenario.

5.5.2.2 Monte Carlo Simulations

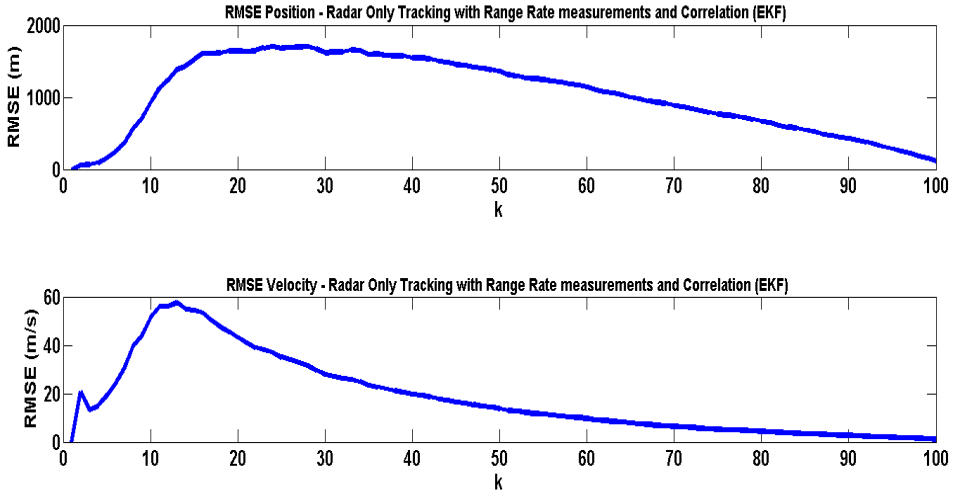

Figure 5.45: Radar EKF with range rate measurements and correlation – RMSE in position and velocity.

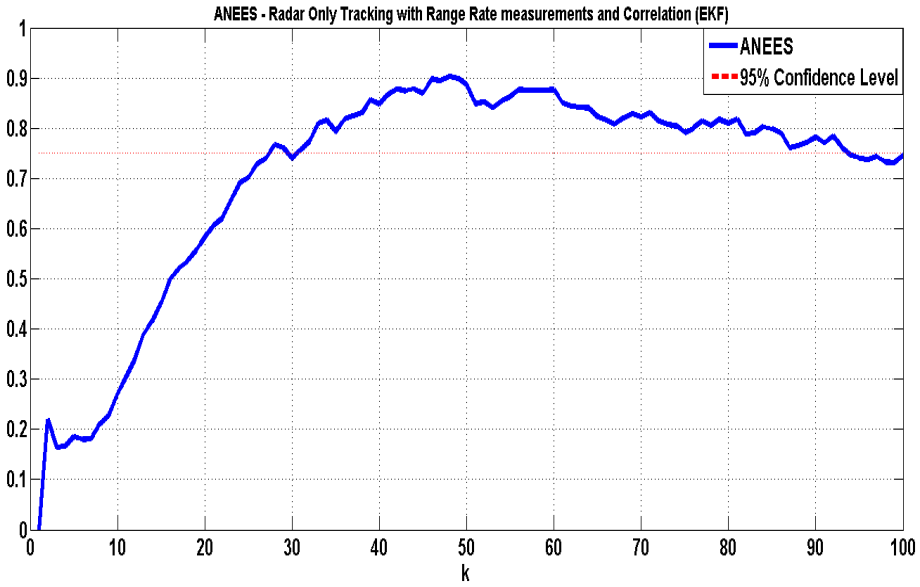

Figure 5.46: Radar EKF with range rate measurements and correlation – ANEES.

5.5.3 Summary

In summary, both tracking filters (Case 1 and Case 2) are able to track the threat under the high noise conditions with the use of range rate measurements; although not particularly well. The range rate measurements increase the accuracy of the velocity estimates in both filter cases. In Case 1, the position tracking error is quite large, until the filter reaches the steady state towards the end of the simulation. This is as a result of the linear approximation of the measurement equation. Case 2 displays similar large position estimation errors. However, the steady state is reached sooner than Case 1 and the estimation error is significantly reduced, as can be seen in Figure 5.42.

5.6 Fusion Filter

One of the objectives of this thesis, stated in Chapter 1, is to demonstrate that a fusion filter more accurately tracks the threat than a single sensor tracking filter. The single sensor filters developed in the previous sections of this chapter satisfy half of this objective. Therefore, in what follows, the design of tracking filters using the measurements from multiple, redundant sensors are detailed.

The TTU is composed of a fusion filter (or fusion architecture). Three fusion architectures are designed, named FA1-FA3, to perform data fusion using an EO camera and a Doppler radar with range rate measurements as set out in Chapter 1. The tracking filters developed thus far form the building blocks from which the fusion architectures are built up with. This section describes all three architectures and presents their results so that, ultimately, a performance comparison can take place to identify the fusion architecture to be used in the TTU.

For a UAV SAS a centralised architecture would, at first thought, make most sense, as sensors and computing platforms are all on-board the UAV.

However, future SAA systems might make use of a network of distributed sensors (for example a network of aiding ground-based radars). Furthermore, current computing platforms provide the potential for running multiple independent filters in parallel. Therefore, distributed fusion architectures have been researched and developed in this section.

5.6.1 Synchronising Sensor Sampling Periods

As stated in section 2.1, the EO camera operates at a higher data rate than the radar. This implies that for each radar measurement received, the TTU would have received $\Delta T^R/\Delta T^C$ many camera measurements. This occurrence can be seen in Figure 5.47. In order to account for this and to synchronise the tracking filter with measurements from radar and camera, data compression is performed as explained in subsection 2.2.4. Data compression is performed in all fusion architectures.

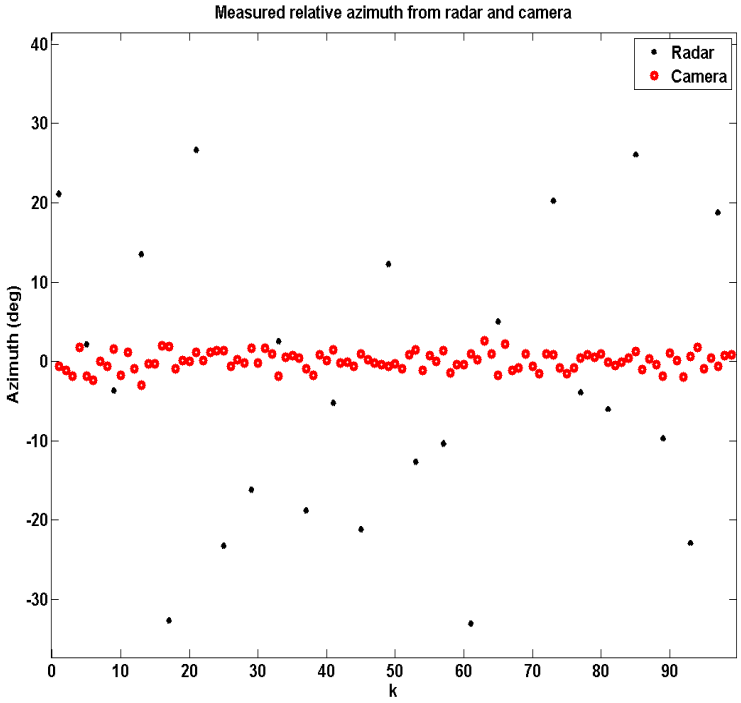

Figure 5.47: Plot of radar and camera azimuth measurements illustrating their contrasting sampling periods. The camera provides measurements four times faster than the radar.

5.6.2 FA1

FA1 is based on a centralised architecture – raw sensor data, from both radar and camera, arrives at the FC which are fused. This fused measurement is then tracked to produce an estimate of the state vector.

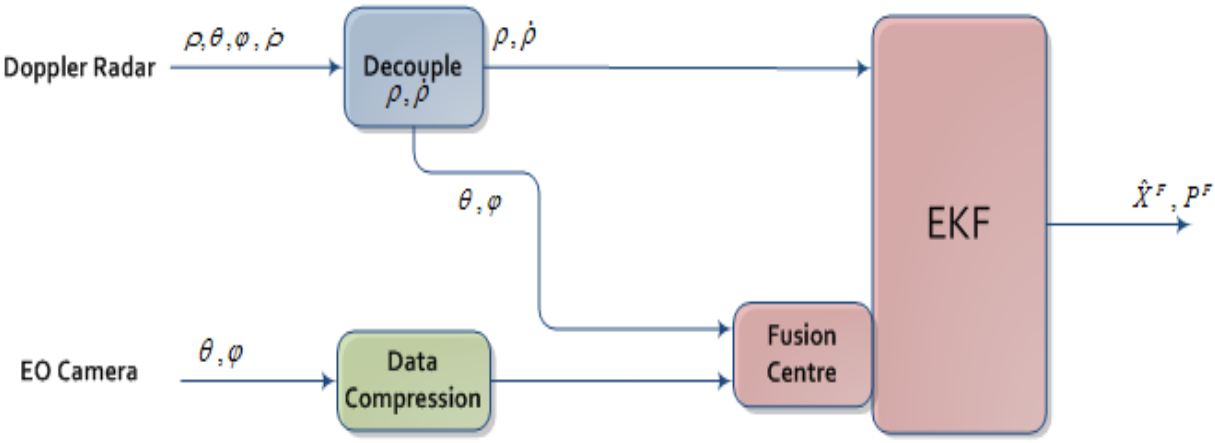

Figure 5.48: Fusion Architecture 3. Measurement fusion is performed in this centralised approach.

Data compression is performed on the camera observations to synchronise its frequency with that of the radar's data rate. Then the range and range rate measurements are temporarily decoupled from the radar observation vector. The reason for doing this is so that the angular components (threat azimuth and elevation) from the radar and camera can be used. The FC fuses this redundant data by utilising measurement fusion, identified in the Literature Review, using Equations (2.1) and (2.2).

The fused measurement and its covariance are then augmented by placing the previously decoupled range and range rate data (and their covariance) back into the fused matrices. These fused-augmented matrices are then input into an EKF for tracking. The EKF implementation used in FA1 is that which was developed in 5.5.1.

5.6.2.1 Results

The results from running FA1 in the simulation environment over 100 discrete time-steps are presented below.

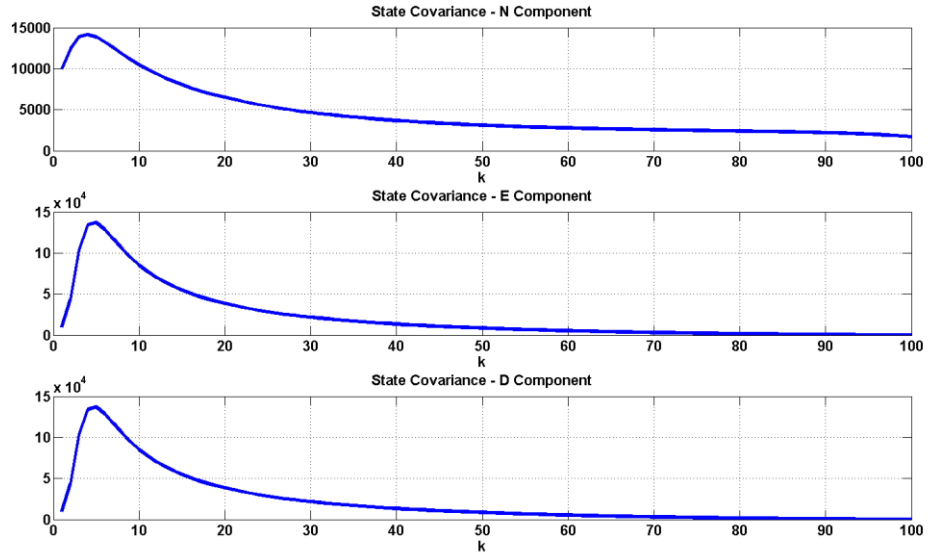

Figure 5.49: FA1 – state error covariance in N, E and D directions.

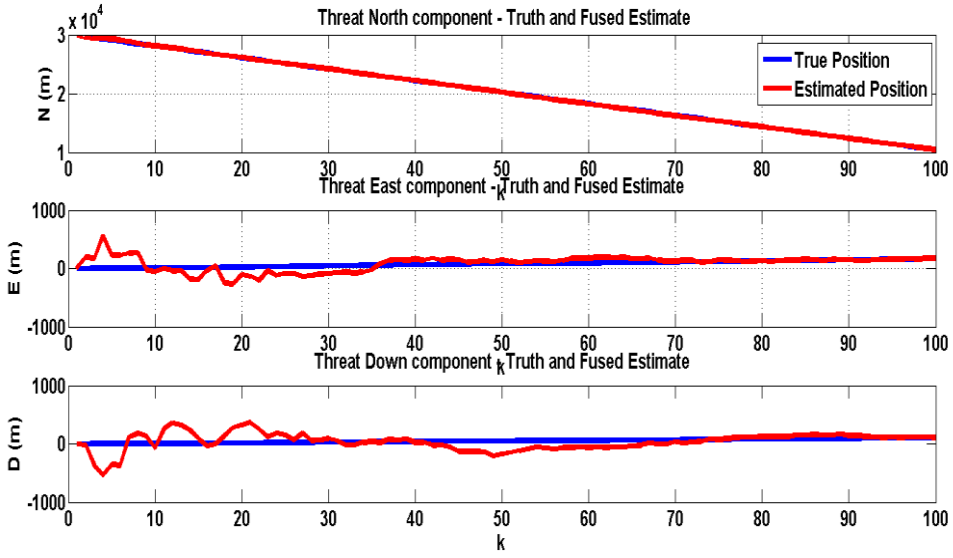

Figure 5.50: FA1 – true and estimated threat position in N, E and D directions.

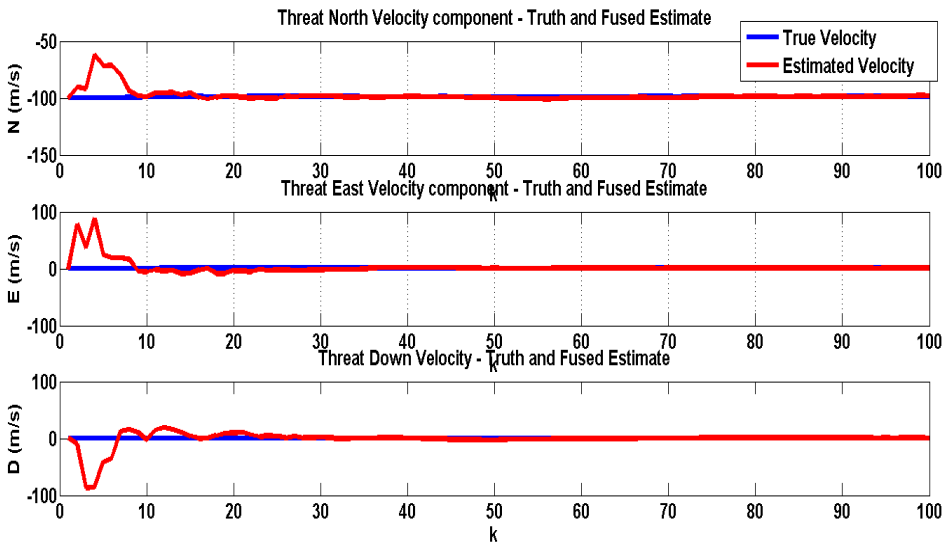

Figure 5.51: FA1 – true and estimated threat velocity in N, E and D directions.

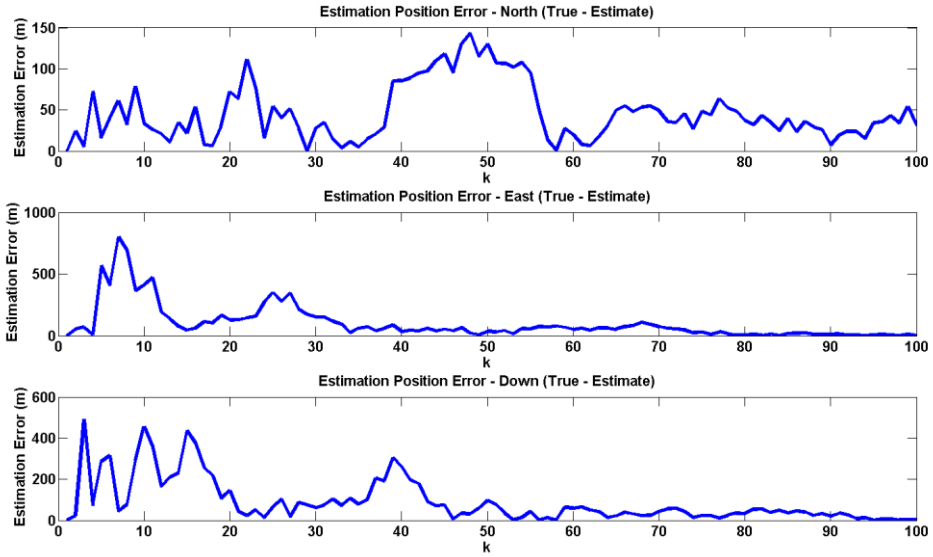

Figure 5.52: FA1 – estimation error in N, E and D directions in a single flight scenario.

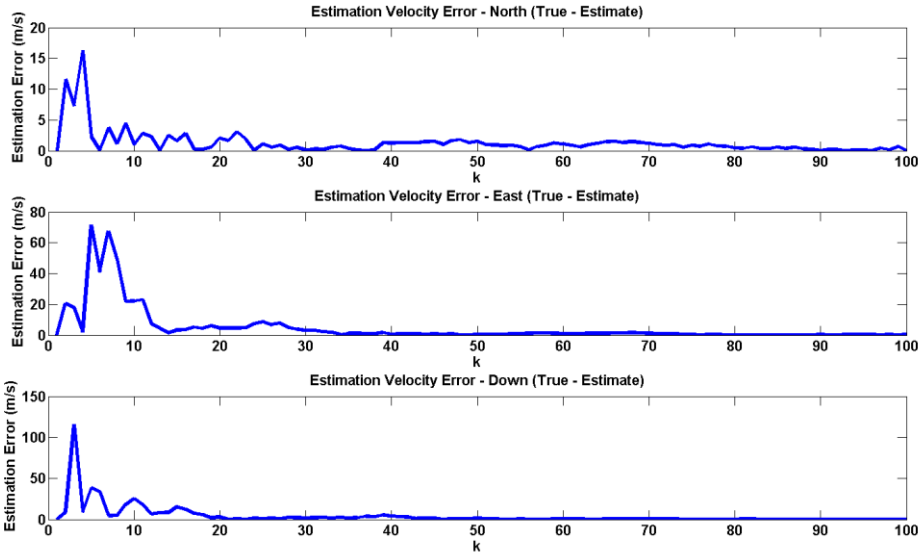

Figure 5.53: FA1 – velocity estimation error in N, E and D directions in a single flight scenario.

5.6.2.2 Monte Carlo Simulations

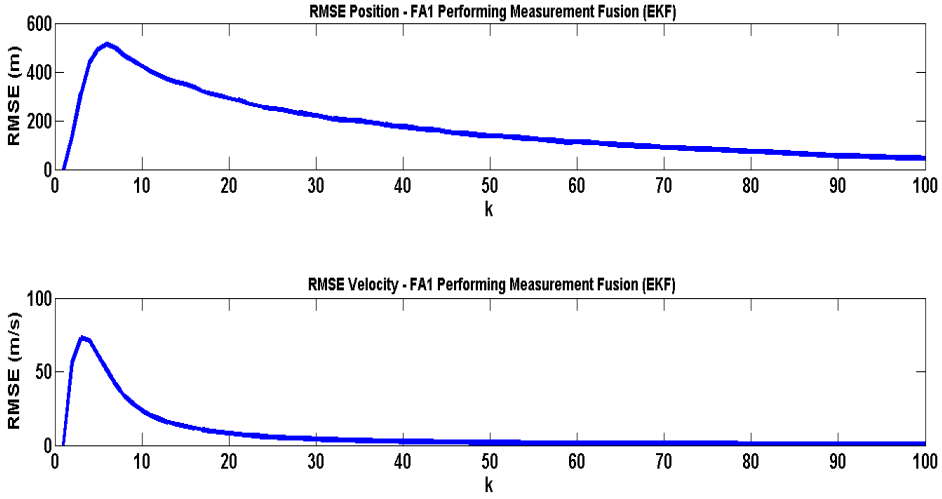

Figure 5.54: FA1 centralised fusion - RMSE in position and velocity.

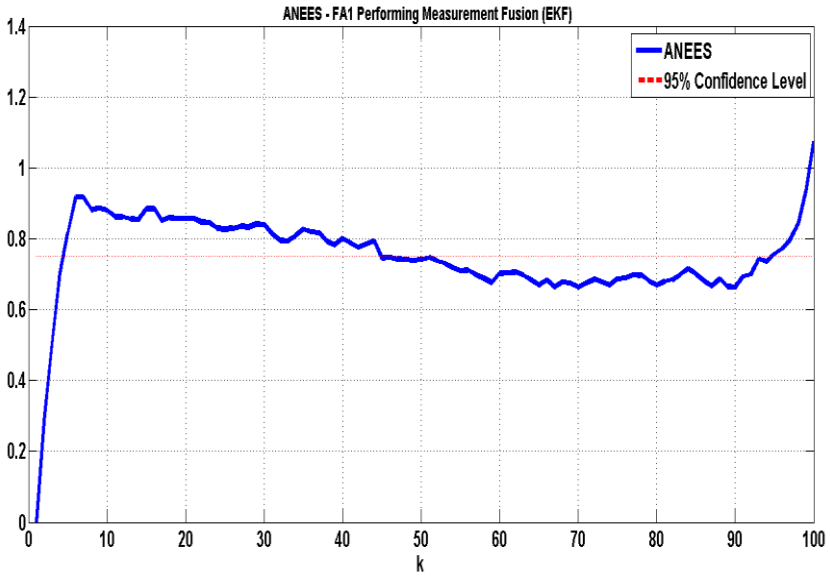

Figure 5.55: FA1 centralised fusion - ANEES.

5.6.3 FA2

FA2 is a distributed fusion architecture (see Section 2.1.1) employing state vector fusion with partial feedback. The measurements from radar and camera are used directly in the target state estimation and therefore FA2 consists of two EKFs – one for each local filter. FA2 is illustrated in Figure 5.56.

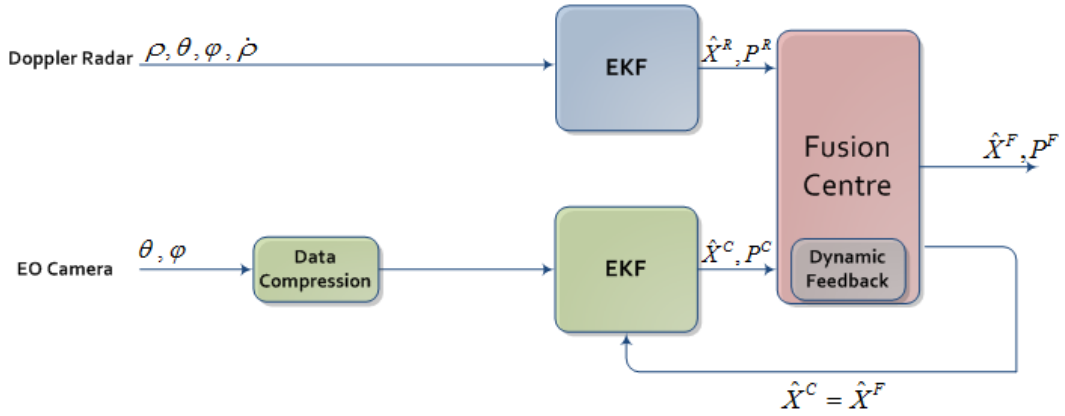

Figure 5.56: Fusion Architecture 2. Both radar and camera local filter are EKFs.

As sensor measurements are nonlinear with respect to the state variables and that, in FA1, they are not manipulated in any way, two EKFs are employed for both radar and camera local filters.

Centralised architectures produce the globally optimal estimate, by minimising the theoretical MSE (an efficient estimator). Distributed architectures can achieve this through the technique of feedback. This entails setting the local estimate to the globally fused estimate. In FA2 we set the camera local filter's state estimates to the globally fused one:

$$\hat{X}^{C}_{k|k} = \hat{X}^{F}_{k|k}. \tag{5.25}$$

The reason for doing this is twofold: first so that the architecture does not have a single source of failure, i.e. the whole system does not rely on the FC. If the FC were to fail during operation, then the radar local filter is still able to function correctly and perform tracking. Second, the feedback helps the camera's deficiency in being able to track with its angular only measurements.

The radar local filter is that as designed in Section 5.5.1 and the camera local filter is that designed in Section 5.3.

5.6.3.1 Results

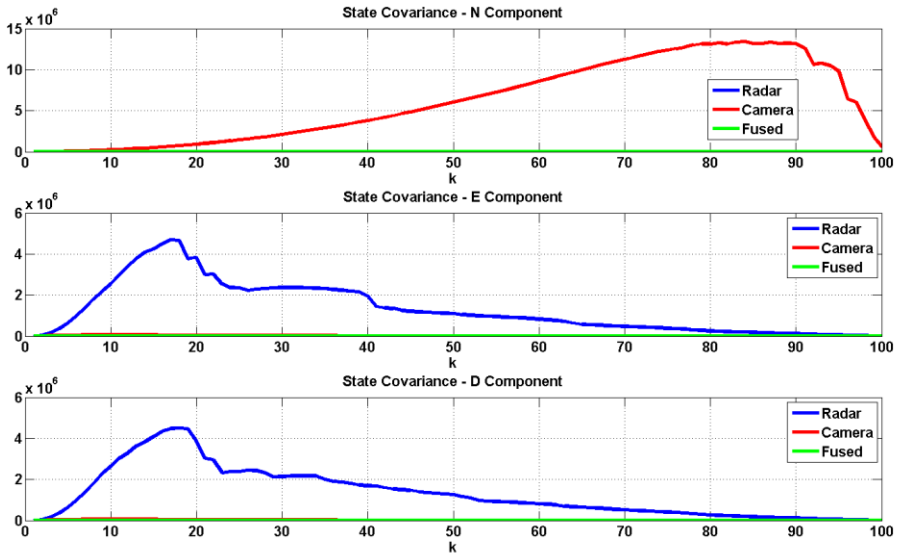

Figure 5.57: FA2 - radar, camera and fused error covariances in N, E and D directions.

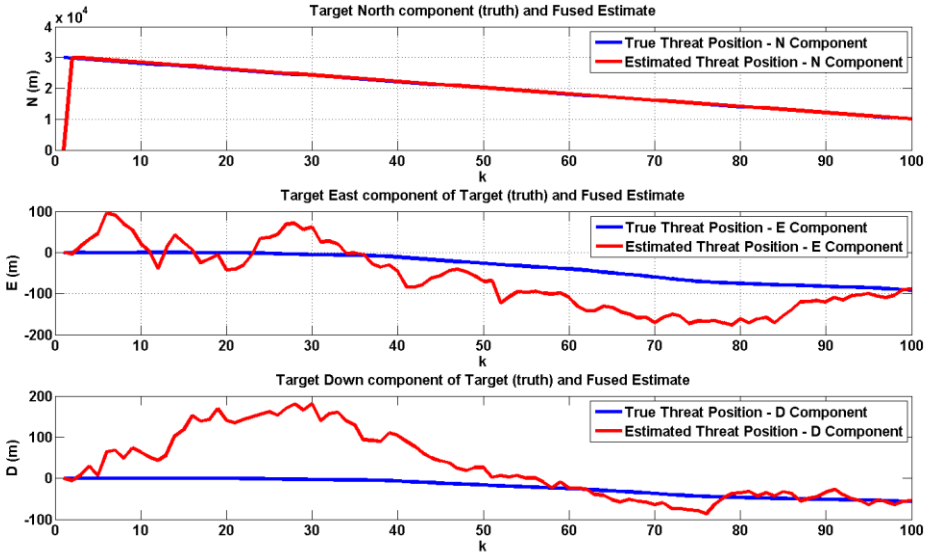

Figure 5.58: FA2 – true and estimated threat position in N, E and D directions.

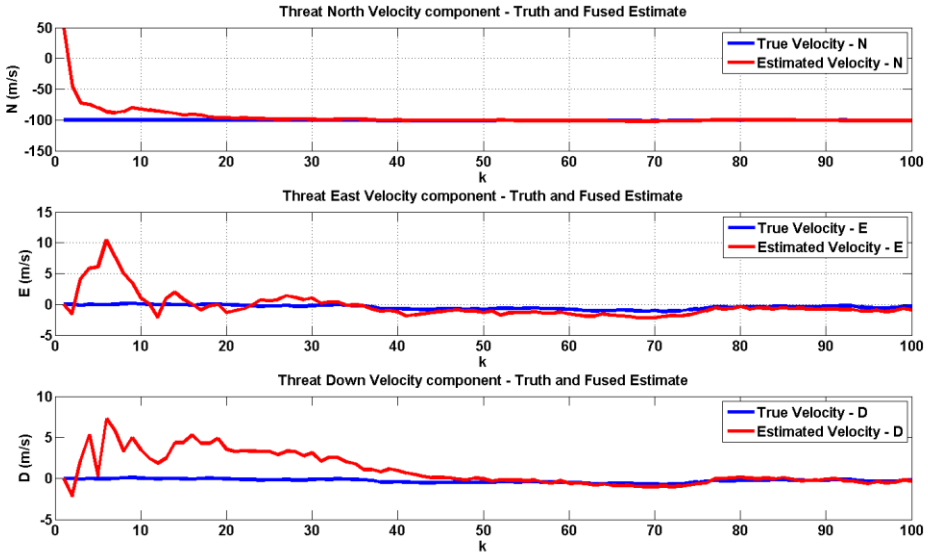

Figure 5.59: FA2 – true and estimated threat velocity in N, E and D directions.

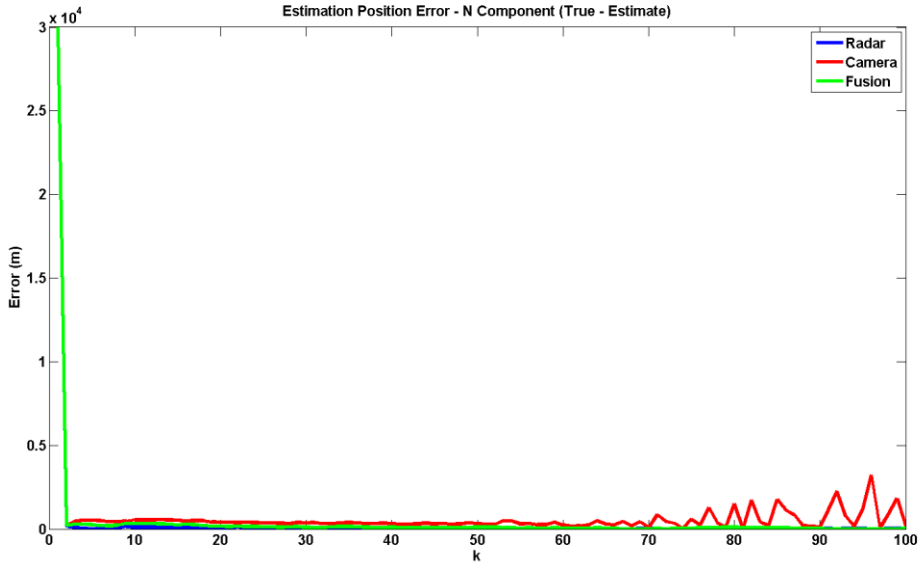

Figure 5.60: FA2 - radar, camera and fused position estimation errors in the N direction.

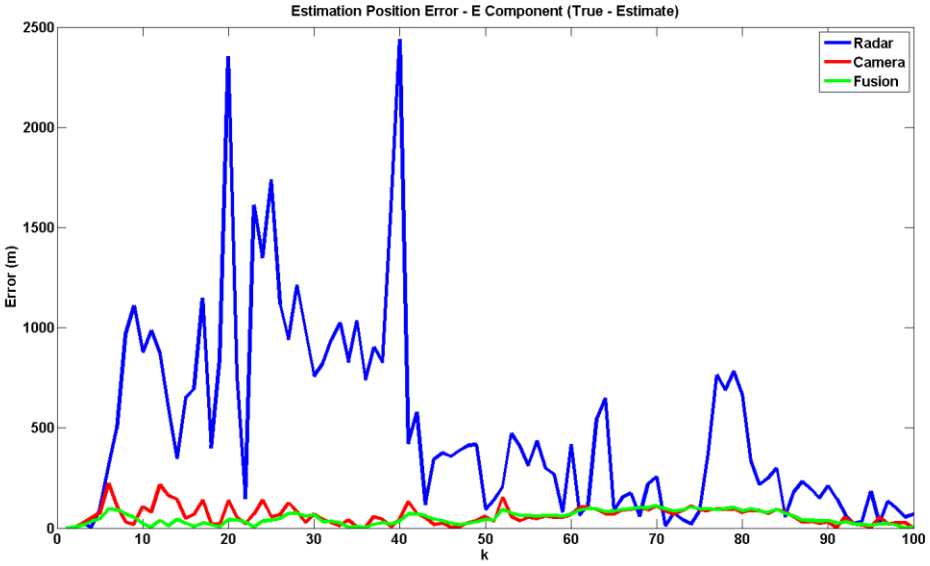

Figure 5.61: FA2 - radar, camera and fused position estimation errors in the E direction.

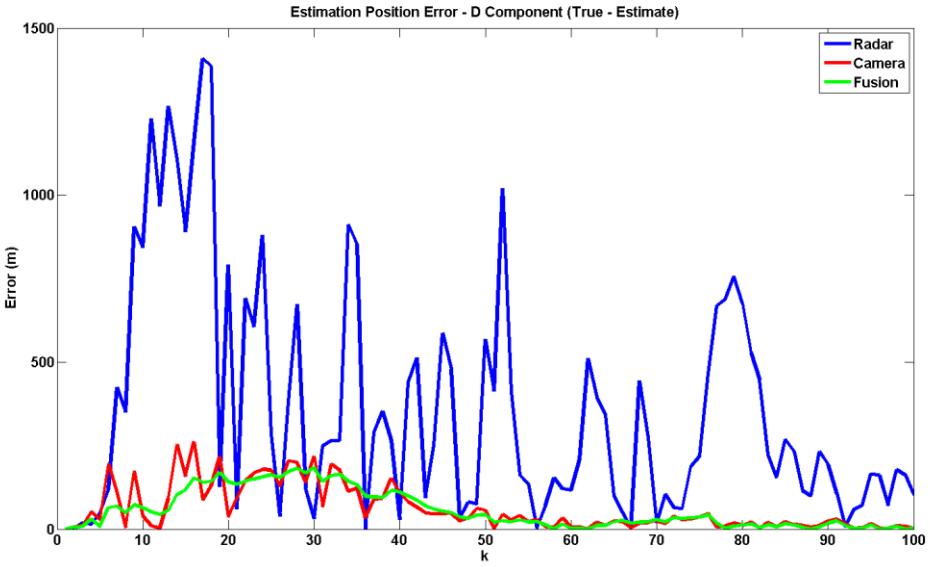

Figure 5.62: FA2 - radar, camera and fused position estimation errors in the D direction.

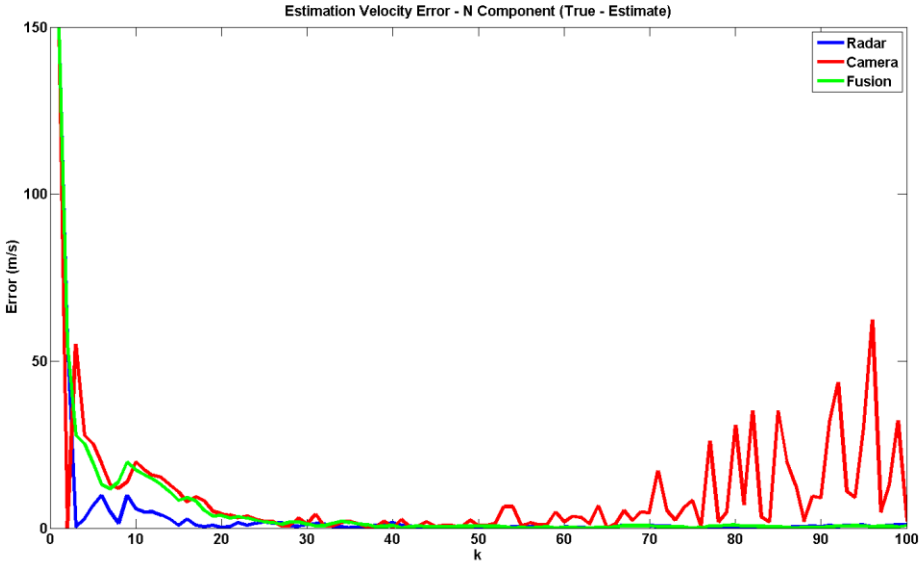

Figure 5.63: FA2 - radar, camera and fused velocity estimation errors in the N direction.

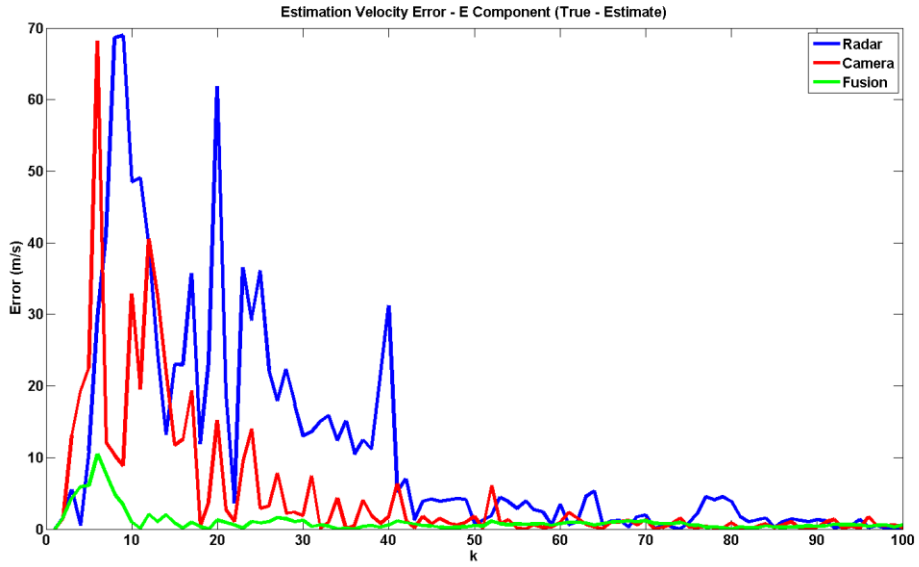

Figure 5.64: FA2 - radar, camera and fused velocity estimation errors in the E direction.

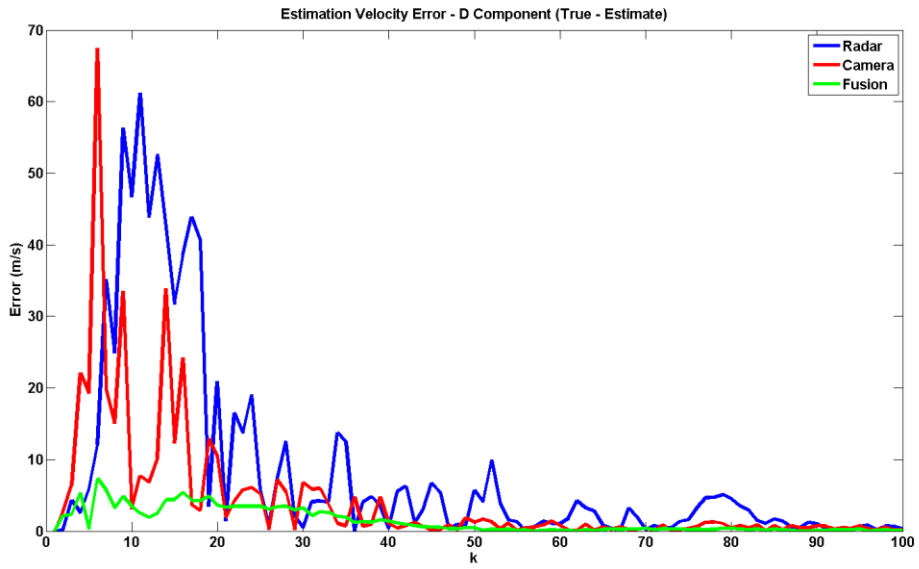

Figure 5.65: FA2 - radar, camera and fused velocity estimation errors in the D direction.

5.6.3.2 Monte Carlo Simulations

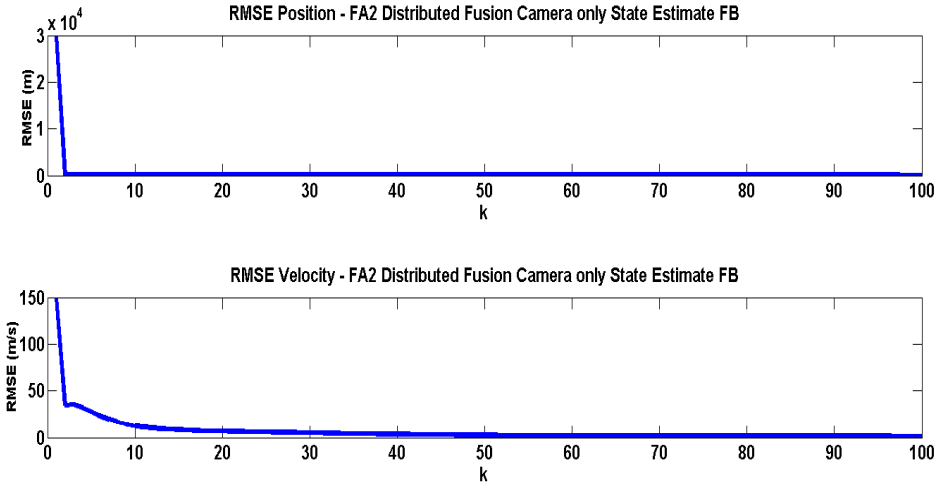

Figure 5.66: FA2 - RMSE in position and velocity.

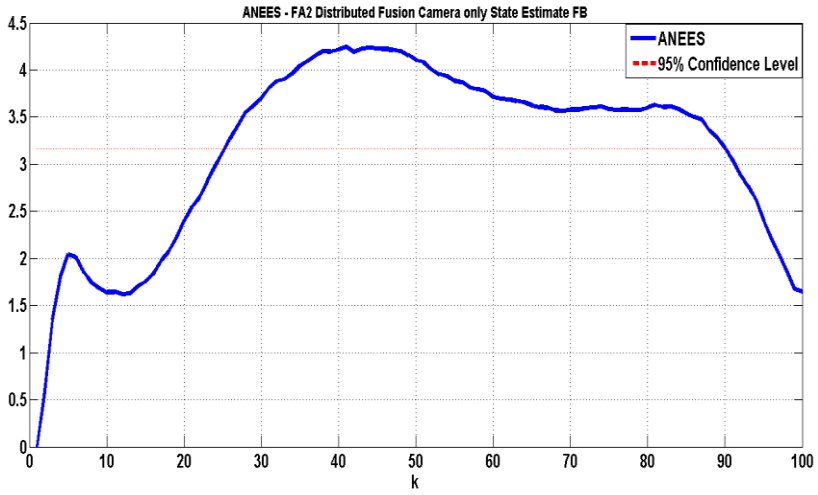

Figure 5.67: FA2 - ANEES.

5.6.4 FA3

Following the fusion system design in [**29**], we adopt a similar approach albeit with a few modifications. FA3 is another distributed fusion architecture employing state vector fusion with dynamic feedback. However, in contrast to FA2, Radar and camera pseudo measurements are formed that allow us to make use of a linear KF for the radar local filter and an EKF for the EO camera local filter.

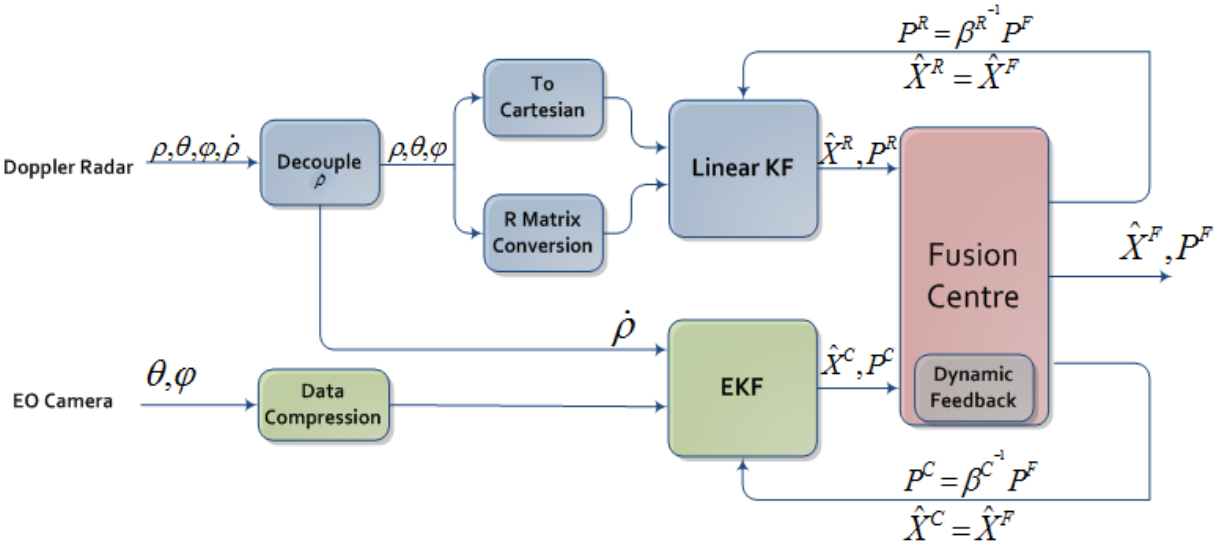

Figure 5.68: Fusion Architecture 3. It consists of a linear KF, an EKF and the Fusion Centre.

FA3 was developed in an attempt to reduce the computational load of FA2, which is comprised of two EKFs. In order to achieve this end, radar and camera pseudo measurements were formed; thus making it possible to replace the nonlinear radar local filter in FA2 with a linear KF.

First range rate measurements are decoupled from the radar camera observation vector and then concatenated with the camera observation vector. This process is illustrated in Figure 5.68. The result of this is the formation of new radar and camera observation vectors in Equation (5.26) and (5.27): radar (pseudo) measurements consists of only relative range azimuth and elevation; and the new camera measurements consists of relative azimuth, elevation and range rate. Therefore a linear KF filter and an EKF are used for the radar and camera local filters respectively:

$$z_k^{PR} = [\rho_k^R \ \theta_k^R \ \varphi_k^R]^T + v_k^R \tag{5.26}$$

and

$$z_k^{PC} = [\theta_k^C \ \varphi_k^C \dot{\rho}_k^R]^T + v_k^C. \tag{5.27}$$

In contrast to FA2, FA3 assumes that the correlation of errors in range and range rate exists and therefore needs to be accounted for. However, through the formation of the pseudo measurements the correlation between range and range rate does not exist anymore as they are a part of separate tracking filters. Therefore, unlike the filter developed in section 5.5.2, no mathematical techniques are required to account for the correlation. Furthermore, this solution sees a decrease in system complexity as a linear filter is used for the radar local filter.

In contrast to FA2, full feedback is performed in FA3, i.e.:

$$\hat{X}_{k|k}^R = \hat{X}_{k|k}^C = \hat{X}_{k|k}^F. \tag{5.28}$$

Furthermore the estimated state covariance is proportionally fed back based on the information distribution coefficient β [**72**]:

$$P_{k|k}^{R} = \beta^{R^{-1}} P_{k|k}^{F},\ P_{k|k}^{C} = \beta^{C^{-1}} P_{k|k}^{F}. \tag{5.29}$$

Where:

$$\beta^{R} + \beta^{C} = 1. \tag{5.30}$$

Instead of fixing the radar and camera information distribution coefficients, the coefficients are dynamically updated. The FC determines whether the uncertainties in radar or camera local filter estimates are higher. Initially the radar and camera distribution coefficients are set to 0.6 and 0.4 respectively. If the principal uncertainty in the radar local filter's estimate is greater than that of the camera, then β^{R} is more heavily weighted. If the camera filter's uncertainty is greater, then β^{C} is more heavily weighted. This alteration to the values of the coefficients is performed at each time-step just prior to the feedback step of Equation (5.29).

The linear KF (radar local) used in this architecture was developed in section 5.4.2. However, the EKF (camera local filter) has not yet been as it is only used in FA3. As before, the camera measurement matrix is determined by finding the Jacobian of Equation (5.27) with respect to the state variables in Equation (5.3) and evaluated around the current posterior estimate:

$$h^{PC}(X) = z_k^{PC} = \begin{bmatrix} \tan^{-1}\left(\frac{E}{N}\right) \\ -\sin^{-1}\left(\frac{D}{\sqrt{N^2+E^2+D^2}}\right) \\ \frac{N\dot{N}+E\dot{E}+D\dot{D}}{\sqrt{N^2+E^2+D^2}} \end{bmatrix}, \tag{5.31}$$

$$H_X^{PC}(k) = \left.\frac{\partial h^{PC}(X)}{\partial X}\right|_{X=\hat{X}_{k|k-1}^F}. \tag{5.32}$$

See Appendix A4 for the complete linearized measurement matrix.

Two other variants of FA3 have also been developed in order to quantitatively present the effects of state feedback, covariance feedback and the dynamic weighting of the information distribution coefficients. Therefore the developed variants are:

a) FA3a – FA3 with the FC providing no feedback at all to local filters.
b) FA3b – full feedback (to radar and camera), i.e. state and covariance with constant gain coefficients.

The results of FA3 are presented below.

5.6.4.1 Results

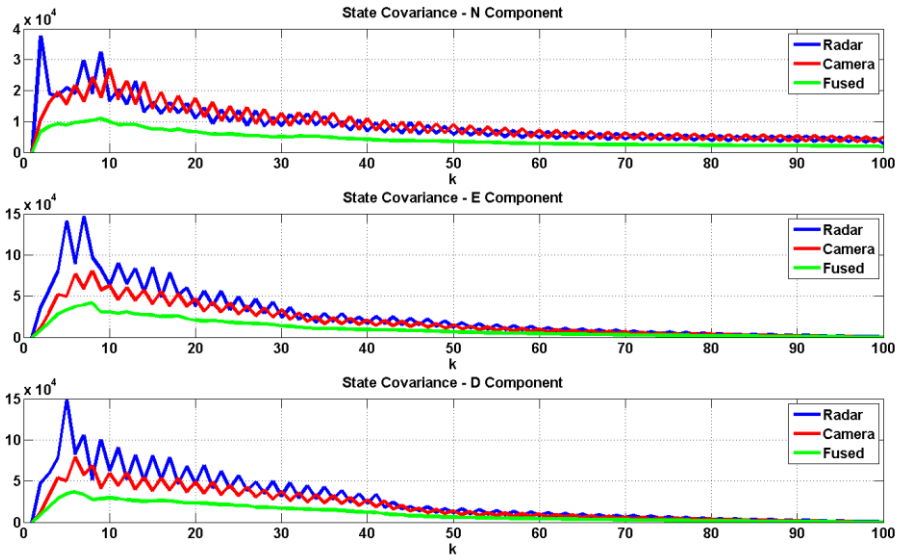

Figure 5.69: FA3 - radar, camera and fused error covariances in N, E and D directions.

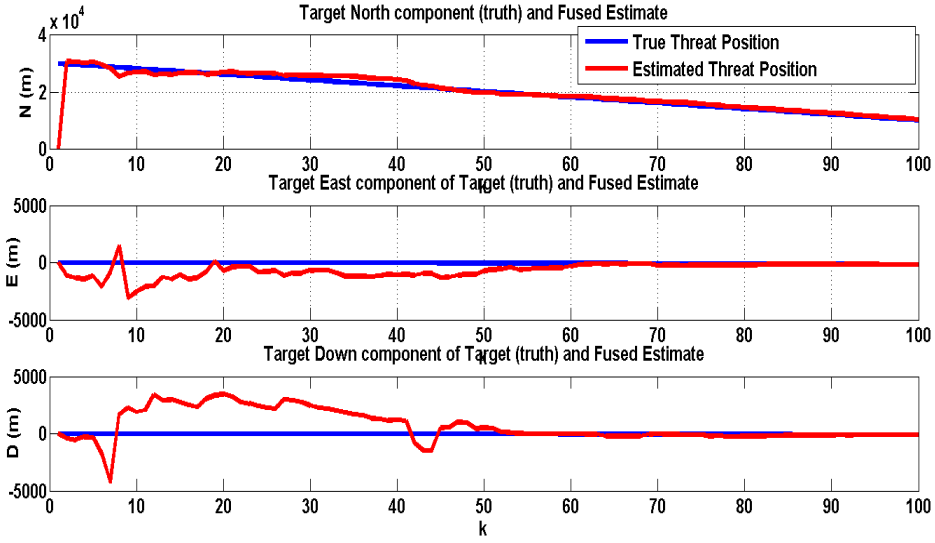

Figure 5.70: FA3 – true and estimated threat position in N, E and D directions.

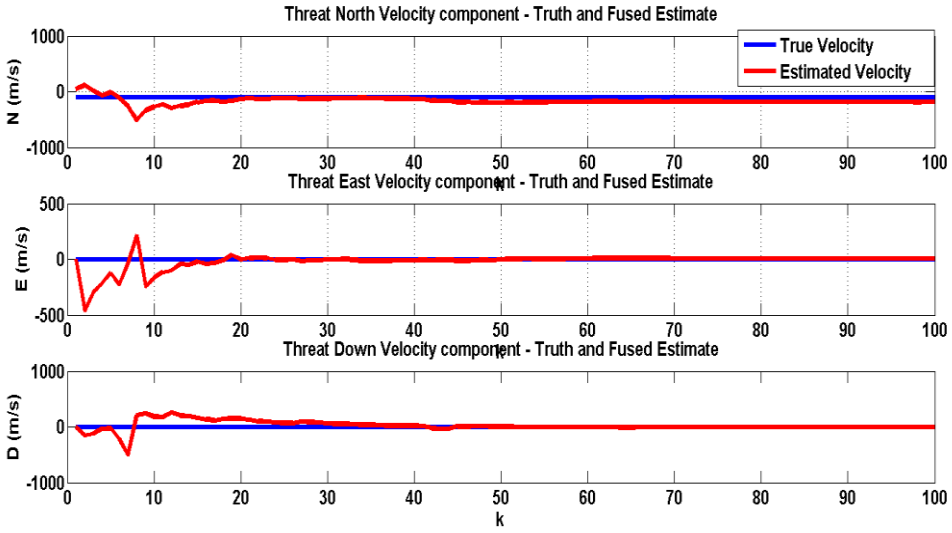

Figure 5.71: FA3 – true and estimated threat velocity in N, E and D directions.

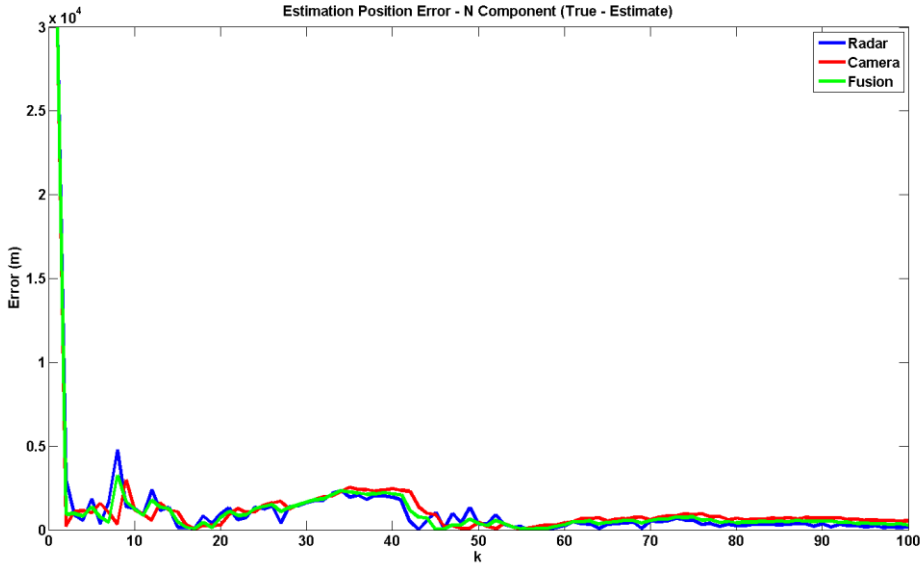

Figure 5.72: FA3 - radar, camera and fused position estimation errors in the N direction.

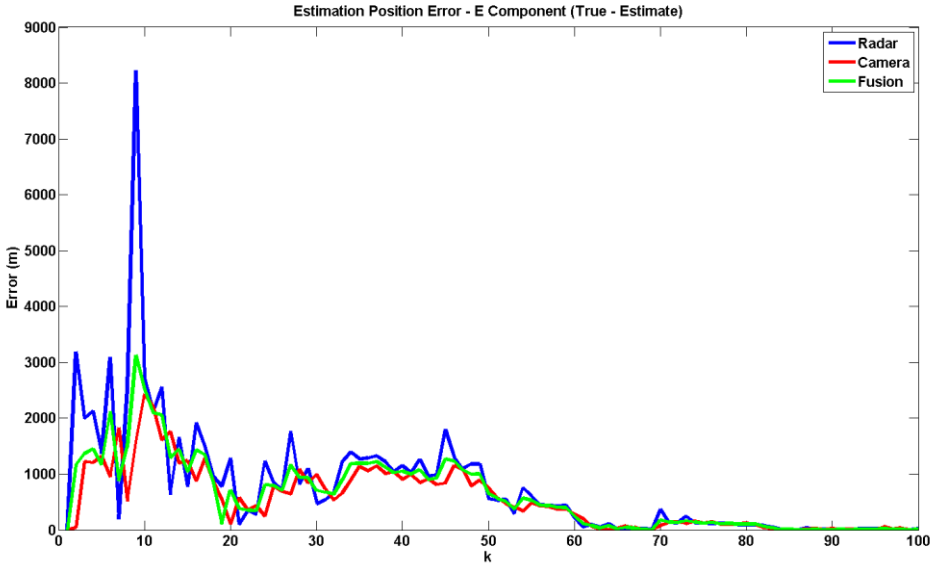

Figure 5.73: FA3 - radar, camera and fused position estimation errors in the E direction.

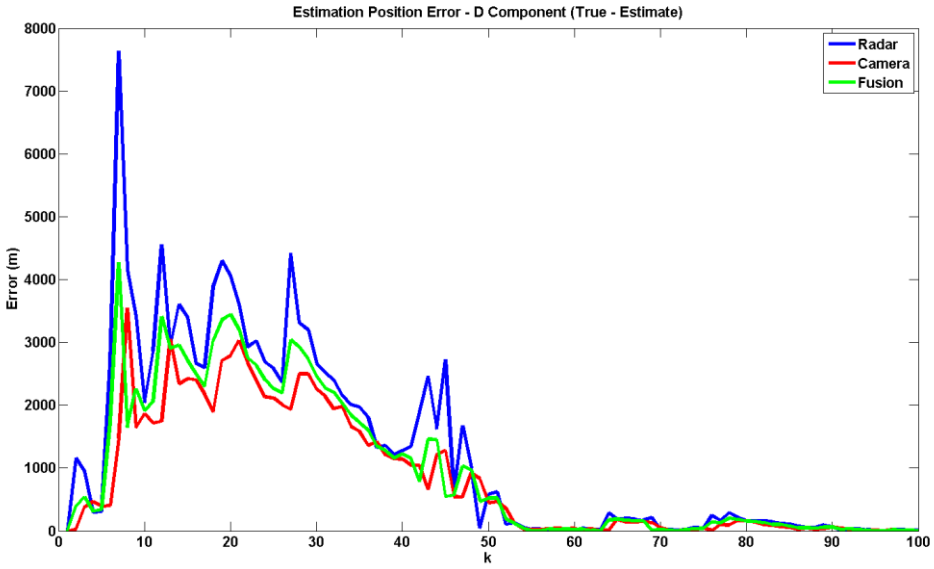

Figure 5.74: FA3 - radar, camera and fused position estimation errors in the D direction.

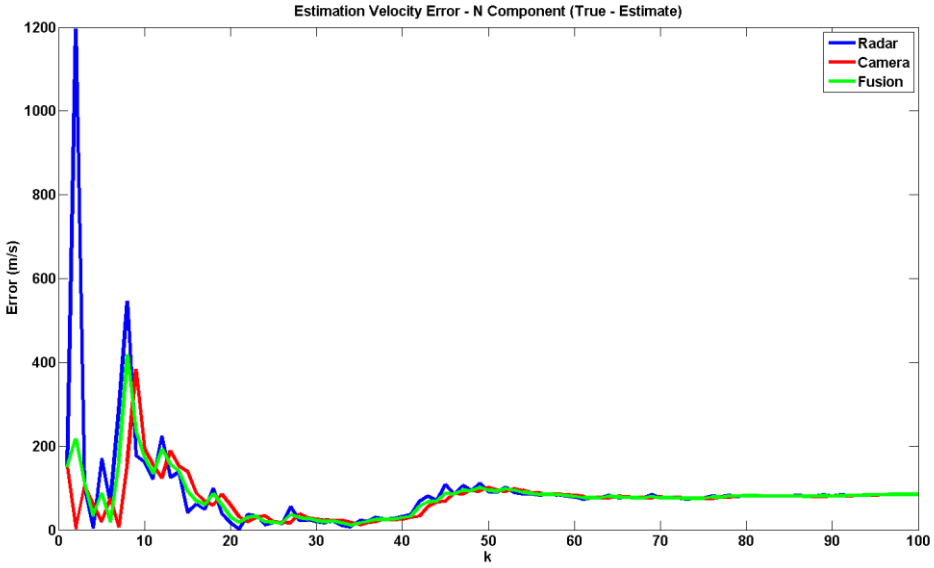

Figure 5.75: FA3 - radar, camera and fused velocity estimation errors in the N direction.

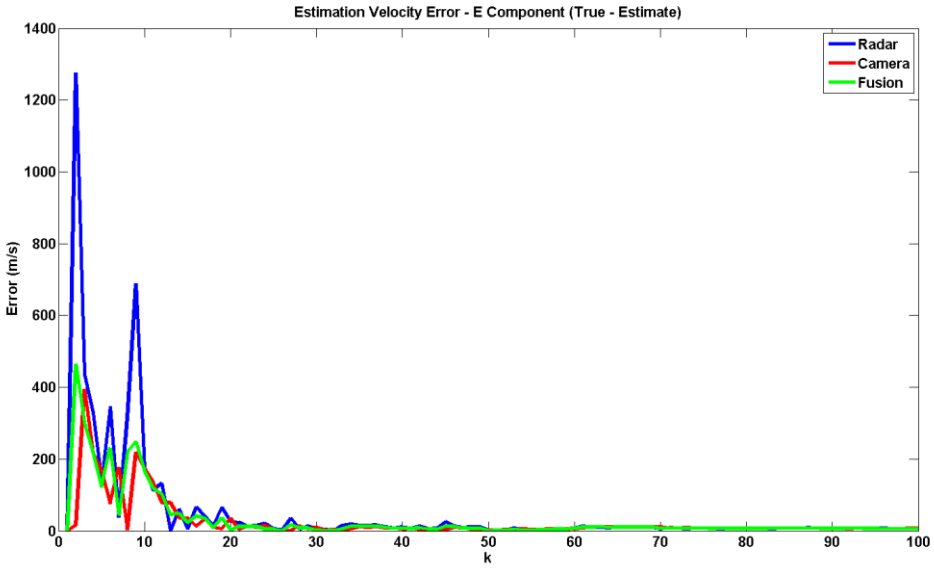

Figure 5.76: FA3 - radar, camera and fused velocity estimation errors in the E direction.

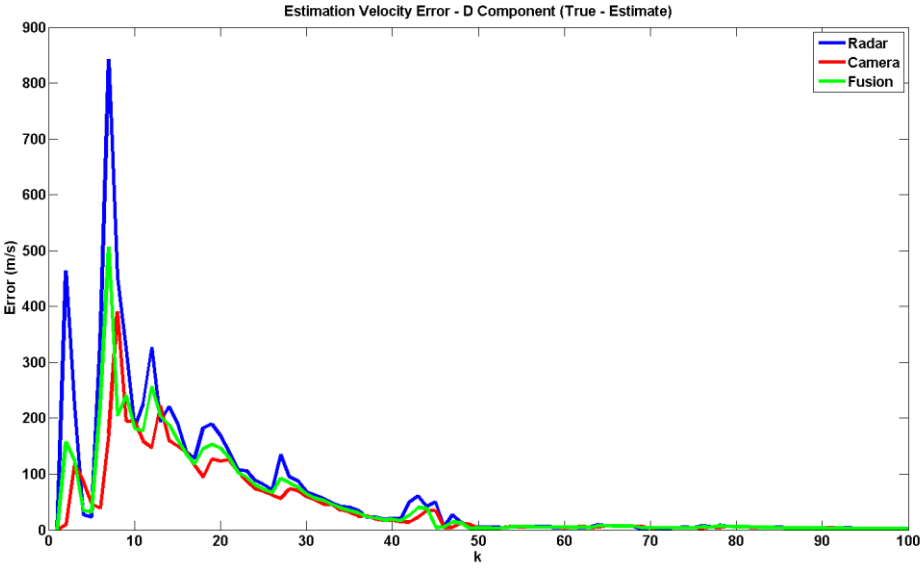

Figure 5.77: FA3 - radar, camera and fused velocity estimation errors in the D direction.

5.6.4.2 Monte Carlo Simulations

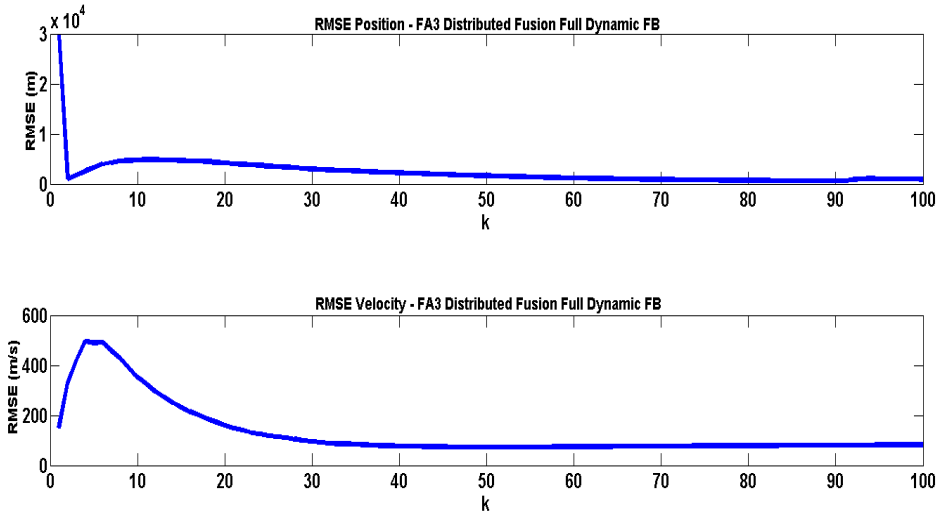

Figure 5.78: FA3 - RMSE in position and velocity.

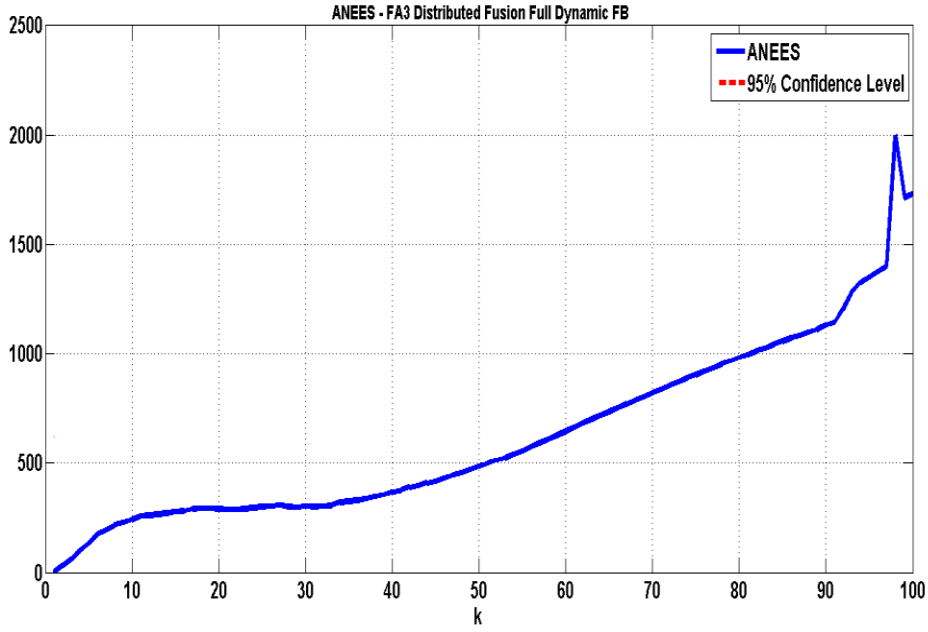

Figure 5.79: FA3 - ANEES.

5.6.5 Summary

In summary, FA1 displayed the most accurate state estimates. FA2 displayed slightly less accurate results. Furthermore, FA3 was substantially less accurate than FA1. These results will be discussed in detail in the following chapter.

Chapter 6: Discussions

This chapter discusses the results of the tracking filters developed in Chapter 5. Furthermore comparisons between like filters are performed.

The comparisons are made between the following airborne tracking filters:

- Range radar linear and nonlinear variants.
- Doppler radar with and without correlation.
- FA1, FA2 and FA3.

6.1 Single Sensor Tracking Filters

The single sensor tracking filters of EO camera, range radar and Doppler radar, presented in sections 5.3, 5.4 and 5.5 respectively, were developed so that their results would form a baseline with which the multisensor filters can be compared against. In addition to this, to prove that filters employing MSDF are more accurate than their single sensor counterparts.

6.1.1 EO Camera

It is evident from Figure 5.5 that the EO camera-only filter diverges, despite its very accurate angular measurements. This is because our state vector is not observable with only the angular measurements provided by the EO camera. This lack of observability makes the camera-only filter, on its own, unable to track the motion of the threat. In fact, the effects of this can be seen by illustrating the true threat motion versus the filter estimated motion in the NE plane.

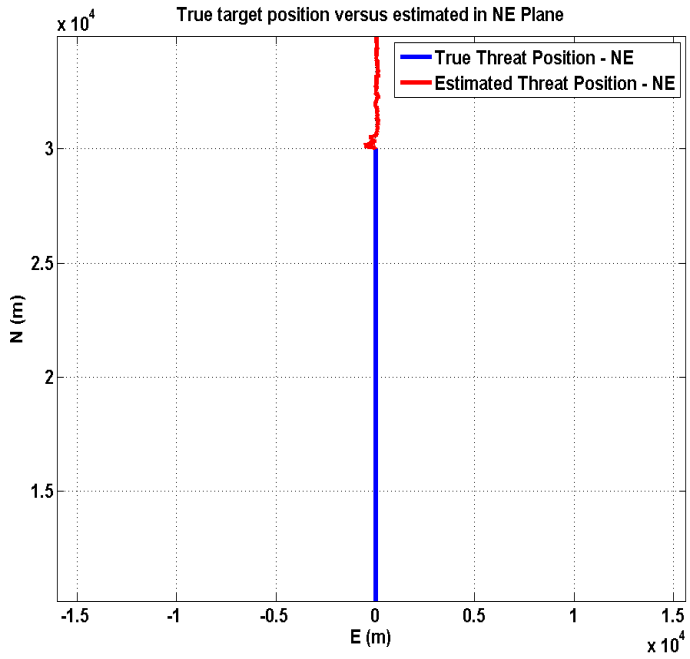

Figure 6.1: Camera filter - True and estimated threat position in the NE plane.

The filter starts tracking the threat correctly, but soon after the filter estimates that the threat is flying due north when it is actually travelling due south. This is the case even though the threat's east and down position components are adequately tracked as seen in Figure 5.6. This makes sense as the position of the threat at every discrete time is a 3-vector of position in the north, east and down directions respectively. Therefore if the N component is estimated incorrectly, as this filter does, then it implies that the threat is in an entirely different position than it actually is.

The MCS further confirms this behaviour, Figure 5.9 shows that the RMSE in position (RMSEP) is $20\ km$.

It is worth noting that, other than the camera tracking filter, all other filters converge to the steady state.

6.1.2 Radar

6.1.2.1 Without Range Rate Measurements

Two radar tracking filters were developed in sections 5.4.1 and 5.4.2 respectively: first a nonlinear filter based on an EKF and the other based on a linear KF.

The nonlinear filter was developed first as it is the most intuitive – one that uses the measurements directly from the radar; determines the linearized measurement matrix; computes the Kalman gain; and performs state and covariance estimation. Nonlinear filters are, however, computationally intensive. Therefore, the linear approach was developed to investigate its performance relative to that of the EKF based approach.

It can be seen that from the true and estimated position and velocity components, the nonlinear radar filter is adequately tracking the threat, once the steady state has been reached, in Figure 5.17 and Figure 5.18. This is reflected by projecting the true and estimated position in the NE plane in Figure 6.2. Furthermore, as the radar provides measurements in the sensor frame (BRF), the relative range, azimuth and elevation measured by the radar is contrasted against the threat's same components in the BRF in Figure 5.16. It is evident that the EKF filters out the radar sensor's noise to produce accurate estimates.

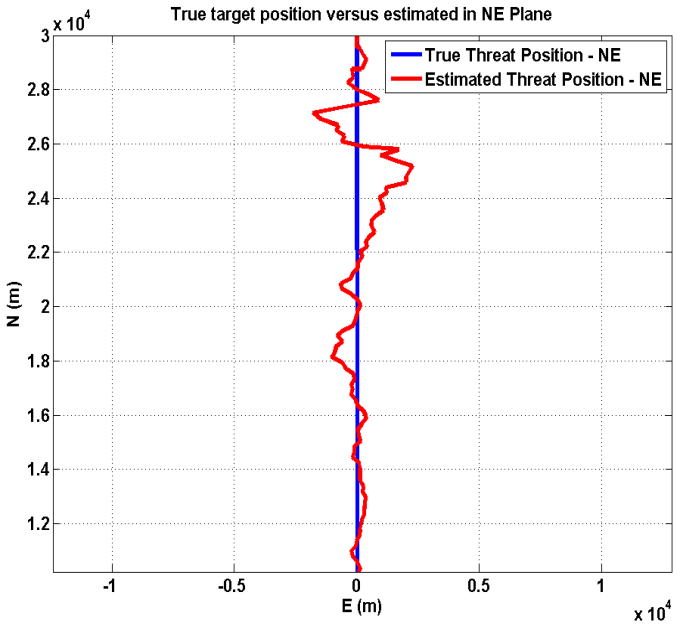

Figure 6.2: Radar nonlinear filter (EKF) - True and estimated threat position in the NE plane.

Similar conclusions can be drawn from the linear KF, whose estimated position and velocity accurately track the true threat motion in Figure 5.25 and Figure 5.26. However, the earlier estimates produced by the filter are quite far from the truth. This is attributed to the introduction of errors as a result of the coordinate transform to Cartesian coordinates. Considering that this conversion, in our simulation environment, is the last in line of many transformations and conversions (see Chapter 4), these errors eventually compound resulting in the above observation. Furthermore, as explained at the beginning of this chapter, the radar measurement error covariance matrix is set to reflect high noise in the radar measurements, and this matrix is also transformed to Cartesian space. This effect can easily be seen in Figure 6.3. This projection introduces further errors. In spite of this, the filter is able to recover from these effects and eventually produce accurate estimates.

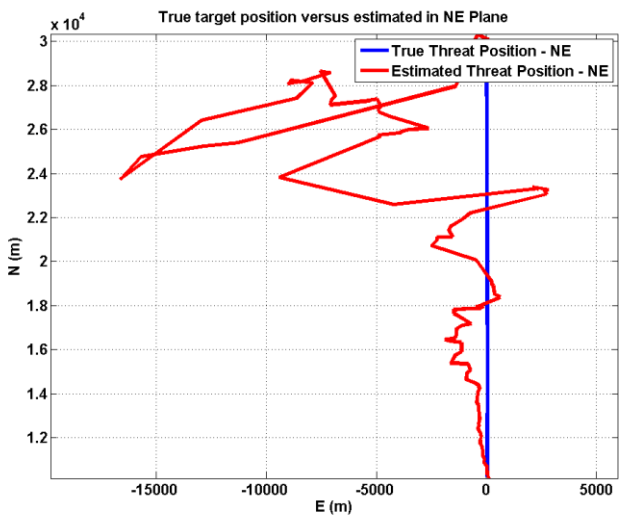

Figure 6.3: Radar linear filter (KF) tracking with Cartesian coordinates - True and estimated threat position in the NE plane.

The RMSE of the EKF in Figure 5.20 and KF in Figure 5.29 shows that both filters are accurate as the estimation error reaches an acceptable level. Moreover, it is evident that the error decreases every time-step. Therefore both filters are *consistent,* that is the filters produce increasingly accurate estimates over time.

The state error covariance of the nonlinear EKF and linear KF in Figure 5.15 and Figure 5.30 show that both filters converge. However, the KF reaches the steady-state at around $k = 40$, a lot sooner than the EKF. At this point the filter believes it is producing accurate state estimates and therefore more heavily weights the predicted measurement (opposed to the real measurement) in the estimation of the state vector; a consequence of the dynamic Kalman gain calculation.

That being said, care should be taken when interpreting the error covariance. The error covariance is the filter's self-assessment of how well it believes it is performing. The EKF's and KF's ANEES are illustrated in Figure 5.21 and Figure 5.30 respectively. The ANEES provides a measure of how true the filter produced error covariance is. It is clear that the EKF is credible as the

ANEES is approximately equal to 1. However, the KF is non-credible. Therefore the estimated KF state covariance matrix is not a reflection of the truth. In fact, the filter is optimistic; it believes the estimation error is smaller than it actually is. This is apparent from the filter's estimated error covariance in Figure 5.23. Note that this does not mean that the estimates produced by the filter are not accurate, but rather that the filter is not aware of how accurate (or inaccurate) the estimates actually are.

6.1.2.2 With Range Rate Measurements

Two radar tracking filters with range rate measurements were developed in Chapter 5: one where a correlation between measurement errors in range and rate exists and the other where the correlation does not exist. Both filters make use of an EKF due to the high nonlinearity of the range rate measurement in Cartesian space; recall that these filters track with mixed coordinates. That is, the relative motion dynamics is modelled in Cartesian coordinates, whereas the radar measurements are in spherical sensor coordinates.

It is evident from the true and estimated position and velocity components, that both filters are adequately tracking the threat once the steady state has been reached, when correlation does not exist and when accounted for in Figure 5.34, Figure 5.35 and Figure 5.42, Figure 5.43 respectively. This is reflected by projecting the true and estimated position in the NE plane of both filters in Figure 6.4 and Figure 6.5.

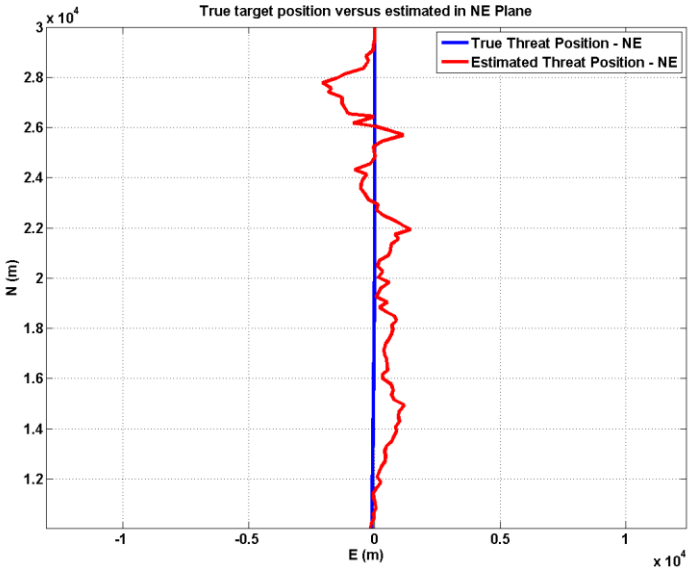

Figure 6.4: Radar EKF with range rate measurements - true and estimated threat position in the NE plane.

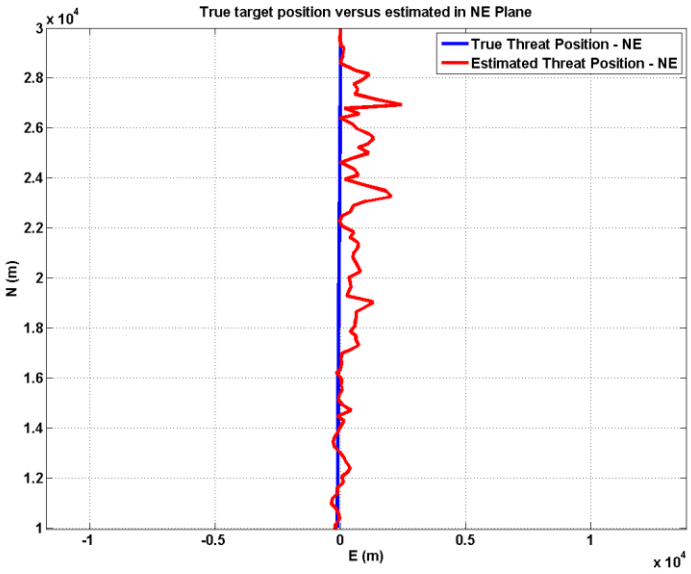

Figure 6.5: Radar EKF with range rate measurements and correlation - true and estimated threat position in the NE plane.

It is interesting to note that range rate estimates are very stable and more accurate for decreasing distance, i.e. as k moves towards 100 (relative range decreases from the start of the simulation). The range rate is illustrated in Figure 5.37.

The RMSE of both filters in Figure 5.38 and Figure 5.45 decrease over time and therefore they are consistent. Furthermore, the RMSE illustrates that their performance is very similar to the nonlinear range radar filter. Chapter 2 stated that the inclusion of range rate measurements greatly increases tracking performance [**55**]. However, in Chapter 5, it has been proved that under high noise conditions, the inclusion of range rate measurements produces performance similar to the filter where the range rate is excluded. Furthermore, velocity errors are reduced minimally, however the minimisation occurs at a sooner time than filters without velocity measurements. Note that the filter which accounts for correlation is the most computationally intensive filter.

The ANEES in Figure 5.39 and Figure 5.46 characterises that both filters are credible. Therefore, both filter's self-assessments are reliable. However, the filters are underestimating the actual estimation error.

6.1.3 Single Sensor Summary

It was proved in subsection 6.1.1 that the EO camera is not able to estimate the required state vector of a threat. This shortcoming is attributed to the fact that the camera provides only relative angular measurements.

When range rate measurements are not available in a radar tracking filter, tracking can be performed by using a nonlinear filter or a linear one by converting the measurements to Cartesian space. Despite the increased accuracy and shorter settling time exhibited by the nonlinear range radar filter shown above, it is not desirable to make use of a nonlinear filter. It is significantly more computationally intensive than its linear counterpart which was proved to yield similar performance. For our application the credibility of the filter is not of paramount importance, but rather the accuracy of a threat's state estimates. Therefore, the linear range radar filter tracking with Cartesian coordinates only is preferred when no range rate measurements are available.

Two filters were discussed that performed tracking with Doppler information: one that assumed no correlation between measurement errors in range and range rate; and the other filter assumed and accounted for the existence of the correlation by forming a pseudo measurement which decorrelated the errors. These two filters exhibited similar performance, although the latter is more computationally intensive. Furthermore, in comparison to the range radar filters, they provide increased tracking accuracy. Table 6.1 provides the RMSE in position and velocity as well as the ANEES for all the single sensor tracking filters developed.

Table 6.1: Single sensor tracking filter's RMSE in position and velocity and ANEES.

Tracking Filter	**Position RMSE** $(\boldsymbol{m})$	**Velocity RMSE** $(\boldsymbol{m/s})$	**Credibility**	
			Credible?	ANEES
EO camera	Divergent	37.21	No	N/A
Radar KF (Cartesian only)	576.5	12.8	No	N/A
Radar EKF	134.6	1.5	Yes	1.05
Radar EKF with range rate measurements	136.77	1.5	Yes	1.1
Radar EKF with range rate measurements and correlation	105	1.2	Yes	0.74

In summation, all the radar filters that were developed are able to adequately track the threat. However, the camera filter is not able to. It would be a tragedy if the availability of the accurate camera measurements would go to

waste. The next section discusses how the advantages of each sensor are utilised in the fusion architectures developed.

6.2 Multisensor Tracking Filters

Thesis objective 5, outlined in Chapter 1, required the design and implementation of an airborne tracking filter using MSDF. However, the approach adopted in this thesis was to develop multiple airborne tracking filters using MSDF. The reason for this was to investigate the performance of different approaches; to ultimately arrive at the best solution and consequently the selection for use in the TTU. Thus, three FAs were developed. FA1 is a centralised architecture employing measurement fusion. FA2 is a distributed architecture and was the first attempt at mitigating the camera-only tracking filter's deficiencies, while making good use of its accurate measurements. This was achieved through partial feedback of the fused estimate to the camera local filter. FA3 is built on FA2 by introducing state covariance feedback, using dynamically weighted information distributed coefficients, and full feedback. Furthermore, FA3 was developed in order to reduce to the computational requirements by FA2.

In spite of the fact that the radar EKF with range rate measurements and correlation accounted for, exhibited the best performance of all the single sensor tracking filters; the radar EKF which did not account for the correlation was preferred for use in the fusion architectures. The reason for this is that it requires less computing resources, especially since it forms part of a larger fusion/tracking system.

6.2.1 FA1

FA1 is the first FA developed and exhibits similarities to the single sensor filters as it contains a single tracking filter only. Theoretically, centralised fusion provides the most accurate filters. Indeed, the results from FA1 reflect

very accurate estimation in Figure 5.50 to Figure 5.54. This accurate estimation can be visualised by plotting the threat's motion in the NE plane with its filter-produced estimate (see Figure 6.6).

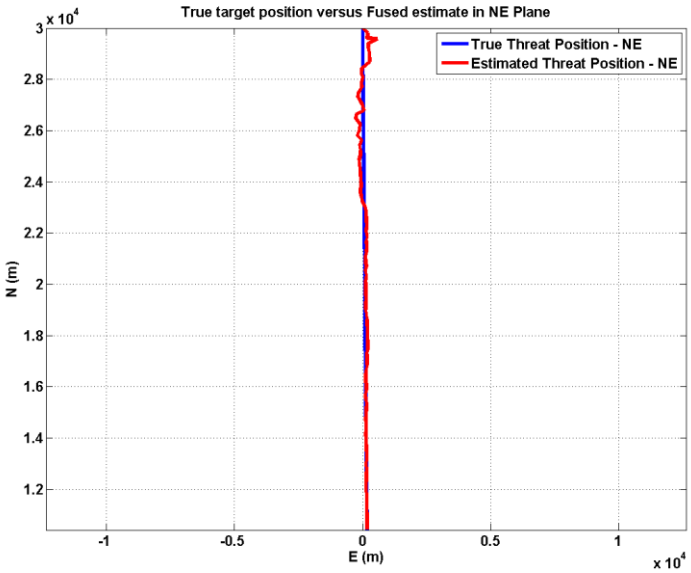

Figure 6.6: FA1 – true and estimated threat position in the NE plane.

The covariance in Figure 5.49 reaches the steady-state around $k = 40$. But unlike the single sensor KF tracking in Cartesian coordinates, this filter self-assessment is very reliable. The conclusion about the filter's reliability of its estimated error covariance is based on the evaluation of its ANEES in Figure 5.55; which shows a value of approximately 1. Therefore FA1 is highly credible. Furthermore the ANEES tells us that when $k < 40$ the filter is optimistic and pessimistic thereafter. This is what is expected as the estimation error starts out high and, when the filter settles, becomes very low.

6.2.2 FA2

FA2 is the first distributed architecture developed. It consists of two EKFs: one for the radar local filter and one for the camera local filter.

The state error covariance in Figure 5.57 provides insight into the working of this distributed architecture. In section 5.3 it was proved that the camera filter diverges; more specifically, the North component. This effect occurs again in FA2. However, very interestingly, towards the end of the simulation, the camera's North error covariance starts converging. This is as a result of the performance of partial feedback; where the local camera state estimate is set to the globally fused estimate, after radar and camera state vector fusion has been performed. This provides a means of mitigating the camera's deficiencies, while the overall tracking system reaps the benefits from the accurate camera angular measurements.

Figure 5.57 also displays the fused and radar local filter's error covariance. It is clear that both reach the steady-state. Through fusion, the fused state error covariance is minimised and stays constant practically throughout the simulation.

Figure 5.60 to Figure 5.62 and Figure 5.63 to Figure 5.65 shows the estimation error of radar, camera and fusion in position and velocity respectively, of a single flight scenario (simulation run). The radar yields the expected error; the same as that of the single sensor radar EKF developed in subsection 5.5.1. The errors in the states produced by the camera, however, are very low. The reason for this is since the performance of partial feedback enables the camera filter to track the threat, the camera's highly accurate angular measurements lead to minimal errors in state estimation. Furthermore, the fused estimation error is even lower. It can be seen that as a result of fusion, the fused estimation error is less than both the radar and camera local filters'.

The RMSE in Figure 5.66 shows that FA2 produces very accurate estimates of the threat's state vector. In fact the RMSE in position settles at $94\ m$: more accurate than all the single sensor filters. The reason for this is that fusion is performed. However, the camera only provides angular measurements and

should not greatly affect the errors in range. Despite this, we see a decrease in error which is a result of the coupling between range and angles in the spherical – Cartesian equations used by the EKF. Lastly, the ANEES in Figure 5.67 shows that, although not as close to unity as FA1, FA2 is credible.

6.2.3 FA3

FA3 is the result of a few modifications to FA2. The radar local filter in FA3 is a linear KF tracking with Cartesian coordinates only. The use of a nonlinear filter is possible as a pseudo measurements were formed by decoupling the range rate measurement from the radar observation vector. At the FC, full feedback is performed, viz. to radar and camera local filters. Furthermore, the fused error covariance matrix is fed back to both local filters using dynamically weighted information distribution coefficients.

In contrast to FA2, the camera local filter does not diverge at any point during the simulation. Figure 5.69 shows the fused as well as radar and camera local filter's estimated error covariance. In particular, this figure highlights the effects of performing dynamic feedback. Dynamic feedback causes the local filter's error covariance matrix to be brought closer to the fused error covariance matrix. At each time-step it is determined which local filter believes it is producing worse estimates. Then that filter's information distribution coefficient is increased. This results in the sawtooth shape seen in the figure and causes both local filters error covariance to converge to the fused error covariance.

Figure 5.72 to Figure 5.74 and Figure 5.75 to Figure 5.77 shows the estimation error of radar, camera and fusion in position and velocity respectively, of a single flight scenario. The result of performing full feedback as well as error covariance feedback is that estimation errors from each local filter (and consequently the FC) are sufficiently close to one another.

The RMSE in Figure 5.78 shows the tracking filter's accuracy, which settles at about $900\,m$ in position. This error is fairly high in comparison to the other fusion architectures developed and is attributed to the conversion involved in tracking with Cartesian coordinates only in the radar local filter, as well as the high process noise of the system. As stated earlier, FA3 was developed in order to reduce the computational load of FA2 and achieves this through the replacement of the nonlinear EKF in FA2 with a linear KF. Therefore the trade-off exists between computational load and tracking filter accuracy.

Figure 5.79 shows the ANEES of FA3. It is clear that its value is far from unity. Therefore the filter is non-credible. In fact it is highly optimistic and the estimated state error covariance matrix is not reliable. Note that despite this the estimates produced by FA3 are still reliable.

Recall that two variants of FA3 were developed to illustrate the importance of feedback and the dynamic weighting of the distribution coefficients. The four variants that were developed are as follows:

a) FA3a – FA3 with the FC providing no feedback at all to local filters.
b) FA3b – full feedback (to radar and camera), i.e. state and covariance with **constant** gain coefficients.

Thus, the table below lists the RMSE in position and velocity as well as the ANEES for FA3 and its four variants.

Table 6.2: Performance comparison of FA3 and its two variants.

Fusion Architecture	Position RMSE (m)	Velocity RMSE (m/s)	Credibility	
			Credible?	ANEES
FA3	940	82.5	No	N/A
FA3a	2530	36.8	No	N/A
FA3b	1900	101	No	N/A

FA3a is clearly the least accurate of the FA3 variants. Thus it quantitatively proves that through the performance of feedback more accurate filters result. In FA3b full feedback is performed but with constant information distribution coefficients. Therefore with feedback (FA3b), we see a decrease of 25% in position error from that of no feedback (FA3a); and a further decrease 50% with full dynamic feedback (FA3).

6.3 Comparison of Fusion Architectures

Before the results from the FAs are quantitatively presented, the measurement errors of the EO Camera and Doppler radar are restated from their respective design sections in Chapter 5.

Table 6.3: Measurement errors of Doppler Radar and EO Camera.

Sensor	Measurement Error	Value
Doppler Radar	Range	$200\ m$
	Azimuth	$17.19°$
	Elevation	$17.19°$
	Range Rate	$50\ m/s$
EO Camera	Azimuth	$1.15°$
	Elevation	$1.15°$

The following table summarises the accuracy and credibility of the fusion architectures developed.

Table 6.4: Performance comparison of fusion architectures. Filter accuracies and credibility. The radar-only filter has been included in the table to easily compare the fusion with the single sensor performance.

Tracking Filter	**Position RMSE** $(\boldsymbol{m})$	**Velocity RMSE** $(\boldsymbol{m/s})$	**Credibility**	
			Credible?	ANEES
FA1	45	0.9	Yes	1.07
FA2	93.8	1.36	Yes	1.64
FA3	940	82.5	No	N/A
Radar EKF	136.77	1.5	Yes	1.1

It has been proved that all three fusion architectures can accurately track an airborne threat. Moreover, in-line with thesis objective 7, that FA1 and FA2 does so more accurately than all of the single sensor tracking filters.

FA1 is clearly the most accurate architecture from Table 6.4; this was expected as centralised fusion is employed in FA1. However there is only a $50\,m$ difference in the distributed FA2's position error. This could possibly be reduced further through performing dynamic feedback. The attempt at reducing the computational load of FA2 resulted in FA3. Although FA3 is able to track the threat, its accuracy is a lot worse than the other FAs. Its position error is $\sim 1\,km$; which is too high for our application, but might be adequate for certain non-critical applications where minimising the computational load is of utmost importance.

By inspecting the original sensor measurement errors in Table 6.3, we see that the radar's measurement errors in range and range rate are $200\,m$ and

$50\ m/s$ respectively. Through fusion both FA1 and FA2 produces position and velocity errors which are considerably more accurate than the sensor's measurement errors.

To quantify the effect on fusion tracking accuracy when one of the sensors measurement errors are more accurate, Table 6.5 lists, for each architecture, the position and velocity RMSE when the radar's angular measurement accuracies are the same as the camera's and when the camera's is more accurate by a factor of about 10.

Table 6.5: Position and Velocity RMSE when both sensor's angular measurement errors are the same and when the camera's are more accurate.

Architecture	**Angle Measurement Erros**			
	$\boldsymbol{\sigma_\theta^R, \sigma_\varphi^R, \sigma_\theta^C, \sigma_\varphi^C = 0.3rad}$		$\boldsymbol{\sigma_\theta^R, \sigma_\varphi^R = 0.3rad,\ \sigma_\theta^C, \sigma_\varphi^C = 0.02rad}$	
	$\boldsymbol{RMSE^{pos}}$ (m)	$\boldsymbol{RMSE^{vel}}$ (m/sec)	$\boldsymbol{RMSE^{pos}}$ (m)	$\boldsymbol{RMSE^{vel}}$ (m/sec)
Radar-only EKF	136.77	1.5	136.77	1.5
FA1	95.5	1.1	45	0.9
FA2	135.2	1.8	93.8	1.36
FA3	970	87.5	940	82.5

Table 6.5 makes it clear that the fusion of accurate angular measurements from the camera improves the overall tracking performance.

Considering the above information, albeit FA1 is the most accurate of the architectures, FA2 is the best compromise between filter accuracy,

computational load and robustness. Therefore it has been selected as the best candidate for use in the TTU.

6.4 Summary

This chapter provided a discussion of the results derived from the simulation of single sensor tracking filters and the fusion architectures developed. The comparison between the three fusion architectures showed that FA1 is the most accurate filter for an airborne tracking system. Furthermore, that the difference in accuracy exhibited by FA2 in comparison to FA1 is negligible.

It is a known fact that centralised architectures yield the best performance, yet distributed architectures are not without merits: robustness - the FC is not a single point of failure in the system as in the centralised architecture; and they are less computationally intensive.

Taking this into account, it was concluded that FA2 is the superior architecture for an airborne threat tracking system and hence the TTU.

In summation, it has been proved that:

- All three FAs can reliably track a threat.
- Centralised FA1 is the best performing tracking filter.
- The Distributed FA2’s accuracy is also very good.
- Both FA1 and FA2 are more accurate than any single sensor tracking filter.
- FA3 is not nearly as accurate as FA1 and FA2, but it does have a lower computational burden of the lot.

Chapter 7: Conclusions

The overall aim of this research was to design an airborne tracking filter using MSDF. This chapter will summarise the findings from the work performed in this thesis to achieve this end and offer conclusions based on the findings. Additionally, recommendations for future research will be discussed, with regards to the progression of this study.

7.1 Research Objectives: Conclusions

This thesis focussed on the design of an airborne tracking filter using MSDF. MSDF fuses data from multiple sensors to provide more meaningful information than from a single sensor. The advantages and disadvantages of radar and camera sensors were analysed, which made it clear that by combining the two a superior sensor would result. A progressive design structure was adopted, where first single sensor tracking filters were developed, followed by multisensor tracking filters. Statistical analysis of the filters was performed based on numerical simulations.

Three fusion architectures were developed in order perform target tracking while making use of the measurements from multiple, redundant sensors. The architectures were termed FA1, FA2 and FA3.

FA1 was a centralised architecture performing measurement fusion, whereas FA2 and FA3 were distributed architectures employing state vector fusion. It was proved that all fusion architectures could reliably track an airborne threat under high measurement and process noise.

As this thesis is concerned with airborne tracking, the fusion architectures tracked relative motion between UAV and threat. It was decided that tracking was to be performed in the inertial NED system. This choice stemmed from the fact that relative motion in the BRF depends on attitude dynamics,

whereas the motion's projection in NED does not. Thus tracking in NED is considerably less complex than that in the BRF.

It was proved that FA1 with measurement fusion was the most accurate and the most credible tracking filter. The design of FA2 was the first attempt at mitigating the camera tracking filter's deficiency – its inability to track a threat using high accuracy angular measurements. This was successfully achieved through partial feedback and proved that FA2 was very accurate in tracking the threat in both position and velocity.

FA3 built on FA2 by introducing state covariance feedback, using dynamically weighted information distributed coefficients. Another extension was full feedback – the global fused estimate was fed back to both radar and local filters. Furthermore, FA3 was developed in order to reduce to the computational requirements by FA2. This was achieved through the decoupling of the range rate measurements from the radar observation vector, which resulted in the formation of radar and camera pseudo measurements.

The new radar pseudo measurement allowed the use of a linear KF, thereby reducing system complexity. However, the cost of this reduced system complexity is the trade-off with filter accuracy. Therefore, although requiring the least computational resources, FA3 was the worst performing architecture.

Single sensor tracking filters were developed to quantify the performance gain resulting from MSDF. These included:

- EO camera filter. An EKF was used as the angular measurements are nonlinear with respect to the state variables.
- A nonlinear and a linear radar filter. The nonlinear filter was the EKF; and the linear a KF tracking with Cartesian coordinates only. The sensor measurements were converted to Cartesian space and then debiased, which allowed the use of a linear filter.

- Two Doppler radar filter with range rate measurements. The one was designed under the assumption that no correlation exists between measurement errors in range and range rate; and the other based on the assumption that the correlation exits.

Furthermore, a simulation environment was created in which the developed filters were simulated. This simulation environment generated the UAV and threat aircraft's motion in the inertial NED system as well the Doppler radar and EO camera sensor measurements with respect to the BRF. The pre-processing step to filtering was the transformation of the sensor measurements to the NED system. This was achieved using direction cosines.

The application of MSDF demonstrated large improvements in tracking accuracy. Despite FA1 being the most accurate architecture, FA2 was assessed to be a good compromise between filter accuracy and credibility, computational load and robustness. Therefore it has been selected as the tracking filter to make up the TTU; and in-turn be integrated with the overall UAV SAS.

7.2 Recommendations for Future Work

Although the fusion architectures developed in this work accurately track a threat in inertial space, in practical applications a number of challenges exist that are not addressed in this thesis. For example, in civil airspace the UAV is likely to be surrounded by multiple aircraft. Thus, a natural extension to the filters developed would be to account for multiple targets. This would entail adding data association techniques into the filter to pair observations of target's to existing tracks. Once an observation has been discriminated against a target, tracking of the target's state vector can occur.

Another challenge not addressed in this thesis is that of manoeuvring targets. Therefore the extension to this work to add the ability to track manoeuvring threats is recommended. The widely used method of tracking manoeuvring targets makes use of the IMM filter; where multiple KF models are used for each manoeuvre stage. Then the estimates from each KF are combined with the use of a Markov model. Furthermore, to simulate the threats more accurately a Singer acceleration model could be used.

Improved sensor models are also recommended; where the sensor's FOV is taken into account. That is measurements are only returned if a target is in the sensor's FOV. Furthermore, noise should be added to the measurement model to make it more realistic in that measurements do not always arrive at the tracking filter.

Other than the sensor's FOV, the detection ranges of the sensors should also be incorporated into the fusion algorithm. If an EO camera can generally detect a target within 3 km, then data fusion should only occur when a target is within 3 km of the UAV. This can be further extended by realising the fact that the radar is not able to reliably detect a target at very close ranges. Therefore, the fusion algorithm will decide the appropriate filter, depending on the distance to the target.

This thesis deals with target tracking using simulated data. In order to validate the performance of the fusion filters with practical measurements, it is recommended that a real-world test be conducted. However, practical data for this application would be particularly hard to come by; for example, performing flight tests. An intermediate step towards that end could be a laboratory setup where a 3D graphics flight simulation is generated on one computer. The simulation should be generated from the UAV's first-person perspective. A camera-in-the-loop system should then be set up on another computer, which performs image processing algorithms and feeds the TTU with real sensor data.

Finally, the TTU is to be integrated alongside control algorithms performing CA functions; where the TTU provides the control system with threat position and velocity estimates. The amalgamation of these two subsystems sees the realisation of a UAV SAS.

Appendix A1: EO Camera Linearized Measurement Matrix

$$H_X^C(k) = \begin{bmatrix} H_{11} & 0 & H_{13} & 0 & 0 & 0 \\ H_{21} & 0 & H_{23} & 0 & H_{25} & 0 \end{bmatrix} \quad \text{(A.1)}$$

Where:

H_{11}	$\dfrac{-E}{N^2 + E^2}$
H_{13}	$\dfrac{N}{N^2 + E^2}$
H_{21}	$\dfrac{N \times D}{\left(\sqrt{\left(\dfrac{1 - D^2}{N^2 + E^2 + D^2}\right)}\,(N^2 + E^2 + D^2)^{3/2}\right)}$
H_{23}	$\dfrac{E \times D}{\left(\sqrt{\left(\dfrac{1 - D^2}{N^2 + E^2 + D^2}\right)}\,(N^2 + E^2 + D^2)^{3/2}\right)}$
H_{25}	$\dfrac{-(N^2 + E^2)}{\left(\sqrt{\left(\dfrac{1 - D^2}{N^2 + E^2 + D^2}\right)}\,(N^2 + E^2 + D^2)^{3/2}\right)}$

Appendix A2: Radar Linearized Measurement Matrix

$$H_X^R(k) = \begin{bmatrix} H_{11} & 0 & H_{13} & 0 & H_{15} & 0 \\ H_{21} & 0 & H_{23} & 0 & 0 & 0 \\ H_{31} & 0 & H_{33} & 0 & H_{35} & 0 \end{bmatrix} \quad \text{(A.2)}$$

Where:

H_{11}	$\frac{N}{\sqrt{N^2 + E^2 + D^2}}$
H_{13}	$\frac{E}{\sqrt{N^2 + E^2 + D^2}}$
H_{15}	$\frac{D}{\sqrt{N^2 + E^2 + D^2}}$
H_{21}	$\frac{-E}{N^2 + E^2}$
H_{23}	$\frac{N}{N^2 + E^2}$
H_{31}	$\frac{N \times D}{\left(\sqrt{\left(\frac{1 - D^2}{N^2 + E^2 + D^2}\right)} (N^2 + E^2 + D^2)^{3/2}\right)}$
H_{33}	$\frac{E \times D}{\left(\sqrt{\left(\frac{1 - D^2}{N^2 + E^2 + D^2}\right)} (N^2 + E^2 + D^2)^{3/2}\right)}$
H_{35}	$\frac{-(N^2 + E^2)}{\left(\sqrt{\left(\frac{1 - D^2}{N^2 + E^2 + D^2}\right)} (N^2 + E^2 + D^2)^{3/2}\right)}$

Appendix A3: Radar Linearized Measurement Matrix With Range Rate Measurements

$$H_X^R(k) = \begin{bmatrix} H_{11} & 0 & H_{13} & 0 & H_{15} & 0 \\ H_{21} & 0 & H_{23} & 0 & 0 & 0 \\ H_{31} & 0 & H_{33} & 0 & H_{35} & 0 \\ H_{41} & H_{42} & H_{43} & H_{44} & H_{45} & H_{46} \end{bmatrix} \quad \text{(A.3)}$$

Where:

H_{11}	$\frac{N}{\sqrt{N^2+E^2+D^2}}$
H_{13}	$\frac{E}{\sqrt{N^2+E^2+D^2}}$
H_{15}	$\frac{D}{\sqrt{N^2+E^2+D^2}}$
H_{21}	$\frac{-E}{N^2+E^2}$
H_{23}	$\frac{N}{N^2+E^2}$
H_{31}	$\frac{N \times D}{\left(\sqrt{\left(\frac{1-D^2}{N^2+E^2+D^2}\right)}(N^2+E^2+D^2)^{3/2}\right)}$
H_{33}	$\frac{E \times D}{\left(\sqrt{\left(\frac{1-D^2}{N^2+E^2+D^2}\right)}(N^2+E^2+D^2)^{3/2}\right)}$
H_{35}	$\frac{-(N^2+E^2)}{\left(\sqrt{\left(\frac{1-D^2}{N^2+E^2+D^2}\right)}(N^2+E^2+D^2)^{3/2}\right)}$
H_{41}	$\frac{\dot{N}E^2 - N\dot{E}E + \dot{N}D^2 - N\dot{D}D}{(N^2+E^2+D^2)^{3/2}}$
H_{42}	$\frac{N}{\sqrt{N^2+E^2+D^2}}$

H_{43}	$\frac{\dot{E}N^2 - \dot{N}EN + \dot{E}D^2 - E\dot{D}D}{(N^2 + E^2 + D^2)^{3/2}}$
H_{44}	$\frac{E}{\sqrt{N^2 + E^2 + D^2}}$
H_{45}	$\frac{\dot{D}N^2 - \dot{N}DN + \dot{D}E^2 - E\dot{E}D}{(N^2 + E^2 + D^2)^{3/2}}$
H_{46}	$\frac{D}{\sqrt{N^2 + E^2 + D^2}}$

Appendix A4: Camera Pseudo Linearized Measurement Matrix

$$H_X^C(k) = \begin{bmatrix} H_{11} & 0 & H_{13} & 0 & 0 & 0 \\ H_{21} & 0 & H_{23} & 0 & H_{25} & 0 \\ H_{31} & H_{32} & H_{33} & H_{34} & H_{35} & H_{36} \end{bmatrix} \quad \text{(A.4)}$$

Where:

H_{11}	$\frac{-E}{N^2+E^2}$
H_{13}	$\frac{N}{N^2+E^2}$
H_{21}	$\frac{N \times D}{\left(\sqrt{\left(\frac{1-D^2}{N^2+E^2+D^2}\right)}(N^2+E^2+D^2)^{3/2}\right)}$
H_{23}	$\frac{E \times D}{\left(\sqrt{\left(\frac{1-D^2}{N^2+E^2+D^2}\right)}(N^2+E^2+D^2)^{3/2}\right)}$
H_{25}	$\frac{-(N^2+E^2)}{\left(\sqrt{\left(\frac{1-D^2}{N^2+E^2+D^2}\right)}(N^2+E^2+D^2)^{3/2}\right)}$
H_{31}	$\frac{\dot{N}E^2 - N\dot{E}E + \dot{N}D^2 - N\dot{D}D}{(N^2+E^2+D^2)^{3/2}}$
H_{32}	$\frac{N}{\sqrt{N^2+E^2+D^2}}$
H_{33}	$\frac{\dot{E}N^2 - \dot{N}EN + \dot{E}D^2 - E\dot{D}D}{(N^2+E^2+D^2)^{3/2}}$
H_{34}	$\frac{E}{\sqrt{N^2+E^2+D^2}}$
H_{35}	$\frac{\dot{D}N^2 - \dot{N}DN + \dot{D}E^2 - E\dot{E}D}{(N^2+E^2+D^2)^{3/2}}$

H_{36}	$\frac{D}{\sqrt{N^2 + E^2 + D^2}}$

References

[1] Department of Defence - United States of America, "FY2009-2034 Unmanned Systems Integrated Roadmap," 2009.

[2] Federal Aviation Administration, "Sense and Avoid (SAA) for Unmanned Aircraft Sytems (UAS)," SAA Workshop, Final Report 2009.

[3] Net Resources International. Army Technology. [Online]. http://www.army-technology.com/projects/rq1-predator/rq1-predator3.html

[4] D.L. Hall and S.A.H. McMullen, *Mathematical techniques in multisensor data fusion*.: Artech House Publishers, 1992.

[5] T. Hutchings, S. Jeffryes, and SJ Farmer, "Architecting UAV sense \& avoid systems," in *IET*, 2007, pp. 1-8.

[6] Federal Aviation Administration. (2010, August) FAA. [Online]. http://adsb.tc.faa.gov/TCAS.htm

[7] ADS-B Technologies. (2011) ADS-B Technologies. [Online]. http://www.ads-b.com/

[8] The MITRE Corporation, "Airspace Integration Alternatives for Unmanned Aircraft," in *AUVSI Unmanned Systems Asia-Pacific*, Singapore, 2010.

[9] NATO Naval Armaments Group, "Sense Aand Avoid Requirements for Unmanned Aerial Vehicle Systems," Joint Capability Group On Unmanned Aerial Vehicles, 2008.

[10] A.B. Hamza, Y. He, H. Krim, and A. Willsky, "A multiscale approach to pixel-level image fusion," *Integrated Computer Aided Engineering*, vol. 12, no. 2, pp. 135-146, 2005.

[11] C. Kopp, "Sensors, Weapons and Systems for a 21st Century Counter-Air Campaign," , 1999, pp. 15-16.

[12] S. Reece, S. Roberts, C. Claxton, and D. Nicholson, "Multi-sensor fault recovery in the presence of known and unknown fault types," in *IEEE*,

2009, pp. 1695-1703.

[13] T.G. Lee, "Centralized Kalman filter with adaptive measurement fusion: its application to a GPS/SDINS integration system with an additional sensor," *International Journal Of Control Automation And Systems*, vol. 1, pp. 444-452, 2003.

[14] E. Waltz, J. Llinas, and others, *Multisensor data fusion*.: Artech House Norwood, MA, 1990, vol. 685.

[15] WJ Harlin and DA Cicci, "Ballistic missile trajectory prediction using a state transition matrix," *Applied mathematics and computation*, vol. 188, no. 2, pp. 1832-1847, 2007.

[16] GR Zabriskie, CA Cummings, and EA Rogers, "Multisensor fusion using a new procedure for ballistic path estimation," in *IEEE*, 1994, pp. 521-528.

[17] A. Gad and M. Farooq, "Data fusion architecture for maritime surveillance," in *IEEE*, vol. 1, 2002, pp. 448-455.

[18] Y. Fischer and A. Bauer, "Object-oriented sensor data fusion for wide maritime surveillance," in *IEEE*, 2010, pp. 1-6.

[19] O. Hellwich, "An alternative paradigm for data evaluation in remote sensing using multisensor data fusion," in *IEEE*, vol. 1, 1999, pp. 299-301.

[20] ASP Bhogall et al., "Extraction of forest attribute information using multisensor data fusion techniques: A case study for a test site on Vancouver island, British Columbia," in *IEEE*, vol. 2, 2001, pp. 674-680.

[21] E. Shahbazian, L. Gagnon, J.R. Duquet, M. Macieszczak, and P. Valin, "Fusion of imaging and non-imaging data for surveillance aircraft," *Sensor Fusion: Architecture, Algorithms, and Applications, SPIE Aerosense*, vol. 97, pp. 179-189, 1997, In [], a system is proposed that makes of FLIR,.

[22] S.G. Narasimhan and S.K. Nayar, "Vision and the atmosphere," in *ACM*, 2008, p. 69.

[23] The Boeing Company, "Sensor Data Fusion Technologies for Precision Strike," in *RTO SCI Symposium on Sensor Data Fusion and Integration of the Human Element*, Ottawa, Canada, 1998.

[24] S.S. Blackman, "Multiple-target tracking with radar applications," *Dedham, MA, Artech House, Inc., 1986, 463 p.*, vol. 1, 1986.

[25] S.S. Blackman and R. Popoli, *Design and analysis of modern tracking systems*.: Artech House Noorwood, MA, 1999, vol. 685.

[26] D.L. Hall, "Dirty secrets in multisensor data fusion," DTIC Document, 2001.

[27] JA Roecker and CD McGillem, "Comparison of two-sensor tracking methods based on state vector fusion and measurement fusion," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 24, no. 4, pp. 447-449, 1988.

[28] Q. Gan and C.J. Harris, "Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 37, no. 1, pp. 273-279, 2001.

[29] W. Qingchao and W. Wenfei, "Tracking method based on separation and combination of the measurements for radar and IR fusion system," *Journal of Systems Engineering and Electronics*, vol. 20, no. 2, pp. 241-246, 2009.

[30] X. Tian and Y. Bar-Shalom, "Sequential track-to-track fusion algorithm: exact solution and approximate implementation," , vol. 6969, 2008, p. 696910.

[31] H.Z. Qiu, H.Y. Zhang, and H. Jin, "Fusion algorithm of correlated local estimates," *Aerospace science and technology*, vol. 8, no. 7, pp. 619-626, 2004.

[32] Australian Centre for Field Robotics, University of Sydney, Multi Sensor Data Fusion, 2001, Course Notes.

[33] M.E. Liggins et al., "Distributed fusion architectures and algorithms for target tracking," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 95-107, 1997.

[34] L.Y. Pao, "Distributed multisensor fusion," *Am. Inst. of Aeronautics and Astronautics*, 1994.

[35] Carol Anway-Wiese, Adrienne Miller, and Stephen Beardslee, "The Effects of Centralized Vs. Distributed Data Fusion on Mission Outcome in a Cruise Missile Defense Scenario," in *6th International Command and Control Research and Technology Symposium*, 2001.

[36] E. Brookner and J. Wiley, *Tracking and Kalman filtering made easy*.: Wiley Online Library, 1998.

[37] D. Simon, *Optimal state estimation: Kalman, H [infinity] and nonlinear approaches*.: John Wiley and Sons, 2006.

[38] Y. Bar-Shalom, "Multitarget-multisensor tracking: advanced applications," *Norwood, MA, Artech House, 1990, 391 p.*, vol. 1, 1990.

[39] Y. Bar-Shalom and X.R. Li, "Estimation and tracking- Principles, techniques, and software," *Norwood, MA: Artech House, Inc, 1993.*, 1993.

[40] S.J. Julier and J.K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Spie Bellingham, WA*, vol. 3, 1997, p. 26.

[41] A. Gupta, "Nonlinear Geometric and Differential Geometric Guidance of UAVs with Vision Sensing for Reactive Obstacle Avoidance," DTIC Document, 2010.

[42] P.R. Mahapatra and K. Mehrotra, "Mixed coordinate tracking of generalized maneuvering targets using acceleration and jerk models," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 36, no. 3, pp. 992-1000, 2000.

[43] W.F. Denham and S. Pines, "Sequential estimation when measurement function nonlinearity is comparable to measurement error(Sequential linear estimation when measurement function is nonlinear, not

considering system dynamics, applied to space trajectories)," *AIAA; AIAA JOURNAL*, vol. 4, pp. 1071-1076, 1966.

[44] Z. Duan, X.R. Li, C. Han, and H. Zhu, "Sequential unscented Kalman filter for radar target tracking with range rate measurements," in *IEEE*, vol. 1, 2005, pp. 8--pp.

[45] M. Lei and C. Han, "Sequential nonlinear tracking using UKF and raw range-rate measurements," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 43, no. 1, pp. 239-250, 2007.

[46] S. Du, Z. Shi, W. Zang, and K. Chen, "Using interacting multiple model particle filter to track airborne targets hidden in blind Doppler," *Journal of Zhejiang University-Science A*, vol. 8, no. 8, pp. 1277-1282, 2007.

[47] X.R. Li and V.P. Jilkov, "A survey of maneuvering target tracking—Part III: Measurement models," , vol. 4473, 2001, pp. 423-446.

[48] G. Fasano, "Multisensor based Fully Autonomous Non-Cooperative Collision Avoidance System for UAVs," University of Naples, Faculty of Engineering, 2008.

[49] W. Premerlani and P. Bizard, "Direction cosine matrix IMU: theory," *DIY DRONE: USA*, pp. 13-15, 2009.

[50] JB Pearson and E.B. Stear, "Kalman filter applications in airborne radar tracking," *Aerospace and Electronic Systems, IEEE Transactions on*, no. 3, pp. 319-329, 1974.

[51] W. Kuhn, "Semantic reference systems," *International Journal of Geographical Information Science*, vol. 17, no. 5, pp. 405-409, 2003.

[52] GV Trunk, "Automatic detection, tracking, and sensor integration," DTIC Document, 1988.

[53] S. Zollo and B. Ristic, "On the choice of the coordinate system and tracking filter for the track-while-scan mode of an airborne pulse Doppler radar," 1999.

[54] X. Rong Li and V.P. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 39, no. 4, pp. 1333-1364, 2003.

[55] Z. Duan, C. Han, and X.R. Li, "Sequential nonlinear tracking filter with range-rate measurements in spherical coordinates," in *Citeseer*, vol. 1, pp. 599-605.

[56] Y.J. Zhang and G.Z. Xu, "Bearings-only target motion analysis via instrumental variable estimation," *Signal Processing, IEEE Transactions on*, vol. 58, no. 11, pp. 5523-5533, 2010.

[57] K. Dogancay and others, "Reducing the bias of a bearings-only TLS target location estimator through geometry translations,"," , 2004, pp. 1123-1126.

[58] K. Radhakrishnan, A. Unnikrishnan, and KG Balakrishnan, "Bearing only Tracking of Maneovuring Targets Using a Single Coordinated Turn Model," *International Journal of Computer Applications IJCA*, vol. 1, no. 1, pp. 29-36, 2010.

[59] S. Sadhu, M. Srinivasan, S. Mondal, and TK Ghoshal, "Bearing only tracking using square root sigma point Kalman filter," in *IEEE*, 2004, pp. 66-69.

[60] M.S. Arulampalam, B. Ristic, N. Gordon, and T. Mansell, "Bearings-only tracking of manoeuvring targets using particle filters," *EURASIP Journal on Applied Signal Processing*, vol. 2004, pp. 2351-2365, 2004.

[61] S.C. Nardone and V.J. Aidala, "Observability criteria for bearings-only target motion analysis," *Aerospace and Electronic Systems, IEEE Transactions on*, no. 2, pp. 162-166, 1981.

[62] Brett Bethke, "Persistent Vision-Based Search and Track Using Multiple UAVs," Massachusetts Institute of Technology, MSc Thesis 2005.

[63] Y. Watanabe, "Stochastically optimized monocular vision-based navigation and guidance," Georgia Institute of Technology, 2008.

[64] D.F. Bizup and D.E. Brown, "The over-extended kalman filter-don't use it!," , vol. 1, 2003, pp. 40-46.

[65] X. Yuan, C. Han, and Z. Duan, "Performance analysis and correlation selection with Doppler measurements," in *IEEE*, pp. 373-379.

[66] Y. Bar-Shalom, "Negative correlation and optimal tracking with Doppler measurements," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 37, no. 3, pp. 1117-1120, 2001.

[67] P. Suchomski, "Explicit expressions for debiased statistics of 3D converted measurements," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 35, no. 1, pp. 368-370, 1999.

[68] JG Wang, T. Long, and PK He, "A New Method of Incorporating Radial Velocity Measurement into Kalman Filter," *Proccedings of signal Processing*, no. 18, pp. 414-416, 2002.

[69] N. Bellotto and H. Hu, "Multisensor-based human detection and tracking for mobile service robots," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, no. 1, pp. 167-181, 2009.

[70] E. Richter, R. Schubert, and G. Wanielik, "Radar and vision based data fusion-advanced filtering techniques for a multi object vehicle tracking system," in *IEEE*, 2008, pp. 120-125.

[71] A. Farina and F.A. Studer, "Radar data processing," 1985.

[72] Y. Xu and X. Liang, "Information fusion for radar/infrared compound seeker based on federated filter," *International Journal of Digital Content Technology and its Applications*, vol. 5, no. 4, 2011.

[73] C. Blanc, L. Trassoudaine, and J. Gallice, "EKF and particle filter track-to-track fusion: a quantitative comparison from radar/lidar obstacle tracks," in *IEEE*, vol. 2, 2005, pp. 7--pp.

[74] C. Blanc, L. Trassoudaine, Y. Le Guilloux, and R. Moreira, "Track to track fusion method applied to road obstacle detection," in *Citeseer*, 2004.

[75] N. Chumerin and M.M. Van Hulle, "Cue and Sensor Fusion for Independent Moving Objects Detection and Description in Driving Scenes," *Signal Processing Techniques for Knowledge Extraction and Information Fusion*, vol. 3, pp. 161-180, 2008.

[76] Y. Zhu, Z. You, J. Zhao, K. Zhang, and X.R. Li, "The optimality for the distributed Kalman filtering fusion with feedback," *Automatica*, vol. 37, no. 9, pp. 1489-1493, 2001.

[77] I. Leibowicz, P. Nicolas, and L. Ratton, "Radar/ESM tracking of constant velocity target: comparison of batch (MLE) and EKF performance," in *IEEE*, vol. 1, 2000, pp. TUC2--3.

[78] Arkadiy Turevskiy, Stacey Gage, and Craig Buhr, "Model-Based Design of a New Light-weight Aircraft," 2007.

[79] Amir Patel, "UAV Collision Avoidance: A Specific Acceleration Matching Approach," University of Cape Town, Cape Town, MSc Thesis 2011.

[80] Z. Zhao and X.R. Li, "Probabilistic model distortion measure and its application to model-set design of multiple model approach," in *IEEE*, vol. 2, 2004, pp. 2146-2150.

[81] Ian Reid, "Estimation II ," University of Oxford, Lecture Notes 2001.

[82] Iain Peddle, "Aircraft Dynamics," University of Stellenbosch, Course Notes 2009.

[83] Y. Gao, W.J. Jia, X.J. Sun, and Z.L. Deng, "Self-tuning multisensor weighted measurement fusion Kalman filter," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 45, no. 1, pp. 179-191, 2009.

[84] W. Khawsuk and L.Y. Pao, "Decorrelated state estimation for distributed tracking of interacting targets in cluttered environments," in *IEEE*, vol. 2, 2002, pp. 899-904.

[85] D.L. Hall and S.A.H. McMullen, *Mathematical techniques in multisensor data fusion*.: Artech House Publishers, 2004.

[86] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*.: Cambridge Univ Press, 2000, vol. 2.

[87] GSW Klein and TW Drummond, "Tightly integrated sensor fusion for robust visual tracking," *Image and Vision Computing*, vol. 22, no. 10, pp. 769-776, 2004.

[88] S. You and U. Neumann, "Fusion of vision and gyro tracking for robust augmented reality registration," in *IEEE*, 2001, pp. 71-78.

[89] T. Auer and A. Pinz, "The integration of optical and magnetic tracking for multi-user augmented reality," *Computers \& Graphics*, vol. 23, no. 6, pp. 805-808, 1999.