



UNIVERSITEIT • STELLENBOSCH • UNIVERSITY
jou kennisvenoot • your knowledge partner

Measurement Correlation in a Target Tracking System Using Range and Bearing Observations

by

Morné Pistorius



Thesis presented in partial fulfilment of the requirements for the degree of

Master of Science

at Stellenbosch University

Prof. B.M. Herbst Mr. P.J. Wolfaardt
Department of Reutech Radar
Applied Mathematics Systems

December 2006

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature:



Morné Pistorius

Date:

19 – 03 – 2006



Abstract

Measurement Correlation in a Target Tracking System Using Range and Bearing Observations

Morné Pistorius

Thesis: M.Sc.

Department of Applied Mathematics
University of Stellenbosch
Private Bag X1, 7602 Matieland, South Africa
December 2006

In this work we present a novel method to do measurement correlation between target observations made by two or more radar systems. Some of the most common radar sensors available are those measuring only range (distance to the target) and bearing (azimuth angle). We use these measurements to determine the correlation between two different sensors observing the same target. As a by-product of the correlation algorithm, we find a way to estimate the target height for a target observed by at least two radar sensors. The correlation method is expounded upon, where we discuss measurement correlation for moving targets. Targets are tracked using a Kalman Filter, and correlation is done between new observations and existing target tracks. Finally, the correlation algorithm is implemented in an interactive 3D computer simulation. Results obtained indicate a high success rate, with false correlations only obtained where sensor accuracy is the limiting factor.

Opsomming

Metingskorrelasie in Radar Stelsels wat Afstand en Rigtingshoek Meet

Morné Pistorius

Tesis: M.Sc.

Departement Toegepaste Wiskunde
Universiteit van Stellenbosch
Privaatsak X1, 7602 Matieland, Suid Afrika
Desember 2006

In hierdie tesis stel ons 'n metode voor om metingskorrelasie te doen tussen waarnemings van 'n teiken, gemaak deur meer as een radar stelsel. Van die mees algemene radar sensors beskikbaar is dié wat metings maak van slegs die afstand na die teiken en die rigtingshoek tussen die sensor en die teiken. Dit word dan gebruik om die korrelasie te bepaal tussen metings van twee verskillende sensors wat dieselfde teiken waarneem. As 'n byproduk van die korrelasie algoritme vind ons 'n manier om ook die hoogte van 'n teiken te beraam wat deur ten minste twee sensors waargeneem word. Daar word verder uitgebrei op die korrelasie metode deur metingskorrelasie te beskryf vir bewegende teikens. Teikens word gevolg met behulp van 'n Kalman Filter en korrelasie word gedoen tussen nuwe waarnemings van die teiken en reeds bestaande teiken trajekte. Laastens word die korrelasie algoritme geïmplementeer in 'n interaktiewe 3D rekenaarsimulasie. Goeie korrelasie resultate word getoon, met vals korrelasies wat verkry word slegs wanneer die akkuraatheid van die sensors die beperkende faktor is.

Contents

Introduction	1
1 Measurement Correlation	2
1.1 Sensor properties	2
1.1.1 Beam profile	2
1.1.2 Sensor measurements	3
1.1.3 Measurement errors	3
1.2 Measurement correlation	6
1.2.1 Target association	6
1.2.2 The correlation algorithm	6
1.3 Target altitude estimation	13
2 Tracking	17
2.1 The Kalman Filter	17
2.1.1 Some simple probability theory	18
2.1.2 The Linear Kalman Filter	23
2.1.3 Example	25
2.2 Target tracking and correlation	27
3 Simulation	31
3.1 Introduction	31
3.2 Terrain modelling	31
3.3 Radar modelling	34
3.3.1 Sensors	34
3.3.2 Measurement volume analysis	36
3.4 Target modelling	37
3.5 Measurement correlation	38

3.6 Experimental results	39
Conclusion	41
Appendix A – SimRadar v1.0: Simulation User Guide	43
Adding radars	43
Adding airplanes	44
Changing simulation and rendering options	45
Generating a new terrain model	46
Appendix B – Derivation of the Kalman Filter Equations	47



Introduction

Radar tracking systems are very common and necessary parts of any country's aviation and defense systems. Radar sensors observe a target and use these observations to update tracks that estimates each target's current position. Due to measurement errors, these positions are always estimates of the actual target position and are given with a measure of uncertainty, called the variance. It is the task of the tracker to combine these measurements with a target's current estimated position in such a way as to give the optimal new estimated target position. Before this can be done though, we need to determine to which target any specific radar observation belongs. Consider, for example, a squadron of airplanes flying in close formation. If the measurement errors for a sensor are large, the uncertainty in the measurement could be large enough to encompass more than one target. We therefore need a way to associate a particular observation with a single target. This process is called measurement correlation, and we present a novel method to do this.

In the first chapter, we discuss measurement correlation for stationary targets. Some of the most common radar sensors available are those measuring range (distance to the target) and bearing (azimuth angle). We use these measurements to determine the correlation between two different sensors observing the same target. As a by-product of the correlation algorithm, we find a way to estimate the target height for a target observed by at least two radar sensors. The correlation method is expounded upon in the second chapter, where we discuss measurement correlation for moving targets. Targets are tracked using a Kalman Filter, and correlation is done between new observations and existing target tracks. Finally, the correlation algorithm is implemented in an interactive 3D computer simulation, discussed in chapter three. It is written in C++ and OpenGL, a rendering application programming interface used for scientific visualisation. Results obtained indicate a high success rate, with false correlations obtained only where sensor accuracy is the limiting factor.

Chapter 1

Measurement Correlation

1.1 Sensor properties

1.1.1 Beam profile

We can model a radar sensor as a large shaped beam revolving at a constant speed around the z-axis. This beam has properties like maximum range, maximum height, maximum angle of sight and rotational speed. These values can vary over a large range, depending on the intended use for the radar sensor. Typically one would find that radars designed for long range detections (several tens of kilometers) would have a slower rotational speed and would also introduce larger errors in their measurements. On the other hand, radar sensors designed for short range detections (typically only a few kilometers), tend to be more accurate and could provide measurements with a shorter detection time. Hybrid sensors are also possible where the sensors have, for example, two beams with different measurement capabilities. Such sensors can provide twice the number of detections per sweep, since the radar would only have to rotate half a turn per measurement.

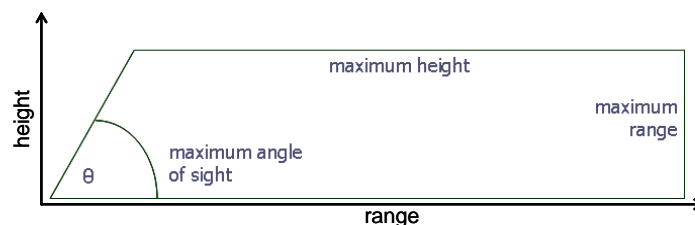


Figure 1: Typical beam profile

1.1.2 Sensor measurements

Radar sensors can be designed to measure a range of different target characteristics. Among the most common measurements made are target range, bearing, height and range rate (Doppler). Due to the large costs involved in developing and manufacturing such sensors, one will seldom use sensors that are able to measure all of these characteristics. Instead, we will only consider sensors that are able to measure target range and bearing (azimuth angle).

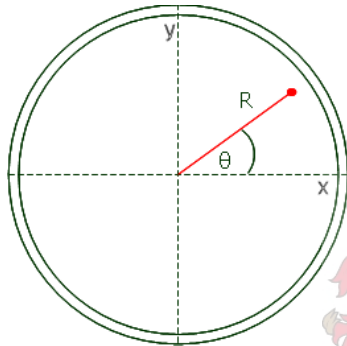


Figure 2: Sensor measuring target range and azimuth angle.

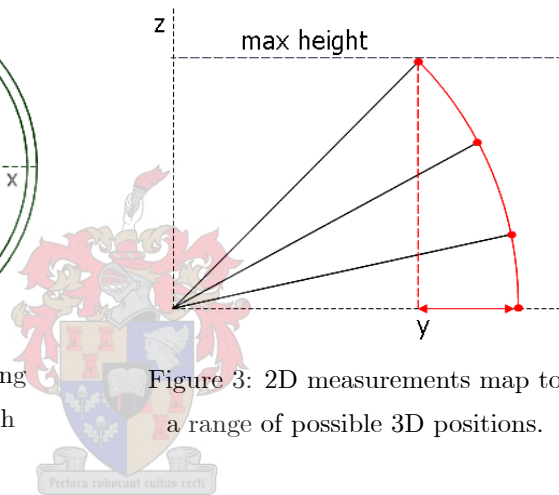


Figure 3: 2D measurements map to a range of possible 3D positions.

Note, such measurements contain no height information. The measurement only gives us the distance from the sensor to the target, and the direction of the target relative to a base reference axis. This means that, for a measurement of $r = 10000$ m, the target can be anywhere on the arc with radius r . There is no one-to-one mapping between sensor measurements and actual target positions – each measurement describes a whole range of possible target positions in Cartesian coordinates. We refer to this as the Height Problem, and present a solution in section (1.3).

1.1.3 Measurement errors

As with all such devices, we are not able to make a completely accurate measurement of the target position – all measurements contain some errors. The range error is usually

small and almost negligible compared to the maximum measurable range (about $10^{-4}\%$). The bearing error accounts for the biggest uncertainty in a measurement (especially at large distances), since this error is directly proportional to the measured target range.

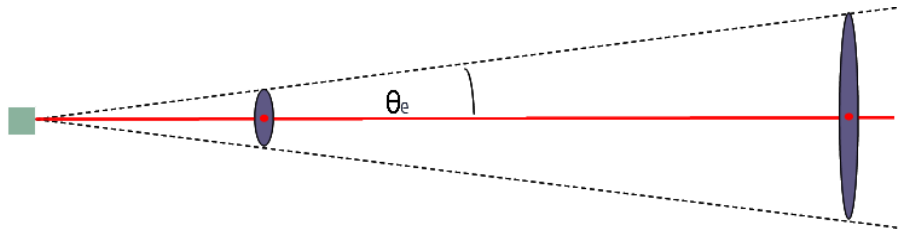


Figure 4: Uncertainty in a measurement is proportional to target range.

Derivation of the measurement covariance matrix

It is often desirable to describe a measurement and its associated uncertainty in terms of a covariance matrix. While some radar systems work in spherical coordinates, it is often easier to implement target trackers and other systems in Cartesian coordinates. We therefore need to convert the measurement errors (given in polar coordinates) to their Cartesian equivalents. Given a measurement with range (R) and azimuth angle (θ), the Cartesian equivalents are formed through the equations:

$$x = R \cos \theta, \quad y = R \sin \theta \quad (1.1)$$

with the measurement covariance matrix given by

$$C = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{bmatrix} \quad (1.2)$$

For the covariances in Cartesian coordinates, let us assume θ and R are independent. Also, where $\bar{x} = \langle x \rangle$ denotes the population mean, we have

$$\begin{aligned} \overline{\cos \theta} &= \langle \cos \theta \rangle \\ &= \langle \cos (\bar{\theta} + \theta - \bar{\theta}) \rangle \\ &\approx \langle \cos \bar{\theta} - (\theta - \bar{\theta}) \sin \bar{\theta} \rangle \end{aligned} \quad (1.3)$$

after a first order Taylor expansion. For σ_x^2 we have

$$\begin{aligned}
\sigma_x^2 &= \langle (R \cos \theta - \bar{R} \cos \bar{\theta})^2 \rangle \\
&= \langle [(\bar{R} + R - \bar{R}) \cos \theta - \bar{R} \cos \bar{\theta}]^2 \rangle
\end{aligned} \tag{1.4}$$

Then, using equation (1.3), we have

$$\begin{aligned}
\sigma_x^2 &\approx \langle ([\bar{R} + (R - \bar{R})] (\cos \bar{\theta} - (\theta - \bar{\theta}) \sin \bar{\theta}) - \bar{R} \cos \bar{\theta})^2 \rangle \\
&= \langle [\bar{R} \cos \bar{\theta} + (R - \bar{R}) \cos \bar{\theta} - \bar{R} (\theta - \bar{\theta}) \sin \bar{\theta} - (R - \bar{R}) (\theta - \bar{\theta}) \sin \bar{\theta} - \bar{R} \cos \bar{\theta}]^2 \rangle \\
&= \langle [(R - \bar{R}) \cos \bar{\theta} - \bar{R} (\theta - \bar{\theta}) \sin \bar{\theta} - (R - \bar{R}) (\theta - \bar{\theta}) \sin \bar{\theta}]^2 \rangle \\
&= \left\langle \begin{aligned} &(R - \bar{R})^2 \cos^2 \theta - 2(R - \bar{R})^2 (\theta - \bar{\theta}) \cos \theta \sin \theta - \\ &2R(R - \bar{R}) (\theta - \bar{\theta}) \cos \theta \sin \theta + 2R(R - \bar{R}) (\theta - \bar{\theta})^2 \sin^2 \theta + \\ &R^2 (\theta - \bar{\theta})^2 \sin^2 \theta + (R - \bar{R})^2 (\theta - \bar{\theta})^2 \sin^2 \theta \end{aligned} \right\rangle
\end{aligned} \tag{1.5}$$

Note,

$$\begin{aligned}
\langle R - \bar{R} \rangle &= \langle R \rangle - \langle \bar{R} \rangle = 0 \\
\text{and } \langle (R - \bar{R})^2 \rangle &= \sigma_R^2
\end{aligned}$$

Similarly,

$$\begin{aligned}
\langle \theta - \bar{\theta} \rangle &= \langle \theta \rangle - \langle \bar{\theta} \rangle = 0 \\
\text{and } \langle (\theta - \bar{\theta})^2 \rangle &= \sigma_\theta^2
\end{aligned}$$

with σ_R^2 and σ_θ^2 the range and azimuth angle measurement covariances respectively. Equation (1.5) then simplifies to

$$\begin{aligned}
\sigma_x^2 &\approx \sigma_R^2 \cos^2 \theta + R^2 \sigma_\theta^2 \sin^2 \theta + \sigma_R^2 \sigma_\theta^2 \sin^2 \theta \\
&= \sigma_R^2 \cos^2 \theta + (R^2 + \sigma_R^2) \sigma_\theta^2 \sin^2 \theta
\end{aligned} \tag{1.6}$$

Because $\sigma_R^2 \ll R^2$, we can write this as

$$\sigma_x^2 \approx \sigma_R^2 \cos^2 \theta + R^2 \sigma_\theta^2 \sin^2 \theta \tag{1.7}$$

Similarly,

$$\sigma_y^2 \approx \sigma_R^2 \sin^2 \theta + R^2 \sigma_\theta^2 \cos^2 \theta \tag{1.8}$$

$$\sigma_{xy}^2 \approx \frac{1}{2} (\sigma_R^2 - R^2 \sigma_\theta^2) \sin 2\theta \tag{1.9}$$

1.2 Measurement correlation

1.2.1 Target association

One of the main problems for tracking systems is that of data correlation. Simply put, this is the association of every new observation with one or more known targets, where the observation is classified as one of these existing targets, or a new, previously unidentified target. Due to measurement errors, it is often difficult to associate a measurement with one single target, especially if the targets are close together as with airplanes flying in formation. Some trackers classify this as a group target, where the single measurement is associated with the whole group of planes. Others try to correlate the measurement with only one plane in the group, following a 'best-correlation-wins' strategy. This could lead to false correlations, but since the planes are flying close together, it could be argued that false correlations in such a situation have a negligible effect on the tracker. Regardless of the method used for target association, all tracking systems need a way to evaluate and correlate new measurements with existing targets. One such method is presented here.

1.2.2 The correlation algorithm

Outline

As input, the algorithm takes two 2D vectors - a range and bearing measurement from two different radar sensors, and their respective covariance matrices. In this case both observations describe coordinates only in the $R\theta$ -plane. Although (as we will see in the next chapter) the algorithm generalises easily to the third dimension, for illustrative purposes we will keep to the two dimensional case. For the given radius R , we assume a certain target height, z , and project the (R, θ) measurements onto the XY-plane, as illustrated in Figure 5. The covariance matrices for these observations are combined through a Monte Carlo-like simulation to form a correlation covariance matrix (the correlation window). Using the calculated correlation covariance matrix and one observation as a mean position, we form a 2D Gaussian distribution in the XY-plane. This is used to calculate the likelihood that the second observation correlates with the first and we get this likelihood by evaluating this Gaussian at the position of the second observation. If this evaluates to a value higher than a given correlation threshold, then we can associate both measurements with the same target. Recall that we do not get any height information from an observation. The correlation likelihood will vary depending on the height that we assume and an uncorrelated result for a specific z will not necessarily

be uncorrelated for all z and we have to test all heights up to the maximum range a sensor can measure. As a by-product of the correlation algorithm, we can approximate the target height as the chosen z that yields the largest correlation likelihood.

Inputs

Often we already have an estimated position for one or more targets. These positions are usually given as the output of some sort of tracker, for example a Kalman filter. We will look at this situation in more detail in the next chapter. For now, let us assume that we are looking at a static target, and we want to correlate observations from two different radar sensors. As input, we only have range and bearing observations from two different radar sites, $\mathbf{A} = (R_a, \theta_a)$, and $\mathbf{B} = (R_b, \theta_b)$, with covariance matrices C_a and C_b .

We also assume a specific target height, z . Keep in mind that the actual target position is in 3D Cartesian space. The estimated target position (on the XY-plane) is not completely determined by just a range and bearing measurement, and is represented by a straight line ranging from the projected position at zero height to the projected position at the maximum height for the sensor.

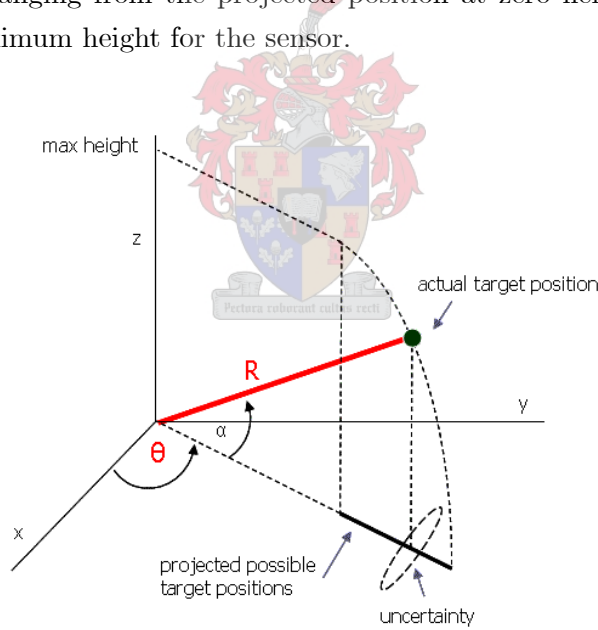


Figure 5: Possible target positions on the XY-plane from a single measurement

Algorithm

1. Assume a target height, z .
2. Convert observation \mathbf{A} plus m samples drawn from the normal distribution $N(\mathbf{A}, C_a)$ to sensor \mathbf{B} coordinates, and form the $m + 1$ sample set $\mathbf{S}_{R\theta}$.
3. For every sample $\mathbf{s}_m = (R_m, \theta_m)$ in $\mathbf{S}_{R\theta}$
 - i) draw n samples from the normal distribution $N(\mathbf{s}_m, C_b)$
 - ii) convert to XY-coordinates with

$$\alpha = \sin^{-1} \left(\frac{z}{R_n} \right) \quad (1.10)$$

$$\mathbf{s}_n = \begin{bmatrix} s_{nx} \\ s_{ny} \end{bmatrix} = \begin{bmatrix} R_n \cos(\alpha) \cos(\theta_n) \\ R_n \cos(\alpha) \sin(\theta_n) \end{bmatrix} \quad (1.11)$$

to form the $n \times (m + 1)$ sample set, \mathbf{S}_{XY} in the XY-plane.

4. From all the samples in \mathbf{S}_{XY} , calculate the mean \mathbf{x}_c and covariance matrix C_c . This is a Gaussian distribution that describes our correlation window.
5. Take observation $\mathbf{B} = (R_b, \theta_b)$ and using (1.11), convert it to XY-coordinates $\mathbf{B}' = (x_b, y_b)$.
6. Calculate the likelihood that $\mathbf{B}' \in N(\mathbf{x}_c, C_c)$.

Note that for a static target it is not necessary to project to the XY-plane - correlation could just as easily be done in the $R\theta$ -coordinate system. However, when we want to estimate the height of a target or if we want to correlate an observation with a track, this becomes a necessary step. We will discuss this further in the next section.

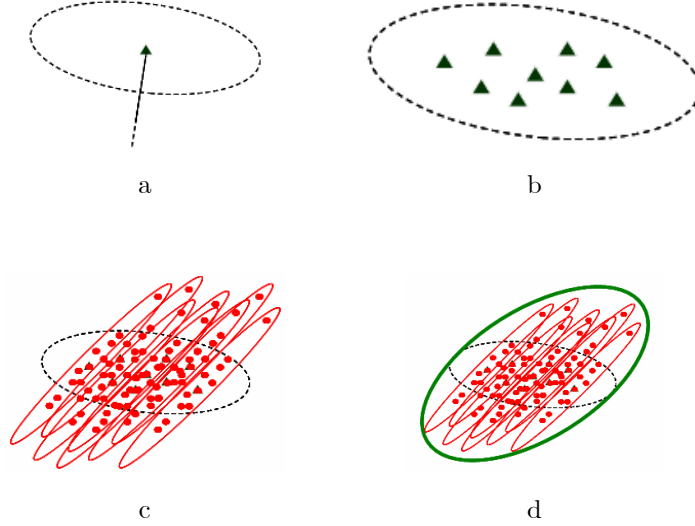


Figure 6: a) Distribution for the first observation. b) Forming the first $m + 1$ sample set. c) Forming the second $n \times (m + 1)$ sample set. d) Constructed correlation window

Scoring

Step 6 in the correlation algorithm requires that we calculate the likelihood that $\mathbf{B}' \in N(\mathbf{x}_c, C_c)$. This can be done by evaluating the Gaussian distribution described by \mathbf{x}_c and C_c at the point \mathbf{B}' .

$$p(\mathbf{B}' | \mathbf{x}_c, C_c) = \frac{1}{(2\pi)^{D/2} |C_c|^{1/2}} e^{-\frac{1}{2} (\mathbf{B}' - \mathbf{x}_c)^T C_c (\mathbf{B}' - \mathbf{x}_c)} \quad (1.12)$$

where D is the dimension of \mathbf{B}' , in this case 2. Equation (1.12) is best evaluated by calculating the log-likelihood given by

$$\ln p(\mathbf{B}' | \mathbf{x}_c, C_c) = -\frac{1}{2} (\mathbf{B}' - \mathbf{x}_c)^T C_c (\mathbf{B}' - \mathbf{x}_c) - \frac{D}{2} \ln(2\pi) - \frac{1}{2} \ln(|C_c|) \quad (1.13)$$

We measure the correlation by calculating

$$s = \frac{p(\mathbf{B}'|\mathbf{x}_c, C_c)}{p(\mathbf{x}_c|\mathbf{x}_c, C_c)} \quad (1.14)$$

where $0 \leq s \leq 1$. For a correlation threshold, T_c , we have

$$\begin{cases} s > T_c & \text{measurement correlates} \\ s \leq T_c & \text{measurement uncorrelated} \end{cases}$$

Recall that our original assumption was that the target was at height z . An uncorrelated result means only that the measurement is uncorrelated for a target at this specific height. We therefore have to test the whole range of heights between 0 and maximum height before we can decide if a target is truly uncorrelated or not. If $s > T_c$ for any height, then the measurement is correlated.

Example

Suppose we have two radar sensors located in the XY-plane at $\mathbf{A}_o = (10000, 10000)$ and $\mathbf{B}_o = (130000, 125000)$ as in the figure below. Radar A has a range of 80 km, with $\sigma_R^A = 20$ m and $\sigma_\theta^A = 0.014$. Radar B has a range of 30 km, with $\sigma_R^B = 15$ m and $\sigma_\theta^B = 0.0087$. Now suppose we have a target at the location marked with $\times = (108000, 130000)$, flying at an altitude of 4000 m. Both radars make measurements containing errors and return, $\mathbf{A}_{R\theta} = (31327 \text{ m}, 1.315)$ and $\mathbf{B}_{R\theta} = (22931 \text{ m}, 2.925)$. All angles are in radians. Take the correlation threshold as $T_c = 0.6$.

Using equations (1.7) to (1.9) to compute the measurement covariance matrices in the XY-plane, we have for Radar A

$$\begin{aligned} C_A &= \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{bmatrix} \\ &= \begin{bmatrix} \sigma_R^2 \cos^2 \theta + R^2 \sigma_\theta^2 \sin^2 \theta & \frac{1}{2} \sin 2\theta (\sigma_R^2 - R^2 \sigma_\theta^2) \\ \frac{1}{2} \sin 2\theta (\sigma_R^2 - R^2 \sigma_\theta^2) & \sigma_R^2 \sin^2 \theta + R^2 \sigma_\theta^2 \cos^2 \theta \end{bmatrix} \\ &= \begin{bmatrix} 180063 & -46986 \\ -46986 & 12688 \end{bmatrix} \end{aligned}$$

Similarly, by substituting the measurements and sensor parameters for Radar B, we have

$$C_B = \begin{bmatrix} 2052.7 & 8306.1 \\ 8306.1 & 37972 \end{bmatrix}$$

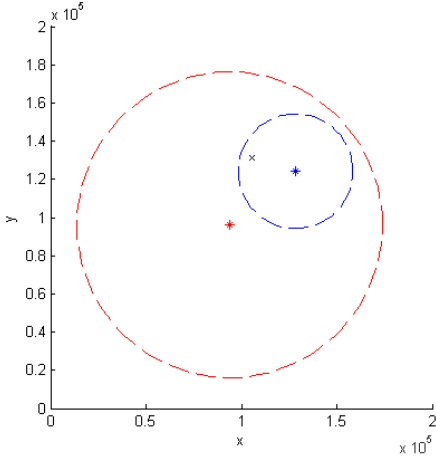


Figure 7: Two radar sites with a target positioned at \times

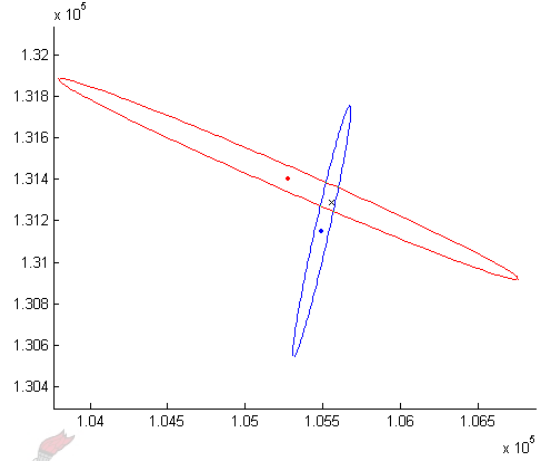


Figure 8: Observation uncertainties at 3σ . \times marks the actual target location.

Following steps 1 to 4 in the correlation algorithm described earlier, we construct the correlation window at ground level, and compute the covariance as

$$C_c = \begin{bmatrix} 140974 & -26733 \\ -26733 & 46950 \end{bmatrix}$$

Now, assume the target altitude is zero, $z = 0$. Convert the measurements from both sensors to world coordinates in the XY-plane:

$$\begin{aligned} \alpha &= \sin^{-1} \left(\frac{z}{R_A} \right) = 0 \\ \mathbf{A}_{xy} &= \begin{bmatrix} R_A \cos(\alpha) \cos(\theta_A) \\ R_A \cos(\alpha) \sin(\theta_A) \end{bmatrix} + \mathbf{A}_o \\ &= \begin{bmatrix} 107861 \\ 130060 \end{bmatrix} \end{aligned}$$

Similarly,

$$\mathbf{B}_{xy} = \begin{bmatrix} 107948 \\ 129852 \end{bmatrix}$$

To determine if both radars measured the same target, we calculate the correlation score using equations (1.13) and (1.14). We take \mathbf{A}_{xy} as the mean and evaluate the Gaussian distribution at \mathbf{B}_{xy} to give us a correlation score:

$$\ln p(\mathbf{B}_{xy}|\mathbf{A}_{xy}, C_c) = -\frac{1}{2}(\mathbf{B}_{xy} - \mathbf{A}_{xy})^T C_c (\mathbf{B}_{xy} - \mathbf{A}_{xy}) - \ln(2\pi) - \frac{1}{2} \ln(|C_c|)$$

$$s = \frac{p(\mathbf{B}_{xy}|\mathbf{A}_{xy}, C_c)}{p(\mathbf{A}_{xy}|\mathbf{A}_{xy}, C_c)} = 0.068$$

Calculating the score yields $s = 0.068 < T_c$, therefore uncorrelated. This is not surprising, since we know the target is at $z = 4000$ m. However, if we calculate $\mathbf{A}_{xy}, \mathbf{B}_{xy}$ and the score for different heights, we reach $s = T_c$ at $z = 3895$ m. Below is a graph of correlation score vs. height.

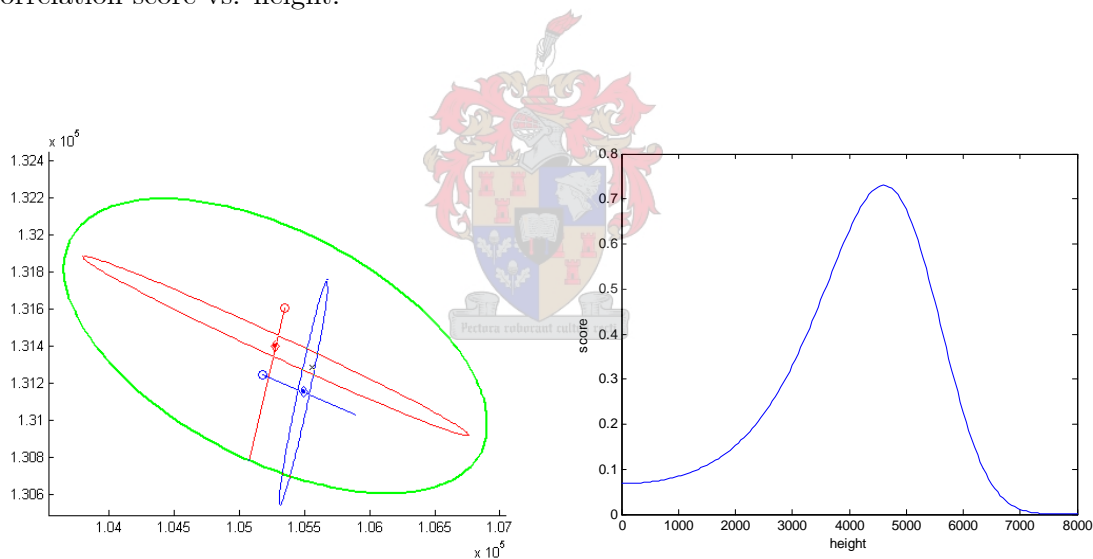


Figure 9: Correlation window for $z = 4000$

Figure 10: Correlation score at different heights

From the graph one can see that the correlation score increases as we come closer to the target height. Indeed, for perfect measurements the score would be 1 at the actual target altitude. This is an important result, and the topic of the next section.

1.3 Target altitude estimation

With measurements from just one radar station, it is impossible to determine the altitude of a target. The only bounds we have on a altitude estimate is ground level and the maximum height that a sensor can measure. To estimate target altitude, we need measurements from at least two different sensors. If these measurements were completely accurate, the altitude estimation could be solved exactly by computing the intersection between possible target positions for each sensor. Unfortunately, we don't have perfect measurements. In a worst case scenario where both measurements contain large errors, the possible target positions do not even intersect.

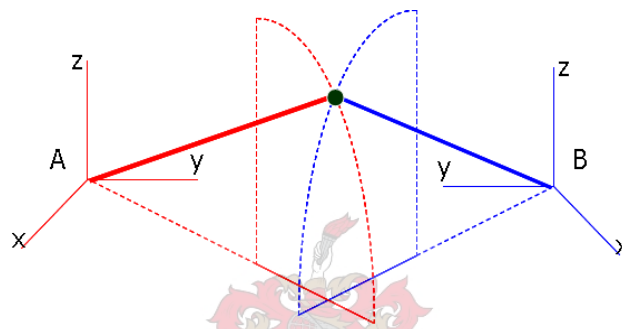


Figure 11: Projected target locations intersect on the XY-plane.

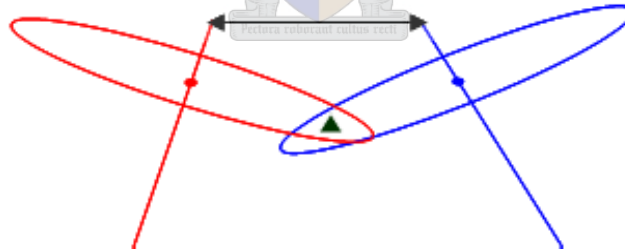


Figure 12: No intersection for two bad measurements.

As a by-product, the correlation algorithm described in the previous section also gives us an estimate of target altitude. We simply take this estimate as the altitude corresponding to the highest score. But, as we have just seen, this altitude is not always reliable, and we need some way to scale our confidence in the altitude estimation.

One way would be to set the confidence in the altitude to the correlation score at this altitude. At first glance, this seems like a good approach, since the correlation score is directly dependent on the accuracy of the measurements. The more accurate the measurements are, the better the altitude estimate would be. Interestingly, in some circumstances it is possible to get a relatively accurate altitude estimation even for bad measurements. This can be done by looking at the position of the target relative to the two radar sensors.

To illustrate this, consider how the correlation graph changes with the position of the target. Let us describe the position of the target as a function of the shortest distance from the target to the line connecting the two sensors, i.e. the perpendicular distance from the target to the line connecting the two radars.

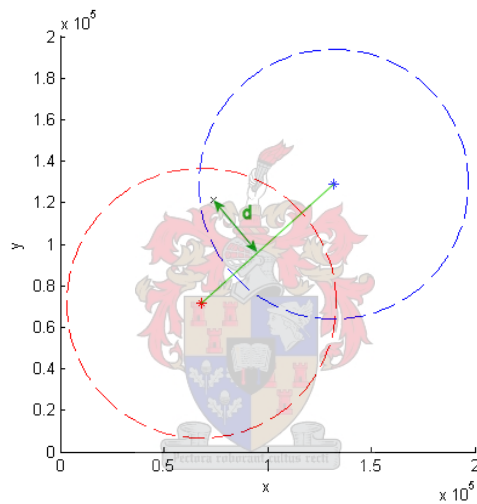


Figure 13: Target position described by perpendicular distance to line connecting sensors.

First let us consider a target situated at a large perpendicular distance, at a height of 4000 m.

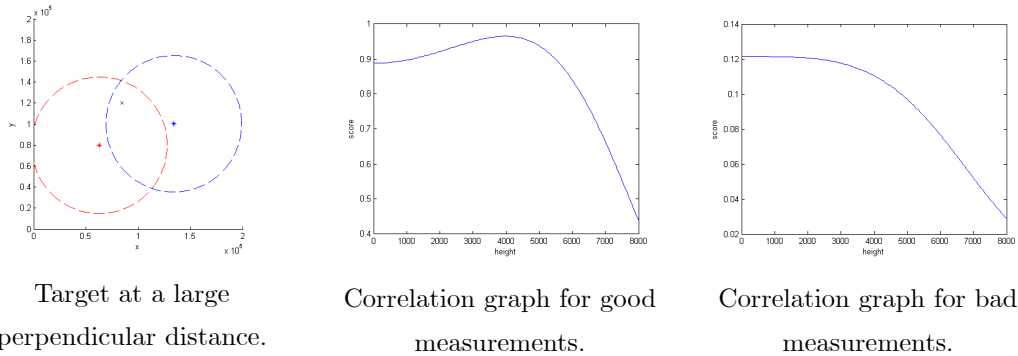


Figure 14: Correlation score for a target at a large perpendicular distance

If we had accurate measurements from both sensors, the correlation graph would peak at a height very close to 4000 m. However, the scores associated with heights between around 2000 m and 5500 m are very close to the highest score that we had at 4000 m. Correlation score is therefore not a good indicator of the validity of our height estimation. If one measurement had only slightly larger errors, it would still have had a high correlation score, but the height estimation would have been inaccurate. In the third diagram above, where both measurements contained large errors, the estimated height at ground level is completely inaccurate.

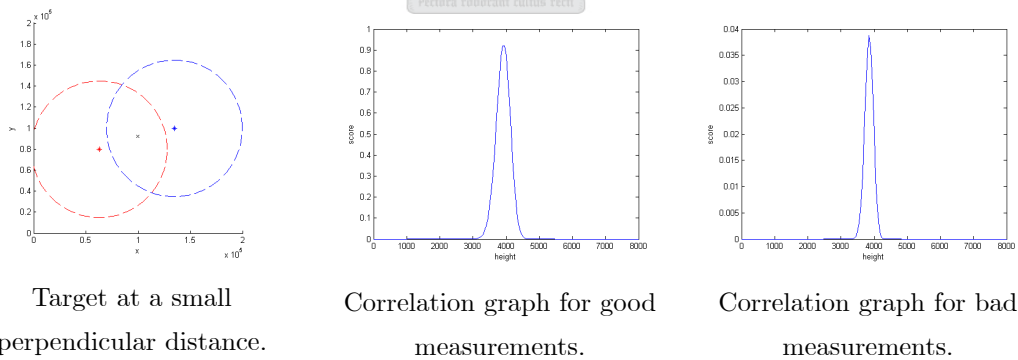


Figure 15: Correlation score for a target at a small perpendicular distance

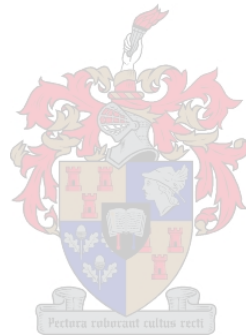
For targets that are situated close to the line connecting the two sensors, our height estimation is much more reliable. As is evident in the diagrams above, the correlation

score follows a Gaussian distribution with a small variance around the estimated height. Even for cases where we have two bad measurements, the height can still be determined reliably. The perpendicular distance of the target to the line connecting the radar sensors can therefore be used as a good measure of confidence in our height estimation. This confidence, say h_c , is given as

$$h_c = 1 - \frac{d}{R_{\max}}$$

where d is the perpendicular distance and R_{\max} is the maximum range of the two radar sensors, with $0 \leq h_c \leq 1$.

As a practical side note, the above discussion gives us good incentive to use more radar sensors near sensitive areas. The more sensors we have, the better the chances are to make observations where d is short, giving us a better height estimate of the target.



Chapter 2

Tracking

Target tracking is the process where the position of a target is predicted at any time, based on past measurements of the target and a motion model associated with the target. There are many trackers available and the choice of tracker largely depends on the problem at hand. One of the best known trackers for radar based tracking is the Kalman filter. In this chapter we discuss the use of a Kalman filter for target tracking and how the measurement correlation method discussed previously is used as input for a Kalman filter.

2.1 The Kalman Filter

The Kalman filter was developed to estimate the propagation of a dynamic process contaminated by noise. To illustrate this concept, consider the motion of a plane as it travels through the sky. It is important that air traffic controllers on the ground know the position of every plane in their air space – a position that they can measure by radar. However, these measurements are contaminated by measurement noise and are imprecise. Furthermore, the radar has to sweep through an angle of 360° and measurements of the plane's current position can only be taken at fixed intervals. We therefore need some way to estimate the position of the plane between observations. This can be done by combining the observations with knowledge about the motion of the plane. We use a mathematical model to describe the movement of the plane, called the motion model. The motion model also contains some inaccuracies, and we have to find the best way to combine the information from the motion model with our observations. This is exactly what the Kalman filter does.

Sherlock and Herbst [4] give a good discussion of the Kalman filter – we follow a

similar approach.

2.1.1 Some simple probability theory

Assumptions

First of all, we assume that the dynamic evolution of the process as well as the measurement relations, are *linear*. This seems to be a very restrictive assumption, since this seldom happens in practice. There are, however, some very effective techniques to linearise a non-linear problem and are offered as extensions to the Linear Kalman filter, e.g. the Extended Kalman Filter [3]. Another improvement on the Linear Kalman Filter is the Unscented Kalman Filter [4], but these extensions will not be discussed here.

Another assumption is that the process noise and measurement noise is zero-mean, white, Gaussian noise. Therefore, if we know their correlation matrices, we know their Gaussian distributions completely. Recall that we want to know the best possible estimate of the object's position. Under our Gaussian assumption this becomes unambiguous and can simply be defined as the *mean* of the density function.

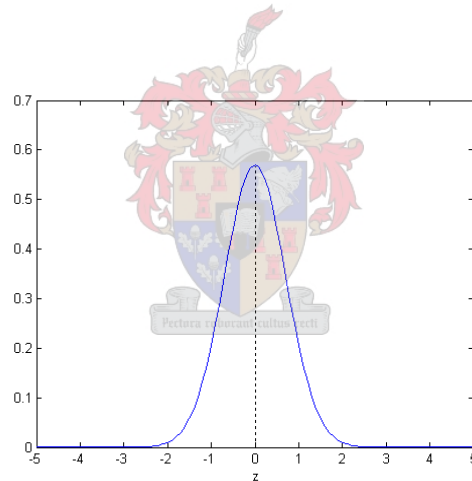


Figure 16: Gaussian density function

Applied to a static problem

We illustrate the mechanics of the Kalman filter by continuing with our aeroplane example. Suppose we have a variable x that describes the current state of the plane and contains information of its position. As we are now considering a static problem, let us assume that the plane is still on the runway and preparing for takeoff. Also, for the

sake of simplicity, let us keep it one dimensional and only consider the radial distance from, say, the control tower. We will look at the general case a bit later.

Ideally, with perfect equipment, all measurements of the plane's position at this stage should be the same. We define our history of measurements as the set

$$Z^k = \{z_i, i \leq k\} \quad (2.1)$$

where Z^k is the set of all measurements up to and including the one made at time t_k . We will describe the estimate of the state vector as the conditional mean

$$x(j|k) = E[x_j | Z^k], \quad (2.2)$$

that is, the expected state x_j at time t_j , given all our measurements up to time t_k . Similarly, the conditional covariance of the error in the estimate is given by

$$\sigma_x^2(j|k) = E[(x_j - x(j|k))(x_j - x(j|k))^T | Z^k]. \quad (2.3)$$

Suppose that we have no other information about the whereabouts of the plane but the last measurement, z_0 . Based only on our one observation, the expected position of the plane at time t_1 is

$$x(1|0) = z_0 \quad (2.4)$$

with the variance of the error in the estimate given by the variance of the error in the first measurement:

$$\sigma_x^2(1|0) = \sigma_{z_0}^2 \quad (2.5)$$

Now suppose we have another radar also making a measurement z_1 on the position of the plane at time t_1 . However, this radar is able to make more accurate measurements, with variance $\sigma_{z_1}^2 < \sigma_{z_0}^2$. Note, a smaller variance implies less uncertainty in the measurement and therefore gives a more accurate measurement. The following figure shows the conditional density functions, each one conditioned just on a single measurement.

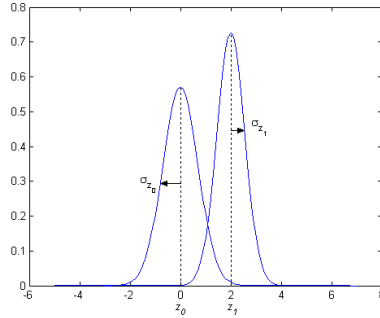


Figure 17: Conditional densities for z_0
and z_1

To obtain the best possible estimate of the position from the two measurements, we need to combine them somehow. To see exactly how this is done, refer to the appendix. For now, accept that the new combined density function is again Gaussian with mean μ_x and variance σ_x^2 given by

$$\mu_x = \left(\frac{\sigma_{z_1}^2}{\sigma_{z_0}^2 + \sigma_{z_1}^2} \right) z_0 + \left(\frac{\sigma_{z_0}^2}{\sigma_{z_0}^2 + \sigma_{z_1}^2} \right) z_1 \quad (2.6)$$

and

$$\frac{1}{\sigma_x^2} = \frac{1}{\sigma_{z_0}^2} + \frac{1}{\sigma_{z_1}^2}. \quad (2.7)$$

The inverse relationship between the variances in equation (2.7) tells us that by combining the variances for both measurements, we get a variance σ_x^2 that is smaller than either $\sigma_{z_0}^2$ or $\sigma_{z_1}^2$. Therefore, the uncertainty in the new combined estimate is less and thus more accurate than either of the previous two.

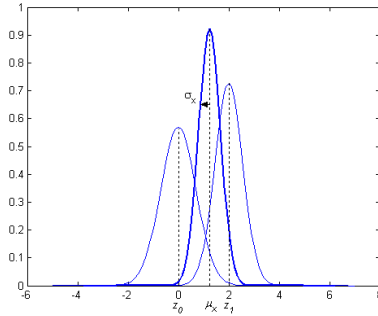


Figure 18: Combining two conditional densities

The best estimate for state x_1 then becomes

$$\hat{x}(1|1) = \mu_x \quad (2.8)$$

with variance

$$\sigma_x^2(1|1) = \left(\frac{1}{\sigma_{z_0}^2} + \frac{1}{\sigma_{z_1}^2} \right)^{-1} \quad (2.9)$$

Applied to a dynamic problem

Let us now consider the case where the plane is moving. Still applying it to the one dimensional case, let us assume a motion model given by

$$\frac{dx}{dt} = u + w \quad (2.10)$$

where u is the velocity and w the process noise, with variance σ_w^2 . Recall that we only get new measurements of the plane's position at fixed intervals. To obtain estimates for the position in the blind intervals, we use the best estimate of the previous position and update it with our knowledge of the system (2.10). This gives us a best estimate (in the absence of any other measurements) at time $t_2 = t_1 + \tau$,

$$\hat{x}(2|1) = \hat{x}(1|1) + \tau u. \quad (2.11)$$

We can continue to estimate the new position for the plane in this way, but the variance of the error will increase rapidly with time. This is in accordance with our increasing uncertainty as to the exact position of the plane in the absence of any new measurements.

The variance for the new estimated position is given by

$$\sigma_x^2(2|1) = \sigma_x^2(1|1) + \tau\sigma_w^2 \quad (2.12)$$

(see the appendix for details on how this is obtained). Clearly we need a new measurement to improve the accuracy of our estimate.

Eventually we obtain a new measurement z_2 from the radar, with variance $\sigma_{z_2}^2$. As with the case for the static problem, we again have two Gaussian densities that contain information on the plane's current position. The one is $\hat{x}(2|1)$, that contains all the information that we have up to the new measurement, and the other is the new measurement z_2 itself. Again we can combine the two to form the best new estimate of the plane's position and update $\hat{x}(2|1)$ and $\sigma_x^2(2|1)$ exactly as before,

$$x(2|2) = \left[\frac{\sigma_{z_2}^2}{\sigma_x^2(2|1) + \sigma_{z_2}^2} \right] \hat{x}(2|1) + \left[\frac{\sigma_x^2(2|1)}{\sigma_x^2(2|1) + \sigma_{z_2}^2} \right] z_2 \quad (2.13)$$

$$\sigma_x^2(2|2) = \left(\frac{1}{\sigma_x^2(2|1)} + \frac{1}{\sigma_{z_2}^2} \right)^{-1}. \quad (2.14)$$

These equations are more conveniently written as

$$\begin{aligned} x(2|2) &= \left[\frac{\sigma_{z_2}^2}{\sigma_x^2(2|1) + \sigma_{z_2}^2} \right] \hat{x}(2|1) + \left[\frac{\sigma_x^2(2|1)}{\sigma_x^2(2|1) + \sigma_{z_2}^2} \right] z_2 \\ &= x(2|1) + W(t_2)(z_2 - \hat{x}(2|1)) \end{aligned} \quad (2.15)$$

with

$$W(t_2) = \frac{\sigma_x^2(2|1)}{\sigma_x^2(2|1) + \sigma_{z_2}^2}, \quad (2.16)$$

and

$$\sigma_x^2(2|2) = \sigma_x^2(2|1) - W(t_2)\sigma_x^2(2|1). \quad (2.17)$$

Equation (2.16) for $W(t_2)$ is known as the Kalman gain and is used to scale the correction term that updates the previous best estimate to obtain the new best estimate. Let us consider for a moment the exact role that the Kalman gain plays. If our measurement is noisy and its variance $\sigma_{z_2}^2$ is large, we don't have much confidence in the measurement and, accordingly, $W(t_2)$ is small ($W(t_2) \rightarrow 0$ in the limiting case). Our best estimate according to the process model is therefore not updated by much. On the other hand, if we have little confidence in the accuracy of our dynamic process model, the process

noise σ_w^2 is large and so is the variance $\sigma_x^2(2|1)$ (see equation (2.12)). Accordingly, the Kalman gain and the measurement is weighted more heavily ($W(t_2) \rightarrow 1$ in the limiting case).

In summary, given an estimate value for the current state x_k with variance $\sigma_{x_k}^2$, we are able to obtain a new estimate for the state x_{k+1} with variance $\sigma_{x_{k+1}}^2$ by combining information gathered from a prediction of the state x_{k+1} , given all measurements made up to time t_k , and a new measurement made at t_{k+1} . All that remains to be done, is to derive similar equations for the general case. This is exactly the Linear Kalman filter and is described in the next section.

2.1.2 The Linear Kalman Filter

Deriving the equivalent equations for the general case is a lengthy (and somewhat tedious) exercise. For the brave, a full derivation is included in Appendix B – here we list the pertinent equations and give an explanation of the steps in the algorithm.

Let us assume that the dynamic evolution of the system is given by

$$\mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k \mathbf{u}_k + \mathbf{w}_k \quad (2.18)$$

where \mathbf{x}_k is a vector that describes the state (position, velocity, etc.) of the object at time t_k . A_k is a square matrix that models the motion of the object and is known. The vector \mathbf{u}_k is an external control term and describes the state of some external input (e.g., in relation to our radar tracker, this could describe the position of a radar sensor mounted on a moving object). Similarly, B_k models the motion of the external control and is known and (possibly) time-varying. The vector \mathbf{w}_k describes the system noise, which is zero-mean, white, Gaussian noise with covariance

$$Q_k = E [\mathbf{w}_k \mathbf{w}_k^T]. \quad (2.19)$$

We also gather information from measurements. These measurements can access the state variables through

$$\mathbf{z}_{k+1} = H_{k+1} \mathbf{x}_{k+1} + \mathbf{v}_{k+1}, \quad (2.20)$$

where H_{k+1} is a rectangular matrix that relates a measurement to the state, and \mathbf{v}_{k+1} describes the measurement noise (zero-mean, white, Gaussian) with covariance

$$R_k = E [\mathbf{v}_k \mathbf{v}_k^T]. \quad (2.21)$$

The matrices H , Q and R are also known. We want to propagate the probability density function (pdf) of \mathbf{x} . From this pdf we are able to estimate the value of \mathbf{x} and also obtain a measure of the accuracy of our estimate in terms of the variance of the pdf. The vector and matrix equivalents of equations (2.2) and (2.3) for these values are the conditional mean,

$$\mathbf{x}(j|k) = E \left[\mathbf{x}_j | Z^k \right], \quad (2.22)$$

and associated conditional variance

$$P(j|k) = E \left[(\mathbf{x}_j - \mathbf{x}(j|k)) (\mathbf{x}_j - \mathbf{x}(j|k))^T | Z^k \right]. \quad (2.23)$$

Algorithm

The estimation algorithm starts with the initial estimate $\mathbf{x}(0|0)$ and associated initial covariance $P(0|0)$, assumed to be known. Practical procedures to obtain the initial estimate and initial covariance are discussed in [3].

Given the estimate $\mathbf{x}(k|k)$ and its covariance $P(k|k)$ at time t_k ,

1. Compute the predicted state at time t_{k+1} prior to the measurement at t_{k+1} ,

$$\mathbf{x}(k+1|k) = A_k \mathbf{x}(k|k) + B_k \mathbf{u}_k. \quad (2.24)$$

This prediction is made entirely from our knowledge of the dynamic process model and earlier observations of the object. No new measurements have been made yet. This step accounts for the bodily drift of the pdf.

2. Compute the predicted covariance of the estimated state at time t_{k+1} , prior to the measurement at t_{k+1}

$$P(k+1|k) = A_k P(k|k) A_k^T + Q_k. \quad (2.25)$$

Note the effect of the process noise contained in Q_k . The added noise makes our prediction less certain. This can be thought of as a diffusion process that models our uncertainty in the amount of drift caused by equation (2.24).

3. Compute the *predicted* measurement at time t_{k+1}

$$\mathbf{z}(k+1|k) = H_k \mathbf{x}(k+1|k). \quad (2.26)$$

We want to combine all possible information we have to obtain the best estimate. Our Gaussian assumption guarantees that such a combination would be optimal. Equation (2.20) relates measurements to a state and could also be used to relate it to a *predicted* state. This represents new information and is assimilated in our estimate to provide better accuracy.

4. Compute the covariance of our predicted measurement at time t_{k+1}

$$S(k+1|k) = H_{k+1}P(k+1|k)H_{k+1}^T + R_{k+1}. \quad (2.27)$$

Once again, our predicted measurement contains inaccuracies that are influenced by the accuracy of the estimated state (covariance $P(k+1|k)$) and the measurement noise contained in R_{k+1} .

5. From the above covariances, compute the Kalman gain

$$W_{k+1} = P(k+1|k)H_{k+1}^T S(k+1|k)^{-1}. \quad (2.28)$$

This is used to scale the effect the new measurement has on our estimated state.

6. Finally, given the new measurement \mathbf{z}_{k+1} taken at time t_{k+1} , use it to update the estimated state

$$\mathbf{x}(k+1|k+1) = \mathbf{x}(k+1|k) + W_{k+1}(\mathbf{z}_{k+1} - \mathbf{z}(k+1|k)). \quad (2.29)$$

By combining information from all the predicted estimates and the new measurement, we obtain the new optimal estimation $\mathbf{x}(k+1|k+1)$.

7. As a measure of accuracy, compute the updated state error covariance $P(k+1|k+1) = P(k+1|k) - W_{k+1}S(k+1|k)W_{k+1}^T$.

With this last step, we have successfully propagated the pdf for \mathbf{x} from $\mathbf{x}(k|k)$, $P(k|k)$ to $\mathbf{x}(k+1|k+1)$, $P(k+1|k+1)$ and our algorithm is complete.

2.1.3 Example

As an example of the practical efficacy of the Kalman filter, we simulate the motion of a particle following a predefined curve, and compare it with the estimated motion of the filter. The motion of a particle is given by $y = \frac{1}{2}x^2 + 10$, $-2\pi \leq x \leq 2\pi$, which becomes our ground truth. This movement is divided into $T = 200$ steps with a measurement

made at each step. We simulate measurements by adding zero-mean, white, Gaussian noise, with variance $\sigma_m^2 = 0.1$, to the ground truth (see figure 15).

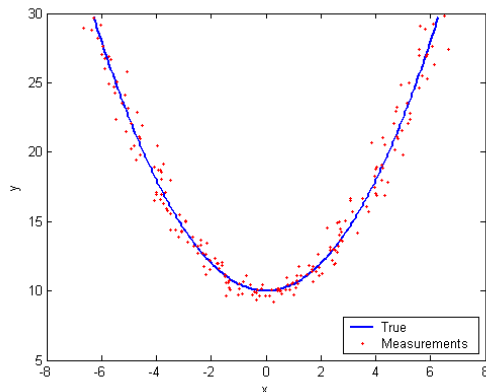


Figure 19: Simulated particle movement with superimposed measurements.

We represent the state vector as a four dimensional vector that describes the position and velocity of the particle at time t_k ,

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}. \quad (2.30)$$

The dynamic evolution of the system is given by

$$\mathbf{x}_{k+1} = A_k \mathbf{x}_k + \mathbf{w}_k, \quad (2.31)$$

with (assuming $\Delta t = 1$ sec)

$$A_k = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.32)$$

and \mathbf{w}_k zero-mean, white, Gaussian process noise with variance $\sigma_p^2 = 0.05$. Note, we have no external input, hence the absence of $B\mathbf{u}_k$ in equation (2.31). Additional parameters

are

$$Q = \begin{bmatrix} 0.05 & 0 & 0 & 0 \\ 0 & 0.05 & 0 & 0 \\ 0 & 0 & 0.05 & 0 \\ 0 & 0 & 0 & 0.05 \end{bmatrix}, \quad R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \quad \text{and} \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

The tracker is initialised with $\mathbf{x} = [x_0 \ y_0 \ 0 \ 0]^T$, where (x_0, y_0) is the initial point of the ground truth, and an initial variance of 1. The result of running this simulation for 200 time steps is shown in the following figure

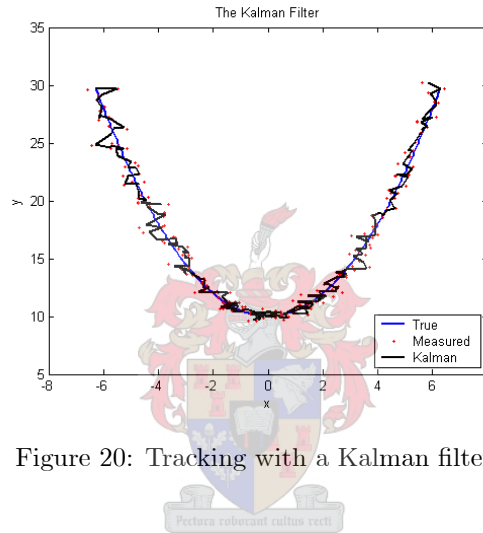


Figure 20: Tracking with a Kalman filter.

The tracked curve follows the true position of the particle somewhat erratically, but still gives better estimates of the particle's position than using just the measurement data. For this simulation, the mean error made by the measurements is $\bar{\varepsilon}_m = 0.4039$, compared to the Kalman filter with $\bar{\varepsilon}_k = 0.3286$. The situation can, of course, be improved by improving our process model to rely less on the measurements. For example, for a process model with process noise $\sigma_p^2 = 0.005$, we get a mean Kalman error of $\bar{\varepsilon}_k = 0.2630$ – a 20% decrease.

2.2 Target tracking and correlation

A Kalman filter can easily be applied to target tracking using radar sensors. For example, given estimates of target velocity (\mathbf{v}), acceleration (\mathbf{a}) and time difference (t),

the Cartesian target position, \mathbf{x} , can be determined from the simple linear equation

$$\mathbf{x}(k+1) = \mathbf{x}(k) + t\mathbf{v}(k) + \frac{t^2}{2}\mathbf{a}(k). \quad (2.33)$$

This is used to form the process equation for predicting future positions. Measurements from radar sensors are then used to update the predicted positions. Before we update the Kalman filter with a new measurement, we must first associate new measurements with current or new tracks. This is done exactly as with the measurement correlation method described in Chapter 1, except for one difference: instead of correlating two observations from two radar sensors, we correlate a single observation with an estimated position from our target tracker. The Kalman filter gives us the exact input needed for target correlation - an estimated position $\mathbf{x}(k+1)$ and associated covariance matrix $P(k+1)$. The correlation covariance matrix, C_c (correlation window) is computed as described in the in the algorithm below. The correlation score is then computed using

$$s = \frac{p(\mathbf{X}|\mathbf{x}(k+1), C_c)}{p(\mathbf{x}(k+1)|\mathbf{x}(k+1), C_c)}$$

where \mathbf{X} is the target observation in Cartesian coordinates.

By applying the correlation algorithm, we get a score measuring the correlation between the new measurement and a given target track. As we could be tracking multiple targets, we need to correlate the new observation to all our current estimated target positions. If most targets are flying relatively far apart, the correlation score would likely exceed the correlation threshold T_c for only one target track and we update that track with the new observation. Where targets are flying in close formation and we have relatively inaccurate measurements (e.g. when the targets are just entering the airspace observed by the radar sensor and we have large measurement errors), it could be that the new observation correlates with more than one target track. In this case, we update the track with the largest correlation score. However, if the correlation score is less than T_c for every target track, then we are probably observing a new target for the first time and we start a new track for that target.

Up until now we only correlated 2D measurements and track positions. This can be extended to 3D track positions and measurements by including a height estimate in the predicted position and measurement. The Kalman filter is initialised with a height estimate with a large initial variance (between 0 and maximum height). This is improved upon with the first measurement from a radar sensor. In the next chapter

we see how analysing the terrain around the radar can help to further improve these estimates.

Correlation algorithm for moving targets

For a target position estimate $\mathbf{T} = (x_t, y_t, z_t)$ with covariance matrix C_T and radar sensor at $\mathbf{X}_o = (x_r, y_r, z_r)$ with target observation $\mathbf{X}_{R\theta} = (R, \theta)$:

1. Take z as the tracker height estimate (if available), otherwise assume a target height, z with variance σ_z^2 .
2. Calculate the covariance matrix for observation $\mathbf{X}_{R\theta}$ in Cartesian coordinates (see section (1.1.3)) to form C_R .

$$\begin{aligned} C_R &= \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{bmatrix} \\ &= \begin{bmatrix} \sigma_R^2 \cos^2 \theta + R^2 \sigma_\theta^2 \sin^2 \theta & \frac{1}{2} \sin 2\theta (\sigma_R^2 - R^2 \sigma_\theta^2) \\ \frac{1}{2} \sin 2\theta (\sigma_R^2 - R^2 \sigma_\theta^2) & \sigma_R^2 \sin^2 \theta + R^2 \sigma_\theta^2 \cos^2 \theta \end{bmatrix} \end{aligned}$$

3. Take the position estimate \mathbf{T} plus m samples drawn from $N(\mathbf{T}, C_T)$ and form the $m + 1$ sample set \mathbf{S}_1 .
4. For every sample $\mathbf{s}_m = (x_m, y_m, z_m)$ in \mathbf{S}_1 draw n samples from $N\left(\mathbf{s}_m, \begin{bmatrix} C_R & \mathbf{0} \\ \mathbf{0} & \sigma_z^2 \end{bmatrix}\right)$ to form the $n \times (m + 1)$ sample set, \mathbf{S}_2 in Cartesian coordinates.
5. From all the samples in \mathbf{S}_2 , calculate the covariance matrix C_c . Using the estimated target position as the mean $\mathbf{x}_c = \mathbf{T}$, this gives us a Gaussian distribution that describes our correlation window.
6. Take observation $\mathbf{X}_{R\theta} = (R, \theta)$ and convert it to Cartesian coordinates \mathbf{X}_{xyz}

$$\begin{aligned} \alpha &= \sin^{-1}\left(\frac{z}{R}\right) \\ \mathbf{X}_{xyz} &= \begin{bmatrix} R \cos(\alpha) \cos(\theta) \\ R \cos(\alpha) \sin(\theta) \end{bmatrix} + \mathbf{X}_o \end{aligned}$$

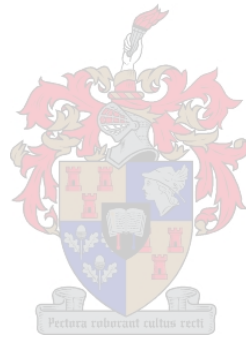
7. Calculate the likelihood that $\mathbf{X}_{xyz} \in N(\mathbf{x}_c, C_c)$. For a 3 dimensional Gaussian distribution, we use

$$\ln p(\mathbf{X}_{xyz} | \mathbf{x}_c, C_c) = -\frac{1}{2} (\mathbf{X}_{xyz} - \mathbf{x}_c)^T C_c (\mathbf{X}_{xyz} - \mathbf{x}_c) - \frac{3}{2} \ln(2\pi) - \frac{1}{2} \ln(|C_c|)$$

to calculate the correlation likelihood

$$s = \frac{p(\mathbf{X}_{xyz}|\mathbf{x}_c, C_c)}{p(\mathbf{x}_c|\mathbf{x}_c, C_c)}.$$

If s is greater than a certain correlation thresholds T_c , then the the observation $\mathbf{X}_{R\theta}$ and the estimated target position \mathbf{T} are correlated and describe the same target.



Chapter 3

Simulation

3.1 Introduction

In order to test the correlation algorithm discussed in Chapter 1, the author developed an application that closely simulates real life scenarios. Real terrain data is used to construct a 3D model of some geographical region, with aerial photography texture mapped onto the model. Radar sensors are simulated according to their design specifications, with accurate measurement capabilities and errors. Targets are simulated as different aircraft models, each flying at a constant speed on some predefined flight path. Target observations are simulated in real time, with observations made only where the targets are in the sensor measurement volume and not obstructed by the surrounding terrain. Each observation is either correlated with an existing target track, or uncorrelated and marked as a possible new target. The simulation was written by the author in C++ and OpenGL, a programming API used to render 3D graphics.

3.2 Terrain modelling

To be able to simulate realistic radar observations, it is necessary to construct an accurate model of the terrain in the sensor's measurement volume. For example, even if a target is flying in the sensor's measurement volume, it might be obscured from view by a mountain and the radar should not be able to make an observation until the target is visible. Accurate terrain data can be found on the internet for many regions, or from the local Directorate of Surveys and Mapping. It is usually in the form of digital elevation maps (DEMs), which are text files containing latitude, longitude and height values. These coordinates are mapped onto a spheroid with specific radii to the equator

and polar regions, called a *datum*. In South Africa, we use a modified Clarke 1880 datum and, more recently, the Hartbeesthoek94 datum which is very similar to the international WGS84 datum. Most commercial mapping software packages are able to read such DEMs and reproject them onto a flat surface (2D map). This map is then used to register aerial photography and combine several photos to form one large georeferenced image.

In the simulation, a 2×2 arc degree model was built of the Cape Town region ($33^\circ - 34^\circ$ S, $18^\circ - 19^\circ$ E). Various DEMs, obtained from the Chief Directorate of Surveys and Mapping, were combined to create a single DEM of the entire region. The original DEMs contain height data on a regular grid with resolution of 25m, 50m, 200m and 400m. The high resolution DEMs are only available for densely populated and mountainous regions. The combined DEM was created from the highest resolution data available for any specific region. Along with the combined height map (DEM), a texture map was also formed for the entire region. Aerial photography is used for areas where it is available, while other regions were colored according to their height. *Global Mapper*, a commercial mapping software package, was used to create these files.

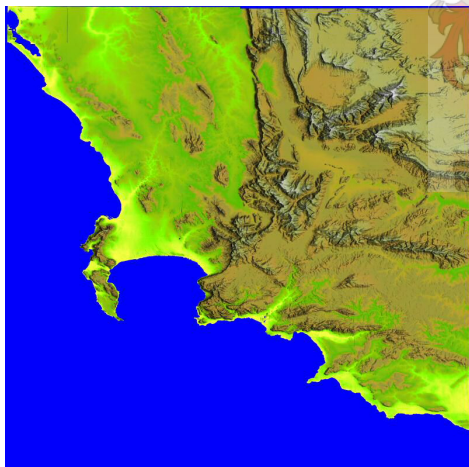


Figure 21: Digital Elevation Map (DEM) for the Cape Town region, coloured according to height.

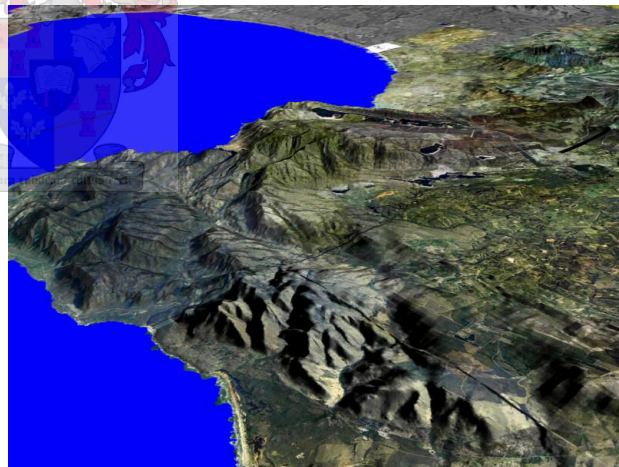


Figure 22: 3D model with texture mapped aerial photography.

The combined data for the height maps and imagery amount to about 10 GB. It is

impossible to create and display such a large model on any desktop computer currently available. To render such a model at acceptable frame rates, it is necessary to break it down into smaller, manageable pieces. To do this, we use a technique called Paged Level Of Detail (pagedLOD). With pagedLOD, we break the model down to thousands of different resolution tiles and only render parts of the terrain currently visible in the view frustum. For example, we start at level 0 by creating one tile, which consist of a greatly simplified version of the entire model. This is rendered when the camera is far away from the model. As the camera moves closer to the model, we divide the original tile into four, higher resolution, slightly less simplified tiles – these are level 1 tiles. Each of these tiles are also divided into four higher resolution tiles (level 3), etc. This is a recursive process until we reach the highest resolution tiles which are not simplified at all. At any one stage, only the tiles which are currently in the view frustum are loaded and displayed. This enables us to keep the amount of data sent to the graphics subsystem to a minimum, while keeping constant frame rates at any resolution. The number of tiles created for any level l is given by $n = 4^l$. The model for the Cape Town region is generated to a level 8 resolution, amounting to a total of 87 381 files. It takes approximately three and a half days to generate this model on a 3.0GHz Pentium 4 with 1 GB of RAM, but this is a once off process. Details of how to generate such a model is given in Appendix A.

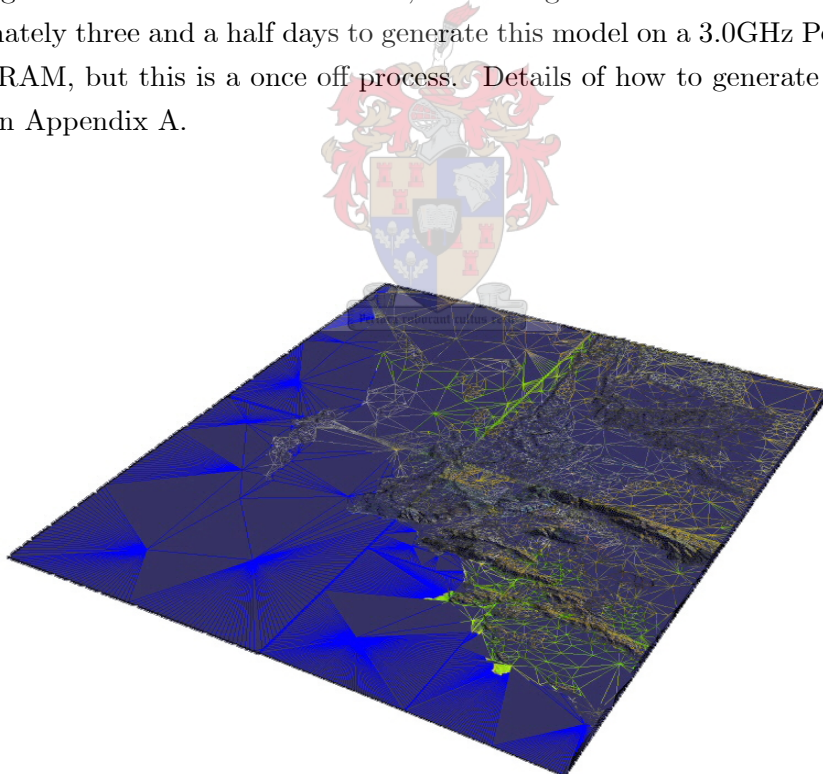


Figure 23: Level 1 tiles with level 2 tiles in the fourth quadrant.

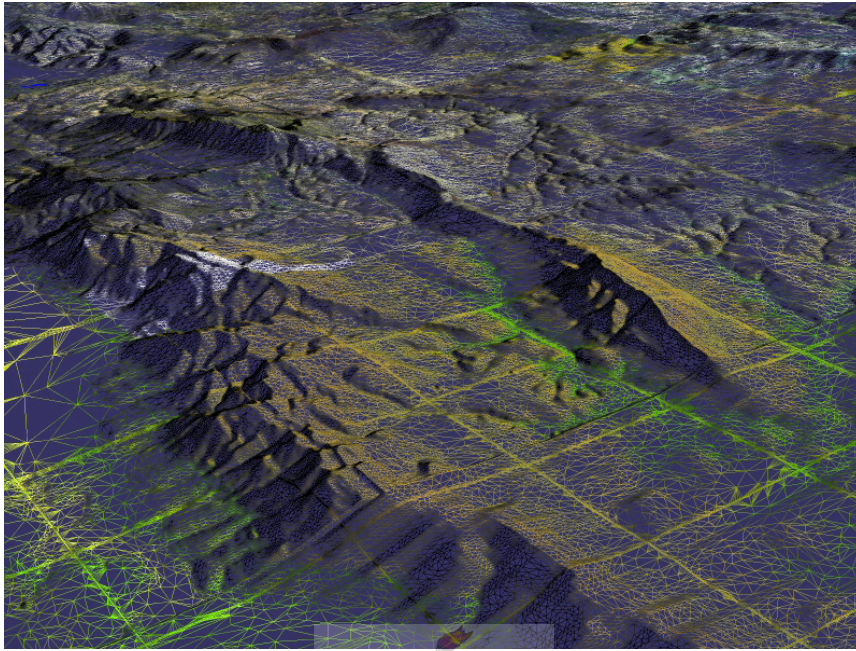


Figure 24: Higher resolution level 6 tiles.

3.3 Radar modelling

3.3.1 Sensors

Radar sensors are modelled as described in Chapter 1, as a flat beam with maximum range, maximum height, maximum angle of sight and rotational speed. These sensors can be placed at any given latitude and longitude, or by moving them to the desired position on the terrain with the mouse. Range and bearing measurements are made of any visible target, with measurement errors incorporated by drawing random samples from $N(R, \sigma_R^2)$ and $N(\theta, \sigma_\theta^2)$. When adding new radar sensors, the initial beam direction is set. Multi-faced radars are modelled by placing different radars with the same rotational speed at the same position, and by setting their initial beam angles to the desired separation. For each observation, the line connecting the sensor and target is checked for terrain intersections. If this line intersects the terrain, the target is invisible to the sensor and the observation is discarded.

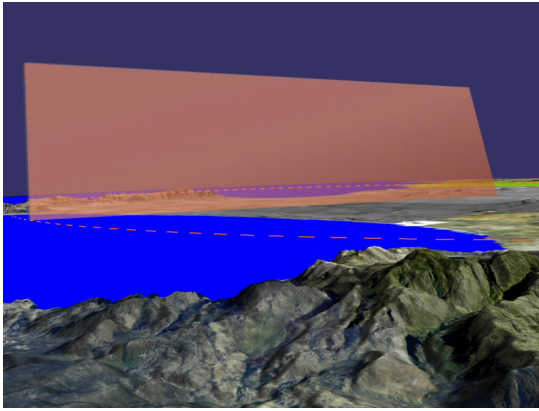


Figure 25: Beam profile for a typical sensor.

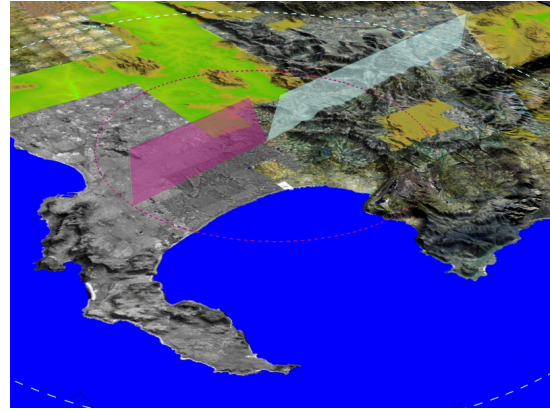


Figure 26: Two faced radar sensor

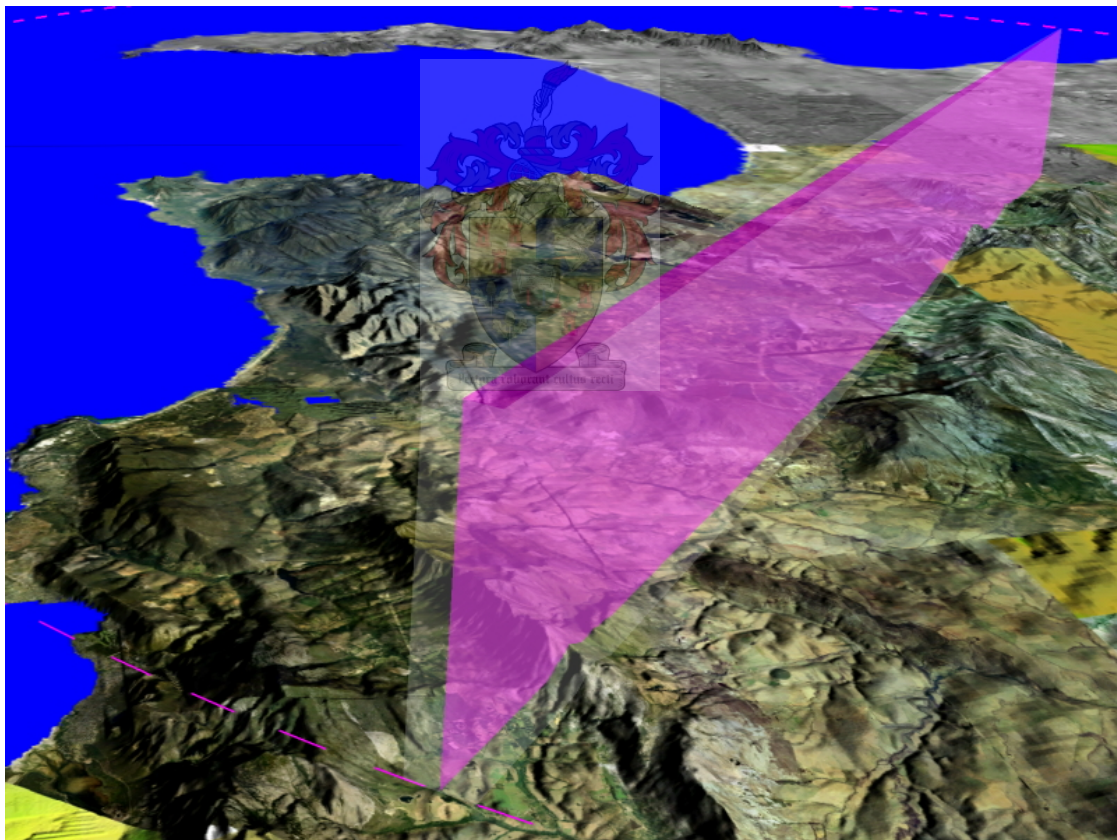


Figure 27: Radar sensor with surrounding transparent box showing measurement uncertainties.

3.3.2 Measurement volume analysis

For any radar tracking system to work effectively, it is necessary to get continued updates of a target position through new observations. It is therefore desirable to position radars in such a way that as much as possible of the radar's scan volume is visible to the radar and not occluded by terrain features. A measurement volume analysis for a sensor is done by sweeping through the entire volume at regular azimuth and elevation angle intervals. For each interval, a straight line from the sensor to the maximum range or maximum height boundary is drawn, and checked for terrain intersections. For each such line, the segment from the sensor to the first terrain intersection is marked as visible, while the remainder of the segment is marked as invisible. From this analysis, convex hulls can be formed for all the regions invisible to the radar by creating boundaries at ground level, maximum range and the last line still containing an invisible segment.

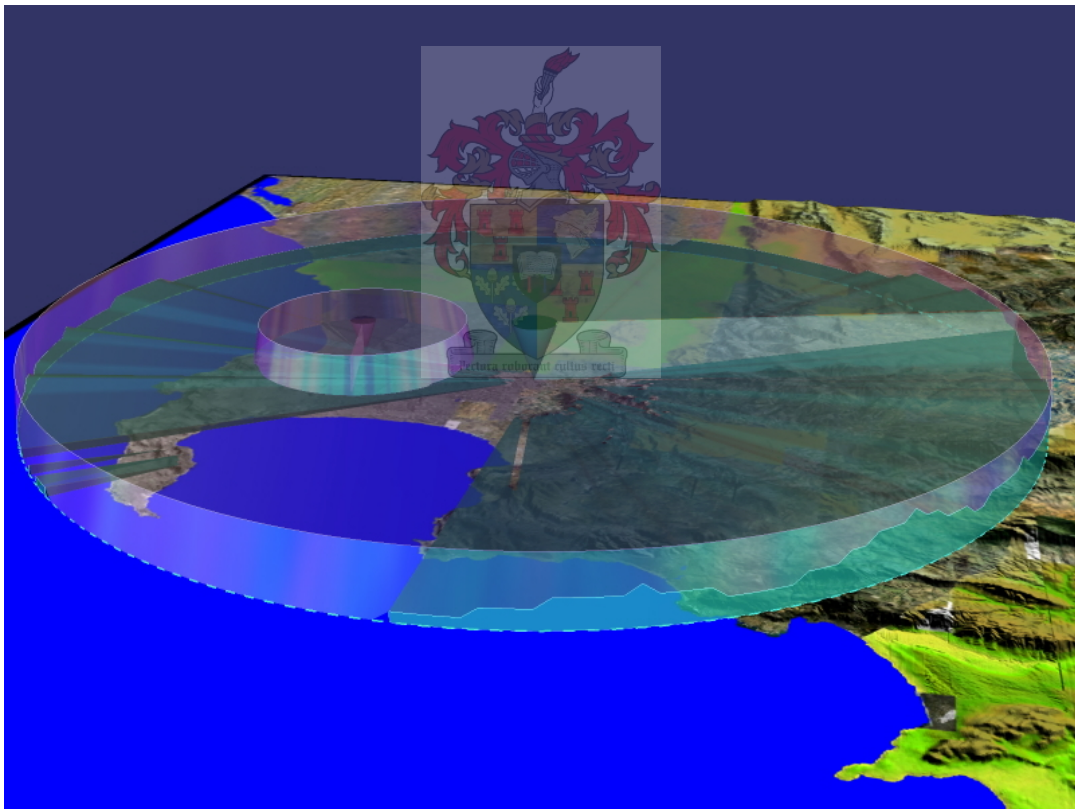


Figure 28: Measurement volume analysis for two radar sensors.

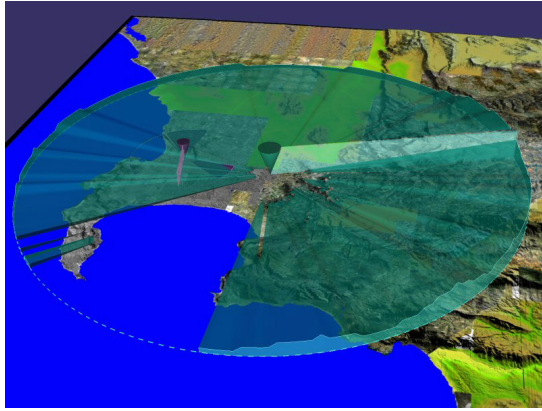


Figure 29: Invisible measurement volumes.

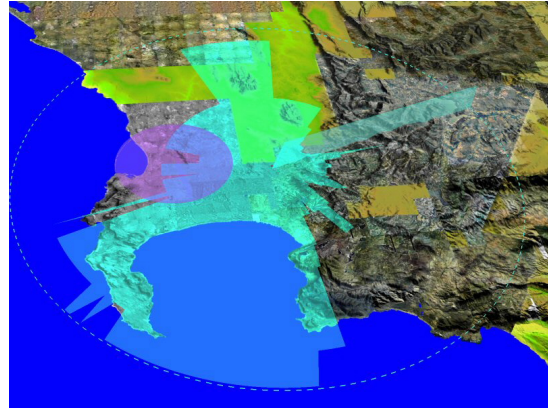


Figure 30: Visible areas at 1000 m.

3.4 Target modelling

Airplanes are modelled as point targets with their only discerning features their speed and associated flight path. Each airplane moves at an assigned constant speed, on a predetermined flight path. A flight path is built from nodes which can be a starting point node and a combination of straight section and constant g turn nodes. As a starting point, any latitude, longitude, altitude and initial heading is chosen. For a straight section, the distance traveled is chosen. For constant g turns, the final heading and altitude is chosen, as well as the load factor (gravitational acceleration scale factor). Such a constructed flight path does not represent a completely realistic model of all airplane flight and maneuvering capabilities, but gives an approximation that should suffice for most planes.

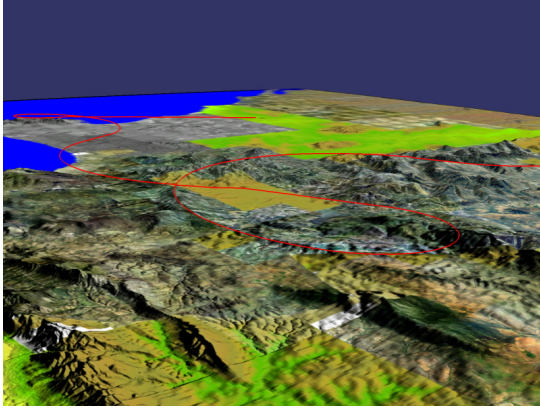


Figure 31: Flight path with many turns.

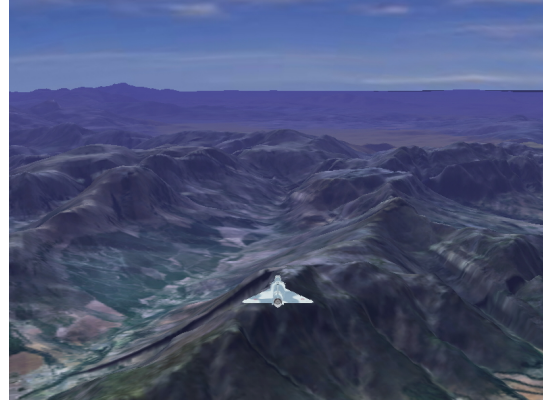


Figure 32: Low flying aircraft.

3.5 Measurement correlation

Correlation of target observations to target tracks is done as described in Chapter 2. Due to time constraints, a Kalman filter was not implemented in the simulation. Instead, estimated track positions are taken from a normal distribution with covariance $C_T = \begin{bmatrix} 10000 & 0 \\ 0 & 10000 \end{bmatrix}$ around the actual target position. Correlation is done only in two dimensions. For each observation, information about the target position, range and bearing measurements, range and bearing measurement errors, and the correlation result is displayed. The correlation result can either be a true correlation, for when an observation is correlated with the correct target, a false correlation, for when an observation is correlated with a different target than the one the radar measured, or uncorrelated, for when the observation could not be correlated with any of the known targets. The correlation score is displayed along with the result as a percentage value. Ellipses for the uncertainty in the sensor measurement, the uncertainty in the track position and the correlation window is drawn for each observation.

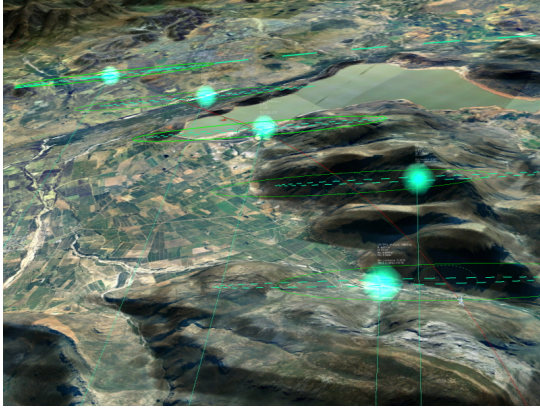


Figure 33: Radar observations with uncertainty ellipses and correlation windows shown.

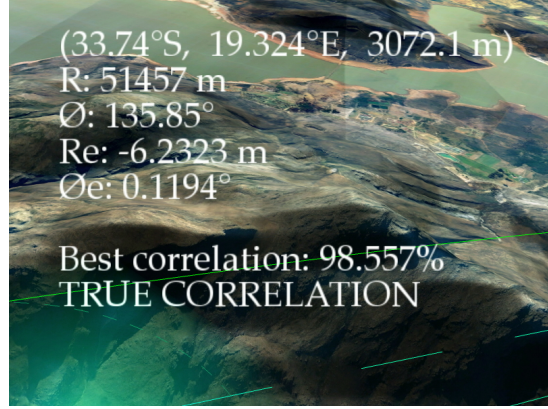


Figure 34: Measurement info displayed with each observation.

3.6 Experimental results

We test the correlation algorithm by simulating three similar airplanes flying the same flight path in an equilateral triangle formation. We let the airplanes fly in the measurement volume of a single radar sensor and determine the number of false correlations compared to the total number of measurements. We use a radar sensor with the following properties

Maximum range	65000 m
Maximum height	8000 m
Maximum angle of sight	22.3°
Range error	18 m
Bearing error	0.7°

Table 1: Radar sensor properties

We also simulate a tracker for each target, where the estimated target position is given as the actual target position with a random error ($\sigma_x^2 = 10000$, $\sigma_y^2 = 10000$) added to it. We take the correlation threshold at $T_c = 0.1$. The experiment is run for airplanes at different distances from each other.

Distance between planes	Total correlations	False correlations	Accuracy
20 m	309	161	52.1 %
125 m	312	44	85.9 %
225 m	315	17	94.6 %

Table 2: Correlation results

As one would expect, for targets very close to each other, we don't get very accurate correlations. But considering the inaccuracies in the sensor as well as the simulated tracker, targets as close as 20 m apart would be considered a single target - we just did not use an accurate enough sensor to tell them apart. We actually get very good results for airplanes a little further apart. For example, at the outer edge of the measurement volume, our correlation window (at 3 standard deviations) is an ellipse with a 600 m axis in the range dimension and a 4764 m axis in the bearing dimension. (This is calculated by taking the maximum deviation from either the sensor or the tracker at the maximum range). For such a large possible correlation volume, planes flying at 125 m apart can be considered as targets that are close together, and we get very good correlation results.



Conclusion

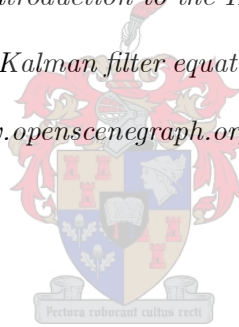
We were faced with the problem of correlating measurements with targets in such a way that we minimise the effect of the inaccuracies contained in the measurements and target trackers. We presented a novel algorithm for measurement correlation of a stationary target between multiple sensors. As a side effect of this correlation algorithm, we are able to estimate target height from two sensors measuring only range and bearing. We extended the correlation algorithm for correlation between moving targets (target tracks) and sensor measurements. A highly visual and interactive OpenGL simulation was developed by the author to test the algorithm and assist in analysing sensor placement in the surrounding terrain. Results from the simulation indicate a high success rate in true target to track correlation for targets that share the same measurement uncertainty volume.

Acknowledgements

The author would like to acknowledge Reutech Radar Systems and the NRF who funded this work. He is also grateful for insightful discussions with Prof. Ben Herbst and Mr Pieter-Jan Wolfaardt. Furthermore, he acknowledges Dr. Barry Sherlock for his work on deriving the Kalman filter equations.

Bibliography

- [1] S. Blackman, *Multiple-Target Tracking with Radar Applications*, Artech House, 1986.
- [2] Y. Bar-Shalom, *Multitarget-Multisensor Tracking: Advanced Applications*, Artech House, 1990.
- [3] Y. Bar-Shalom, X. Rong Li and T. Kirubarajan, *Estimation with applications to tracking and navigation*, Wiley Interscience NY, 2001.
- [4] B. Sherlock and B. Herbst, *Introduction to the Kalman filter and applications*, 2003
- [5] B. Sherlock, *Derivation of the Kalman filter equations*, Colloquim lecture notes, 2003.
- [6] OpenSceneGraph, <http://www.openscenegraph.org>



Appendix A – SimRadar v1.0: Simulation User Guide

Adding radars

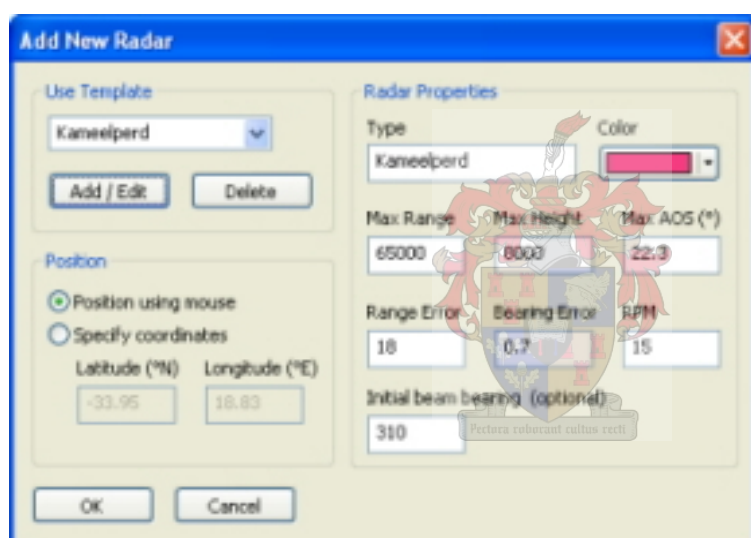


Figure 35: Add new radar dialog box.

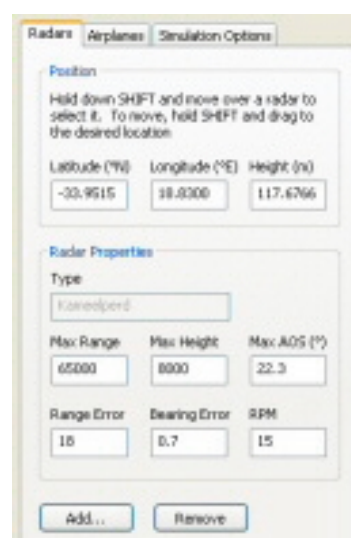


Figure 36: Radars tab.

Clicking the Add button on the Radars tab brings up the Add New Radar dialog box. Predefined sensor templates are available through the 'use template' combo box, and will load default values for all sensor properties. To change any of these properties, click the Add/Edit button. This will enable the property text boxes and allow one to change any of the values. If the sensor type is changed, a new radar template will be created which will also be available the next time a radar is added. The radar is positioned at the specified coordinates. It is also possible to change the radar position by holding down the Shift key and dragging the sensor to the desired position, or by

changing the radar coordinates on the Radars tab. Any of the sensor properties can be changed at runtime time by entering new values on the Radars tab. Different radars are selected by holding down the Shift key and moving the mouse into their scan volumes. To remove a radar, first select it holding down shift and click on the remove button.

Adding airplanes



Figure 37: Airplanes tab.

Airplanes are added by going to the Airplanes tab and clicking the Add button. Four airplane models are available in the "SimRadar/data/models/Planes" directory. These are all 3D Studio Max models, and more models can be added by downloading *.3ds models from the internet and copying them in this directory. Once a model is loaded in the simulation, set its desired speed and scale factor (the scale factor is necessary because not all models are created at the same scale, and the models need

to be rescaled to a suitable size). A flight path can be attached to the model by loading one of the predefined plight paths - click the load button and select a file from the "SimRadar/data/flight path" directory. Otherwise, build a custom flight path by selecting either the start position, straight section or constant g turn, and enter the appropriate values and click add. At any stage, the flight path can be changed by selecting the appropriate node, change its values and click apply. To hide the flight path in the simulation screen, remove the check from the "Show flight path" check box.

Changing simulation and rendering options

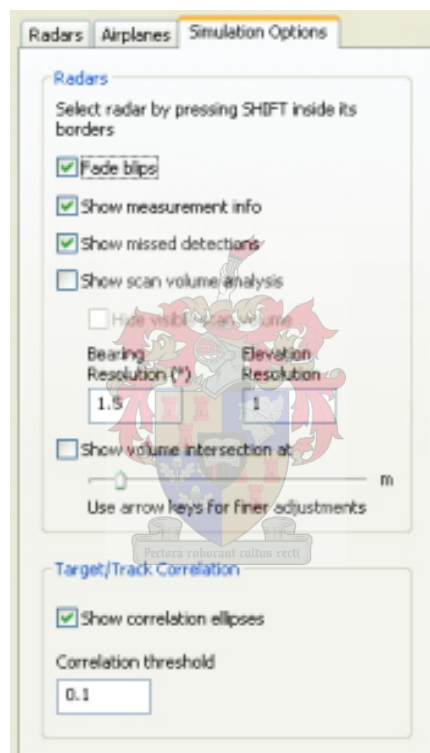


Figure 38: Simulation options

Several simulation and rendering options can be set on the Simulation Options tab. A description of each follows:

- *Fade blips* – New radar observations will be faded out after a couple of seconds
- *Show measurement info* – Info from each observation is displayed alongside the blip

- *Show missed detections* – Missed detections due to terrain occlusion will be shown in a red line from the radar to the target position
- *Show scan volume analysis* – The visible and invisible sensor scan volumes are displayed. Only the areas invisible to the radar is displayed if "*Hide visible scan volume*" is ticked. The bearing and elevation resolution can be changed in the appropriate text boxes.
- *Show volume intersection* – An intersection of the scan volume for a sensor can be displayed showing only the visible areas at a certain height. Select the height with the track bar or use the arrow keys for small changes.

For measurement correlations, the value for the correlation threshold, T_c , can be set in the provided text box. Remove the tick from the "*Show correlation ellipses*" to stop displaying the correlation window, measurement and track uncertainties.

Generating a new terrain model

A Paged Level Of Detail model for the terrain is generated by using the *osgdem* program situated in the "SimRadar/Win32" directory. This is a command line program that takes as input georeferenced height maps and imagery, combines them and splits them up into the desired number of levels. Typical usage is as follows:

```
>osgdem -d cape_elev.tif -t CapeTex -o cape.ive -l 8 --geocentric
```

where `-d` specifies the height map (this could also be a directory containing many different maps), `-t` specifies the texture imagery (or directory containing multiple images), `-o` specifies the generated output database (has to be iva format), `-l` specifies the desired number of levels and `--geocentric` specifies that it is mapped onto a spheroid, not a flat earth model. The generated terrain model can be viewed with

```
>osgviewer cape.ive
```

Appendix B – Derivation of the Kalman Filter Equations

A complete derivation of the Kalman filter equations was done by Sherlock [5]. Before we delve into those, we need to show some properties of matrices.

Inverse of a partitioned matrix.

Suppose

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix},$$

where A_{11} and A_{22} are square matrices. Then

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

$$\Rightarrow \begin{cases} A_{11}B_{11} + A_{12}B_{21} = I & (1) \\ A_{11}B_{12} + A_{12}B_{22} = 0 & (2) \\ A_{21}B_{11} + A_{22}B_{21} = 0 & (3) \\ A_{21}B_{12} + A_{22}B_{22} = I & (4) \end{cases}$$

(1) – $A_{12}A_{22}^{-1}(3)$:

$$\begin{aligned} A_{11}B_{11} + A_{12}B_{21} - A_{12}A_{22}^{-1}A_{21}B_{11} - A_{12}A_{22}^{-1}A_{22}B_{21} &= I - 0 \\ \Rightarrow (A_{11} - A_{12}A_{22}^{-1}A_{21})B_{11} &= I \\ \Rightarrow B_{11} &= (A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1} \end{aligned}$$

(4) – $A_{21}A_{11}^{-1}(2)$:

$$\begin{aligned}
& A_{21}B_{12} + A_{22}B_{22} - A_{21}A_{11}^{-1}A_{11}B_{12} - A_{21}A_{11}^{-1}A_{12}B_{22} = I - 0 \\
& \Rightarrow (A_{22} - A_{21}A_{11}^{-1}A_{12})B_{22} = I \\
& \Rightarrow B_{22} = (A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1}
\end{aligned}$$

(2) :

$$\begin{aligned}
& A_{11}B_{12} + A_{12}B_{22} = 0 \\
& \Rightarrow A_{11}B_{12} = -A_{12}B_{22} \\
& \Rightarrow B_{12} = -A_{11}^{-1}A_{12}B_{22}
\end{aligned}$$

(3) :

$$\begin{aligned}
& A_{21}B_{11} + A_{22}B_{21} = 0 \\
& \Rightarrow A_{22}B_{21} = -A_{21}B_{11} \\
& \Rightarrow B_{21} = -A_{22}^{-1}A_{21}B_{11}
\end{aligned}$$

Remark 1 By symmetry between $A = B^{-1}$ and $B = A^{-1}$, we can swap A 's and B 's in the above equations to get:

$$\begin{aligned}
& A_{11} = (B_{11} - B_{12}B_{22}^{-1}B_{21})^{-1} \\
& A_{22} = (B_{22} - B_{21}B_{11}^{-1}B_{12})^{-1} \\
& A_{12} = -B_{11}^{-1}B_{12}A_{22} \\
& A_{21} = -B_{22}^{-1}B_{21}A_{11}
\end{aligned}$$

To be shown: Given jointly Gaussian vectors x and z , then $p_{x|z}(x|z)$ is also Gaussian, and has mean $E[x | z] = E[x] + P_{xz}P_{zz}^{-1}(z - E[z])$ and covariance $P_{xx|z} = P_{xx} - P_{xz}P_{zz}^{-1}P_{zx}$.

Given vectors x and z , suppose $y = \begin{bmatrix} x \\ z \end{bmatrix}$ (stacked vector) is Gaussian.

The mean of y is

$$\bar{y} = E(y) = \begin{bmatrix} E(x) \\ E(z) \end{bmatrix} = \begin{bmatrix} \bar{x} \\ \bar{z} \end{bmatrix}.$$

The covariance of y is

$$\begin{aligned}
P_{yy} &= E[(y - \bar{y})(y - \bar{y})^T] \\
&= E \left[\left(\begin{pmatrix} x \\ z \end{pmatrix} - \begin{pmatrix} \bar{x} \\ \bar{z} \end{pmatrix} \right) \left(\begin{pmatrix} x \\ z \end{pmatrix} - \begin{pmatrix} \bar{x} \\ \bar{z} \end{pmatrix} \right)^T \right] \\
&= E \left(\begin{bmatrix} x - \bar{x} \\ z - \bar{z} \end{bmatrix} \begin{bmatrix} x - \bar{x} \\ z - \bar{z} \end{bmatrix}^T \right) \\
&= E \left(\begin{bmatrix} x - \bar{x} \\ z - \bar{z} \end{bmatrix} \begin{bmatrix} (x - \bar{x})^T & (z - \bar{z})^T \end{bmatrix} \right) \\
&= E \left(\begin{bmatrix} (x - \bar{x})(x - \bar{x})^T & (x - \bar{x})(z - \bar{z})^T \\ (z - \bar{z})(x - \bar{x})^T & (z - \bar{z})(z - \bar{z})^T \end{bmatrix} \right) \\
&= \begin{bmatrix} P_{xx} & P_{xz} \\ P_{zx} & P_{zz} \end{bmatrix}
\end{aligned}$$

which is a partitioned covariance matrix.

Any Gaussian vector has

$$p_x(x) = \frac{1}{\sqrt{\det(2\pi P_{xx})}} \exp\left(-\frac{1}{2}(x - \bar{x})^T P_{xx}^{-1}(x - \bar{x})\right)$$

(definition of multivariate Gaussian)

Now,

$$\begin{aligned}
p_{x|z}(x|z) &= \frac{p_{x,z}(x, z)}{p_z(z)} \\
&= \frac{p_y(y)}{p_z(z)} \quad \text{since } x \text{ and } z \text{ are jointly Gaussian} \\
&= \frac{\frac{1}{\sqrt{\det(2\pi P_{yy})}} \exp\left(-\frac{1}{2}(y - \bar{y})^T P_{yy}^{-1}(y - \bar{y})\right)}{\frac{1}{\sqrt{\det(2\pi P_{zz})}} \exp\left(-\frac{1}{2}(z - \bar{z})^T P_{zz}^{-1}(z - \bar{z})\right)} \\
&= \sqrt{\frac{\det(2\pi P_{zz})}{\det(2\pi P_{yy})}} \exp\left(-\frac{1}{2} \left[(y - \bar{y})^T P_{yy}^{-1}(y - \bar{y}) - (z - \bar{z})^T P_{zz}^{-1}(z - \bar{z}) \right] \right) \\
&= \sqrt{\frac{\det(2\pi P_{zz})}{\det(2\pi P_{yy})}} \exp\left(-\frac{1}{2} \left\{ \begin{bmatrix} x - \bar{x} \\ z - \bar{z} \end{bmatrix}^T \begin{bmatrix} P_{xx} & P_{xz} \\ P_{zx} & P_{zz} \end{bmatrix}^{-1} \begin{bmatrix} x - \bar{x} \\ z - \bar{z} \end{bmatrix} - (z - \bar{z})^T P_{zz}^{-1}(z - \bar{z}) \right\} \right) \\
&= \sqrt{\frac{\det(2\pi P_{zz})}{\det(2\pi P_{yy})}} \exp\left(-\frac{1}{2} \left\{ \begin{bmatrix} x - \bar{x} \\ z - \bar{z} \end{bmatrix}^T \begin{bmatrix} T_{xx} & T_{xz} \\ T_{zx} & T_{zz} \end{bmatrix} \begin{bmatrix} x - \bar{x} \\ z - \bar{z} \end{bmatrix} - (z - \bar{z})^T P_{zz}^{-1}(z - \bar{z}) \right\} \right)
\end{aligned}$$

where $T_{yy} = P_{yy}^{-1}$, inverse of partitioned matrix P_{yy} . Hence, using the “partitioned matrix inverse” shown earlier, we identify:

$$T_{xx} = (P_{xx} - P_{xz}P_{zz}^{-1}P_{xz})^{-1} \quad (*)$$

$$P_{zz} = (T_{zz} - T_{zx}T_{xx}^{-1}T_{xz})^{-1} \quad (**)$$

$$\begin{aligned} P_{xz} &= -T_{xx}^{-1}T_{xz}P_{zz} \quad (***) \\ &\Rightarrow T_{xx}^{-1}T_{xz} = -P_{xz}P_{zz}^{-1} \end{aligned}$$

Therefore, for $\xi = x - \bar{x}$, $\zeta = z - \bar{z}$

$$\begin{aligned} p_{x|z}(x|z) &= \sqrt{\frac{\det(2\pi P_{zz})}{\det(2\pi P_{yy})}} \exp\left(-\frac{1}{2} \left\{ \begin{bmatrix} \xi \\ \zeta \end{bmatrix}^T \begin{bmatrix} T_{xx} & T_{xz} \\ T_{zx} & T_{zz} \end{bmatrix} \begin{bmatrix} \xi \\ \zeta \end{bmatrix} - \zeta^T P_{zz}^{-1} \zeta \right\}\right) \\ &= \sqrt{\frac{\det(2\pi P_{zz})}{\det(2\pi P_{yy})}} \exp\left(-\frac{1}{2} \left\{ \begin{bmatrix} \xi^T & \zeta^T \end{bmatrix} \begin{bmatrix} T_{xx} & T_{xz} \\ T_{zx} & T_{zz} \end{bmatrix} \begin{bmatrix} \xi \\ \zeta \end{bmatrix} - \zeta^T P_{zz}^{-1} \zeta \right\}\right) \\ &= \sqrt{\frac{\det(2\pi P_{zz})}{\det(2\pi P_{yy})}} \exp\left(-\frac{1}{2} \left\{ \xi^T T_{xx} \xi + \xi^T T_{xz} \zeta + \zeta^T T_{zx} \xi + \zeta^T T_{zz} \zeta - \zeta^T P_{zz}^{-1} \zeta \right\}\right) \\ &= \sqrt{\frac{\det(2\pi P_{zz})}{\det(2\pi P_{yy})}} \exp\left(-\frac{1}{2} \left\{ \xi^T T_{xx} \xi + \xi^T T_{xz} \zeta + \zeta^T T_{zx} \xi + \zeta^T (T_{zz} - P_{zz}^{-1}) \zeta \right\}\right) \\ &= \sqrt{\frac{\det(2\pi P_{zz})}{\det(2\pi P_{yy})}} \exp\left(-\frac{1}{2} \left\{ \xi^T T_{xx} \xi + \xi^T T_{xz} \zeta + \zeta^T T_{zx} \xi + \zeta^T (T_{zz} - (T_{zz} - T_{zx}T_{xx}^{-1}T_{xz})) \zeta \right\}\right) \text{ by } (***) \\ &= \sqrt{\frac{\det(2\pi P_{zz})}{\det(2\pi P_{yy})}} \exp\left(-\frac{1}{2} \left\{ \xi^T T_{xx} \xi + \xi^T T_{xz} \zeta + \zeta^T T_{zx} \xi + \zeta^T (+T_{zx}T_{xx}^{-1}T_{xz}) \zeta \right\}\right) \\ &= \sqrt{\frac{\det(2\pi P_{zz})}{\det(2\pi P_{yy})}} \exp\left(-\frac{1}{2} \left\{ \xi^T T_{xx} \xi + \xi^T T_{xz} \zeta + \zeta^T T_{xz}^T T_{xx}^{-1} T_{xz} \zeta + \zeta^T T_{zx} T_{xx}^{-1} T_{xz} \zeta \right\}\right) \end{aligned}$$

since $T_{zx} = T_{xz}^T = T_{xz}^T (T_{xx}^{-1} T_{xx}) = T_{xz}^T T_{xx}^{-T} T_{xx}$.

$$p_{x|z}(x|z) = \sqrt{\frac{\det(2\pi P_{zz})}{\det(2\pi P_{yy})}} \exp\left(-\frac{1}{2} \left\{ \xi^T T_{xx} \xi + \xi^T T_{xz} \zeta + \zeta^T T_{xz}^T T_{xx}^{-T} T_{xx} \xi + \zeta^T T_{xz}^T T_{xx}^{-T} T_{xz} \zeta \right\}\right)$$

since $T_{xx}^{-T} = T_{xx}^{-1}$ and $T_{zx} = T_{xz}^T$.

$$\begin{aligned} p_{x|z}(x|z) &= \sqrt{\frac{\det(2\pi P_{zz})}{\det(2\pi P_{yy})}} \exp\left(-\frac{1}{2} \left\{ (\xi + T_{xx}^{-1} T_{xz} \zeta)^T T_{xx} \xi + (\xi + T_{xx}^{-1} T_{xz} \zeta)^T T_{xz} \zeta \right\}\right) \\ &= \sqrt{\frac{\det(2\pi P_{zz})}{\det(2\pi P_{yy})}} \exp\left(-\frac{1}{2} \left\{ (\xi + T_{xx}^{-1} T_{xz} \zeta)^T T_{xx} \xi + (\xi + T_{xx}^{-1} T_{xz} \zeta)^T T_{xx} (T_{xx}^{-1} T_{xz} \zeta) \right\}\right) \end{aligned}$$

since $T_{xx} T_{xx}^{-1} = I$.

$$p_{x|z}(x|z) = \sqrt{\frac{\det(2\pi P_{zz})}{\det(2\pi P_{yy})}} \exp\left(-\frac{1}{2} \left\{ (\xi + T_{xx}^{-1} T_{xz} \zeta)^T T_{xx} (\xi + T_{xx}^{-1} T_{xz} \zeta) \right\}\right)$$

which is a quadratic form

Now substitute back:
$$\begin{cases} \xi = x - \bar{x} \\ \zeta = z - \bar{z} \\ T_{xx} = (P_{xx} - P_{xz} P_{zz}^{-1} P_{zx})^{-1} \quad \text{i.e. (*)} \\ T_{xx}^{-1} T_{xz} = -P_{xz} P_{zz}^{-1} \quad \text{i.e. (***)} \end{cases}$$

Let $K = \sqrt{\frac{\det(2\pi P_{zz})}{\det(2\pi P_{yy})}}$, then

$$\begin{aligned} p_{x|z}(x|z) &= K \exp\left(-\frac{1}{2} \left\{ ((x - \bar{x}) - P_{xz} P_{zz}^{-1} (z - \bar{z}))^T (P_{xx} - P_{xz} P_{zz}^{-1} P_{zx})^{-1} ((x - \bar{x}) - P_{xz} P_{zz}^{-1} (z - \bar{z})) \right\}\right) \\ &= K \exp\left(-\frac{1}{2} \left\{ (x - [\bar{x} + P_{xz} P_{zz}^{-1} (z - \bar{z})])^T (P_{xx} - P_{xz} P_{zz}^{-1} P_{zx})^{-1} (x - [\bar{x} + P_{xz} P_{zz}^{-1} (z - \bar{z})]) \right\}\right) \end{aligned}$$

which is of form $(x - \text{mean})^T (\text{Covariance})^{-1} (x - \text{mean})$.

Hence the mean

$$E(x | z) = \hat{x} = \bar{x} + P_{xz} P_{zz}^{-1} (z - \bar{z})$$

And covariance

$$P_{xx|z} = P_{xx} - P_{xz} P_{zz}^{-1} P_{zx}$$

Summary statement:

Given jointly Gaussian vectors x and z , then $p_{x|z}(x|z)$ is also Gaussian, and has mean $E[x | z] = E[x] + P_{xz}P_{zz}^{-1}(z - E[z])$ and covariance $P_{xx|z} = P_{xx} - P_{xz}P_{zz}^{-1}P_{zx}$.

Covariance matrices

Innovation covariance matrix

$$\begin{aligned}
P_{zz} &= E \left[(z_k - \hat{z}_k^-) (z_k - \hat{z}_k^-)^T \right] \\
&= E \left[(z_k - H\hat{x}_k^-) (z_k - H\hat{x}_k^-)^T \right] \quad \text{since } \hat{z}_k^- = H\hat{x}_k^- \\
&= E \left[(Hx_k + v_k - H\hat{x}_k^-) (Hx_k + v_k - H\hat{x}_k^-)^T \right] \quad \text{since } z_k = Hx_k + v_k \\
&= E \left[(H(x_k - \hat{x}_k^-) + v_k) (H(x_k - \hat{x}_k^-) + v_k)^T \right] \\
&= E \left[H(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T H^T + H(x_k - \hat{x}_k^-)v_k^T + v_k(x_k - \hat{x}_k^-)^T H^T + v_k v_k^T \right] \\
&= HE \left[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T \right] H^T + HE \left[(x_k - \hat{x}_k^-)v_k^T \right] + E \left[v_k(x_k - \hat{x}_k^-)^T \right] H^T + E \left[v_k v_k^T \right] \\
&= HP_k^- H^T + H0 + 0H^T + R \\
&= HP_k^- H^T + R
\end{aligned}$$

Covariance between state and measurement

$$\begin{aligned}
P_{xz} &= E \left[(x_k - \hat{x}_k^-) (z_k - \hat{z}_k^-)^T \right] \\
&= E \left[(x_k - \hat{x}_k^-) (Hx_k + v_k - H\hat{x}_k^-)^T \right] \quad \text{since } \hat{z}_k^- = H\hat{x}_k^- \text{ and } z_k = Hx_k + v_k \\
&= E \left[(x_k - \hat{x}_k^-) (H(x_k - \hat{x}_k^-) + v_k)^T \right] \\
&= E \left[(x_k - \hat{x}_k^-) (x_k - \hat{x}_k^-)^T H^T + (x_k - \hat{x}_k^-) v_k^T \right] \\
&= P_k^- H^T
\end{aligned}$$

State covariance matrix

$$\begin{aligned}
P_{xx}(k+1) &\equiv P_{k+1}^- = E \left[(x_{k+1} - \hat{x}_{k+1}^-)(x_{k+1} - \hat{x}_{k+1}^-)^T \right] \\
&= E \left[((Ax_k + Bu_k + w_k) - (A\hat{x}_k + Bu_k)) ((Ax_k + Bu_k + w_k) - (A\hat{x}_k + Bu_k))^T \right]
\end{aligned}$$

since $x_{k+1} = Ax_k + Bu_k + w_k$ and $\hat{x}_{k+1}^- = A\hat{x}_k + Bu_k$. Therefore

$$\begin{aligned}
P_{xx}(k+1) &= E \left[(A(x_k - \hat{x}_k) + w_k)(A(x_k - \hat{x}_k) + w_k)^T \right] \\
&= E \left[A(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T A^T + A(x_k - \hat{x}_k)w_k^T + w_k(x_k - \hat{x}_k)^T A^T + w_k w_k^T \right] \\
&= AE \left[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T \right] A^T + A0 + 0A^T + E[w_k w_k^T] \\
&= AP_k^- A + Q \\
&\Rightarrow P_{k+1}^- = AP_k^- A + Q
\end{aligned}$$

Given jointly Gaussian vectors x and z , then $p_{x|z}(x | z)$ is also Gaussian, and has mean $E[x | z] = E[x] + P_{xz}P_{zz}^{-1}(z - E[z])$ and covariance $P_{xx|z} = P_{xx} - P_{xz}P_{zz}^{-1}P_{zx}$.

This immediately gives the Kalman filter:

$x \equiv x_k$ is the current state

$z \equiv z_k$ is the new output

$E[x] \equiv \hat{x}_k^-$ is the predicted estimate of the state (not given z_k)

$E[x|z] \equiv \hat{x}_k$ is the corrected estimate of the state (given z_k)

$E[z] \equiv \hat{z}_k^-$ is the estimate of the output

$P_{xx} \equiv P_k^-$ is the predicted estimate of the state covariance

$P_{xx|z} \equiv P_k$ is the corrected estimate of the state covariance

So, rewriting $E[x | z] = E[x] + P_{xz}P_{zz}^{-1}(z - E[z])$ gives:

$$\hat{x}_k = \hat{x}_k^- + P_{xz}P_{zz}^{-1}(z_k - \hat{z}_k^-)$$

i.e.

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{z}_k^-)$$

This is the corrector equation for the state. Therefore,

$$K_k \equiv P_{xz}P_{zz}^{-1} = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

is the equation for the Kalman gain.

Now the covariance:

$$\begin{aligned}
P_k &\equiv P_{xx|z} \\
&= P_{xx} - P_{xz}P_{zz}^{-1}P_{zx} \\
&= P_{xx} - P_{xz}P_{zz}^{-1}P_{xz}^T \\
&= P_k^- - (P_k^- H^T)(HP_k^- H^T + R)^{-1}(P_k^- H^T)^T \\
&= P_k^- - K_k H (P_k^-)^T
\end{aligned}$$

$$\Rightarrow P_k = (I - K_k H)P_k^-$$

This is the corrector equation for the covariance.

The covariance corrector can also be written as

$$\begin{aligned}
P_k &= (I - K_k H)P_k^- \\
&\equiv (I - K_k H)P_k^- + P_k^- H^T K_k^T - P_k^- H^T K_k^T \\
&\equiv (I - K_k H)P_k^- + P_k^- H^T (HP_k^- H^T + R)^{-1} (HP_k^- H^T + R) K_k^T - P_k^- H^T K_k^T \\
&= (I - K_k H)P_k^- + K_k (HP_k^- H^T + R) K_k^T - P_k^- H^T K_k^T \quad \text{by equation for Kalman gain} \\
&\equiv (I - K_k H)P_k^- + K_k R K_k^T + K_k H P_k^- H^T K_k^T - P_k^- H^T K_k^T \\
&\equiv K_k R K_k^T + (I - K_k H)P_k^- - (I - K_k H)P_k^- H^T K_k^T \\
&\equiv K_k R K_k^T + (I - K_k H)P_k^- (I - H^T K_k^T)
\end{aligned}$$

$$P_k = K_k R K_k^T + (I - K_k H)P_k^- (I - H K_k)^T$$