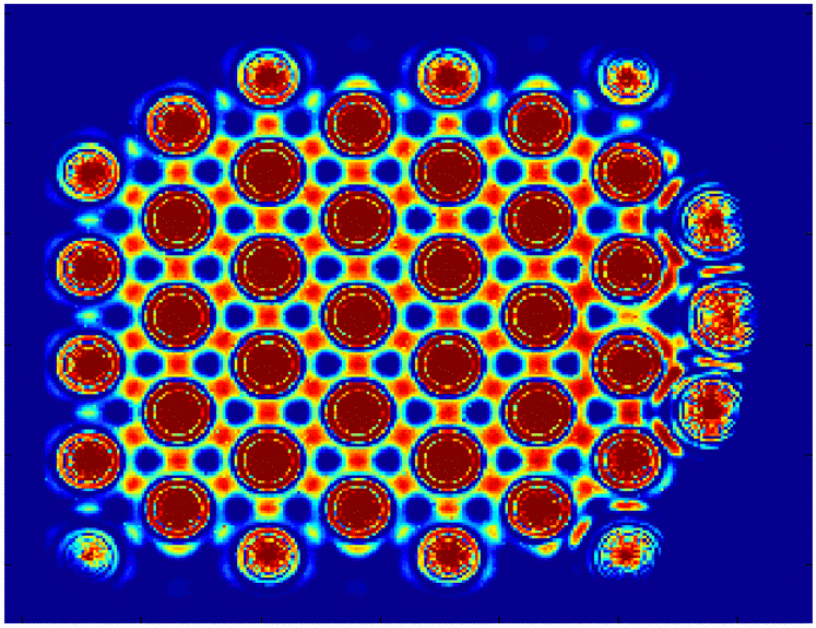
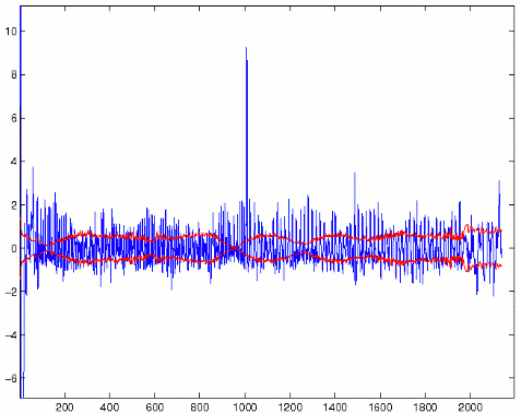
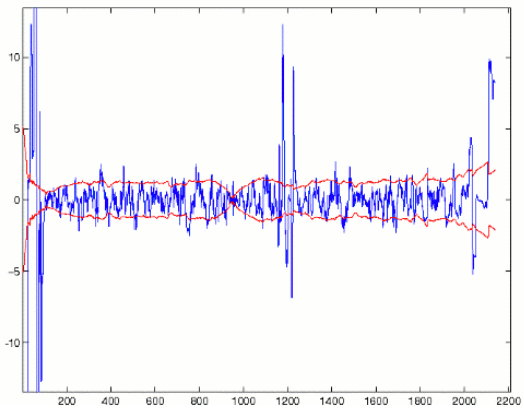
FOI-R—1400--SE
November 2004
ISSN 1650-1942
Technical report

Gustav Haapalahti, Daniel Murdin

Tracking a missile in a multistatic radar network







Sensor Technology
SE-581 11 Linköping

SWEDISH DEFENCE RESEARCH AGENCY

Sensor Technology
P.O. Box 1165
SE-581 11 Linköping

FOI-R—1400--SE

November 2004

ISSN 1650-1942

Technical report

Gustav Haapalahti, Daniel Murdin

Tracking a missile in a multistatic radar network

Issuing organization FOI – Swedish Defence Research Agency Sensor Technology P.O. Box 1165 SE-581 11 Linköping	**Report number, ISRN** FOI-R—1400--SE	**Report type** Technical report
	Research area code 4. C4ISTAR	
	Month year November 2004	**Project no.** E3048
	Sub area code 42 Above water surveillance, target acquisition and reconnaissance	
	Sub area code 2	
Author/s (editor/s) Gustav Haapalahti Daniel Murdin	**Project manager** Mats Pettersson	
	Approved by Björn Larsson	
	Sponsoring agency Armed forces	
	Scientifically and technically responsible Mats Pettersson	

Report title

Tracking a missile in a multistatic radar network

Abstract (not more than 200 words)

The performance that can be achieved when tracking a missile that flies through a radar network is investigated using simulation models and filters. A network of radar stations that can measure monostatic and bistatic range and Doppler is used and the data from each station pair is fed into a Kalman filter. The performance of the tracking is evaluated and compared with the results using only momentaneous measurements.

The comparison between momentaneous measurements and tracking shows that tracking significantly increases accuracy for position and velocity estimates. With low flying targets we find that a regular Kalman filter jumps between ambiguous solutions above and below ground, but with a few restrictions that problem is eliminated.

Keywords

radar cross-section network aasr tracking

Further bibliographic information	**Language** English
ISSN 1650-1942	**Pages** 25 p.
	Price acc. to pricelist

Utgivare	Rapportnummer, ISRN	Klassificering
Totalförsvarets Forskningsinstitut – FOI	FOI-R—1400--SE	Teknisk rapport
Sensorteknik	**Forskningsområde**	
Box 1165	4. Ledning, informationsteknink och sensorer	
581 11 Linköping	**Månad, år**	**Projektnummer**
	November 2004	E3048
	Delområde	
	42 Spaningssensorer	
	Delområde 2	

Författare/redaktör	Projektledare
Gustav Haapalahti	Mats Pettersson
Daniel Murdin	**Godkänd av**
	Björn Larsson
	Uppdragsgivare/kundbeteckning
	Försvarsmakten
	Tekniskt och/eller vetenskapligt ansvarig
	Mats Pettersson

Rapportens titel (i översättning)

Följning av en missil i ett multistatiskt radarnätverk

Sammanfattning (högst 200 ord)

Prestandan som kan uppnås med målföljning på en missil som flyger genom ett radarnätverk undersöks genom att kombinera olika simuleringsmodeller och filter. Ett nätverk av radarstationer som kan mäta monostatiskt och bistatiskt avstånd samt Doppler används och data från varje stationspar matas in i ett Kalman-filter. Prestanda för målföljningsfiltret analyseras och jämförs med resultat som bara använder momentana mätningar.

Jämförelsen mellan momentana mätningar och målföljning visar att målföljning förbättrar noggrannheten i positions- och hastighets-skattningar betydligt. Med lågtflygande mål kan vi konstatera att ett vanligt Kalman-filter hoppar mellan tvetydiga lösningar ovan och under jord, men med några restriktioner elimineras det problemet.

Nyckelord

radar målarea nätverk aasr målföljning

Övriga bibliografiska uppgifter	**Språk** Engelska
ISSN 1650-1942	**Antal sidor:** 25 s.
Distribution enligt missiv	**Pris: Enligt prislista**

Contents

1 Introduction

The idea of guiding a missile, whose job is to destroy a cruise missile, with a ground based radar network is an idea that is interesting because the missile can get accurate positions from the network and know exactly how the cruise missile moves without activating its own radar. Only when the missile gets very close to the cruise missile does it need to turn on its radar to get precision measurements. The radar on the missile can be very simple because it is only needed in the terminal phase of its attack. Also, because the missile is quiet until the last seconds it will not be detected and can do its job much more effectively. Reacquisition of the target is also possible with aid from the radar network.

In this report we will assume a network with radar stations that have omnidirectional antennas in azimuth. Both monostatic and bistatic measurements are used. The available data thus consists of mono- and bistatic range and Doppler measurements. Calculations of the real radar response from a cruise missile, bistatic and monostatic RCS, will be used. We will in this report discuss how the the mesurements can be used for tracking and show the performance that the radar network can achieve when tracking the cruise missile.

2 Momentaneous Estimates

To begin with we shall consider the more basic problem of calculating the position and velocities of a target using only range and Doppler measurements at a specific point in time. Later on the extension to filtering over time will be added.

2.1 Linearized solution

One way to find solutions is to linearize about some nominal solution and then solve the linearized problem around this point. Further improvements can be achieved by iteration, i.e. linearizing around new solution.

Let the targets "true" state be $\boldsymbol{z}$ and the station positions $\boldsymbol{s}_j$. The radar measurements are written $d_i = f_i(\boldsymbol{z}, \boldsymbol{s}) + v_i$, where v_i is measurement error (noise).

Linearize about $\boldsymbol{z}_0$, where $\boldsymbol{z} = \boldsymbol{z}_0 + \boldsymbol{x}$. Using Taylor expansion:

$$f_i(\boldsymbol{z}) = f_i(\boldsymbol{z}_0 + \boldsymbol{x}) \approx f_i(\boldsymbol{z}_0) + f_i'(\boldsymbol{z}_0) \cdot \boldsymbol{x} \tag{1}$$

Defining $y_i = d_i - f_i(\boldsymbol{z}_0)$, and using (1) we get

$$y_i = f_i'(\boldsymbol{z}_0) \cdot \boldsymbol{x} + v_i \tag{2}$$

We can put all these equations into a matrix equation as:

$$\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{v} \tag{3}$$

The least squares solution to this system of equations is:

$$\hat{\boldsymbol{x}} = (A^*A)^{-1}A^*\boldsymbol{y} \tag{4}$$

In order to examine the error in this estimate we proceed by inserting (3)

$$\hat{\boldsymbol{x}} = \boldsymbol{x} + (A^*A)^{-1}A^*\boldsymbol{v} \tag{5}$$

The error in $\hat{\boldsymbol{x}}$, $\tilde{\boldsymbol{x}}$, is thus

$$\tilde{\boldsymbol{x}} = \hat{\boldsymbol{x}} - \boldsymbol{x} = (A^*A)^{-1}A^*\boldsymbol{v} \tag{6}$$

Assume uncorrelated, equally distributed measurement errors, i.e. $E[\boldsymbol{v}\boldsymbol{v}^*] = \sigma^2 I$, where I is the identity matrix. This also implies that the solution is an MMSE (Minimum Mean Square Error) estimate. The variance of the error is then

$$E[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^*] = (A^*A)^{-1}A^*E[\boldsymbol{v}\boldsymbol{v}^*]A(A^*A)^{-1} = \sigma^2(A^*A)^{-1} \tag{7}$$

The matrix $(A^*A)^{-1}$, is the "dilution of precision matrix", that only depends on the geometry of the stations and the target.

For the case of non-equally distributed or correlated errors the MMSE solution is

$$\hat{\boldsymbol{x}} = (A^*R^{-1}A)^{-1}A^*R^{-1}\boldsymbol{y} \tag{8}$$

And in this case the variance of the error is

$$E[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^*] = (A^*R^{-1}A)^{-1} \tag{9}$$

This cannot be factored into a measurement noise and geometry dependent part.

2.1.1 Range measurements

For range measurements we have as state the position of the target $\boldsymbol{p}$.

The monostatic measurement is:

$$f_i(\boldsymbol{p}) = ||\boldsymbol{p} - \boldsymbol{s}_i|| \tag{10}$$

and

$$f_i'(\boldsymbol{p}) = \nabla_p f_i(\boldsymbol{p}) = \frac{\boldsymbol{p} - \boldsymbol{s}_i}{||\boldsymbol{p} - \boldsymbol{s}_i||} \tag{11}$$

For bistatic measurements

$$f_i(\boldsymbol{p}) = \frac{1}{2}(||\boldsymbol{p} - \boldsymbol{s}_{i_1}|| + ||\boldsymbol{p} - \boldsymbol{s}_{i_2}||) \tag{12}$$

and

$$f_i'(\boldsymbol{p}) = \frac{1}{2}(\frac{\boldsymbol{p} - \boldsymbol{s}_{i_1}}{||\boldsymbol{p} - \boldsymbol{s}_{i_1}||} + \frac{\boldsymbol{p} - \boldsymbol{s}_{i_2}}{||\boldsymbol{p} - \boldsymbol{s}_{i_2}||}) \tag{13}$$

2.1.2 Doppler measurements

When we have velocity measurements we expand the state $\boldsymbol{z}$ to include position and velocity, $\boldsymbol{z} = \begin{bmatrix}\boldsymbol{p}^* & \boldsymbol{v}^*\end{bmatrix}^*$.

For monostatic Doppler velocity measurements we have:

$$f_i(\boldsymbol{p}, \boldsymbol{v}) = \frac{\boldsymbol{v} \cdot (\boldsymbol{p} - \boldsymbol{s}_i)}{||\boldsymbol{p} - \boldsymbol{s}_i||} = \boldsymbol{v} \cdot \bar{\boldsymbol{p}}_i \tag{14}$$

and

$$\frac{\partial f_i(\boldsymbol{p}, \boldsymbol{v})}{\partial \boldsymbol{p}} = \frac{\boldsymbol{v}}{||\boldsymbol{p}_i||} - \frac{\bar{\boldsymbol{p}}_i \boldsymbol{v}^* \bar{\boldsymbol{p}}_i}{||\boldsymbol{p}_i||} \tag{15}$$

$$\frac{\partial f_i(\boldsymbol{p}, \boldsymbol{v})}{\partial \boldsymbol{v}} = \bar{\boldsymbol{p}}_i \tag{16}$$

where

$$\bar{\boldsymbol{p}}_i = \frac{\boldsymbol{p}_i}{||\boldsymbol{p}_i||} \tag{17}$$

$$\boldsymbol{p}_i = \boldsymbol{p} - \boldsymbol{s}_i \tag{18}$$

And analogously for bistatic measurements.

2.1.3 Linearization errors

If we include the error term of the Taylor expansion in (1) we get

$$f_i(\boldsymbol{z}) = f_i(\boldsymbol{z}_0 + \boldsymbol{x}) = f_i(\boldsymbol{z}_0) + f_i'(\boldsymbol{z}_0) \cdot \boldsymbol{x} + \boldsymbol{x}^* \frac{f_i''(\boldsymbol{\xi})}{2} \boldsymbol{x} \tag{19}$$

where $\xi \in [\boldsymbol{z}_0, \boldsymbol{z}_0 + \boldsymbol{x}]$

We look at size of the nonlinear error term

$$\|\boldsymbol{x}^* \frac{f_i''(\boldsymbol{\xi})}{2} \boldsymbol{x}\| \leq \|\boldsymbol{x}\|^2 \frac{\|f_i''(\boldsymbol{\xi})\|}{2} \approx \|\boldsymbol{x}\|^2 \frac{\|f_i''(\boldsymbol{z_0})\|}{2} \tag{20}$$

We here have an expression approximating the linearization error. We will next evaluate this for range and Doppler measurements.

Range measurements For monostatic measurements

$$\|f_i''(\boldsymbol{p})\| = \frac{I}{\|\boldsymbol{p}_i\|} - \frac{\bar{\boldsymbol{p}}_i \bar{\boldsymbol{p}}_i^*}{\|\boldsymbol{p}_i\|} \tag{21}$$

Let T be an orthogonal transformation T with first row $\bar{\boldsymbol{p}}_i$. We now get

$$\begin{aligned} \|f_i''(\boldsymbol{p})\| = \frac{1}{\|\boldsymbol{p}_i\|} \|TT^* - T \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^* \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} T^*\| \leq \\ \frac{1}{\|\boldsymbol{p}_i\|} \|T\| \|I - \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^* \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\| \|T^*\| = \frac{1}{\|\boldsymbol{p}_i\|} \end{aligned} \tag{22}$$

The linearization error is thus

$$e \lessapprox \frac{\|\boldsymbol{x}\|^2}{2\|\boldsymbol{p}_i\|} \tag{23}$$

If the error in linearization point is limited via:

$$\|\boldsymbol{x}\|^2 \leq 2\|\boldsymbol{p}_i\|\sigma \tag{24}$$

The linearization error is limited as:

$$e \lessapprox \sigma_i \tag{25}$$

And the linearization error is then smaller than the expected error due to noise. This can be used in solving the linearized system. If the estimated error in linearizing point, $\hat{\boldsymbol{x}}$, is larger by a predetermined allowable amount than specified in (24), linearize about the new estimate and recalculate estimate.

Doppler measurements For monostatic Doppler measurements one can show that the linearization error is limited by

$$e \lessapprox \frac{3\|\boldsymbol{x_p}\|^2}{2\|\boldsymbol{p}_i^2\|} + \frac{\|\boldsymbol{x_p}\| \|\boldsymbol{x_v}\|}{\|\boldsymbol{p}_i\|} \tag{26}$$

2.2 Non-linear optimization

Besides the linearization method discussed above a MMSE solution can be sought of the full non-linear problem. The problem may be solved with linear(izing) methods, but the problem statement is the full non-linear one. The problem can be formulated as

$$\min_{\boldsymbol{p}} \sum (d_i - \|\boldsymbol{p} - \boldsymbol{s}_j\|)^2 \tag{27}$$

The gradient, which is useful in many optimization methods, is

$$2\sum(1-\frac{d_j}{\|\boldsymbol{p}-\boldsymbol{s}_j\|})(\boldsymbol{p}-\boldsymbol{s}_j) \tag{28}$$

3 Kalman Filtering

To improve accuracy, measurements can be processed over time. In order to do this one must also have a model of the target movements. One can then e.g. use a Kalman filter to process the sequence of measurements. Besides improved accuracy, we can also easily handle cases where the available measurements at a certain time are not sufficient to determine the six-dimensional state or when measurements do not coincide exactly in time. The Kalman filter will automatically only adjust its estimate in the measured subspace. Only if there are no measurements in this subspace over a prolonged time, will there be problems in determining the state.

The Kalman filter assumes a linear model, and also assumes a white noise (or white noise that passes through a known filter) as measurement and process noise. When tracking an aircraft the accelerations enter as noise in the model. Accelerations are however more accurately described as "unknown input", and cannot normally be seen as white noise. There are many approaches to handle this, different "whitening" techniques, multiple Kalman filters, adaptive methods (e.g. change detection) to retune noise parameters etc. Many different models for aircraft motion can also be used. We do not intend to cover these here, merely to give a hint towards methods for processing measurements over time.

3.1 State model

A Kalman filter assumes the state space model

$$\boldsymbol{x}_{i+1} = F_i\boldsymbol{x}_i + G_i\boldsymbol{u}_i, \tag{29}$$
$$\boldsymbol{y}_i = H_i\boldsymbol{x}_i + \boldsymbol{v}_i, \tag{30}$$

$$E\begin{bmatrix}\boldsymbol{u}_i\\ \boldsymbol{v}_i\\ \boldsymbol{x}_0\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_j^* & \boldsymbol{v}_j^* & \boldsymbol{x}_0^* & 1\end{bmatrix} = \begin{bmatrix}\begin{bmatrix}Q_i & S_i\\ S_i^* & R_i\end{bmatrix}\delta_{ij} & \begin{matrix}0 & 0\\ 0 & 0\end{matrix}\\ \begin{matrix}0 & 0\end{matrix} & \begin{matrix}\Pi_0 & \xi_0\end{matrix}\end{bmatrix}. \tag{31}$$

Where $\boldsymbol{x}$, F is the state transition matrix, $\boldsymbol{u}$ is "process" noise, G defines which states are affected by noise and by how much, $\boldsymbol{y}$ is the measurement vector, H the measurement matrix and $\boldsymbol{v}$ measurement noise. A subindex denotes the time. The two noise sources u_i and v_i are both assumed to be uncorrelated at different times, they are also normally assumed mutually uncorrelated (i.e. $S_i = 0$). Both these cases can however be handled by defining a new (larger) state resulting in a model with these assumptions fulfilled (see e.g. [9]).

We now introduce some notation. An estimate is denoted with a superscript $\hat{}$, e.g. an estimate of the state is denoted $\hat{\boldsymbol{x}}$. An estimate given $\boldsymbol{y}_0, \boldsymbol{y}_1 \ldots, \boldsymbol{y}_j$ is denoted $\hat{\boldsymbol{x}}_{i|j}$. The simplified notation $\hat{\boldsymbol{x}}_i\check{s} = \hat{\boldsymbol{x}}_{i|i-1}$ is also used. The estimation error is denoted with a superscript $\tilde{}$, e.g. $\tilde{\boldsymbol{x}}_{i|j} = \boldsymbol{x}_{i|j} - \hat{\boldsymbol{x}}_{i|j}$. The estimate error covariance is $P_{i|j} = E[\tilde{\boldsymbol{x}}_{i|j}\tilde{\boldsymbol{x}}_{i|j}^*]$ We define the term "Innovations" for $\boldsymbol{e}_i = \boldsymbol{y}_i - H_i\hat{\boldsymbol{x}}_i$. The innovations are the difference between what we expect to measure and what we actually measure. These tells us the "new" part of the measurements and enters the Kalman Filter. The innovations are also useful for other purposes. If our model is correct the innovations have Gaussian distribution with mean zero and known variance (calculated as part of the Kalman filter mesurement update). Continously monitoring of the innovations can therefore be used to discover when the model becomes invalid, if we e.g. are assuming

the target is flying along a straight track, we can detect when it starts manouvering and change our state model accordingly.

3.2 Kalman filter recursion

The equations for updating the time of The Kalman filter estimate are

$$\hat{\boldsymbol{x}}_{i+1} = F_i \hat{\boldsymbol{x}}_{i|i}, \tag{32}$$
$$P_{i+1} = F_i P_{i|i} F_i^* + G_i Q_i G_i^*. \tag{33}$$

and the equations for incorporating new measurements into our estimate are

$$\hat{\boldsymbol{x}}_{i|i} = \hat{\boldsymbol{x}}_i + K_{f,i} \boldsymbol{e}_i, \tag{34}$$
$$P_{i|i} = P_i - K_{f,i} R_{e,i} K_{f,i}^*, \tag{35}$$
$$R_{e,i} = R_i + H_i P_i H_i^*, \tag{36}$$
$$K_{f,i} = P_i H_i^* R_{e,i}^{-1}. \tag{37}$$

Several other variants also exist.

3.3 Linearized Kalman filters

We have a non-linear system so some adaptations must be done. We will assume the following modification of (29–30) to a non-linear state-space model

$$\boldsymbol{x}_{i+1} = f_i(\boldsymbol{x}_i) + G_i \boldsymbol{u}_i, \tag{38}$$
$$\boldsymbol{y}_i = h_i(\boldsymbol{x}_i) + \boldsymbol{v}_i, \tag{39}$$

As can be seen, the noise is assumed to still enter in a linear fashion.

3.3.1 Time update

In the time update we linearize about the latest state estimate i.e. we rewrite (38) as

$$\boldsymbol{x}_{i+1} \approx f_i(\hat{\boldsymbol{x}}_{i|i}) + f_i'(\hat{\boldsymbol{x}}_{i|i})(\boldsymbol{x}_i - \hat{\boldsymbol{x}}_{i|i}) + G_i \boldsymbol{u}_i \tag{40}$$

The Kalman filter time update equations (32) then become

$$\hat{\boldsymbol{x}}_{i+1} = E[\boldsymbol{x}_{i+1}] = f_i(\hat{\boldsymbol{x}}_{i|i}), \tag{41}$$
$$P_{i+1} = E[\tilde{\boldsymbol{x}}_{i+1} \tilde{\boldsymbol{x}}_{i+1}^*] = F_i P_{i|i} F_i^* + G_i Q_i G_i^* \tag{42}$$

where

$$F_i = f_i'(\hat{\boldsymbol{x}}_{i|i}). \tag{43}$$

3.3.2 Measurement update

In the measurement update we linearize about a point $\boldsymbol{\chi}_i$. i.e. we rewrite (39) as

$$\boldsymbol{y}_i = h_i(\boldsymbol{\chi}_i) + h_i'(\boldsymbol{\chi}_i)(\boldsymbol{x}_i - \boldsymbol{\chi}_i) + \boldsymbol{v}_i, \tag{44}$$

The innovations then become

$$\boldsymbol{e}_i = \boldsymbol{y}_i - \hat{\boldsymbol{y}}_i = \boldsymbol{y}_i - h_i(\boldsymbol{\chi}_i) - h_i'(\boldsymbol{\chi}_i)(\hat{\boldsymbol{x}}_i - \boldsymbol{\chi}_i)$$

Defining

$$H_i = h_i'(\boldsymbol{\chi}_i). \tag{45}$$

the measurement update equations (34–37) are unchanged.

Using the obvious choice of $\boldsymbol{\chi}_i$ as $\boldsymbol{\chi}_i = \hat{\boldsymbol{x}}_i$, one gets the innovations

$$\boldsymbol{e}_i = \boldsymbol{y}_i - h_i(\hat{\boldsymbol{x}}_i) \tag{46}$$

This is the so called EKF (Extended Kalman Filter).

Another choice is to let $\boldsymbol{\chi}_{i,0} = \hat{\boldsymbol{x}}_i$, calculate the state estimate $\hat{\boldsymbol{x}}_{i,0|i}$, and then choose this as a new linearization point. Iteratively one can define a new linearization point as $\boldsymbol{\chi}_{i,k+1} = \hat{\boldsymbol{x}}_{i,k|i}$, where $\hat{\boldsymbol{x}}_{i,k|i}$ is the solution using $\boldsymbol{\chi}_{i,k}$ as a linearization point. This is the IEKF (Iterated Extended Kalman Filter). As stopping point for iteration one can e.g. use a criterion like (24).

3.4 Discretized Kalman filters

Both the state and measurements can be continuous or discrete. Discretization of continuous parts is therefore often important in implementing Kalman filters. In our case the state is continuous, while measurements are discrete, so the time update step needs to be discretized. We will not discuss this further, other than stating the discretized models used.

3.5 Numerical issues

There are some numerical problems with the Kalman filter recursions shown in (32–37). Since some of the matrices are covariance matrices, they are symmetric and positive definite. If this property is lost due to numerical errors the filters can diverge. Symmetry can of course be maintained by only using one half of the matrix or by symmetrizing a matrix via $P_s = 1/2(P + P^*)$. A better solution is to use so-called array algorithms, where one propagates square roots of matrices. To obtain the positive definite symmetric matrix, one can then square the propagated matrix. Other benefits of square root filters include, better conditioning, reduced dynamic range and easier parallel and fast (Chandrasekhar) implementation. See [9] for more on square-root filters.

4 Example study

The goal that we want to achieve with this simulation is to produce monostatic and bistatic range/Doppler radar data from a station network and then feed it into a Kalman filter to track a target. We want to find out the performance that we can expect from a radar system of AASR type in the case of following a missile. Earlier studies has been made on this subject, see [7].

A bistatic radar network of AASR type, see [2, 3], is used in the simulations. This type of network has stations with both a transmitter and a receiver in each station. Every station can receive radar data from its own transmitter as well as all transmitters in the other stations. For this type of network we have a problem to solve. Since we do not know where the targets are and we only measure radial and ellipsoidal distances and speeds we need to find out which measurements that come from the same targets. This is called the association problem. Some patented methods that solve the association problem are described in the papers [4, 5, 6]. In the simulations that we describe in this paper we assume that the association problem has already been solved for all measurements.

4.1 Target route calculations

The target route is described by a series of break points with a position vector and a speed at every breakpoint. The position vectors describing the route are $\boldsymbol{r}_i$ and the speeds are s_i for $i = 1, ..., k$ where k is the number of break points. We define the time $t_1 = 0$ and calculate t_p for $p = 2, ..., k$ in the following manner

$$t_p = \sum_{i=2}^{p} \frac{|\boldsymbol{r}_i - \boldsymbol{r}_{i-1}|}{s_{i-1}} \tag{47}$$

The velocity at time t is calculated as follows. Find the range $[t_l, t_{l+1}]$ that t is in and then calculate the velocity $\boldsymbol{v}$.

$$\boldsymbol{v}_l = \frac{\boldsymbol{r}_{l+1} - \boldsymbol{r}_l}{|\boldsymbol{r}_{l+1} - \boldsymbol{r}_l|} s_l \tag{48}$$

The position that we get at time t is calculated as follows. First find the range $[t_l, t_{l+1}]$ that t is in and then calculate the position vector $\boldsymbol{r}$.

$$\boldsymbol{r} = \boldsymbol{r}_l + (t - t_l)\, \boldsymbol{v}_l \tag{49}$$

4.2 Radar simulation

In the simulation that we perform we have two time parameters. One is the update time for the whole station network, that is how often the whole state of the network is updated. The other time parameter is the integration time for each station. We assume that coherent integration can be achieved for the specified integration time.

For every station pair we calculate a signal to noise ratio that takes into account the noise temperature, the radar equation with antenna diagrams and target RCS values from a database. The database contains simulated values of the RCS for varying polarizations and for varying angles of incoming and return signal in the horizontal plane for a cruise missile, see [7]. To cover different and non-horizontal angles interpolation between angles and polarizations is done with an assumed circular symmetry on the cruise missile. This symmetry assumption also means we can ignore the attitude of the missile as it has no effect. The incoming wave, return wave and missile symmetry axis directions must however be close to coplanar. All the atmosphere loss and other propagation effects that the EREPS software, see [1], calculates are also used. The propagation model in EREPS switches between different calculation modes depending on the circumstances. It chooses the one that is best suited for the parameters that it is fed with. At medium distances an optical model is used. It takes into account the interference between the direct wave and the wave that is reflected on the ground/sea.

The accuracy for each station pair is also calculated. The accuracy is defined as the standard deviation of a measurement that goes from a transmitter to the target and then to the receiver. A monostatic range measurement for example is defined as twice the distance from the station to the target. The formulas 50 and 51 are used to calculate measurement accuracies.

$$r_{acc} = \frac{c}{B\sqrt{2\,(S/N)}} \tag{50}$$

$$d_{acc} = \frac{\lambda}{t_i\sqrt{2\,(S/N)}} \tag{51}$$

The symbols used in formulas 50 and 51 are r_{acc} that is the accuracy for a range measurement, c is the speed of light, B is the bandwidth, d_{acc} is the accuracy for a Doppler measurement, t_i is the integration time, S/N is the signal to noise ratio and λ is the wavelength.

The probability of detection is also calculated for each station pair. The type of Swerling case, Swerling case 1 in this simulation, and the signal to noise ratio is taken into consideration when it is calculated. It is assumed that Gaussian noise is present.

When the probability of detection (PD), signal to noise ratio and accuracy for every station pair has been calculated we start a new part of the simulation. We cast a random number between 0 and 1 for every station pair and check if it is below the calculated PD value. If it is, we say that the station pair sees the target and we save the information for later use.

When all station pairs have been processed we have a simulated state that tells us which station pairs that see the target and also the expected accuracy for all those measurements. With all the monostatic and bistatic ranges and Doppler measurements we can calculate a target position and velocity in a 3D Cartesian coordinate system together with an error covariance matrix (visualizable in part as error ellipsoids) as described in 2.1.

4.3 Statistical simulation

Statistical simulation is done to get a picture of how the combined PD and the average combined error covariance matrix is in an area. By casting random numbers for all measurements that we can get in a point and then calculate the error covariance matrix, as before, we create a single simulated state. We repeat the creation of the simulated state a number of times to get statistics on all the variables.

For every simulated state we check if we have enough measurements to calculate a position. If we do not have enough measurements we assume that we cannot detect the target. If we have enough measurements we calculate the covariance matrix for the state. Assume that we do r number of repetitions and we find that a target can be calculated t number of times. The calculated covariance matrices are C_i, $i = 1, ..., t$ and we want to calculate a total covariance matrix C and a combined probability of detection P_d. Equations (52-53) shows how to calculate P_d and C.

$$P_d = \frac{t}{r} \tag{52}$$

$$C = \frac{1}{t}\sum_{i=1}^{t} C_i \tag{53}$$

To present the covariance matrix as a picture for a number of simulated points we have to decide what to show. In figures 2-7 we show the horizontal and vertical position errors. For more information about the figures, see section 4.7. The vertical error can be read directly from the covariance matrix. The horizontal position error value is the maximum eigenvalue of the submatrix consisting of the two horizontal components from the covariance matrix. The covariance matrix can be divided up into some submatrices

$$C = \begin{bmatrix} H_p & \cdots & \cdots & \cdots \\ \cdots & v_p & \cdots & \cdots \\ \cdots & \cdots & H_v & \cdots \\ \cdots & \cdots & \cdots & v_v \end{bmatrix} \tag{54}$$

where H_p is the 2×2 submatrix containing the horizontal position accuracy information, the scalar v_p is the vertical position accuracy component, H_d is the 2×2 submatrix containing the horizontal velocity accuracy information.and the scalar v_v is the vertical velocity accuracy component.

The values that are used to describe the horizontal errors are

$$a_{p,h} = \sqrt{\lambda_{\max}(H_p)} \tag{55}$$

and

$$a_{v,h} = \sqrt{\lambda_{\max}(H_v)} \tag{56}$$

where $a_{p,h}$ is the accuracy for the position in the horizontal plane, $a_{v,h}$ is the accuracy for the velocity in the horizontal plane and $\lambda_{\max}(A)$ represents the maximum eigenvalue of A. Considering that we use the maximum eigenvalue to represent the horizontal error we get the error in the horizontal direction with largest error. The vertical errors are simply

$$a_{p,v} = \sqrt{v_p} \tag{57}$$

and

$$a_{v,v} = \sqrt{v_v} \tag{58}$$

where $a_{p,v}$ is the accuracy for the position along the vertical axis and $a_{v,v}$ is the accuracy for the velocity along the vertical axis.

4.4 Target model

Sections 2.1.3 and 2.1.2 describe the linearization of measurement and the measurement model used. We will model the state update equation (29) with targets as flying in straight line or alternatively with constant turning rate. The time interval over which we do our time update will be denoted T.

For straight flying model, we will use the six states of position and velocity, with state update matrix

$$F_i = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{59}$$

Process noise is assumed picewise constant on the acceleration of target.

$$u_i = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{60}$$

Also

$$G_i = \begin{bmatrix} T^2/2 & 0 & 0 \\ 0 & T^2/2 & 0 \\ 0 & 0 & T^2/2 \\ T & 0 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix} \tag{61}$$

This shows that the acceleration noise affects velocity state directly (with a factor T), while The position state is effected by the second order effect $T^2/2$. The acceleration in each of the three dimensions is assumed independent and piecewise constant white noise distributed $N(0, 0.1g)$.

For the constant turning rate model we add a seventh state for turning rate ω, i.e. the velocity in the horizontal plane is assumed to be $\boldsymbol{v}_h(t) = v_h(t)\cos(\omega t - \omega_0)\bar{x} + v_h(t)\sin(\omega t - \omega_0)\bar{y}$, where $\bar{x}$ and $\bar{y}$ are unit vectors in the x- and y-directions. The state transition matrix is then non-linear and must be linearized. The state transition matrix becomes

$$F_{\omega i} = \begin{bmatrix} & \frac{Tv_x\cos(\omega T)}{\omega} - \frac{v_x\sin(\omega T)}{\omega^2} - \frac{v_y T\sin(\omega T)}{\omega} + \frac{v_y(1-\cos(\omega T)}{\omega^2} \\ & \frac{Tv_y\cos(\omega T)}{\omega} - \frac{v_y\sin(\omega T)}{\omega^2} + \frac{v_x T\sin(\omega T)}{\omega} - \frac{v_x(1-\cos(\omega T)}{\omega^2} \\ F_i' & 0 \\ & -Tv_x\sin(\omega T) - Tv_y\cos(\omega T) \\ & Tv_x\cos(\omega T) - Tv_y\sin(\omega T) \\ & 0 \\ 0 & 1 \end{bmatrix} \tag{62}$$

where F_i' is a modification of (59) to

$$F_i' = \begin{bmatrix} 1 & 0 & 0 & \frac{\sin\omega T}{\omega} & -\frac{1-\cos\omega T}{\omega} & 0 \\ 0 & 1 & 0 & \frac{1-\cos\omega T}{\omega} & \frac{\sin\omega T}{\omega} & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & \cos\omega T & -\sin\omega T & 0 \\ 0 & 0 & 0 & \sin\omega T & \cos\omega T & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{63}$$

The acceleration noise states are still used (although with lower variance). An additional noise source affecting the turning rate is also introduced. Note that this state transition matrix is not used to update the state, but only for the covariance matrix update as indicated by (41–42). A similar matrix can be formed for use in the state update.

Note that this state transition matrix is a function of the variables v_x, v_y and ω which are part of the state. For the non-linear turning model to work, time updates cannot be done over long time intervals in one step. To update the state ten seconds, one can e.g. do ten one-second time updates. The time-span over which one can accurately do time updates in one step depends on the value of ω (which can be more precisely examined by looking at the second derivative for determining the linearization error).

4.5 Altitude problem

In the described scenario the radar stations all have similar altitudes and are therefore close to lying in the same plane. Two different solutions therefore come close to satisfying the measurements, one below the ground and one above. The filter must therefore be restricted to only use the above-ground solution. The missile also flies at low altitude and is therefore close to being co-planar with the stations. Measurements are therefore very insensitive to altitude, and we will therefore have large altitude errors. This can cause problems with the linearization as well as with the altitude ambiguity.

During time updates, if the current estimate of velocity will make the altitude pass below ground, we will therefore set the vertical velocity to zero and redo the time update.

During measurement updates if the altitude ends up below ground, we will set the vertical velocity to zero and the altitude to 50 m.

4.6 Simulation setup

A number of stations with positions shown in Figure 1 is used. The altitude of most of the stations is 50 m, and few are at 100 m. The same figure shows the path of a missile that we attempt to track, the break-points of the missile are listed in Table 1. RT90 x-y coordinates are used, and the ground is assumed to be level at 0 m.

Other system parameters and settings are used in the simulation are shown in Table 2.

North [m]	East [m]	altitude [m]	velocity [m/s]
1450000.0	6500000.0	350	110
1560000.0	6620000.0	150	120
1670000.0	6700000.0	200	-

Table 1: Missile breakpoints

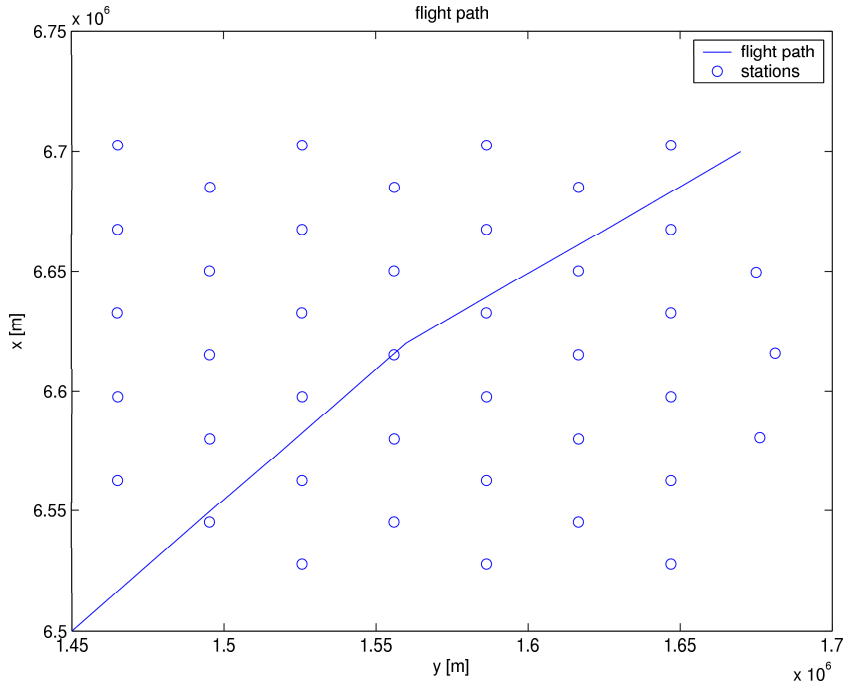


Figure 1: Flight Track and Radar Stations

Parameter	value
Center frequency	500 MHz
Bandwidth	30 MHz
Noise Temperature	500 K
Probability of False Alarm	10^{-6}
Integration time	0.05 s
Measurement Interval	1 s
Mean Power	200 W
Antenna	Dipole
Polarization	Vertical
Swerling case	1
ereps delta (evaporation duct height)	0.0 m
ereps humidity	7.5 g/m^3
ereps sbdht (surface based duct height)	0.0 m
Wind Speed	0.0 m/s

Table 2: System and Simulation Parameters

4.7 Simulation results

4.7.1 Static results

Maps of the PD and the accuracies for the scene at altitudes 150 and 200 meters were produced to get an idea how it looks before tracking. Figures 2-7 shows the results for a target with constant RCS of 1 m^2. The PD and the accuracies are for a single update of the station net, so measurements of that type is basically what the Kalman filter has to work with. See section 4.3 for more information about the way the pictures were created.

As can be seen in figure 2 there are areas at altitude 150 m where the PD is close to zero. At altitude 200 m those areas are almost gone. Position accuracy in the horizontal plane is below 3 m in all areas where the PD/SNR is decent but the position accuracy in the vertical axis is much worse. At altitude 150 m it is not uncommon to have inaccuracies of over 100 m in the vertical axis.

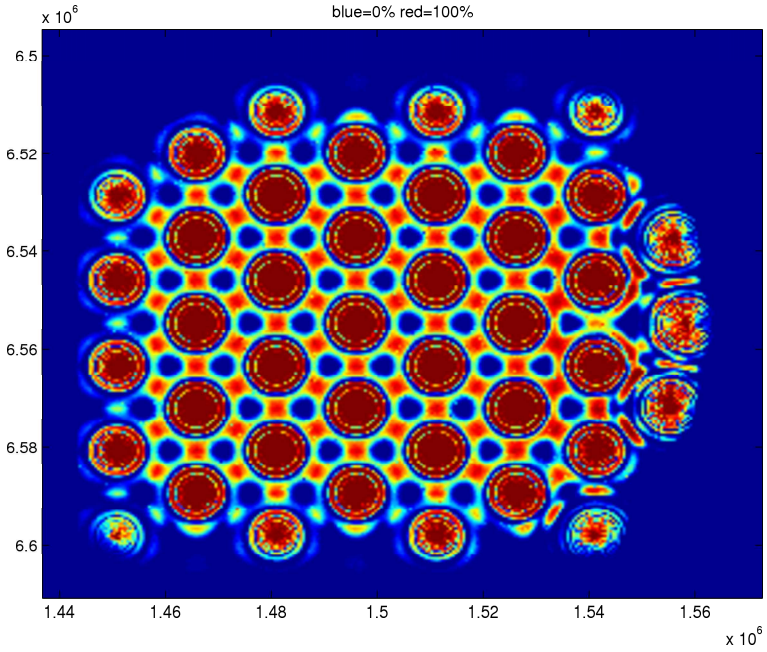


Figure 2: The PD for a target with RCS 1 m at an altitude of 150 m.

4.7.2 Tracking results

We will here show some examples of results from some simulation runs. In Figure 8 the northerly error is shown. The plot also includes the Kalman filter error estimate. If this estimate is correct the error should lie within the estimate 50% of the time. In Figures 9–11, the same parameters are shown for north velocity, altitude and vertical velocity.

The reason the errors grow larger at the end of the simulation is that the missile is leaving the area covered by the radar stations. The Kalman filter thus must rely more on prediction based on extrapolation, and the accuracy of this estimate naturally deteriorates. The short spike in the errors in the middle of the plots are due to the missile maneuver that occurs there. The filter model assumes a straight path and the estimate therefore briefly lags behind. The acceleration noise we have chosen is however fairly large so the lag is very small. The performance in the straight parts can of be improved significantly be assuming lower (or even zero) acceleration-noise. A “real” missile will however not fly perfectly straight. Reducing acceleration noise will also increase the error spike at the maneuver. Multiple models will then be needed

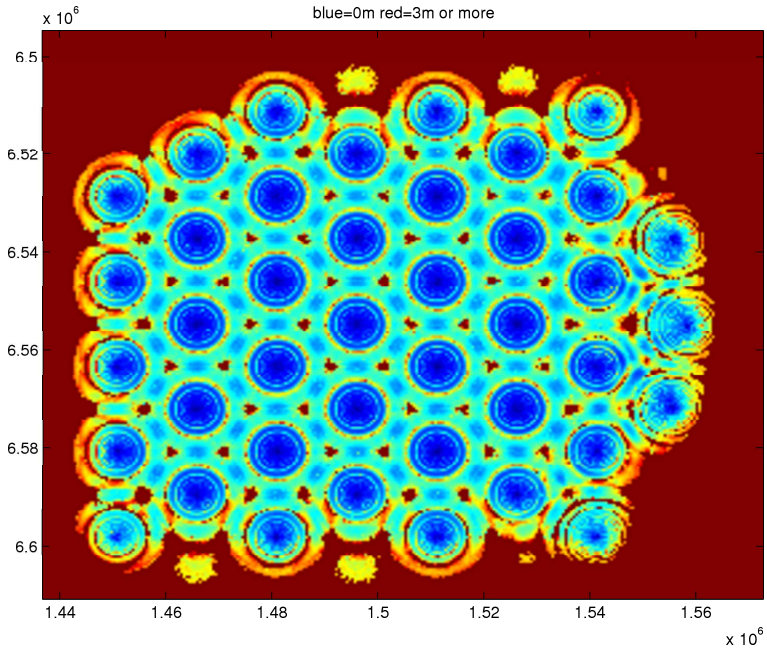


Figure 3: The horizontal accuracy component for a target with RCS 1 m at an altitude of 150 m.

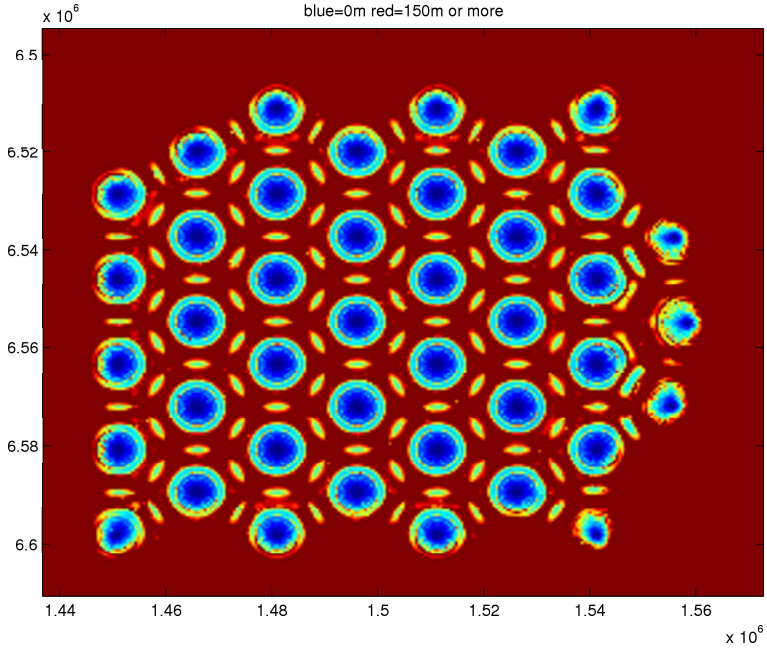


Figure 4: The vertical accuracy component for a target with RCS 1 m at an altitude of 150 m.

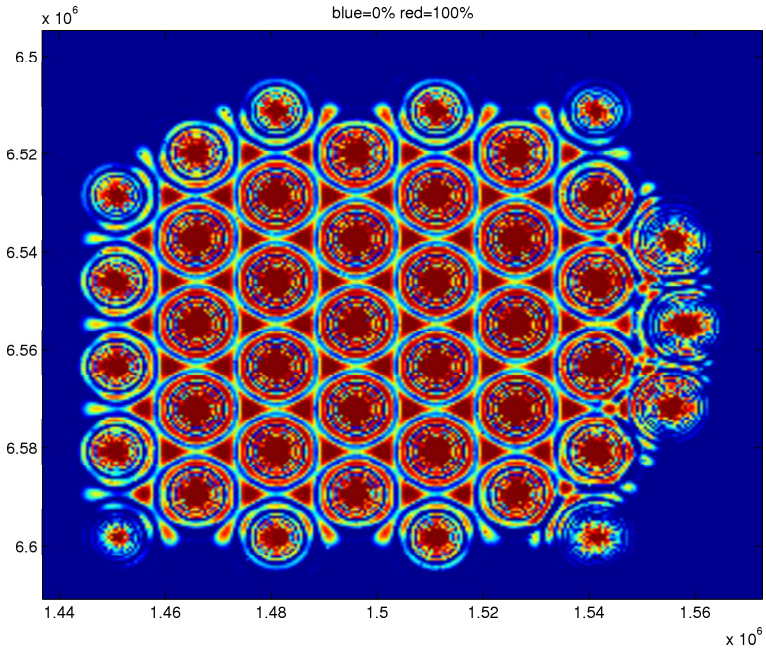


Figure 5: The PD for a target with RCS 1 m at an altitude of 200 m.

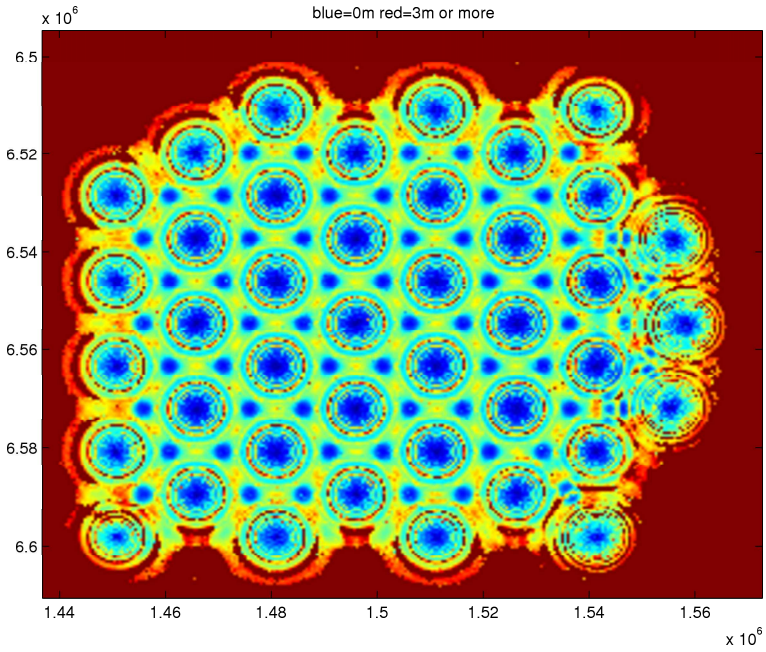


Figure 6: The horizontal accuracy component for a target with RCS 1 m at an altitude of 200 m.

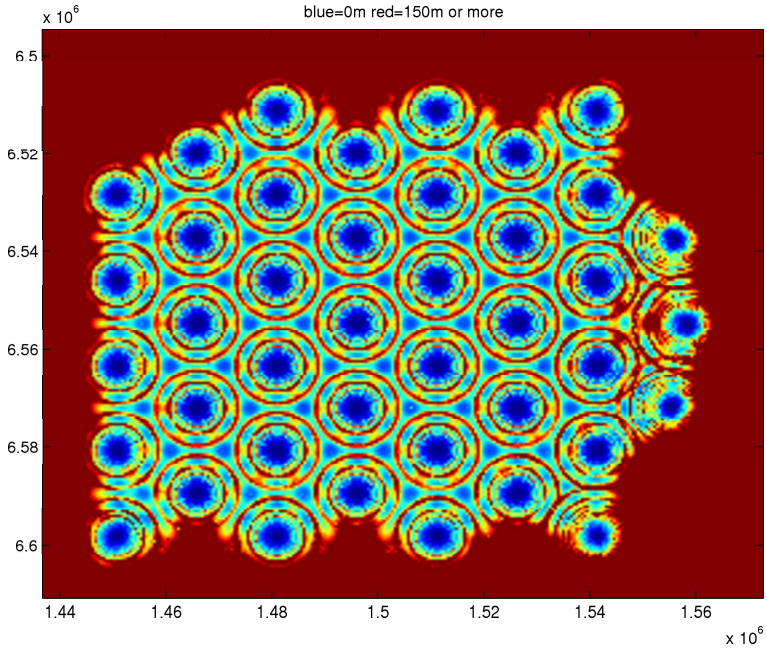


Figure 7: The vertical accuracy component for a target with RCS 1 m at an altitude of 200 m.

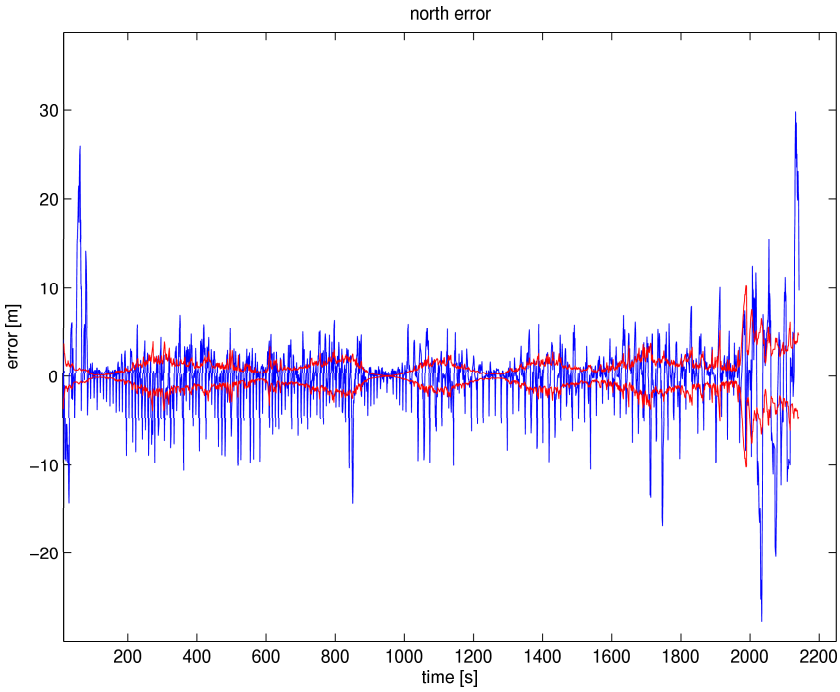


Figure 8: North error

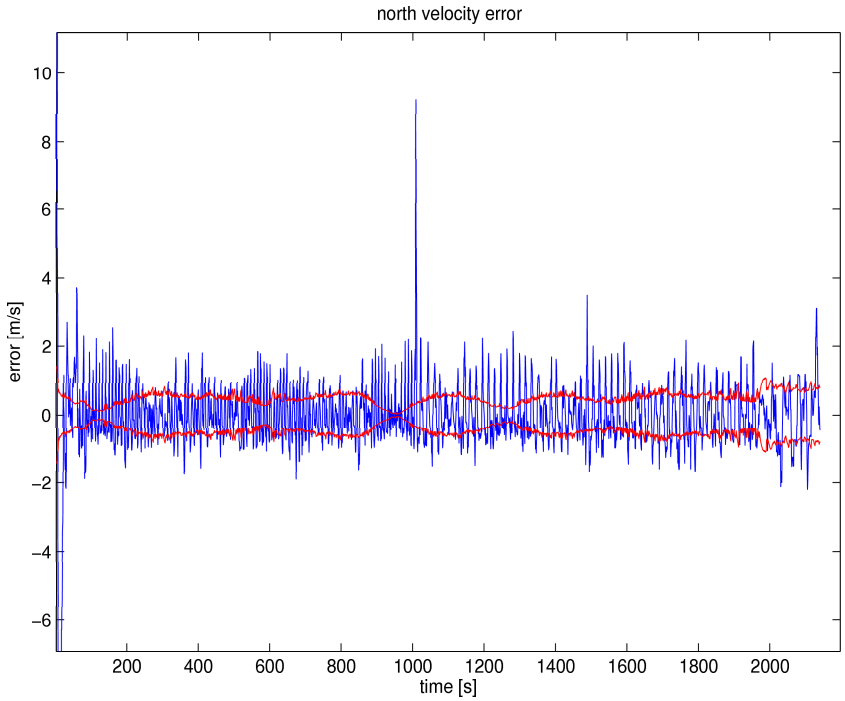


Figure 9: North velocity error

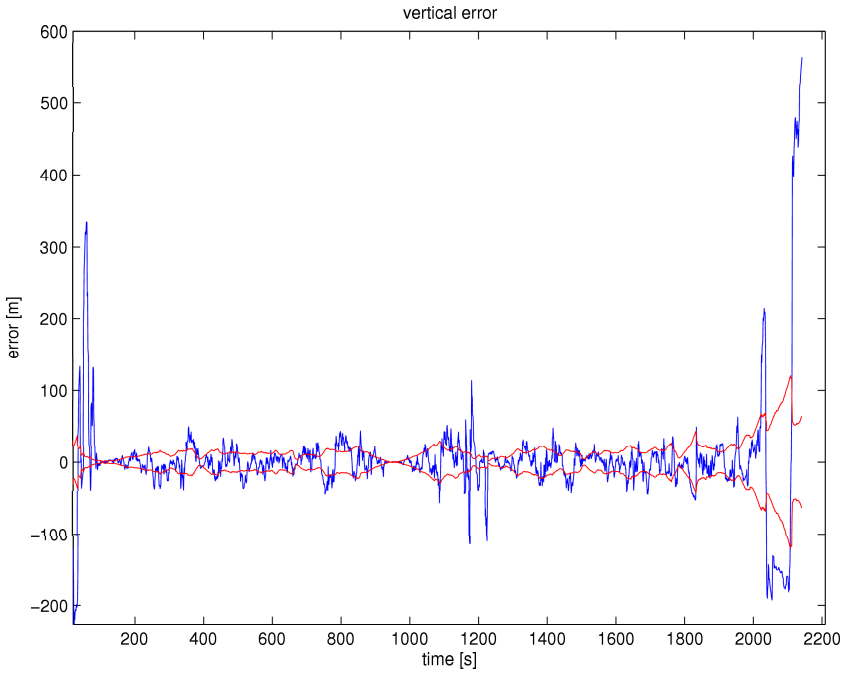


Figure 10: Altitude error

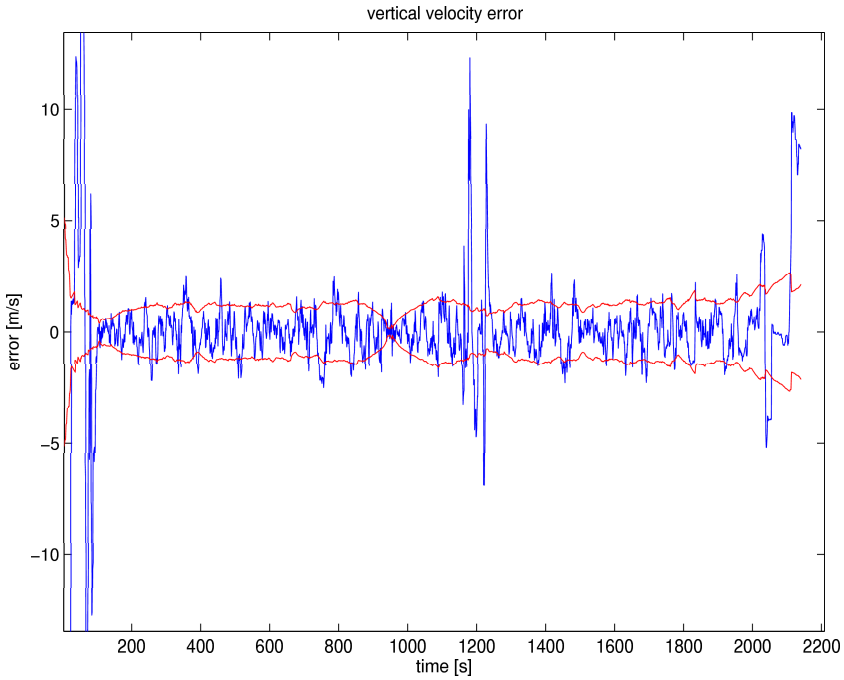


Figure 11: Vertical velocity error

to detect when maneuvers occur. In Figure 12 the maneuver effect on east velocity is shown. The "true" velocity is also included in this plot. As can be seen, the filter follows the true velocity quite well, despite the step change. Some part of the error can be explained by our simple missile simulation that allows for momentaneous changes in velocity (i.e. infinite acceleration). The position simulation does not suffer from this problem and the plots show that position errors do not show such large peaks at the breakpoint. In Figure 13 a part of the altitude plot around the breakpoint is shown. As can be seen the filter follows the true altitude well. Using the constant turning rate model, yields results very close to the results in the plots. The error spike at the the maneuver is however reduced somewhat. Since the turn is done instantaneous, rather than with a constant turning rate as the model assumes, this is expected. The turning rate is by the filter naturally for most of the simulation estimated to be close to zero.

At the end of the altitude plot larger errors start to occur. Measurements here are at larger distances and therefore at relatively lower altitude. The altitude sensitivity therefore goes down. When the altitude error estimate approaches the actual altitude, the linearization errors and altitude ambiguity problems also become more pronounced.

In Figure 14 the benefits of using a Kalman filter over momentaneous pointwise estimates is shown. Note that it this the methods own estimate of the error that is shown, not the actual error from one simulation (which stations that detect the missile is however simulated). The pointwise plot can be seen as a slice through Figure 4 along the flight-path. There are however several differences between the simulations, here the altitude is not constant, we do not take the average over many simulations and the RCS is not a constant 1 m. Besides being lower for the Kalman filter compared to the pointwise estimate, the pointwise error estimate plot has lots of "holes", e.g. times when no estimate is available. A Kalman filter has other benefits, including better interpolation, extrapolation and handling of non-simultaneous measurements. The exact size of the reduced error depends on the ratio of measurement noise to missile acceleration. With stronger modeling on the basically correct straight-line assumption, the filter benefit will be even larger.

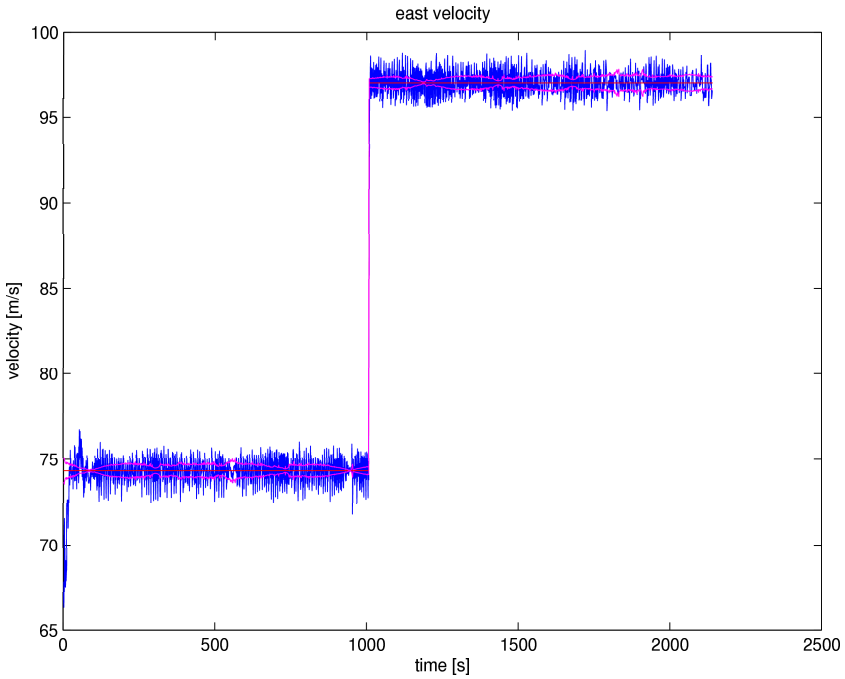


Figure 12: East velocity

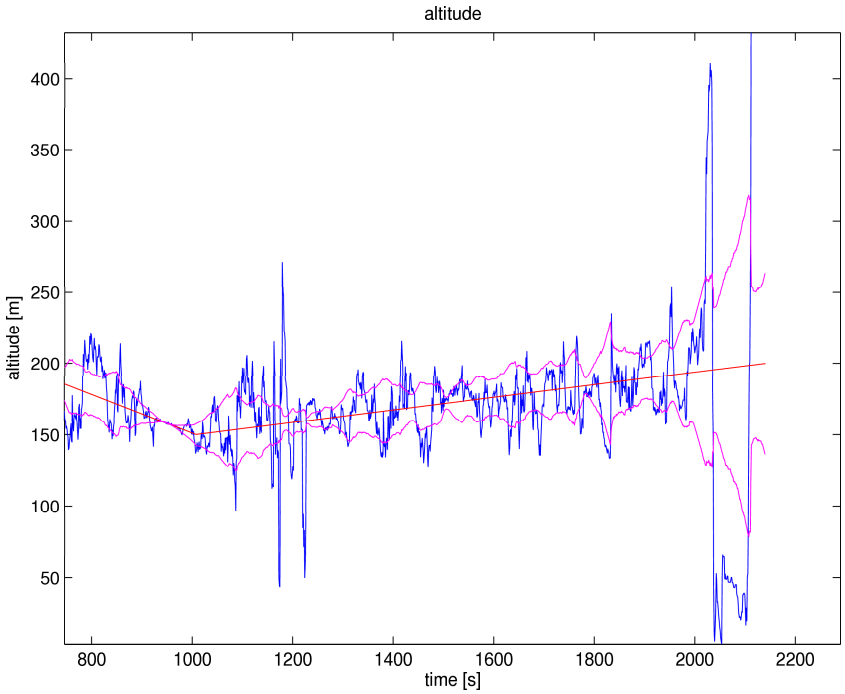


Figure 13: Altitude

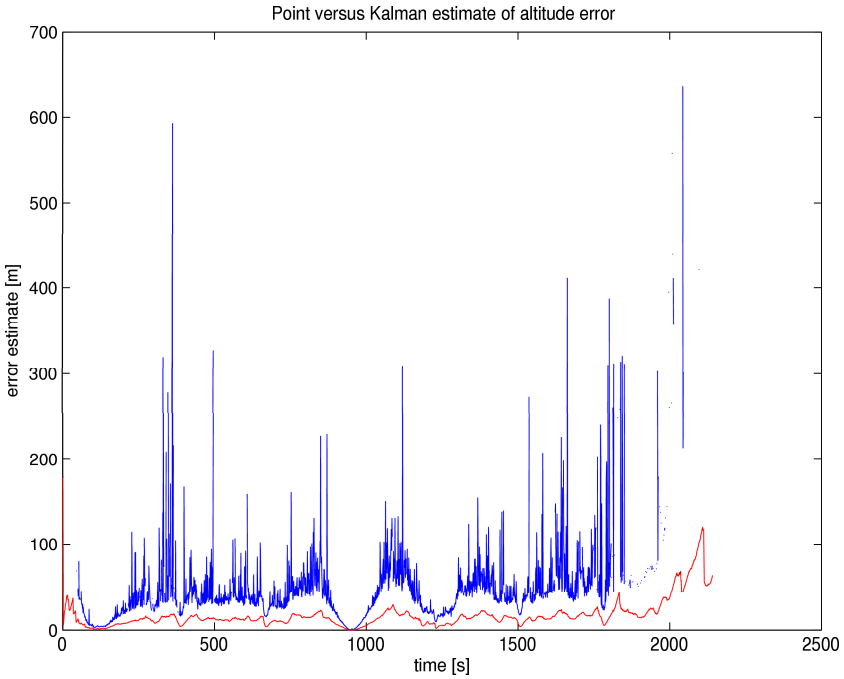


Figure 14: Kalman Filter and Momentaneous estimates of altitude error

5 Conclusions

We have shown how using only mono- and bi-static range and Doppler measurements from a radar network, a cruise missile position and velocity can be calculated. We have examined both momentaneous estimates and filtering over time. A scenario with the difficult case of a low-flying cruise missile was examined, and yielded good tracking results. The low altitude causes ambiguity and sensitivity problems, but we have shown that these can be handled. The simulations show the significantly reduced error that can be achieved when using a Kalman Filter for measurements over time compared to using only momentaneous measurements. The exact amount of improvement depends on the movement behavior, radar response of the missile and system noise. The Kalman filter has additional benefits, including better interpolation, extrapolation and handling of non-simultaneous measurements.

References

[1] W.L. Patterson et.al, *Engineer's Refractive Effects Prediction System (EREPS)* Technical Document 2648, US. Naval Command Control and Ocean Surveillance Center, May 1994

[2] M. Herberthson, *Concept studies Guidance Radar - final report 2003*, User Report FOI-R–1040-SE, December 2003

[3] P. Grahn, M. Herberthson, G. Haapalahti, A. Alm, O. Kärvell, *Assessment of AASR - final report*, FOI report FOI-R–1265–SE, June 2004

[4] H.Hellsten, *System för att med från mål spridda signaler bestämma lägen och hastigheter för målen*, Patent number 0101661-7, 2001

[5] M. Herberthson, *A direct target localization method for AASR - principles and implementation*, FOI report FOI-R–0515-SE, 2002

[6] M. Herberthson, *On determining multiple positions and velocities from bistatic measurements - fast bistatic association in AASR*, FOI report FOI-R–0815-SE, 2003

[7] M. Pettersson, *Halvårsrapport juli 2003 i projektet Avancerade målsökarsystem och koordinatstyrning*, FOI MEMO 03-1833, Augusti 2003

[8] M. Pettersson, *Aspekter på radarnätverk för målföljning av smygmål som en kryssningsmissil på låghöjd*, FOI report in-press, November 2004

[9] Thomas Kailath, Ali H. Sayed, Babak Hassibi, *Linear Estimation*, Prentice Hall, 2000.