LAMU tracker – a tracker for sensors with Large Angular Measurement Uncertainty

Brita H. Hafskjold Gade

Norwegian Defence Research Establishment (FFI)

22. January 2007

FFI-rapport 2006/03576

1060

ISBN 978-82-464-1083-8

Keywords

Tracking

Kalman filter

Measurement uncertainty

Approved by

Johnny Bardal Avdelingssjef/Director

Sammendrag

LAMU (Large Angular Measurement Uncertainty) trackeren har blitt utviklet for sensorer med stor vinkel-usikkerhet, altså stor målestøy for vinkelmålingen. Mange HF (High Frequency) radarer er eksempler på slike sensorer. Tradisjonelle trackere har problemer med å lage gode track for slike sensorer, fordi det sanne usikkerhetsområdet er ulineært, og derfor ikke blir brukt. I LAMU trackeren brukes de sanne måleusikkerhetsområdene, mens det i tradisjonelle trackere brukes et elliptisk usikkerhetsområde. Fordelene med å bruke det sanne usikkerhetsområdet er i denne rapporten vist i simuleringer, der LAMU trackeren lager få falske track, og et track som er svært nær fasit, mens tradisjonell PDA (Probabilistic Data Association) lager mange falske track. LAMU trackeren har også vist gode resultater for virkelige data.

LAMU trackeren er brukt på virkelige data fra en HF radar, med svært gode resultater [1].

LAMU trackeren er utviklet for en 2D sensor som måler avstand, vinkel og hastighet i avstandsretning. Det er svært enkelt å endre LAMU trackeren til å bruke kun avstands- og vinkel-målinger.

MHT (Multi Hypotese Tracking) er den teoretisk optimale trackeren. Ved å videreutvikle LAMU trackeren til å være basert på MHT istedet for å være inspirert av PDA, vil vi sannsynligvis forbedre resultatene ytterligere.

English summary

The LAMU (Large Angular Measurement Uncertainty) tracker has been developed for sensors with large angular measurement uncertainty, such as HF (High Frequency) radars. Traditional trackers do not perform very well for such sensors. The LAMU tracker uses the real uncertainty area of the measurements, while most regular trackers use elliptical uncertainty areas. The advantages of using the real uncertainty areas are shown in simulations, where the LAMU tracker has few false tracks and one track close to the true track, compared to the regular PDA (Probabilistic Data Association) tracker which has many false tracks.

The LAMU tracker has shown very good results when used for tracking real data from an HF radar [1].

The LAMU tracker is developed for a 2D sensor that measures range, azimuth and range rate (e.g. by Doppler measurements). However, the LAMU tracker may easily be changed to fit sensors without range rate measurements.

MHT (Multi Hypothesis Tracker) is the theoretically optimal tracker. Thus, a LAMU tracker based on MHT rather than PDA will probably give even better results than the LAMU tracker described in this report.

Content

1 Introduction

Sensors used for tracking targets typically measure the range and the angle to targets. For 2D sensors, the azimuth angle is measured, and for 3D sensors, both the azimuth and the elevation angles are measured. Some sensors also measure the range rate of the target. Most sensors deliver a mix of true detections (i.e. detections originating from a target) and false detections (usually originating from noise).

The task of the tracking algorithm is to use the detections from the sensor to make tracks of the targets. Two of the recommended trackers for sensors delivering a large amount of false detections, are:

- PDA (Probabilistic Data Association), which tracks relatively well through noisy areas (i.e., areas with a large amount of false detections), uses a relatively low amount of CPU and is easy to implement. PDA uses a "calculated measurement" which is a mixture of all measurements within the gate (i.e., within the uncertainty area of the predicted position of the target) to update the track.
- MHT (Multi Hypothesis Tracker), which is the theoretically optimal tracker, is CPU intensive and relatively complicated to implement. MHT makes a separate track for each of the measurements in the gate, and each of these tracks are used as different hypotheses.

Due to a tight time schedule, MHT has not been used in this report. Instead, we have used the main principles of PDA as a basis to develop a tracker that performs well for sensors with a large angular measurement uncertainty and a large amount of false detections.

Most tracking algorithms use Cartesian linear models of both the measurement process and the movement of the target. This leads to elliptical uncertainty areas. For sensors with a relatively high precision in range and angle, these elliptical uncertainty areas correspond well to the real uncertainty areas. As explained in Chapter 2, these areas do not correspond well for sensors with either large range uncertainty or large angular uncertainty. Therefore, we use a Polar model for most aspects of the tracker. For manoeuvring targets, the elliptical uncertainty area of the predicted position does not correspond well to the real uncertainty area. One way to solve this problem is to use particle filters. This report does not assume highly manoeuvring targets, and therefore uses a single filter.

The LAMU (Large Angular Measurement Uncertainty) tracker described in this report is developed for a 2D sensor that measures:

- Range
- Azimuth
- Range rate (e.g. by Doppler measurements)

However, the LAMU tracker may easily be changed to fit sensors without range rate measurements.

The mathematical deduction of the equations for the LAMU tracker is placed in Chapter 3.

To establish a new track, we use three detections from different points in time, as described in Chapter 4.

We assume a sensor that may deliver many detections (as a mixture of true and false detections) at each point in time. Thus, we need to decide which of these detections that may belong to a certain track. This process, which is called gating, finds the detections that lie within an uncertainty area of the predicted position of the track, as described in Chapter 5.

If the gating finds several simultaneous detections that are believed to belong to the same track, a "calculated measurement" is made as a weighted sum of these detections, as described in Chapter 6.

For scenarios with many false detections, several tracks may be established in the vicinity of the targets, and if these tracks are not merged, several tracks may follow the target. Usually, we do not want more than one track to follow each target. Thus, if more than one track seem to follow the same target, these tracks may be merged by a process called pruning, as described in Chapter 7.

2 Why do sensors with large angular uncertainty need a special tracker?

A sensor with large angular measurement uncertainty will have a real uncertainty area like the green arc shown in Figure 2.1. However, most tracking algorithms use an elliptical uncertainty area, such as the pink ellipse. For small angular measurement uncertainties, the elliptic uncertainty area is a good representation of the real uncertainty area. However, for large angular measurement uncertainties, a large amount of the real uncertainty area is outside of the elliptical uncertainty area, and vice versa. Thus, a tracker is needed that use the real uncertainty area in all aspects of the tracking.

The LAMU tracker uses the real uncertainty area for:

- Gating
- Kalman filtering, including:
    - Measurement uncertainty for range and azimuth
    - Measurement uncertainty for range rate

- Probability calculations of detections, if several detections are inside the gate.
    - For range and azimuth
    - For range rate

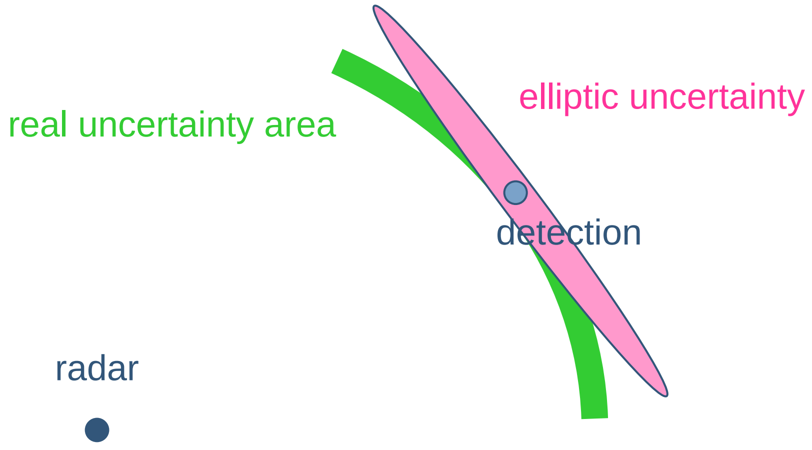

Figure 2.1 The real uncertainty area is used by the LAMU tracker, and the elliptical uncertainty area is used by traditional trackers.

One way to reduce the problem of the elliptic uncertainty area without using the real uncertainty area, is to increase the radial axis of the ellipse. This is further explained in chapter 9, where PDA is simulated with both original and enlarged uncertainty ellipses, and compared to LAMU.

3 Mathematical modelling and Kalman filtering

3.1 Assumptions

- The objects of interest have a speed that is above zero and below a maximum velocity, called max_velocity in the rest of this report.
- The objects of interest may be modelled as having a constant course and speed.
- The objects of interest have an acceleration that is below max_acceleration, which is used to find the uncertainty of the system model.

3.2 Notation

The report uses a subset of the notation defined in [2]. The notation used in this report is summarized in Tables 3.1-3.5.

General symbols used in this report are summarized in Table 3.1.

Variable	Description
t	Current time (continuous)
k	Current time-step (discrete)
h	Time since last time-step

Table 3.1: General symbols

Table 3.2 contains various variants of a general variable *x*.

Mathematical symbol	Description
x	True value
$\tilde{x}$	Measured value
$\hat{x}$	Updated value (from the Kalman filter)
$\overline{x}$	Predicted value (from the Kalman filter)
$\dot{x}$	Derivative (in time)
x_{k+1}	Value valid next time-step
x_{k-1}	Value valid last/previous time-step
δx	Error in a computed or measured version of the variable *x*.

Table 3.2: Variants of a general variable x.

The coordinate frames relevant for this report are defined in Table 3.3.

Symbol	Description
B	***Name:*** **B**ody ***Position:*** The origin is the object's reference point. ***Orientation:*** The *x*-axis points forward, the *y*-axis to the right (starboard) and the *z*-axis in the vehicle's down direction. However, the orientation of *B* is not used in this report. ***Comments:*** The frame is fixed to the vehicle.
M	***Name:*** **M**ap ***Position:*** The origin is earth fixed, at the radar. ***Orientation:*** The *z*-axis is pointing down. The *x*-axis points towards north, and the *y*-axis points towards east (NED – North East Down).

Table 3.3: Coordinate frame definitions

The symbol usage in this report is summarized in Table 3.4.

Symbol	**Description:**	**Example:**
Lowercase letter	Scalar variable	r
Bold lowercase letter	Vector decomposed in a coordinate frame.	$\boldsymbol{v}_{AB}^{A}$ (Velocity of the *B*-system relative to *A*, decomposed in *A*)
Right subscript	Specification of the value (coordinate frame involved, timestep, etc.).	
Right superscript	In which coordinate frame the vector is decomposed.	
Bold uppercase letter	Matrix	$\boldsymbol{D}$ (Measurement matrix)

Table 3.4: Symbol usage.

Table 3.5 shows the notation used to describe relations between two coordinate frames.

Symbol	**Description**
$\boldsymbol{p}_{AB}^{A}$	A vector whose length and direction is such that it goes from the origin of coordinate frame *A* to the origin of coordinate frame *B,* decomposed in *A*.
$\boldsymbol{v}_{AB}^{A}$	The velocity of the origin in coordinate frame *B* relative to coordinate frame *A*, decomposed in *A*.
$\boldsymbol{a}_{AB}^{A}$	The acceleration of the origin in coordinate frame *B* relative to coordinate frame *A*, decomposed in *A*.

Table 3.5: Notation used to describe different relations between two coordinate frames.

3.3 Mathematical modelling

3.3.1 Deducing the state space model

The measurements from the sensor are range, azimuth and range rate, while the typical object has a constant course and speed. There are thus two main choices for the states in the state vector:

- If the state vector contains position and velocity in north and east direction, the system matrix is simple and linear, while the measurement matrix is non-linear. This gives an elliptical uncertainty area, as shown in Figure 2.1, which is quite different from the real uncertainty area.

- If we use range, azimuth, range rate and azimuth rate as states, our state space equation will neither be simple nor linear, but the measurement equation will be simple, linear and give a correct representation of the measurement noise. The uncertainty area of the state vector will be equal to the real uncertainty area illustrated in Figure 2.1.

We have chosen range, azimuth, range rate and azimuth rate as states. In the following, we will calculate the nonlinear state space equation and the nonlinear state matrix needed by the nonlinear Kalman filter.

The position vector from the sensor (*M*) to the observed object (*B*) decomposed in the sensor (*M*) coordinate frame is (only the north and east components are used here, since we assume a 2 dimensional sensor):

$$\boldsymbol{p}_{MB}^{M} = r\begin{bmatrix}\cos(\theta)\\ \sin(\theta)\end{bmatrix} \tag{3.1}$$

where *r* is the range (distance from the radar to the object) and *θ* is the azimuth (clockwise positive angle, which is zero at the north pole).

The velocity vector of the observed object is obtained by time differentiating the position vector:

$$\begin{aligned}\boldsymbol{v}_{MB}^{M} &= \frac{d}{dt}\left(\boldsymbol{p}_{MB}^{M}\right)\\ &= \frac{d}{dt}\left(r\begin{bmatrix}\cos(\theta)\\ \sin(\theta)\end{bmatrix}\right)\\ &= \frac{d}{dr}\left(r\begin{bmatrix}\cos(\theta)\\ \sin(\theta)\end{bmatrix}\right)\frac{dr}{dt}+\frac{d}{d\theta}\left(r\begin{bmatrix}\cos(\theta)\\ \sin(\theta)\end{bmatrix}\right)\frac{d\theta}{dt}\end{aligned} \tag{3.2}$$

$$\Downarrow$$

$$\begin{aligned}\boldsymbol{v}_{MB}^{M} &= \begin{bmatrix}\cos(\theta)\\ \sin(\theta)\end{bmatrix}\dot{r}+r\begin{bmatrix}-\sin(\theta)\\ \cos(\theta)\end{bmatrix}\dot{\theta}\\ &= \begin{bmatrix}\dot{r}\cos(\theta)-r\sin(\theta)\dot{\theta}\\ \dot{r}\sin(\theta)+r\cos(\theta)\dot{\theta}\end{bmatrix}\end{aligned} \tag{3.3}$$

The acceleration vector is obtained by time differentiating the velocity vector:

$$\begin{aligned}\boldsymbol{a}_{MB}^{M} &= \frac{d}{dt}\left(\boldsymbol{v}_{MB}^{M}\right) = \frac{d}{dt}\left(\begin{bmatrix}\cos(\theta)\\ \sin(\theta)\end{bmatrix}\dot{r} + r\begin{bmatrix}-\sin(\theta)\\ \cos(\theta)\end{bmatrix}\dot{\theta}\right)\\ &= \frac{d}{dt}\left(\begin{bmatrix}\cos(\theta)\\ \sin(\theta)\end{bmatrix}\right)\dot{r} + \begin{bmatrix}\cos(\theta)\\ \sin(\theta)\end{bmatrix}\ddot{r} + \dot{r}\begin{bmatrix}-\sin(\theta)\\ \cos(\theta)\end{bmatrix}\dot{\theta} + r\frac{d}{dt}\left(\begin{bmatrix}-\sin(\theta)\\ \cos(\theta)\end{bmatrix}\dot{\theta}\right)\\ &= \begin{bmatrix}-\sin(\theta)\\ \cos(\theta)\end{bmatrix}\dot{\theta}\dot{r} + \begin{bmatrix}\cos(\theta)\\ \sin(\theta)\end{bmatrix}\ddot{r} + \dot{r}\begin{bmatrix}-\sin(\theta)\\ \cos(\theta)\end{bmatrix}\dot{\theta} + r\left(\frac{d}{dt}\left(\begin{bmatrix}-\sin(\theta)\\ \cos(\theta)\end{bmatrix}\right)\dot{\theta} + \begin{bmatrix}-\sin(\theta)\\ \cos(\theta)\end{bmatrix}\ddot{\theta}\right)\\ &= \begin{bmatrix}-\sin(\theta)\\ \cos(\theta)\end{bmatrix}\dot{\theta}\dot{r} + \begin{bmatrix}\cos(\theta)\\ \sin(\theta)\end{bmatrix}\ddot{r} + \dot{r}\begin{bmatrix}-\sin(\theta)\\ \cos(\theta)\end{bmatrix}\dot{\theta} + r\begin{bmatrix}-\cos(\theta)\\ -\sin(\theta)\end{bmatrix}\dot{\theta}^{2} + r\begin{bmatrix}-\sin(\theta)\\ \cos(\theta)\end{bmatrix}\ddot{\theta}\end{aligned} \qquad (3.4)$$

As explained above, we use a model of the observed object with constant course and speed, i.e. $\boldsymbol{a}_{MB}^{M} = \boldsymbol{0} + noise$. Thus, we get:

$$\begin{bmatrix}0\\ 0\end{bmatrix} = \begin{bmatrix}-\sin(\theta)\\ \cos(\theta)\end{bmatrix}\dot{\theta}\dot{r} + \begin{bmatrix}\cos(\theta)\\ \sin(\theta)\end{bmatrix}\ddot{r} + \dot{r}\begin{bmatrix}-\sin(\theta)\\ \cos(\theta)\end{bmatrix}\dot{\theta} + r\begin{bmatrix}-\cos(\theta)\\ -\sin(\theta)\end{bmatrix}\dot{\theta}^{2} + r\begin{bmatrix}-\sin(\theta)\\ \cos(\theta)\end{bmatrix}\ddot{\theta} \qquad (3.5)$$

Equation (3.5) gives the following two equations:

$$\begin{aligned}0 &= -2\sin(\theta)\dot{\theta}\dot{r} + \cos(\theta)\ddot{r} - \cos(\theta)r\dot{\theta}^{2} - \sin(\theta)r\ddot{\theta}\\ 0 &= 2\cos(\theta)\dot{\theta}\dot{r} + \sin(\theta)\ddot{r} - \sin(\theta)r\dot{\theta}^{2} + \cos(\theta)r\ddot{\theta}\end{aligned} \qquad (3.6)$$

By solving the first equation of (3.6) with relation to $\ddot{r}$, we get:

$$\ddot{r} = \frac{2\sin(\theta)}{\cos(\theta)}\dot{\theta}\dot{r} + r\dot{\theta}^{2} + \frac{\sin(\theta)}{\cos(\theta)}r\ddot{\theta} \qquad (3.7)$$

By inserting (3.7) into the second equation of (3.6), we get:

$$0 = 2\cos(\theta)\dot{\theta}\dot{r} + \sin(\theta)\left(\frac{2\sin(\theta)}{\cos(\theta)}\dot{\theta}\dot{r} + r\dot{\theta}^{2} + \frac{\sin(\theta)}{\cos(\theta)}r\ddot{\theta}\right) - \sin(\theta)r\dot{\theta}^{2} + \cos(\theta)r\ddot{\theta} \qquad (3.8)$$

$$\Downarrow$$

$$0 = 2\cos(\theta)\dot{\theta}\dot{r} + \frac{2\sin(\theta)\sin(\theta)}{\cos(\theta)}\dot{\theta}\dot{r} + \sin(\theta)r\dot{\theta}^{2} + \frac{\sin(\theta)\sin(\theta)}{\cos(\theta)}r\ddot{\theta} - \sin(\theta)r\dot{\theta}^{2} + \cos(\theta)r\ddot{\theta} \qquad (3.9)$$

$$\Downarrow$$

$$-\left(\frac{\sin(\theta)\sin(\theta)}{\cos(\theta)} + \cos(\theta)\right)r\ddot{\theta} = 2\cos(\theta)\dot{\theta}\dot{r} + \frac{2\sin(\theta)\sin(\theta)}{\cos(\theta)}\dot{\theta}\dot{r} \qquad (3.10)$$

$$\Downarrow$$

$$-\left(\frac{\sin(\theta)\sin(\theta)+\cos(\theta)\cos(\theta)}{\cos(\theta)}\right)r\ddot{\theta}=2\cos(\theta)\dot{\theta}\dot{r}+\frac{2\sin(\theta)\sin(\theta)}{\cos(\theta)}\dot{\theta}\dot{r} \tag{3.11}$$

$$\Downarrow$$

$$-\left(\frac{r}{\cos(\theta)}\right)\ddot{\theta}=2\cos(\theta)\dot{\theta}\dot{r}+\frac{2\sin(\theta)\sin(\theta)}{\cos(\theta)}\dot{\theta}\dot{r} \tag{3.12}$$

$$\Downarrow$$

$$\ddot{\theta}=2\cos(\theta)\left(\frac{-\cos(\theta)}{r}\right)\dot{\theta}\dot{r}+\frac{2\sin(\theta)\sin(\theta)}{\cos(\theta)}\left(\frac{-\cos(\theta)}{r}\right)\dot{\theta}\dot{r} \tag{3.13}$$

$$\Downarrow$$

$$\ddot{\theta}=-2\cos(\theta)\cos(\theta)\dot{\theta}\frac{\dot{r}}{r}-2\sin(\theta)\sin(\theta)\dot{\theta}\frac{\dot{r}}{r} \tag{3.14}$$

$$\Downarrow$$

$$\ddot{\theta}=-2\dot{\theta}\frac{\dot{r}}{r} \tag{3.15}$$

By inserting equation (3.15) into equation (3.7), we get:

$$\begin{aligned}\ddot{r}&=\frac{2\sin(\theta)}{\cos(\theta)}\dot{\theta}\dot{r}+r\dot{\theta}^2+\frac{\sin(\theta)}{\cos(\theta)}r\left(-2\dot{\theta}\frac{\dot{r}}{r}\right)\\&=\frac{2\sin(\theta)}{\cos(\theta)}\dot{\theta}\dot{r}+r\dot{\theta}^2-\frac{\sin(\theta)}{\cos(\theta)}2\dot{\theta}\dot{r}\end{aligned} \tag{3.16}$$

$$\Downarrow$$

$$\ddot{r}=r\dot{\theta}^2 \tag{3.17}$$

As stated earlier, we have chosen to use range, azimuth, range rate and azimuth rate in the state vector:

$$\boldsymbol{x}=\begin{bmatrix}r\\\theta\\\dot{r}\\\dot{\theta}\end{bmatrix} \tag{3.18}$$

We would now like to find a state space model of the form:

$$\dot{\boldsymbol{x}}(t)=\boldsymbol{f}\left(\boldsymbol{x}(t),t\right)+\boldsymbol{v}(t) \tag{3.19}$$

Equations (3.15) and (3.17) can be used to find the following non-linear state space model:

$$\begin{bmatrix} \dot{r} \\ \dot{\theta} \\ \ddot{r} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{r} \\ \dot{\theta} \\ r\dot{\theta}^2 \\ -2\dot{\theta}\dfrac{\dot{r}}{r} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ v_{\ddot{r}} \\ v_{\ddot{\theta}} \end{bmatrix} \tag{3.20}$$

Here, a change in course and/or speed of the object is modelled by $v_{\ddot{r}}$ in the range direction and $v_{\ddot{\theta}}$ in the cross-range direction.

Thus,

$$\boldsymbol{x}(t) = \begin{bmatrix} r \\ \theta \\ \dot{r} \\ \dot{\theta} \end{bmatrix} \tag{3.21}$$

$$\boldsymbol{f}\left(\boldsymbol{x}(t),t\right) = \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ f_3(\boldsymbol{x}) \\ f_4(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \dot{r} \\ \dot{\theta} \\ r\dot{\theta}^2 \\ -2\dot{\theta}\dfrac{\dot{r}}{r} \end{bmatrix} \tag{3.22}$$

and

$$\boldsymbol{v}\left(t\right) = \begin{bmatrix} 0 \\ 0 \\ v_{\ddot{r}} \\ v_{\ddot{\theta}} \end{bmatrix} \tag{3.23}$$

The spectral density matrix $\boldsymbol{Q}$ of the noise vector is given by:

$$\boldsymbol{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & {\sigma_{q_3}}^2 & 0 \\ 0 & 0 & 0 & {\sigma_{v_{q_4}}}^2 \end{bmatrix} \tag{3.24}$$

where σ_{q_3} is given by equation (3.29) and σ_{q_4} is given by equation (3.30).

The discrete state space equation is:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}\left(\boldsymbol{x}_k\right) + \boldsymbol{v}_k \tag{3.25}$$

where the discrete version of the system noise matrix is:

$$\boldsymbol{v}_k = \int_{t_k}^{t_{k+1}} \boldsymbol{v}(t)\,dt \tag{3.26}$$

The covariance matrix of $\boldsymbol{v}(t)$ is given by:

$$E\left(\boldsymbol{v}(t)\boldsymbol{v}^T(\tau)\right) = \boldsymbol{Q}(t)\delta(t-\tau) \tag{3.27}$$

where $\boldsymbol{Q}(t)$ is the spectral density matrix of $\boldsymbol{v}(t)$.

If we assume that $\boldsymbol{Q}(t) = \boldsymbol{Q}$ is constant, the covariance matrix $\boldsymbol{V}_k$ of the discrete system noise vector $\boldsymbol{v}_k$ is given by:

$$\begin{aligned} \boldsymbol{V}_k &= E\left(\boldsymbol{v}_k\boldsymbol{v}_k^{\;T}\right) = E\left(\int_{t_k}^{t_{k+1}}\int_{t_k}^{t_{k+1}} \boldsymbol{v}(t)\boldsymbol{v}(\tau)\,dtd\tau\right) = \int_{t_k}^{t_{k+1}}\int_{t_k}^{t_{k+1}} E\left(\boldsymbol{v}(t)\boldsymbol{v}(\tau)\right)dtd\tau \\ &= \int_{t_k}^{t_{k+1}}\int_{t_k}^{t_{k+1}} \boldsymbol{Q}\delta(t-\tau)dtd\tau = \int_{t_k}^{t_{k+1}} \boldsymbol{Q}d\tau = \boldsymbol{Q}\cdot\left(t_{k+1}-t_k\right) \end{aligned} \tag{3.28}$$

$\boldsymbol{V}_k$ is the variance of the velocity of the object. Thus $\boldsymbol{Q}$ can be thought of as the increase per second of the variance of the velocity of the object. If we assume an increase of velocity variance of $\left(\Delta\sigma_v\right)^2$ during a time period Δt, we get:

$$\sigma_q^2 = \frac{\left(\Delta\sigma_v\right)^2}{\Delta t} \tag{3.29}$$

The increase of range rate variance per second is thus:

$$\sigma_{q_3}{}^2 = \frac{\left(\Delta\sigma_v\right)^2}{\Delta t} \tag{3.30}$$

Since the connection between azimuth rate and velocity in the cross range direction is $\dot{\theta} = \frac{v_{cross-range}}{r}$, the increase of azimuth rate variance per second can be written as:

$$\sigma_{q_4}{}^2 = \frac{\left(\frac{\Delta\sigma_v}{r}\right)^2}{\Delta t} = \frac{\left(\Delta\sigma_v\right)^2}{\Delta t}\cdot\frac{1}{r^2} \tag{3.31}$$

3.3.2 The linearized version of the system matrix

Since we have a non-linear state space model of the form $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), t\right) + \boldsymbol{w}(t)$, we do not have a system matrix (***A***) in the state space equation. However, we need a linearized version of a system matrix to calculate the covariance matrix of the predicted state vector (for use in Kalman Filter prediction). This linearized matrix is calculated by the following equation (from [3]):

$$\boldsymbol{A} = \frac{\partial \boldsymbol{f}\left(\boldsymbol{x}(t), t\right)}{\partial \boldsymbol{x}(t)} \tag{3.32}$$

By combining equations (3.22) and (3.32), we can find an expression for the linearized system matrix:

$$\boldsymbol{A} = \begin{bmatrix} \frac{\partial f_1(x)}{\partial r} & \frac{\partial f_1(x)}{\partial \theta} & \frac{\partial f_1(x)}{\partial \dot{r}} & \frac{\partial f_1(x)}{\partial \dot{\theta}} \\ \frac{\partial f_2(x)}{\partial r} & \frac{\partial f_2(x)}{\partial \theta} & \frac{\partial f_2(x)}{\partial \dot{r}} & \frac{\partial f_2(x)}{\partial \dot{\theta}} \\ \frac{\partial f_3(x)}{\partial r} & \frac{\partial f_3(x)}{\partial \theta} & \frac{\partial f_3(x)}{\partial \dot{r}} & \frac{\partial f_3(x)}{\partial \dot{\theta}} \\ \frac{\partial f_4(x)}{\partial r} & \frac{\partial f_4(x)}{\partial \theta} & \frac{\partial f_4(x)}{\partial \dot{r}} & \frac{\partial f_4(x)}{\partial \dot{\theta}} \end{bmatrix} \tag{3.33}$$

By using equations (3.22) and (3.33), we can calculate the linearized system matrix ***A***:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \dot{\theta}^2 & 0 & 0 & 2r\dot{\theta} \\ 2\frac{\dot{\theta}\dot{r}}{r^2} & 0 & -2\frac{\dot{\theta}}{r} & -2\frac{\dot{r}}{r} \end{bmatrix} \tag{3.34}$$

3.4 Kalman filtering

3.4.1 Prediction

The Kalman filter prediction equations for the non-linear prediction are:

$$\bar{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_k + \left(t_{k+1} - t_k\right) \cdot \boldsymbol{f}\left(\hat{\boldsymbol{x}}_k\right) \tag{3.35}$$

$$\bar{\boldsymbol{X}}_{k+1} = \boldsymbol{\Phi}_k \hat{\boldsymbol{X}}_k \boldsymbol{\Phi}_k^{\ T} + \left(t_{k+1} - t_k\right) \cdot \boldsymbol{Q} \tag{3.36}$$

where:

$$\boldsymbol{\Phi}_k = \boldsymbol{I} + \left(t_{k+1} - t_k\right) \cdot \boldsymbol{A}_k \tag{3.37}$$

where $\boldsymbol{Q}$ is given by equation (3.23), $\boldsymbol{A}$ is given by equation (3.34), and $\boldsymbol{f}\left(\hat{\boldsymbol{x}}_k\right)$ is given by inserting $\hat{\boldsymbol{x}}_k$ into equation (3.22).

3.4.2 Measurement update

Since all the measured variables are contained in the state vector, we can use equations for a linear Kalman filter update:

$$\hat{\boldsymbol{x}}_k = \bar{\boldsymbol{x}}_k + \boldsymbol{K}_k\left(\boldsymbol{y}_k - \boldsymbol{D}\boldsymbol{x}_k\right) \tag{3.38}$$

where the measurement vector is:

$$\boldsymbol{y}_k = \begin{bmatrix} \tilde{r}_k \\ \tilde{\theta}_k \\ \tilde{\dot{r}}_k \end{bmatrix} \tag{3.39}$$

The Kalman gain matrix is given by:

$$\boldsymbol{K}_k = \bar{\boldsymbol{X}}_k \boldsymbol{D}^T\left(\boldsymbol{D}\bar{\boldsymbol{X}}_k\boldsymbol{D}^T + \boldsymbol{W}_k\right)^{-1} \tag{3.40}$$

The covariance matrix of the measurement noise is:

$$\boldsymbol{W}_k = \begin{bmatrix} \sigma_{\tilde{r}_k}{}^2 & 0 & 0 \\ 0 & \sigma_{\tilde{\theta}_k}{}^2 & 0 \\ 0 & 0 & \sigma_{\tilde{\dot{r}}_k}{}^2 \end{bmatrix} \tag{3.41}$$

where $\sigma_{\tilde{r}_k}$ is the standard deviation of the rate measurement $\sigma_{\tilde{\theta}_k}$ is the standard deviation of the azimuth measurement and $\sigma_{\tilde{\dot{r}}_k}$ is the standard deviation of the range rate measurement.

The measurement matrix is:

$$\boldsymbol{D} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.42}$$

The normal Kalman filter update equation for the covariance matrix

$$\hat{\boldsymbol{X}}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{D})\bar{\boldsymbol{X}}_k \tag{3.43}$$

can lead to numerical problems. Thus, we use the Joseph form (found in [3]), which does not have the same numerical problems:

$$\hat{\boldsymbol{X}}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{D})\bar{\boldsymbol{X}}_k(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{D})^T + \boldsymbol{K}_k\boldsymbol{W}_k\boldsymbol{K}_k{}^T \tag{3.44}$$

4 Track initiation

4.1 Which detections can establish new tracks?

To avoid initiating too many false tracks, three detections are used to establish a new track. These three detections should be at three different timesteps, and the time difference between each of them may be several timesteps.

One drawback of using three detections instead of two detections to establish a new track is the time delay introduced: The track is not established before all three detections have arrived.

To establish a new track, the three detections detection1, detection2 and detection3 must fulfil the following requirements:

- The time difference between detection1 and detection2 and between detection2 and detection3 must be below a maximum limit and above zero.
- The measured range rate of all three detections must be below max_velocity plus three times the standard deviation of the range rate measurement noise.
- The minimum distance between the uncertainty areas of detection1 and detection2, and of detection2 and detection3 and of detection1 and detection3 must be shorter than the distance travelled at max_velocity for the time period between the two detections.

4.2 How is the new track established?

This section describes how a new track is established, based on the three detections from Section 4.1.

First, a new preliminary track is initiated based on only detection1. The state vector is filled with the range, azimuth and range rate from detection1. The azimuth velocity is set to zero. The initial covariance matrix is set to:

$$\boldsymbol{X}\left(t_1\right)=\begin{bmatrix}\sigma_r^2 & 0 & 0 & 0\\ 0 & \sigma_\theta^2 & 0 & 0\\ 0 & 0 & \sigma_{\dot{r}}^2 & 0\\ 0 & 0 & 0 & \sigma_{\dot{\theta}}^2\end{bmatrix} \tag{4.1}$$

where the first three diagonal elements are given by the standard deviation for range, azimuth and range rate. Some sensors provide these numbers for each measurement, while other sensors use a constant standard deviation for range, azimuth and range rate. The standard deviation for the azimuth velocity is set to:

$$\sigma_{\dot{\theta}} = \frac{\text{max\_velocity}}{r} \tag{4.2}$$

i.e. the maximum velocity of the object in interest, divided by the range of detection1.

The state vector and the corresponding covariance matrix are predicted to the time of detection2, updated with detection2, and predicted to the time of detection3.

The gating described in Chapter 5 is then used. If detection3 is outside this gate, the new track is not initiated. If detection3 is inside the gate, the state vector and covariance matrix are updated with detection3, and the track is established as a verified track.

5 Gating

For each new timestep, several detections may arrive, both near and far away from the predicted track position. We use gating to find the subgroup of these detections that can be used to update the track.

The LAMU tracker uses separate gating in azimuth, range and range rate, while traditional trackers use gating in an elliptical uncertainty area.

For each detection, the LAMU tracker performs the following procedure:

- If the range rate of the detection is larger than max_velocity plus three times the standard deviation of the range rate measurement noise, it is regarded as a false detection. If the range rate is below or equal to the max_velocity, the following calculations are performed:
    1. The difference between the new detection and the predicted state vector of the track is calculated separately for range, azimuth and range rate.
    2. The total uncertainty is calculated separately for range, azimuth and range rate, as the sum of the detection measurement variance and the variance of the predicted state vector.
    3. By using the results from steps 1 and 2, the “statistical distance” (the number of sigma, i.e. the number of standard deviations) between the predicted state vector and the detection is calculated separately for range, azimuth and range rate. Steps 1, 2 and 3 are summarized in equations (5.1), (5.2) and (5.3).
    4. If the number of sigma is above a limit (e.g 3 sigma, corresponding to a probability of 99.7 % for one dimension), the detection is outside the gate.

The number of sigma (i.e. number of standard deviations) for range at timestep *k,* $c_{r,k}$, is calculated by:

$$c_{r,k}^{2}=\frac{\left(\tilde{r}_k-\bar{r}_k\right)^2}{\sigma_{\tilde{r},k}^{2}+\sigma_{\bar{r},k}^{2}} \tag{5.1}$$

The number of sigma for azimuth at timestep *k*, $c_{\theta,k}$, is calculated by:

$$c_{\theta,k}^{2}=\frac{\left(\tilde{\theta}_k-\bar{\theta}_k\right)^2}{\sigma_{\tilde{\theta},k}^{2}+\sigma_{\bar{\theta},k}^{2}} \tag{5.2}$$

The number of sigma for range rate at timestep *k*, $c_{\dot{r},k}$, is calculated by:

$$c_{\dot{r},k}^{2}=\frac{\left(\tilde{\dot{r}}_k-\bar{\dot{r}}_k\right)^2}{\sigma_{\tilde{\dot{r}},k}^{2}+\sigma_{\bar{\dot{r}},k}^{2}} \tag{5.3}$$

6 PDA based detection weighting

Like PDA (Probabilistic Data Association) (see e.g. [4]), the LAMU tracker uses all detections that fall inside the gate to form a "measurement" for the Kalman filter update. The normal Cartesian way of weighting detections is not very successful for sensors with large azimuth measurement errors. As can be seen from Figure 2.1, a weighting of range and azimuth would better represent the measurement errors. Thus, we use a statistical weighting for range, azimuth and range rate.

The well-known Gaussian probability for a general variable is given by the equation:

$$g\left(c,\sigma\right)=\frac{1}{\sqrt{2\pi}\cdot\sigma}e^{-\frac{c^2}{2}} \tag{6.1}$$

Where *c* is the number of standard deviations, and σ is the value of the standard deviation.

The weight for each detection is calculated according to the Gaussian probability of the difference between the range, azimuth and range rate of the detection and the predicted range, azimuth and range rate of the track. Thus, the weight w_i for detection number *i* is given by:

$$w_{i,k}=\frac{1}{\sqrt{\sigma_{\tilde{r}_i,k}^{2}+\sigma_{\bar{r}_i,k}^{2}}}e^{\left(\frac{c_{r_i,k}^{2}}{2}\right)}\cdot\frac{1}{\sqrt{\sigma_{\tilde{\theta}_i,k}^{2}+\sigma_{\bar{\theta}_i,k}^{2}}}e^{\left(\frac{c_{\theta_i,k}^{2}}{2}\right)}\cdot\frac{1}{\sqrt{\sigma_{\tilde{\dot{r}}_i,k}^{2}+\sigma_{\bar{\dot{r}}_i,k}^{2}}}e^{\left(\frac{c_{\dot{r}_i,k}^{2}}{2}\right)} \tag{6.2}$$

where $c_{r,k}^{2}$ is given by equation (5.1), $c_{\theta_i,k}^{2}$ is given by equation (5.2) and $c_{\dot{r}_i,k}^{2}$ is given by equation (5.3). Note that the term $\sqrt{2\pi}$ from equation (6.1) is not included in equation (6.2). The reason can be seen in equations (6.3)-(6.8), where the term $\left(\sqrt{2\pi}\right)^3$ would appear both in the nominator and the denominator, if $\sqrt{2\pi}$ was included in each Gaussian equation in (6.2).

The range of the resulting "measurement" is calculated as a weighted sum of the range of the detections within the gate, according to equation (6.3):

$$\tilde{r}_k = \frac{\sum_{i=1}^{n} \left(\tilde{r}_{i,k} \cdot w_{i,k} \right)}{\sum_{i=1}^{n} w_{i,k}} \qquad (6.3)$$

Similarly, the resulting azimuth is calculated as:

$$\tilde{\theta}_k = \frac{\sum_{i=1}^{n} \left(\tilde{\theta}_{i,k} \cdot w_{i,k} \right)}{\sum_{i=1}^{n} w_{i,k}} \qquad (6.4)$$

The resulting range rate is given by:

$$\tilde{\dot{r}}_k = \frac{\sum_{i=1}^{n} \left(\tilde{\dot{r}}_{i,k} \cdot w_{i,k} \right)}{\sum_{i=1}^{n} w_{i,k}} \qquad (6.5)$$

The standard deviations of the range, azimuth and range rate are calculated in the same way:

$$\sigma_{\tilde{r},k} = \frac{\sum_{i=1}^{n} \left(\sigma_{\tilde{r}_i,k} \cdot w_{i,k} \right)}{\sum_{i=1}^{n} w_{i,k}} \qquad (6.6)$$

$$\sigma_{\tilde{\theta},k} = \frac{\sum_{i=1}^{n} \left(\sigma_{\tilde{\theta}_i,k} \cdot w_{i,k} \right)}{\sum_{i=1}^{n} w_{i,k}} \qquad (6.7)$$

$$\sigma_{\tilde{\dot{r}},k} = \frac{\sum_{i=1}^{n} \left(\sigma_{\tilde{\dot{r}}_i,k} \cdot w_{i,k} \right)}{\sum_{i=1}^{n} w_{i,k}} \qquad (6.8)$$

7 Pruning

If more than one track seem to follow the same target, these tracks may be merged. This process, which is called pruning, is described in this chapter.

7.1 Pruning based on common detections

For pruning based on common detections, two tracks that have a certain number (tunable) of detections in common are merged to one track.

7.2 Pruning based on state vector and covariance matrix

The pruning may be based on both the state vector and the covariance matrix. For this pruning ("sigma pruning"), the difference between the two tracks A and B are calculated, based on the state vector and the covariance matrix for both tracks. The number of sigma for range is calculated by the following equation:

$$c_{r,k} = \sqrt{\frac{\left(\bar{r}_{A,k} - \bar{r}_{B,k}\right)^2}{\sigma^2_{\bar{r}_{A,k}} + \sigma^2_{\bar{r}_{B,k}}}} \tag{7.1}$$

The number of sigma for azimuth is calculated by:

$$c_{\theta,k} = \sqrt{\frac{\left(\bar{\theta}_{A,k} - \bar{\theta}_{B,k}\right)^2}{\sigma^2_{\bar{\theta}_{A,k}} + \sigma^2_{\bar{\theta}_{B,k}}}} \tag{7.2}$$

The number of sigma for range rate is calculated by:

$$c_{\dot{r},k} = \sqrt{\frac{\left(\bar{\dot{r}}_{A,k} - \bar{\dot{r}}_{B,k}\right)^2}{\sigma^2_{\bar{\dot{r}}_{A,k}} + \sigma^2_{\bar{\dot{r}}_{B,k}}}} \tag{7.3}$$

The number of sigma for azimuth rate is calculated by:

$$c_{\dot{\theta},k} = \sqrt{\frac{\left(\bar{\dot{\theta}}_{A,k} - \bar{\dot{\theta}}_{B,k}\right)^2}{\sigma^2_{\bar{\dot{\theta}}_{A,k}} + \sigma^2_{\bar{\dot{\theta}}_{B,k}}}} \tag{7.4}$$

If all four of these sigma values are below a certain (selectable) limit (e.g. 3 sigma), the two tracks are merged.

7.3 Which track is the surviving track?

When two tracks are merged, the resulting track may be one of the previous tracks or a mixture of them. In this version of the LAMU tracker, the resulting track is one of the previous tracks. The surviving track is chosen by one of the following methods:

- The oldest track will survive.
- The track that has used the highest number of detections will survive.
- The track with the highest possibility of being true will survive (one possible way to calculate this is described in Section 8).

8 Use of optimal smoothing to distinguish true and false tracks

If the measurement noise is relatively close to being white, one way to distinguish the true tracks from the false tracks is to find the effect spectre of the measurement noise. But to find the exact measurement error of each detection, we need to know the true track, which is never available in practical situations. Optimal smoothing (described in [3]) (which may be thought of as a "Kalman filter run backwards") can however be used to find the best estimate of the true track, with a given delay after real time.

By optimally smoothing the track, and then calculating the effect spectre of the difference between the measurements used by the track and the smoothed estimate of the track, the linear fit of this effect spectre may be used to distinguish between true and false tracks. For white noise, the effect spectre is flat, thus a true track will have an effect spectre that is hopefully close to a flat spectre, while a false track will often have an effect spectre which is high for low frequencies, and falling for higher frequencies.

9 Simulation results

In the simulations, we have used a sensor that measures range, azimuth and range rate, and has the following measurement uncertainties (which may represent an HF radar):

- Standard deviation in range: 2000 m
- Standard deviation in azimuth: 15 deg
- Standard deviation in range rate: 0.05 m/s

To compare the LAMU tracker with PDA, two PDA trackers are implemented and run with the same parameter settings as the LAMU tracker. Thus, the following three trackers are run on the same data sets:

- A Cartesian (original) PDA tracker
- A Cartesian PDA tracker with enlarged uncertainty areas (by increasing the radial axis of the ellipse, for both position and velocity measurements)
- The LAMU tracker

The object to be tracked is simulated as starting at 110 km north of the radar, and moving south with a speed of 10 m/s for 10 000 seconds (i.e. 2.7 hours).

Two simulations are presented; simulation A with no false detections, and simulation B where the number of detections is increased by 500% by adding false detection. In a real situation, the number of false detections may be somewhere in between these two simulations.

9.1 Simulation A

In this simulation, no false detections have been added.

9.1.1 Use of range rate measurement

For the Cartesian PDA, the range rate measurement is divided into measurements of velocity in the east and north directions, with uncertainties given by the rate range uncertainty and the azimuth uncertainty.

Figure 9.1 shows the measured velocity vector, together with the uncertainty areas used by the original PDA and by LAMU. Because of the small range rate uncertainty and the large azimuth uncertainty, the elliptic uncertainty area is not a good representation of the real uncertainty area. A large part of the ellipse is outside the real uncertainty area and vice versa.

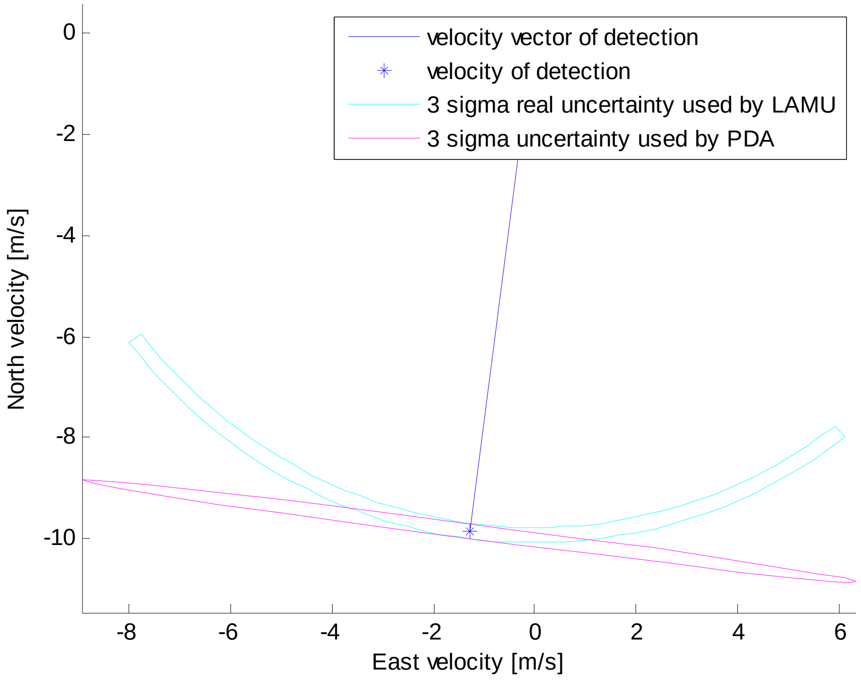

Figure 9.1 Use of range rate measurement for original PDA and for LAMU.

Figure 9.2 shows the same plot for the PDA tracker with enlarged uncertainty areas. A larger part of the real uncertainty area is seen to be covered by the elliptical uncertainty area than was seen in Figure 9.1, but at the cost of a higher range rate uncertainty. The LAMU tracker uses the real uncertainty area.

One way to increase the overlapping area between the ellipse and the real uncertainty area, is to move the center of the ellipse towards origo. This would give a more correct range uncertainty for detections with large azimuth errors, but the increased range rate uncertainty would be unchanged. We have not had time to simulate this type of PDA for this report.

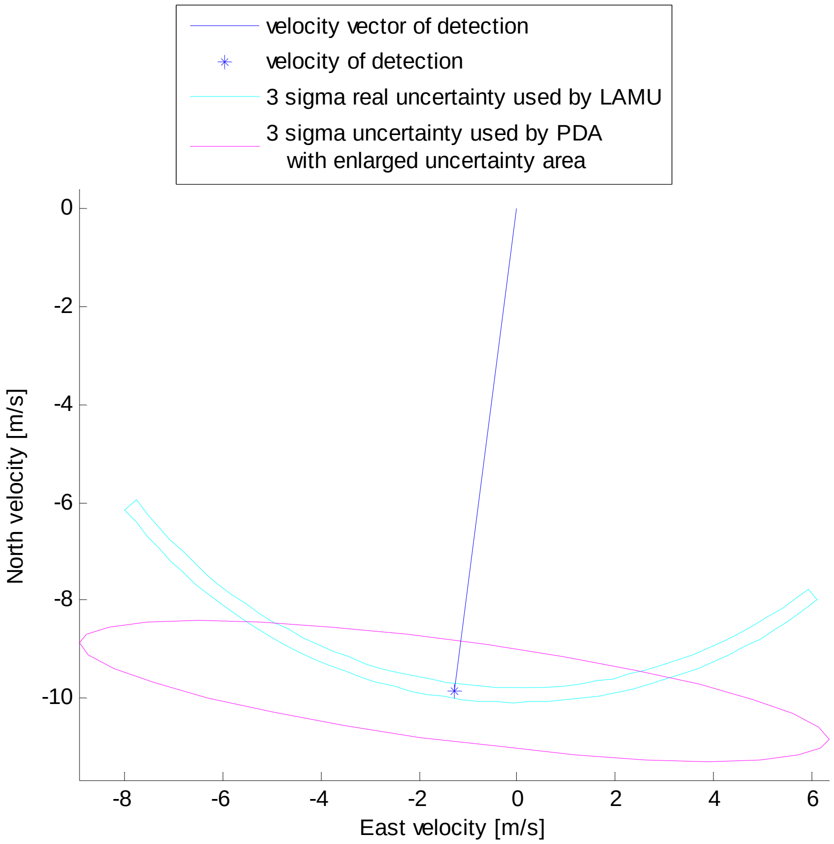

Figure 9.2 Use of range rate measurement for PDA with enlarged uncertainty area, and for LAMU.

9.1.2 Initialization

Three detections (at three different points in time) are used to initialize a new track. This section shows the initialization of a new track for the three different trackers. All trackers use the same three measurements, so the plots may be directly compared to each other.

Figure 9.3 shows the three detections (1, 2 and 3) together with their real uncertainty areas. The elliptical uncertainty areas used by the original PDA are plotted in pink. The resulting uncertainty area of the track with PDA is shown as a yellow filled ellipse. This uncertainty area is smaller than the true uncertainty, which is equal to the LAMU uncertainty area shown in Figure 9.5.

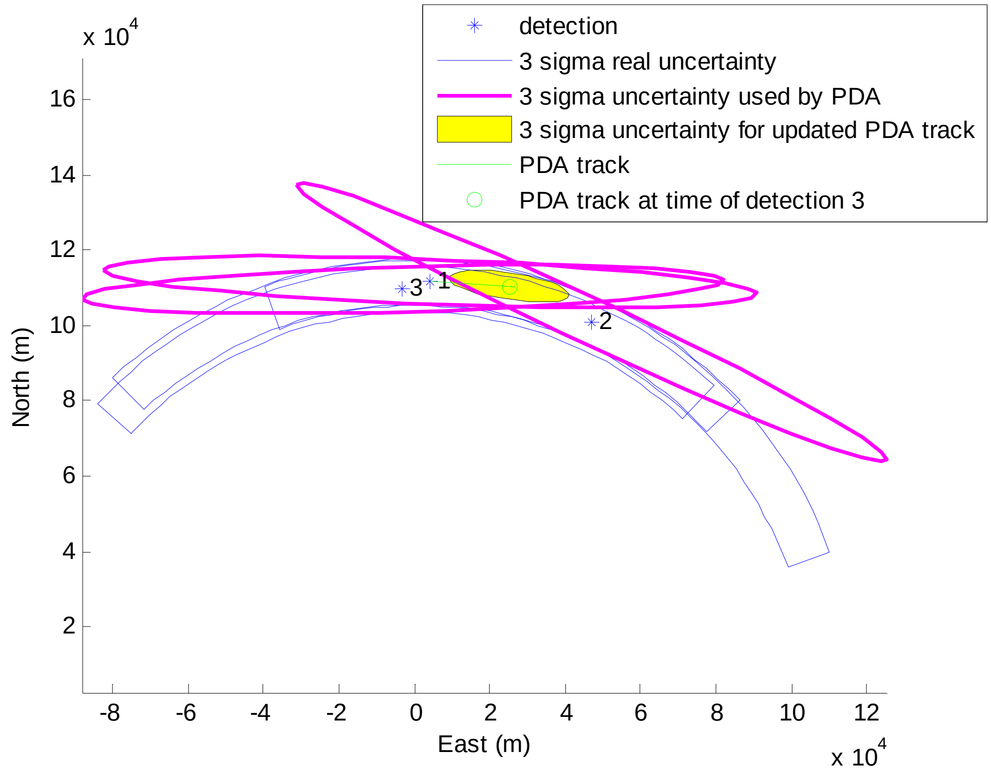

Figure 9.3 Initialization of a new track for original PDA.

Figure 9.4 shows the same plot for PDA with enlarged uncertainty areas. The pink ellipses are wider, and the resulting track uncertainty area is larger than for the original PDA:

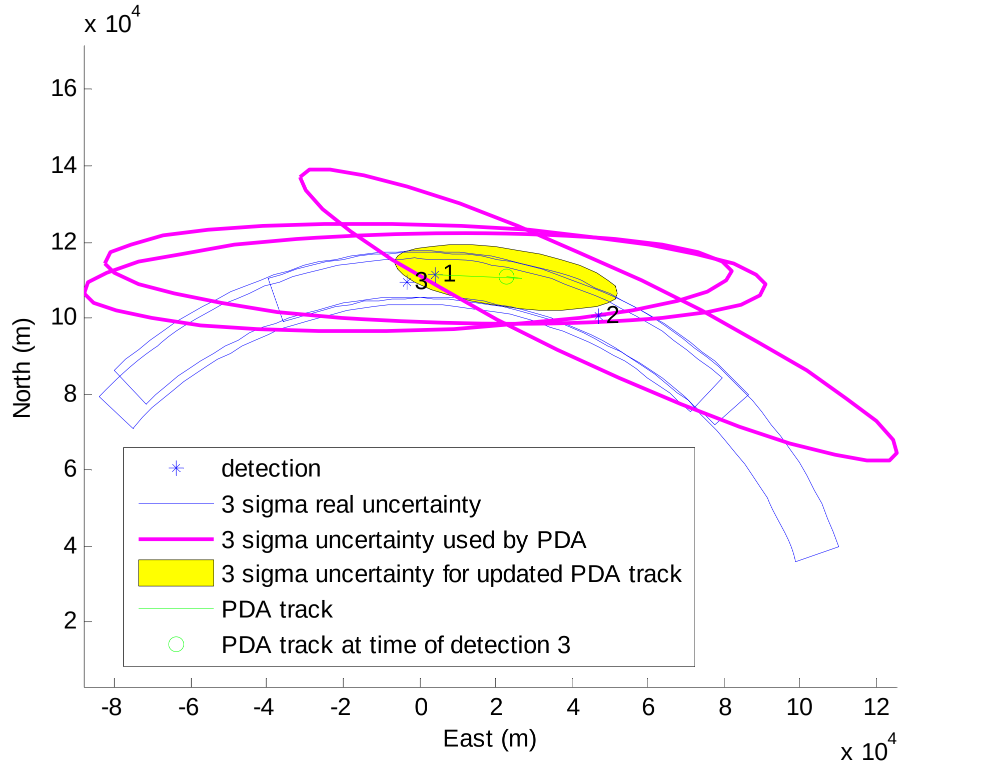

Figure 9.4 Initialization of a new track for PDA with enlarged uncertainty areas.

Figure 9.5 shows the track initialization for the LAMU tracker. The resulting uncertainty area for the new track is shown as a yellow broad arc. By comparing this with the previous two figures, the uncertainty area of the track from the original PDA is seen to have a correct width in the range direction, but a much too small width in the cross-range direction. The track from the PDA with enlarged uncertainty areas has a track uncertainty area that is a bit too wide in the range direction and a bit too small in the cross-range direction. The track from the LAMU tracker has the correct uncertainty area.

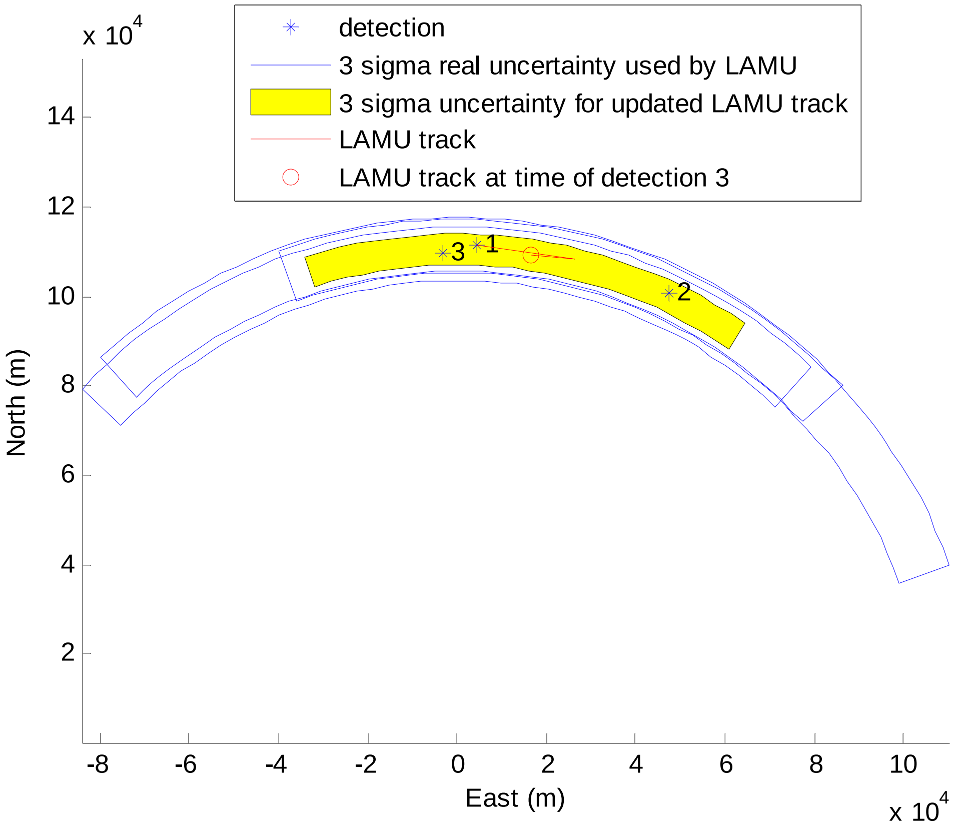

Figure 9.5   Initialization of a new track for LAMU.

9.1.3 Gating

After a track is initiated, gating is used for each measurement, to decide if the measurement belongs to the track or not. In simulation A, no false measurements were added, thus all measurements belong to the track. In this section, plots are shown for the 7th detection, because this is the first detection where the gating difference between the three algorithms is easy to demonstrate.

Figure 9.6 shows the predicted track position and the corresponding uncertainty area for the original PDA, together with the new detection and its uncertainty area. The predicted track uncertainty area is small, but still inside the real uncertainty area of the measurement. The pink ellipse shows the measurement uncertainty area used by the original PDA. No overlap is seen between the pink and green ellipses, hence for the original PDA, the measurement is regarded as being outside the 3 sigma gate.

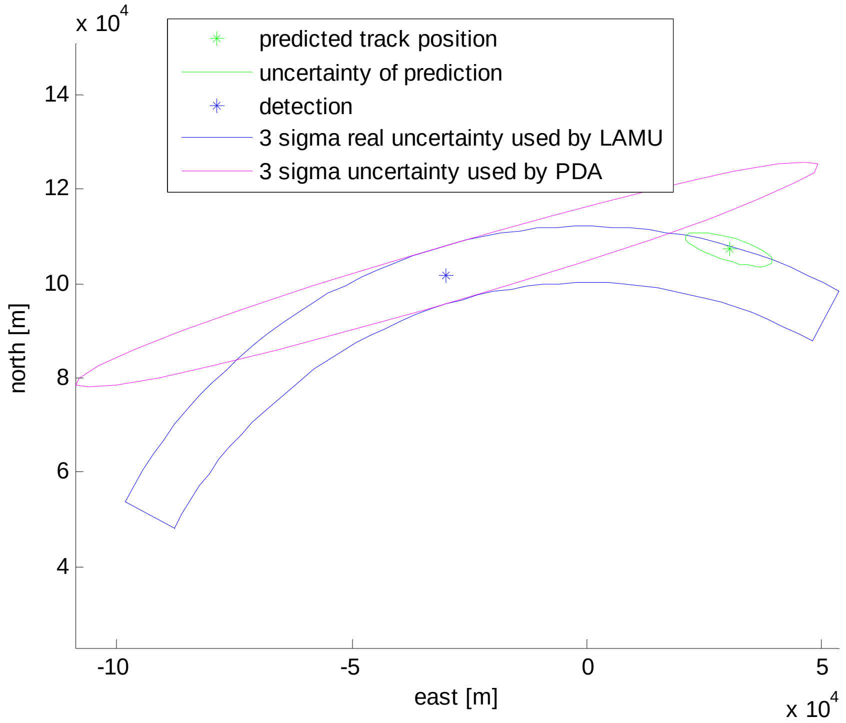

Figure 9.6 Prediction, a new measurement and their uncertainty areas for original PDA.

Figure 9.7 shows the same plot for PDA with enlarged uncertainty ellipses. Both the green and the pink ellipses are larger, and they overlap. Thus, the measurement is inside the gate for the PDA with enlarged uncertainty ellipses. If the measurement had had a larger azimuth error, it would have been outside the gate also for this tracker.

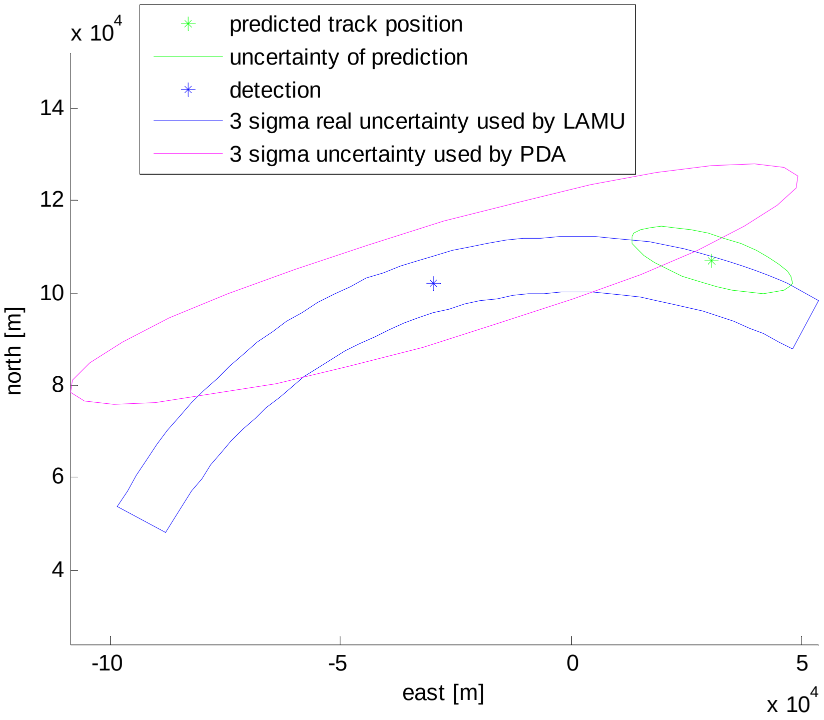

Figure 9.7 Prediction, a new measurement and their uncertainty areas for PDA with enlarged uncertainty areas.

Figure 9.8 shows the predicted track position and the corresponding uncertainty area for LAMU, together with the new detection and its uncertainty area. The overlapping area is very large, and only a small part of the track uncertainty area is outside the measurement uncertainty area. Thus, the measurement is well inside the 3 sigma gate.

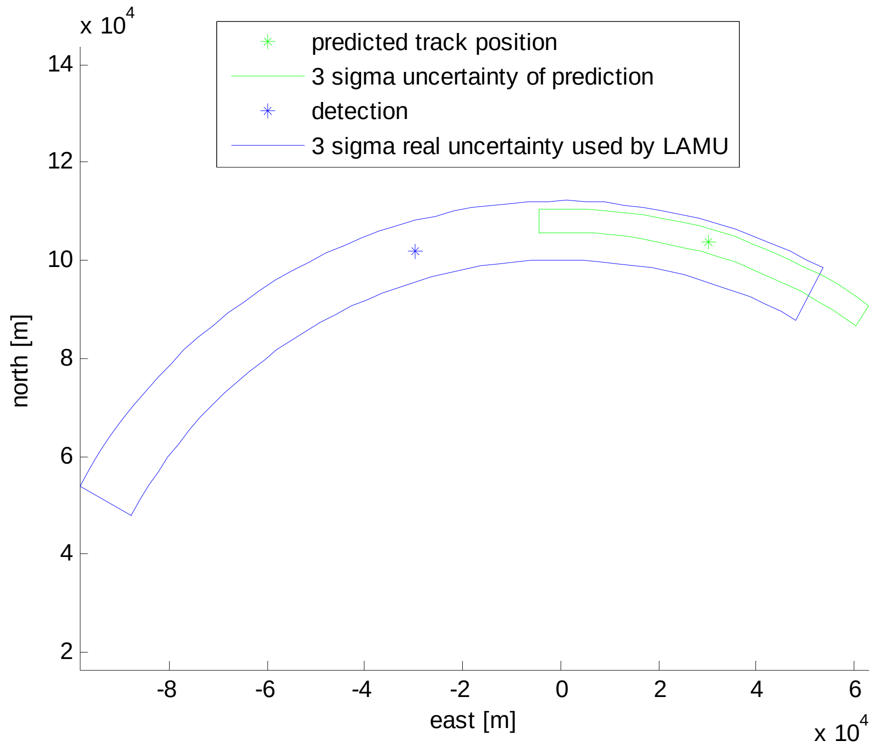

Figure 9.8 Prediction together with a new measurement and their uncertainty areas for LAMU.

9.1.4 Resulting tracks

Figure 9.9 shows the resulting tracks when using the original PDA. Three tracks are shown, and the track closest to the true track lasts for the duration of the true track.

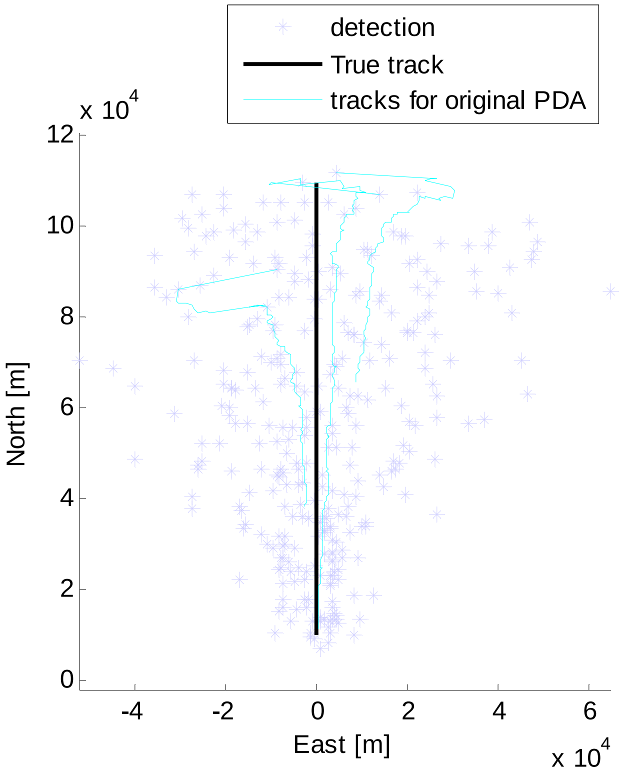

Figure 9.9 Tracking results for original PDA, for the entire time period for simulation A.

Figure 9.10 shows the same plot for PDA with enlarged uncertainty ellipses. Two tracks are seen here.

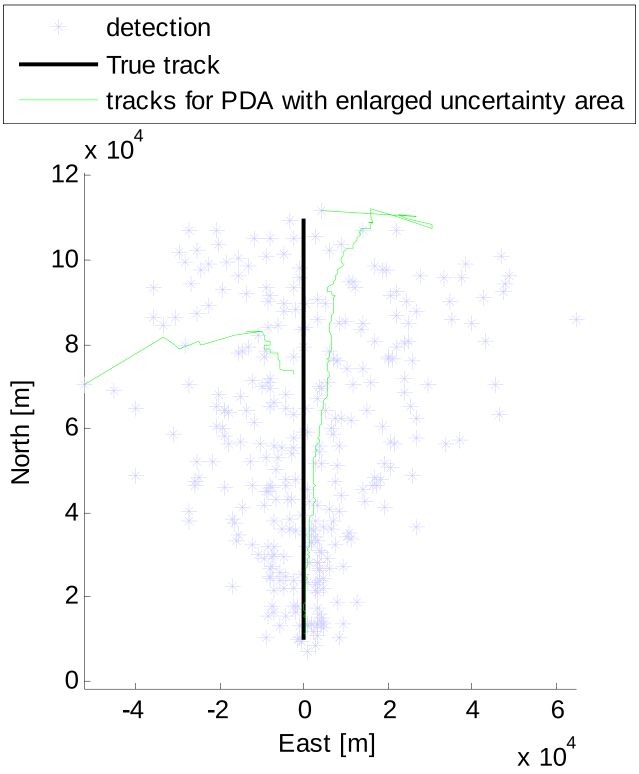

Figure 9.10 Tracking results for the entire time period for PDA with enlarged uncertainty areas for simulation A.

Figure 9.11 shows that with the LAMU tracker, only one track, following the true track, is made.

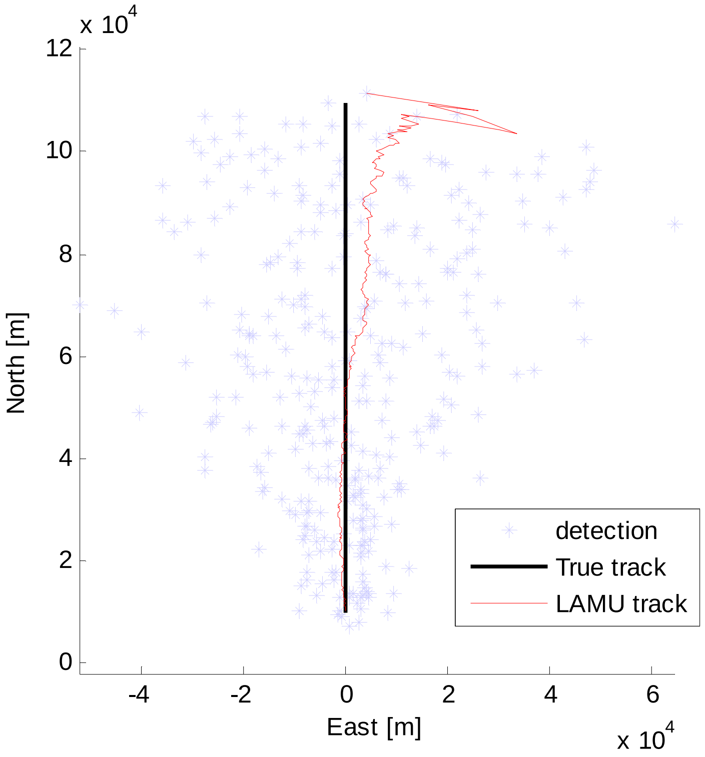

Figure 9.11 Tracking results for the entire time period for LAMU for simulation A.

Figure 9.12 shows the resulting tracks for all three trackers in the same plot.

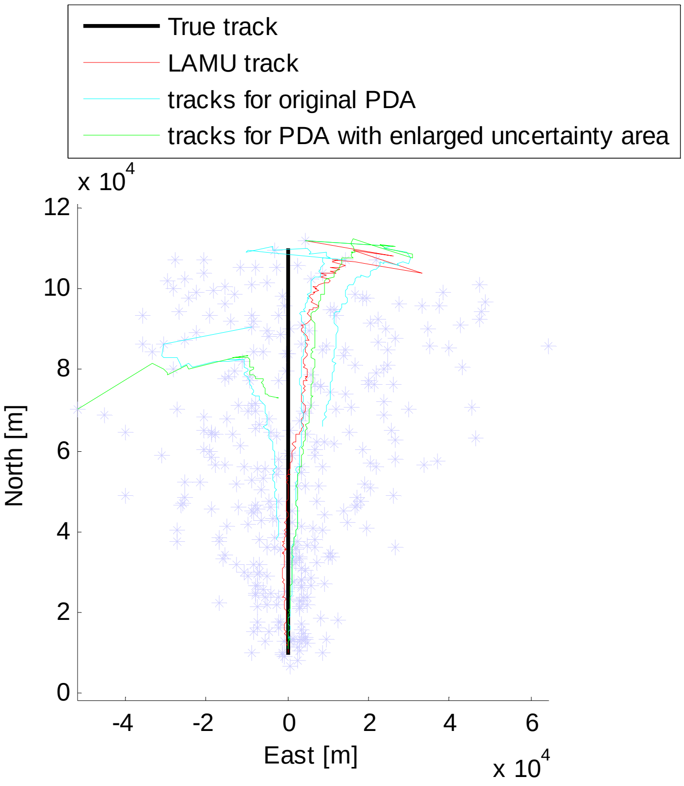

Figure 9.12 Tracking results for all three trackers for simulation A.

Figure 9.13 shows the range error (i.e. the difference between the range of each track and the range of the true track) as a function of time. The range errors of the two PDA trackers are seen to start out at about 4 km, and reduce to about 1 km at the end of the track. The range error of the LAMU tracker is kept below 0.3 km after the initial few detections, and is about 100 m at the end of the track.

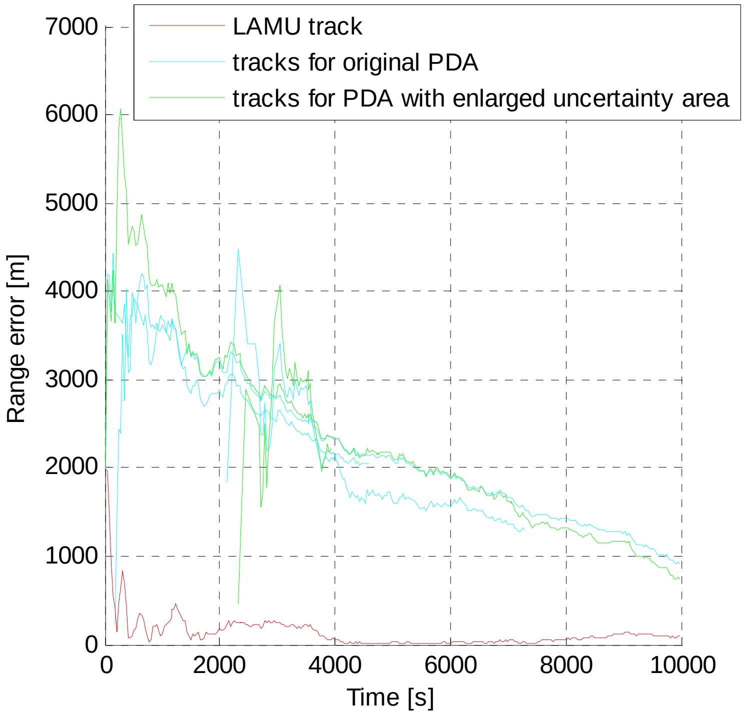

Figure 9.13 Range error as a function of time for the three trackers in simulation A.

Figure 9.14 shows azimuth as a function of time for the three trackers. The azimuth error of the LAMU tracker is not very different from the azimuth error of the best track from the PDA trackers.

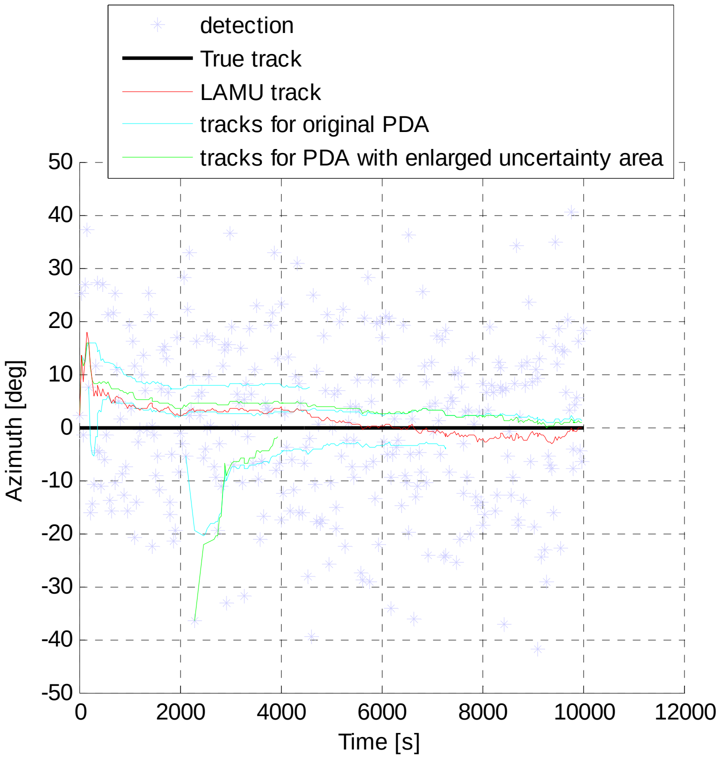

Figure 9.14 Azimuth as a function of time for the three trackers in simulation A.

Figure 9.15 shows the range rate as a function of time for the three trackers. The LAMU tracker uses the range rate directly as a measurement, and therefore has a very small range rate error. The PDA trackers have larger range rate errors, partly because uncertainty of the direction of the range rate measurement cannot be modelled correctly in a Cartesian model.

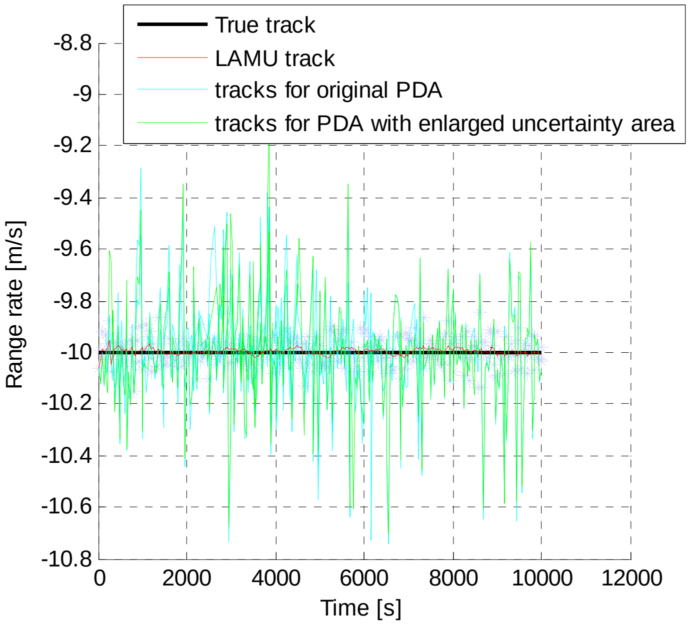

Figure 9.15 Range rate as a function of time for the three trackers in simulation A.

9.2 Simulation B

In this simulation, the number of detections is increased by 500% by adding false detections that are uniformly distributed in range, azimuth and range rate.

9.2.1 Pruning

The pruning of the different algorithms, i.e., the process of merging tracks if more than one track seem to follow the same target, is shown in this section.

The pruning difference between the algorithms could not be clearly observed in simulation A. Thus, pruning is shown here (with simulation B) instead. The pruning limit is set to 4 sigma in simulation B.

Figure 9.16 shows the pruning areas for the original PDA. Track 0 is the first (and oldest) track, and therefore has the smallest pruning area (which is set to 4 sigma in this simulation). Track 3 is a young track with a larger pruning area. The pruning areas (i.e. the 4 sigma uncertainty areas) of the tracks are not overlapping, thus the tracks will not be pruned.

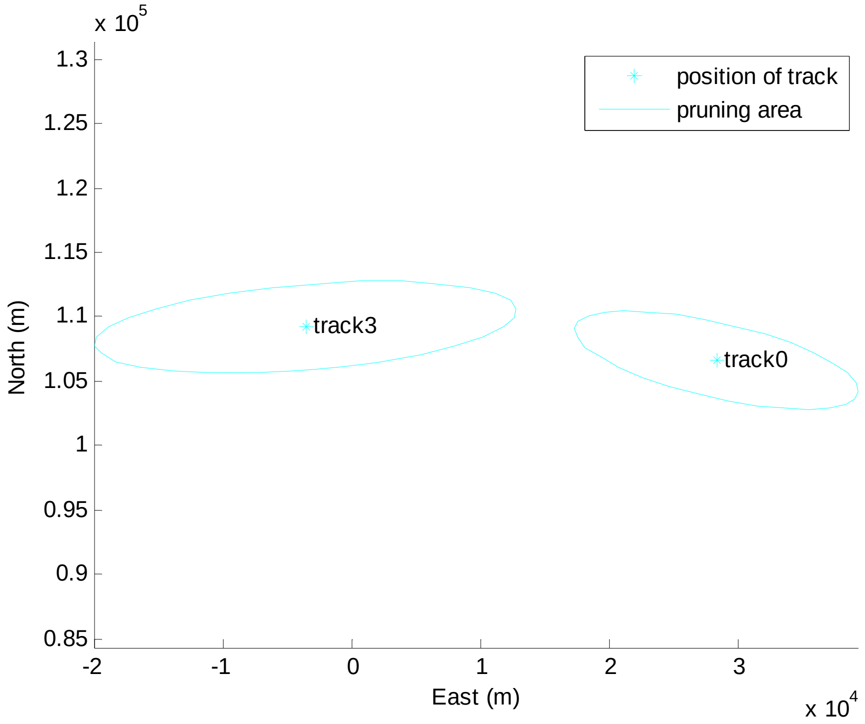

Figure 9.16 Pruning areas for original PDA for simulation B.

Figure 9.17 shows the pruning areas for PDA with enlarged uncertainty areas. Both ellipses are much larger than for the original PDA, and they overlap. Thus, the two tracks are pruned correctly.

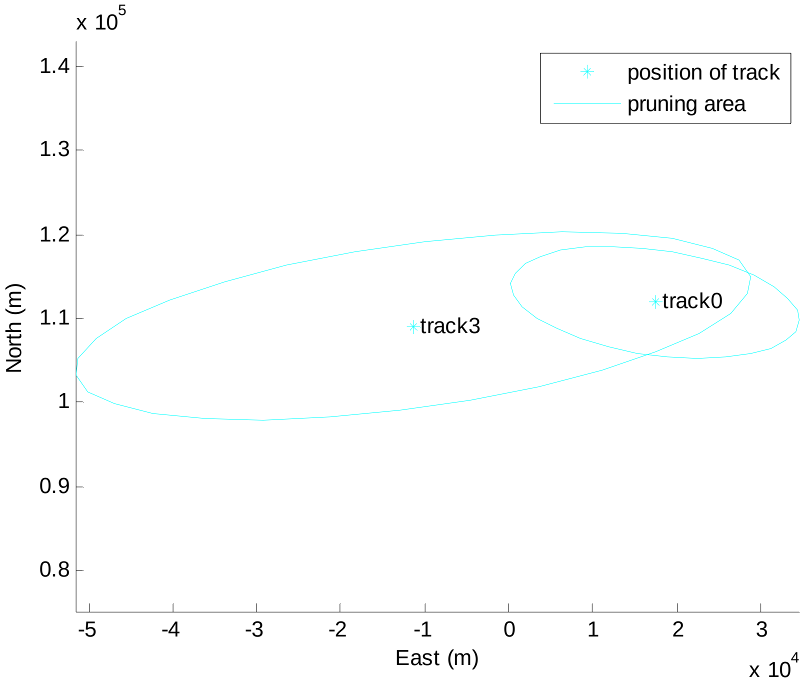

Figure 9.17 Pruning areas for PDA with increased uncertainty areas for simulation B.

Figure 9.18 shows the pruning areas for LAMU. Only a small part of the pruning area of track 0 is outside the pruning area of track 3.

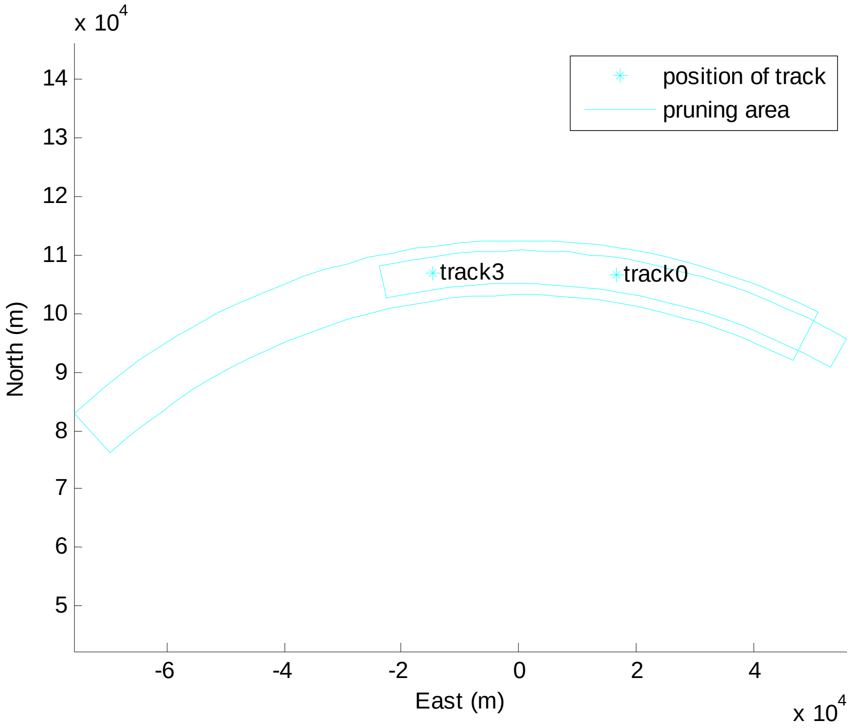

Figure 9.18 Pruning areas for LAMU for simulation B.

This section has shown that LAMU has the most correct pruning areas, and thus prunes the two tracks that are believed to represent the same object. PDA with enlarged ellipses also prunes the two tracks, but the overlapping area of the two pruning areas are smaller than for the LAMU tracker. The original PDA does not prune the two tracks, even though they belong to the same object, and this may lead to several false tracks.

9.2.2 Resulting tracks

Figure 9.19 shows the tracking results when using the original PDA. Two long tracks are seen to follow the true track, and several shorter tracks seem to be false tracks. Due to the large azimuth uncertainty, three detections with a large Cartesian difference may be used to establish a new track. This is the reason for the long line from the first to the second detection (or from the second to the third detection) that is shown for some tracks.

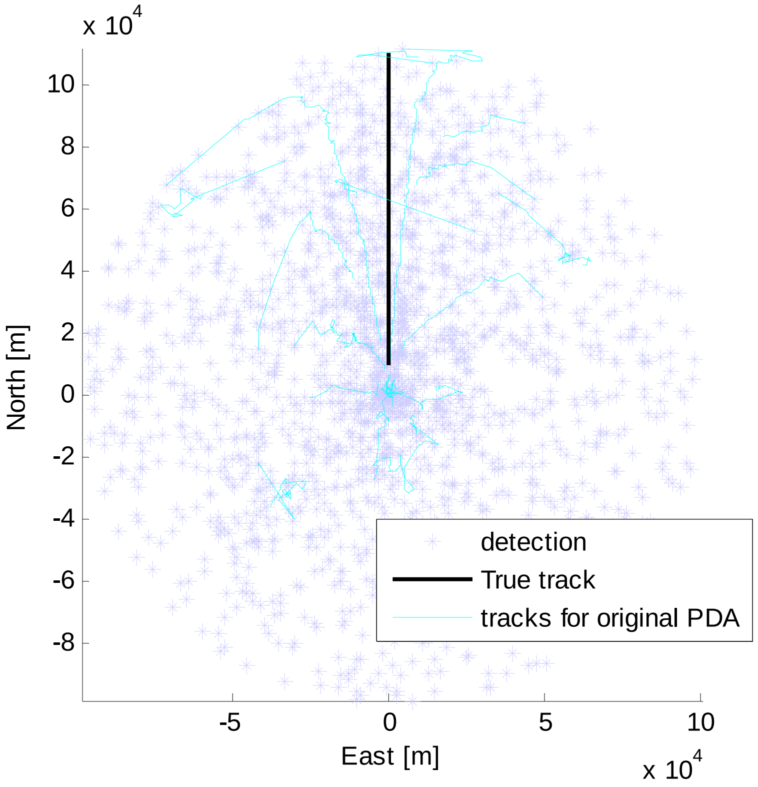

Figure 9.19 Detections and tracking results for original PDA, for simulation B.

Figure 9.20 shows the tracking results for PDA with enlarged uncertainty ellipses. Two long tracks are seen to follow the true track and several relatively short false tracks are found. The number of false tracks is larger than for the original PDA. One reason for this is that increased uncertainty ellipses leads to increased sizes of the gates, and thus an increased number of wrongly initiated tracks and an increased number of track updates from false detections.

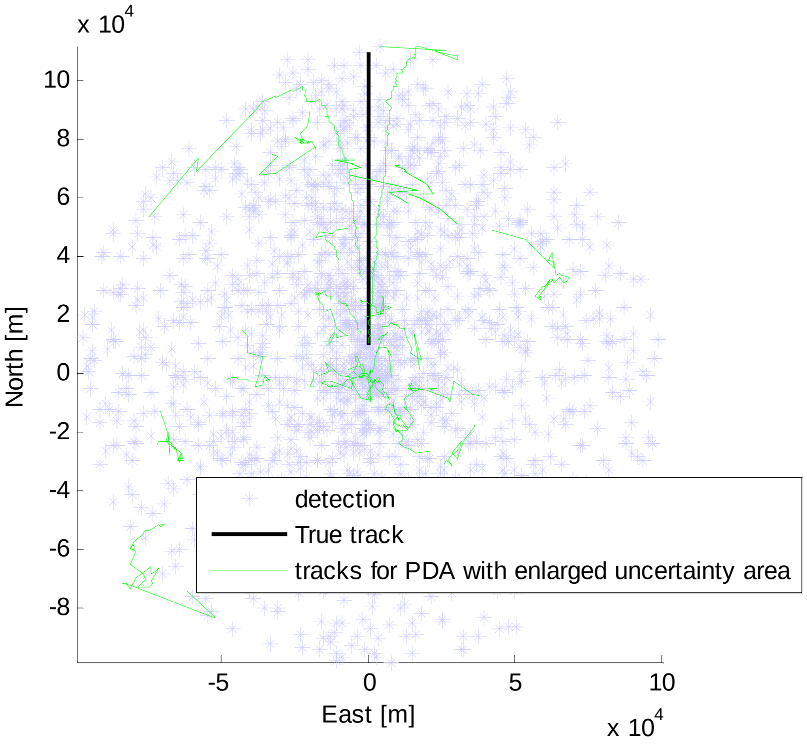

Figure 9.20 Detections and tracking results for PDA with enlarged uncertainty areas, for simulation B.

Figure 9.21 shows the results for the LAMU tracker. One track is seen to follow the true track, and a few false tracks are seen close to the origin, where the false detection density is very high (since the false detections are uniformly distributed in range, azimuth and range rate).

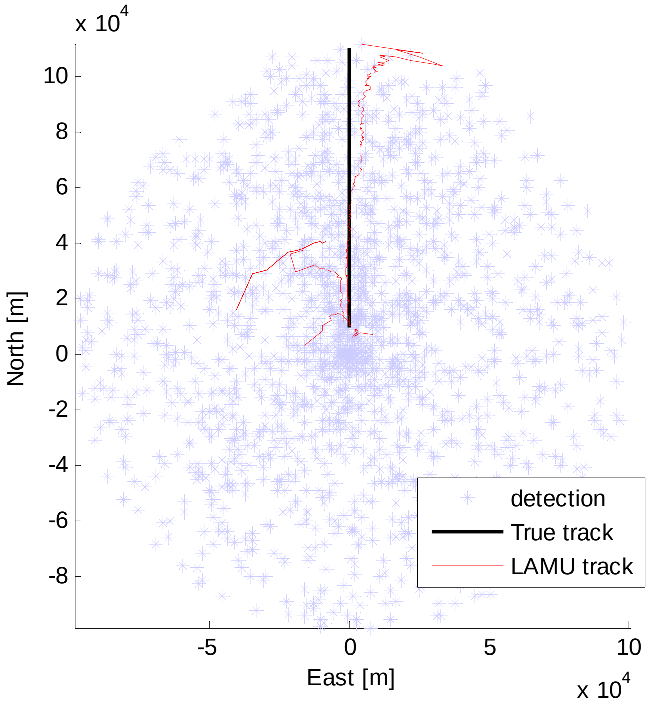

Figure 9.21 Detections and tracking results for LAMU, for simulation B.

Figure 9.22 shows range as a function of time for all detections (true and false) and all tracks for the three trackers. Both PDA trackers have many false tracks with a range far from the range of the true track, and also several tracks with a range close to the true track range. The LAMU tracker has one track close to the true track and a few false tracks.

The false tracks from the PDA trackers have a relatively constant range. This is probably due to the uniformly distribution in range rate of the false detections. Even though each tracks is initiated as having a velocity only in the range direction (equal to the measured range rate), the shape of the uncertainty ellipses will soon allow detections with a different range rate to be used to update the track.

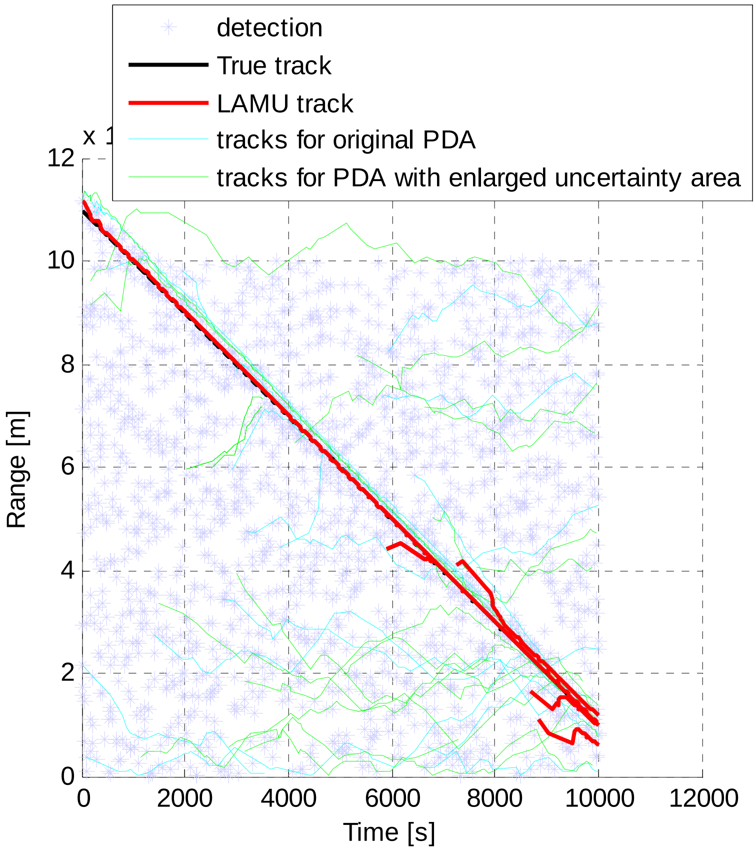

Figure 9.22 Range as a function of time for detections and for tracks for all three trackers

The range error is shown in Figure 9.13 as a function of time. The range errors of the two PDA trackers are large for many tracks which are based on false detections. The LAMU tracker also has a few false tracks, which are relatively close to the true track.

Many tracks are seen with an initial very large range error that is decreasing with time. These tracks are positioned close to the origin where there is a high density of false tracks, and the range error decreases when the true track gets closer to the origin. Other tracks have an error that first decreases, and then increases. These tracks have a relatively constant range that is somewhere between the maximum and minimum range of the true track. The error decreases as the range of the true track decreases, and increases after the true track range gets smaller than the range of the false track. A few tracks have an increasing error. These tracks have a large range, and the range error increases when the true track range decreases.

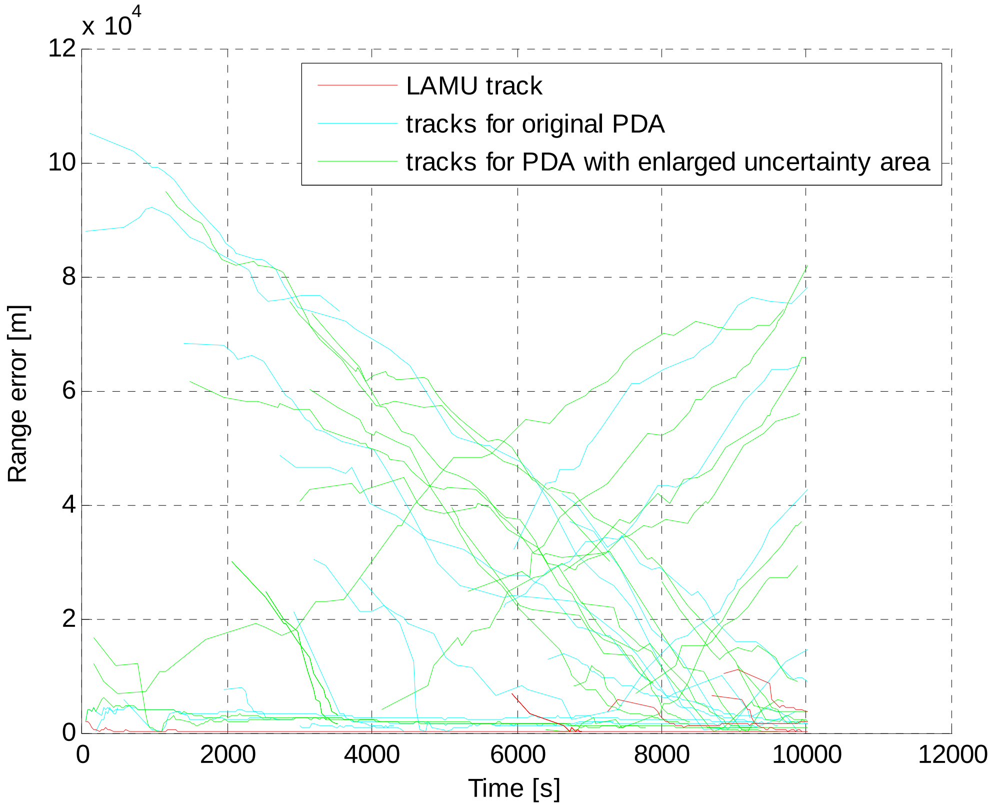

Figure 9.23 Range error as a function of time for the three trackers in simulation B.

Figure 9.24 shows azimuth as a function of time for detections and for tracks for all three trackers. Some of the tracks have a slowly changing azimuth, probably because they have a low total speed, and thus also a low azimuth speed. Some other tracks are positioned at about 180 degrees, and thus show an azimuth value that changes between -180 degrees and 180 degrees. A few tracks have an azimuth value that changes very fast. These tracks are probably very close to the origin, and the large change of azimuth may thus be due to a relatively small change in position.

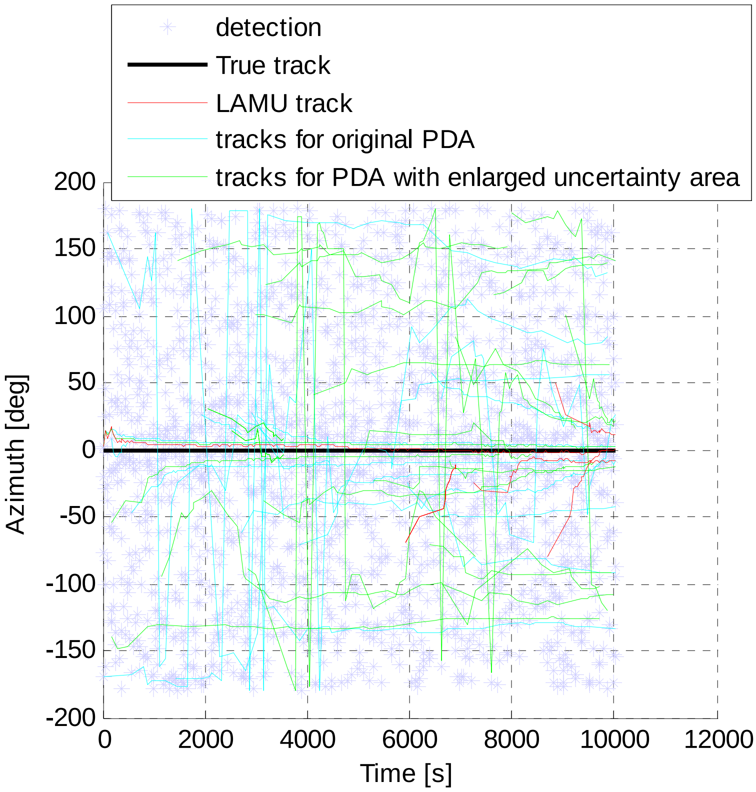

Figure 9.24 Azimuth as a function of time for detections and for tracks for the three trackers

Figure 9.25 shows range rate as a function of time for detections and for tracks for all three trackers. As explained in Section 4.1, detections with a range rate above max_velocity (or below − max_velocity) are not used to establish new tracks. In the simulations in this report, max_velocity is set to 15 m/s. This is the reason why the range rate of the false detections in simulation B is uniformly distributed between -15 and 15 m/s.

Some of the tracks from the original PDA and many tracks from the PDA with enlarged uncertainty ellipses have large and sudden changed in range rate. This is due to the difference between the elliptic uncertainty ellipses used by PDA and the real uncertainty area, as shown in Figure 9.1 and Figure 9.2. For large azimuth measurement errors, the ellipses used for the range rate measurement will not cover the real measurement, thus we have a rate measurement where the true rate is outside of the uncertainty area of the rate measurement. This will lead to large range rate errors for the track.

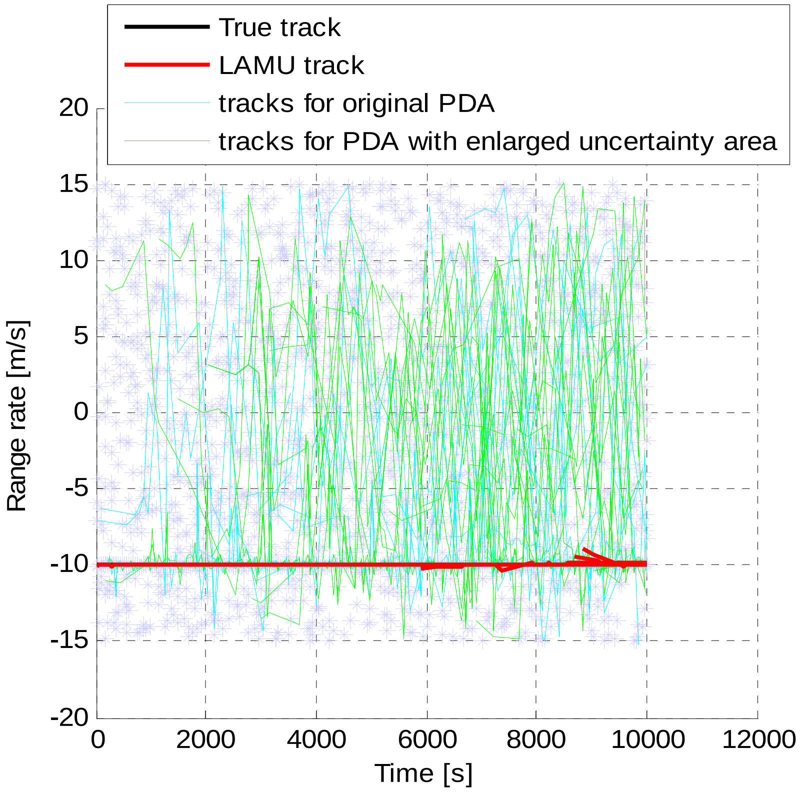

Figure 9.25 Range rate as a function of time for detections and for tracks for all three trackers.

10 Conclusion

The LAMU tracker uses the real uncertainty areas of the measurements, while the PDA trackers use elliptical uncertainty areas. The following advantages of using the real uncertainty areas have been shown in simulations:

- The range error and range rate error of the LAMU track is much lower than the corresponding errors of the PDA tracker.
- In a simulation without false measurements, the LAMU tracker has no false tracks and one track close to the true track, while the PDA trackers have false tracks.
- The LAMU tracker is more robust than PDA in a simulation with many false detections. In this simulation, the LAMU tracker has few false tracks, while the PDA trackers have many false tracks.

11 Future work

The LAMU tracker in this report is based on the principles of PDA. MHT (Multi Hypothesis Tracking) is the theoretically optimal tracker. Thus, a LAMU tracker based on MHT will probably give even better results.

Appendix A Acronyms

2D	Two Dimensional
3D	Three Dimensional
CPU	Central Processing Unit
HF	High Frequency
KF	Kalman Filter
LAMU	Large Angular Measurement Uncertainty
MHT	Multi Hypothesis Tracker
NED	North-East-Down
PDA	Probabilistic Data Association

Appendix B Mathematical calculations

B.1 Extra noise term due to non-linear state space model

The state space equation (3.20) cannot be directly calculated, since we do not have exact knowledge of r, $\dot{r}$ or $\dot{\theta}$.

What we do have is the Kalman filter updated state vector:

$$\hat{\boldsymbol{x}} = \boldsymbol{x} + \delta \boldsymbol{x} \tag{B.1}$$

In this report, the errors (especially the angular error) may be large, and thus we need to model the uncertainty.

Therefore, we write the state space equation (3.19) as:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\hat{\boldsymbol{x}}(t),t\right) - \delta \boldsymbol{f}\left(\hat{\boldsymbol{x}}(t),t\right) + \boldsymbol{v}_{orig}(t) \tag{B.2}$$

where:

$$\delta \boldsymbol{f}\left(\hat{\boldsymbol{x}}(t),t\right) = \boldsymbol{f}\left(\hat{\boldsymbol{x}}(t),t\right) - \boldsymbol{f}\left(\boldsymbol{x}(t),t\right) \tag{B.3}$$

is an extra noise term that is due to the error of the updated states.

After a Kalman filter update, we have the following estimates of the states in the state vector:

$$\hat{r} = r + \delta r \tag{B.4}$$

$$\hat{\theta} = \theta + \delta\theta \tag{B.5}$$

$$\hat{\dot{r}} = \dot{r} + \delta\dot{r} \tag{B.6}$$

$$\hat{\dot{\theta}} = \dot{\theta} + \delta\dot{\theta} \tag{B.7}$$

The delta variables in the above equations represent the difference between the true value and the updated value (from the Kalman filter update).

By using equations (B.4), (B.5) and (B.6), equation (3.22) can be written as:

$$\boldsymbol{f}\left(\boldsymbol{x}(t)\right) = \begin{bmatrix} \hat{\dot{r}} - \delta\dot{r} \\ \hat{\dot{\theta}} - \delta\dot{\theta} \\ \left(\hat{r} - \delta r\right)\left(\hat{\dot{\theta}} - \delta\dot{\theta}\right)^2 \\ -2\left(\hat{\dot{\theta}} - \delta\dot{\theta}\right)\dfrac{\hat{\dot{r}} - \delta\dot{r}}{\hat{r} - \delta r} \end{bmatrix} \tag{B.8}$$

$$\Downarrow$$

$$\boldsymbol{f}\left(\boldsymbol{x}(t)\right) = \begin{bmatrix} \hat{\dot{r}} - \delta\dot{r} \\ \hat{\dot{\theta}} - \delta\dot{\theta} \\ \left(\hat{r} - \delta r\right)\left(\hat{\dot{\theta}}^2 - 2\hat{\dot{\theta}}\delta\dot{\theta} + \left(\delta\dot{\theta}\right)^2\right) \\ -2\dfrac{\hat{\dot{r}}\hat{\dot{\theta}} - \hat{\dot{r}}\delta\dot{\theta} - \hat{\dot{\theta}}\delta\dot{r} + \delta\dot{r}\delta\dot{\theta}}{\hat{r} - \delta r} \end{bmatrix} \tag{B.9}$$

By using the connection:

$$\begin{aligned} \frac{1}{\hat{r} - \delta r} &= \frac{1}{\hat{r}\left(1 - \dfrac{\delta r}{\hat{r}}\right)} \\ &= \frac{1}{\hat{r}} + \frac{\dfrac{\delta r}{\hat{r}^2}}{\left(1 - \dfrac{\delta r}{\hat{r}}\right)} \\ &= \frac{1}{\hat{r}} + \frac{\delta r}{\hat{r}^2 - \hat{r}\delta r} \end{aligned} \tag{B.10}$$

equation (B.9) becomes:

$$\boldsymbol{f}\left(\boldsymbol{x}(t)\right)=\begin{bmatrix} \hat{\dot{r}}-\delta\dot{r} \\ \hat{\dot{\theta}}-\delta\dot{\theta} \\ \hat{\dot{\theta}}^2\hat{r}-\hat{\dot{\theta}}^2\delta r-\hat{r}2\hat{\dot{\theta}}\delta\dot{\theta}+2\hat{\dot{\theta}}\delta\dot{\theta}\delta r+\hat{r}\left(\delta\dot{\theta}\right)^2-\delta r\left(\delta\dot{\theta}\right)^2 \\ -2\left(\hat{\dot{r}}\hat{\dot{\theta}}-\hat{\dot{r}}\delta\dot{\theta}-\hat{\dot{\theta}}\delta\dot{r}+\delta\dot{r}\delta\dot{\theta}\right)\left(\frac{1}{\hat{r}}+\frac{\delta r}{\hat{r}^2-\hat{r}\delta r}\right) \end{bmatrix} \tag{B.11}$$

$$\Downarrow$$

$$\boldsymbol{f}\left(\boldsymbol{x}(t)\right)=\begin{bmatrix} \hat{\dot{r}}-\delta\dot{r} \\ \hat{\dot{\theta}}-\delta\dot{\theta} \\ \hat{\dot{\theta}}^2\hat{r}-\hat{\dot{\theta}}^2\delta r-2\hat{r}\hat{\dot{\theta}}\delta\dot{\theta}+2\hat{\dot{\theta}}\delta\dot{\theta}\delta r+\hat{r}\left(\delta\dot{\theta}\right)^2-\delta r\left(\delta\dot{\theta}\right)^2 \\ \frac{-2\left(\hat{\dot{r}}\hat{\dot{\theta}}-\hat{\dot{r}}\delta\dot{\theta}-\hat{\dot{\theta}}\delta\dot{r}+\delta\dot{r}\delta\dot{\theta}\right)}{\hat{r}}+\frac{-2\left(\hat{\dot{r}}\hat{\dot{\theta}}-\hat{\dot{r}}\delta\dot{\theta}-\hat{\dot{\theta}}\delta\dot{r}+\delta\dot{r}\delta\dot{\theta}\right)\delta r}{\hat{r}^2-\hat{r}\delta r} \end{bmatrix} \tag{B.12}$$

$$\Downarrow$$

$$\boldsymbol{f}\left(\boldsymbol{x}(t)\right)=\begin{bmatrix} \hat{\dot{r}} \\ \hat{\dot{\theta}} \\ \hat{\dot{\theta}}^2\hat{r} \\ \frac{-2\hat{\dot{r}}\hat{\dot{\theta}}}{\hat{r}} \end{bmatrix}+\begin{bmatrix} -\delta\dot{r} \\ -\delta\dot{\theta} \\ -\hat{\dot{\theta}}^2\delta r-2\hat{r}\hat{\dot{\theta}}\delta\dot{\theta}+2\hat{\dot{\theta}}\delta\dot{\theta}\delta r+\hat{r}\left(\delta\dot{\theta}\right)^2-\delta r\left(\delta\dot{\theta}\right)^2 \\ 2\frac{\left(\hat{\dot{r}}\delta\dot{\theta}+\hat{\dot{\theta}}\delta\dot{r}-\delta\dot{r}\delta\dot{\theta}\right)\left(\hat{r}-\delta r\right)}{\hat{r}\left(\hat{r}-\delta r\right)}+2\frac{\left(-\hat{\dot{r}}\hat{\dot{\theta}}\delta r+\hat{\dot{r}}\delta r\delta\dot{\theta}+\hat{\dot{\theta}}\delta r\delta\dot{r}-\delta r\delta\dot{r}\delta\dot{\theta}\right)}{\hat{r}\left(\hat{r}-\delta r\right)} \end{bmatrix} \tag{B.13}$$

By combining equations (B.3) and (B.13), we can see that the extra noise term is:

$$-\delta\boldsymbol{f}\left(\hat{\boldsymbol{x}}(t)\right)=\begin{bmatrix} -\delta\dot{r} \\ -\delta\dot{\theta} \\ -\hat{\dot{\theta}}^2\delta r-2\hat{r}\hat{\dot{\theta}}\delta\dot{\theta}+2\hat{\dot{\theta}}\delta\dot{\theta}\delta r+\hat{r}\left(\delta\dot{\theta}\right)^2-\delta r\left(\delta\dot{\theta}\right)^2 \\ 2\frac{\left(\hat{\dot{r}}\delta\dot{\theta}+\hat{\dot{\theta}}\delta\dot{r}-\delta\dot{r}\delta\dot{\theta}\right)\left(\hat{r}-\delta r\right)}{\hat{r}\left(\hat{r}-\delta r\right)}+2\frac{\left(-\hat{\dot{r}}\hat{\dot{\theta}}\delta r+\hat{\dot{r}}\delta r\delta\dot{\theta}+\hat{\dot{\theta}}\delta r\delta\dot{r}-\delta r\delta\dot{r}\delta\dot{\theta}\right)}{\hat{r}\left(\hat{r}-\delta r\right)} \end{bmatrix} \tag{B.14}$$

$$\Downarrow$$

$$-\delta \boldsymbol{f}\left(\hat{\boldsymbol{x}}(t)\right)=$$
$$\begin{bmatrix} -\delta\dot{r} \\ -\delta\dot{\theta} \\ -\hat{\dot{\theta}}^2\delta r-2\hat{r}\hat{\dot{\theta}}\delta\dot{\theta}+2\hat{\dot{\theta}}\delta\dot{\theta}\delta r+\hat{r}\left(\delta\dot{\theta}\right)^2-\delta r\left(\delta\dot{\theta}\right)^2 \\ 2\dfrac{\hat{r}\hat{\dot{r}}\delta\dot{\theta}-\hat{\dot{r}}\delta\dot{\theta}\delta r+\hat{r}\hat{\dot{\theta}}\delta\dot{r}-\hat{\dot{\theta}}\delta\dot{r}\delta r-\hat{r}\delta\dot{r}\delta\dot{\theta}+\delta r\delta\dot{r}\delta\dot{\theta}-\hat{\dot{r}}\hat{\dot{\theta}}\delta r+\hat{\dot{r}}\delta r\delta\dot{\theta}+\hat{\dot{\theta}}\delta r\delta\dot{r}-\delta r\delta\dot{r}\delta\dot{\theta}}{\hat{r}\left(\hat{r}-\delta r\right)} \end{bmatrix} \tag{B.15}$$

$$\Downarrow$$

$$-\delta \boldsymbol{f}\left(\hat{\boldsymbol{x}}(t)\right)=\begin{bmatrix} -\delta\dot{r} \\ -\delta\dot{\theta} \\ -\hat{\dot{\theta}}^2\delta r-2\hat{r}\hat{\dot{\theta}}\delta\dot{\theta}+2\hat{\dot{\theta}}\delta\dot{\theta}\delta r+\hat{r}\left(\delta\dot{\theta}\right)^2-\delta r\left(\delta\dot{\theta}\right)^2 \\ 2\dfrac{-\hat{\dot{r}}\hat{\dot{\theta}}\delta r+\hat{r}\hat{\dot{\theta}}\delta\dot{r}+\hat{r}\hat{\dot{r}}\delta\dot{\theta}-\hat{r}\delta\dot{r}\delta\dot{\theta}}{\hat{r}\left(\hat{r}-\delta r\right)} \end{bmatrix} \tag{B.16}$$

$$\Downarrow$$

$$-\delta \boldsymbol{f}\left(\hat{\boldsymbol{x}}(t)\right)=\begin{bmatrix} -\delta\dot{r} \\ -\delta\dot{\theta} \\ -\hat{\dot{\theta}}^2\delta r-2\hat{r}\hat{\dot{\theta}}\delta\dot{\theta}+2\hat{\dot{\theta}}\delta\dot{\theta}\delta r+\hat{r}\left(\delta\dot{\theta}\right)^2-\delta r\left(\delta\dot{\theta}\right)^2 \\ 2\dfrac{-\frac{\hat{\dot{r}}}{\hat{r}}\hat{\dot{\theta}}\delta r+\hat{\dot{\theta}}\delta\dot{r}+\hat{\dot{r}}\delta\dot{\theta}-\delta\dot{r}\delta\dot{\theta}}{\left(\hat{r}-\delta r\right)} \end{bmatrix} \tag{B.17}$$

By using a first order approximation, the equation (B.17) can be written as:

$$-\delta \boldsymbol{f}\left(\hat{\boldsymbol{x}}(t)\right)=\begin{bmatrix} -\delta\dot{r} \\ -\delta\dot{\theta} \\ -\hat{\dot{\theta}}^2\delta r-2\hat{r}\hat{\dot{\theta}}\delta\dot{\theta} \\ 2\dfrac{-\frac{\hat{\dot{r}}}{\hat{r}}\hat{\dot{\theta}}\delta r+\hat{\dot{\theta}}\delta\dot{r}+\hat{\dot{r}}\delta\dot{\theta}}{\left(\hat{r}-\delta r\right)} \end{bmatrix} \tag{B.18}$$

The total process noise vector of our system is thus:

$$\boldsymbol{v}=-\delta \boldsymbol{f}\left(\hat{\boldsymbol{x}}(t)\right)+v_{orig}=\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}=\begin{bmatrix} -\delta\dot{r} \\ -\delta\dot{\theta} \\ -\hat{\dot{\theta}}^2\delta r-2\hat{r}\hat{\dot{\theta}}\delta\dot{\theta}+v_{\ddot{r}} \\ 2\dfrac{-\frac{\hat{\dot{r}}}{\hat{r}}\hat{\dot{\theta}}\delta r+\hat{\dot{\theta}}\delta\dot{r}+\hat{\dot{r}}\delta\dot{\theta}}{\left(\hat{r}-\delta r\right)}+v_{\ddot{\theta}} \end{bmatrix} \tag{B.19}$$

By combining the equation (B.18) with equation (3.34), we see that:

$$\delta \boldsymbol{f}\left(\hat{\boldsymbol{x}}\right) = \boldsymbol{A}\left(\hat{\boldsymbol{x}}\right) \cdot \delta \boldsymbol{x} \tag{B.20}$$

where:

$$\delta \boldsymbol{x} = \begin{bmatrix} \delta r \\ \delta \theta \\ \delta \dot{r} \\ \delta \dot{\theta} \end{bmatrix} \tag{B.21}$$

The covariance matrix of the total process noise vector can thus be calculated as (we assume that the elements of $\boldsymbol{v}_{total}$ are uncorrelated):

$$\boldsymbol{Q}_{total} = \begin{bmatrix} {\sigma_{v_1}}^2 & 0 & 0 & 0 \\ 0 & {\sigma_{v_2}}^2 & 0 & 0 \\ 0 & 0 & {\sigma_{v_3}}^2 & 0 \\ 0 & 0 & 0 & {\sigma_{v_4}}^2 \end{bmatrix} \tag{B.22}$$

The standard deviation of the first noise element is:

$$\sigma_{v_1} = \sigma_{\dot{r}} \tag{B.23}$$

where $\sigma_{\dot{r}}$ is given by the covariance matrix of the updated state vector. The standard deviation of the second noise element is:

$$\sigma_{v_1} = \sigma_{\dot{\theta}} \tag{B.24}$$

where $\sigma_{\dot{\theta}}$ is given by the covariance matrix of the updated state vector. The standard deviation of the third noise element is:

$$\begin{aligned} \sigma_{v_3} &= \sqrt{E\left(\left(-\hat{\dot{\theta}}^2 \delta r - 2\hat{r}\hat{\dot{\theta}}\delta\dot{\theta} + v_{\ddot{r}}\right)^2\right)} \\ &\approx \hat{\dot{\theta}}^2 \sigma_r + 2\hat{r}\hat{\dot{\theta}}\sigma_{\dot{\theta}} + \sigma_{v_{\ddot{r}}} \end{aligned} \tag{B.25}$$

where $\sigma_{v_{\ddot{r}}}$ is given by equation (3.29). Note that in the above equation, the (Kalman filter updated) full states are inserted as numbers, and thus are not affected by the expectation operator *E*.

The standard deviation of the fourth noise element is:

$$\begin{aligned}\sigma_{v_4} &= \sqrt{E\left(\left(2\frac{\frac{\hat{\dot{r}}}{\hat{r}}\hat{\dot{\theta}}\delta r+\hat{\dot{\theta}}\delta\dot{r}+\hat{\dot{r}}\delta\dot{\theta}}{\left(\hat{r}-\delta r\right)}+v_{\ddot{\theta}}\right)^2\right)} \\ &\approx \frac{2}{\hat{r}}\left(\frac{\hat{\dot{r}}}{\hat{r}}\hat{\dot{\theta}}\sigma_r+\hat{\dot{\theta}}\sigma_{\dot{r}}+\hat{\dot{r}}\sigma_{\dot{\theta}}\right)+\sigma_{v_{\ddot{\theta}}}\end{aligned} \tag{B.26}$$

where $\sigma_{v_{\ddot{\theta}}}$ is given by equation (3.30). As with the previous equation, note that in the above equation, the (Kalman filter updated) full states are inserted as numbers, and thus are not affected by the expectation operator *E*.

Another way to represent the extra noise term is by using a linearized ***A***-matrix in the covariance prediction equation, as described in Section 3.3.2.

References

[1] B. H. H. Gade, A. L. Gjersøe, and T. Johnsen, "Evaluation of Codar HF radar dataset and tracking performance - Results for LAMU (Large Angular Measurement Uncertainty) Tracker and IMMPDAF Tracker," FFI (Norwegian Research Defence Establishment),FFI/RAPPORT-2006/03575, 2007.

[2] K. Gade, "NAVLAB - Overview and User Guide November 2003," FFI (Norwegian Research Defence Establishment),FFI-RAPPORT 2003/02128, 2003.

[3] A. Gelb, "Applied Optimal Estimation," The M.I.T. press, 1992.

[4] Y. Bar-Shalom and W. D. Blair, *Multitarget-Multisensor Tracking: Applications and Advances Volume III*. London: Artech House, 2000.