A new approach to distributed data fusion

de Waard, H.W.

Citation for published version (APA):
Chapter 7

Artificial measurements

This chapter introduces the concept of artificial measurements to reduce the load on communication channels and platform processing resources by reducing the amount of information which is exchanged between participants. The method starts with calculating so-called tracklets for the different local tracks. Using tracklets directly, reduction of information is only obtained when the number of measurements is ≥ 5. To obtain information reduction for a smaller number of measurements, the tracklets are transformed to artificial measurements using a suitable manoeuvre detector to determine the reduction factor dynamically. Using a set of representative scenarios, it is shown that tracks maintained with artificial measurements are operationally equivalent with tracks maintained with the original measurements. Furthermore, experiments show that it is sufficient to use a constant high reduction factor for linear target trajectories and a constant low reduction factor for circular trajectories.

7.1 Introduction

In section 1.4 the different basic architectures have been identified, which are traditionally used to realize task force data fusion systems. Each of the identified architectures can produce single integrated air pictures (SIAP’s) containing high quality composite tracks. The quality of the composite tracks is expressed in terms of track accuracy, track continuity and track number consistency. In two of the four architectures all the (correlated) measurements are distributed (section 1.4.2 and 1.4.3), which requires the availability of communication channels with high bandwidth. There are situations, where this requirement is not fulfilled. In the two architectures, described in section 1.4.1 and 1.4.4, the different participants exchange (composite) track information (state vector, partial error covariance matrix or Kalman gain). This means that the data transfer requirements may even be higher than for the case that all correlated measurements are exchanged.

Furthermore, the data association problem for a number of data sets ≥ 3, coming from single or multiple sensors, is NP-hard ([51], [112]). In the application domain of interest, it is of the utmost importance that targets are detected
as soon as possible and appropriate measures are taken. For realtime applications, it is required that the data association problem is solved in a few seconds (realtime requirement, section 1.5). Normally, in order to meet these requirements sub-optimal algorithms are used to solve the data association problem.

Many sensors provide only tracks as output and do not provide observations. In two of the identified architectures, the (composite) tracks are distributed. Normally, this means that after a track update, the estimated state vector, a part of the estimated error covariance matrix or the Kalman gain (section 2.3) have to be distributed. Compared with the distribution of (correlated) measurements, the amount of data which has to be distributed is significantly higher. Due to the limited capacity of the available communication channels, the risk of communication delays is also higher.

Most digital data is not stored in the most compact form possible, but in easy-to-use formats [110]. Data compression is the general term to reduce the space needed to store digital data. A compression program is used to convert data from an easy-to-use format to one optimized for compactness. Likewise, a decompression program returns the information to its original form. Data compression techniques found in literature can be divided into two classes

- **lossless techniques**
  Using a lossless technique means that the restored data is identical to the original data. This is absolutely necessary for many types of data, for example: executable code, word processing files, tabulated numbers, etc.

- **lossy techniques**
  In comparison, data files that represent images and other acquired signals do not have to be kept in perfect condition for storage or transmission. All real world measurements inherently contain a certain amount of noise. If the changes made to the signals resemble a small amount of additional noise no harm is done. Compression techniques that allow this type of degradation are called lossy. This distinction is important because lossy techniques are much more effective at compression than lossless methods.

In this thesis it is assumed that the data to be exchanged between participants has been compressed, using one of the lossless or lossy compression techniques. Still, the resulting compression will probably not be sufficient due to the fact that the application domain considered in this thesis is the military domain, in which a large number of communication systems are based on decades-old technology and have a very limited bandwidth. Furthermore, in complex scenarios, where a large number of targets in clutter have to be tracked, the amount of information to be exchanged between participants will be so large that even for state-of-the-art communication systems there will be a risk of unacceptable communication delays.

Compression techniques will reduce the amount of information to be exchanged, but not the number of received measurements/tracks which have to be processed. The necessary computation time for solving the data association problem will be increased slightly due to the need to decompress the received data. The objective of this chapter is to answer the question, if it is possible to either use an existing method or to develop a new method to further reduce/compress the amount of information which has to be distributed and to significantly reduce the amount of computation time necessary to solve the
related data association problem, without significantly reducing the quality of the composite tracks produced by the different contributors to the task force.

An often-heard argument to choose for the distribution of tracks instead of measurements is that the data transfer requirements are lower than when all correlated measurements are exchanged. However, the distribution of tracks induces a new problem: to combine track information track-to-track fusion methods are necessary. In section 7.2 the principles of the different track-to-track fusion methods found in literature and proposed in this thesis are discussed; the question whether certain methods have indeed the advantage of reducing the amount of information to be distributed will be answered. An evaluation of the different methods is given in section 7.3. Since all evaluated methods have certain disadvantages, the new method of artificial measurements which can be used to fuse track information while reducing the amount of information to be distributed is proposed in section 7.4. Section 7.5 discussed the concept of dynamic data reduction. Finally, section 7.6 discussed the application of data reduction when there are communication bandwidth or processing power limitations.

7.2 Track-to-track fusion

In literature only two methods are known to fuse track information. The first method, described in [14, 11, 12, 33], directly combines the track information and will be discussed in section 7.2.1. The other method is based on transforming the track information in the different local track databases to equivalent measurements or tracklets, which can be processed like normal sensor measurements [52, 54]. This approach furthermore provides for the possibility to reduce the amount of track information, which has the advantage that both the load on the communication channels and the platform processing load is reduced. A disadvantage is that there is a larger time interval between track updates. Due to the fact that the first equivalent measurement or tracklet is only based on the first local track state vector and error covariance matrix, there is no delay time introduced with respect to composite track initiation. A number of tracklet approaches is discussed in the sections 7.2.2-7.2.6.

7.2.1 Track fusion

Assume that participant $j$ maintains track $j$. The time of the latest track update is $t_j$. After some time, track $i$ with track update time $t_i$ is received from participant $i$, where $t_j \leq t_i$. First the estimated track state vector $\hat{s}_j$ is predicted to time $t_i$, resulting in the predicted state vector $\bar{s}_j$. The track state vector estimated by platform $i$ at time $t_i$ is $\hat{s}_i$. The corresponding covariance matrices are denoted as $\hat{P}_i$ and $\bar{P}_j$. To determine if the two track hypotheses correspond with the same target, the difference between the two time-aligned state vectors is used

$$\Delta_{ij} = \hat{s}_i - \bar{s}_j$$

(7.1)

The covariance matrix, corresponding with eq. 7.1, is defined by

$$P_{\Delta_{ij}} = E(\Delta_{ij} \times \Delta_{ij}^T)$$

$$= \hat{P}_i + \bar{P}_j - P_{ij} - P_{ji}$$

(7.2)
where
\[ \hat{P}_i = E(\vec{s} - \hat{s}_i) \times (\vec{s} - \hat{s}_i)^T \]
\[ \bar{P}_j = E(\vec{s} - \hat{s}_j) \times (\vec{s} - \hat{s}_j)^T \]
\[ P_{ij} = E(\vec{s} - \hat{s}_i) \times (\vec{s} - \hat{s}_j)^T \]
\[ P_{ji} = E(\vec{s} - \hat{s}_j) \times (\vec{s} - \hat{s}_i)^T \]

\( P_{ij} = P_{ji}^T \) and \( \vec{s} \) is the assumed, but unknown real state vector. The cross-correlation term \( P_{ij} \) can be determined, using the recursion equation (15) from [11].

Assume that \( \Delta_{ij} \) follows a Gaussian distribution. This means that the test statistic
\[ \rho^2 = \Delta_{ij}^T \times (P_{\Delta_{ij}})^{-1} \times \Delta_{ij} \]
follows a \( \chi^2 \)-distribution. If
\[ \rho^2 \leq r^2 \]
the hypothesis that both track hypotheses correspond with the same target is accepted, otherwise the hypothesis is rejected. Using the dimension of the difference vector and accepting a certain risk to make a wrong decision, the threshold \( r^2 \) can be read from the \( \chi^2 \)-distribution table. If the hypothesis is accepted, the state estimate fusion equation becomes [14]
\[ \hat{s} = \hat{s}_i + (\hat{P}_i - P_{ij}) \times (P_{\Delta_{ij}})^{-1} \times (\vec{s}_j - \hat{s}_i) \]

The corresponding error covariance matrix is given by
\[ M = \hat{P}_i - (\hat{P}_i - P_{ij}) \times (P_{\Delta_{ij}})^{-1} \times (\hat{P}_i - P_{ij})^T \]

Due to the fact that for each track hypothesis update at least the estimated state vector and the Kalman gain have to be distributed (section 7.1), the data transfer requirements may even be higher than when all (correlated) measurements are distributed (section 7.3.2). A complication is that it can easily occur that a platform composite track is partially based on the same information as the received (composite) track (section 1.4.1). In that case, the usual assumption of error independence from one update period to another is not valid. Furthermore, it is not possible to correlate and filter the received track information without removing, in a mathematically correct but expensive way, the possible data dependency between a platform composite track hypothesis and the received (composite) track hypothesis for the same object.

### 7.2.2 Track decorrelation

A tracklet \(^1\) results by converting an estimated target state vector and error covariance matrix to remove the cross-correlation between sensor-level tracks and composite tracks and includes only the information from several measurements received after the last time a tracklet for the same track has been calculated [25, 53]. A tracklet is calculated for a number of measurements in such a way that the estimated track using tracklets is the same with only a very limited reduction in accuracy as for an estimate based on measurements and the tracklet

---

\(^1\)The term tracklet has been introduced by Drummond [54].
7.2. TRACK-TO-TRACK FUSION

Errors are not cross-correlated with the errors of any other previously calculated tracklet for the same target [65]. Crosscorrelation due to process noise is neglected. A tracklet is equivalent to a track for a target based on only the most recent measurements since the last time a tracklet has been calculated. Typically a tracklet contains a full state vector estimate, and the corresponding error covariance matrix. The use of tracklets provides the possibility to reduce the amount of information which has to be exchanged between participants. Some delay is introduced, but there is only a limited loss of track accuracy. In Fig. 7.1 an example is shown where the information contained in seven measurements assigned to the track hypothesis is compressed to four equivalent measurements.

It is assumed that with each measurement update a state vector and a covariance matrix are produced and distributed. The resulting composite track is initiated with the first tracklet which is based only on the first measurement. The amount of information to be distributed is not necessarily reduced, however.

![Figure 7.1: The principle of equivalent measurements.](image)

There are a variety of ways to calculate tracklets [52, 53, 54]:

1. Inverse Kalman filter (section 7.2.3)
2. Decorrelated state (section 7.2.4)
3. Inverse information filter (section 7.2.5)
4. Periodic calculation (section 7.2.6)

The main difference between these methods is that a mathematically different approach is taken to guarantee that the tracklet errors are not cross-correlated with errors of previously calculated tracklets, resulting in a mathematically different solution for the tracklet calculation. Due to the difference in the number of mathematical operations, the main difference between the first three methods is that the corresponding implementations on the same computer differ in computation time. Drummond [52] has shown that the decorrelated state approach is equivalent to the inverse Kalman filter and in section 7.2.5 it is shown that the inverse Kalman filter is equivalent to the inverse information filter.
In section 6.13 it will be shown, that the first three methods have the major
disadvantage that during the calculation of a tracklet very easily numerical
problems may occur. The advantage of the periodic calculation method, which
is a new method, is that it is possible to guarantee there will occur no numerical
problems. For all of the enumerated methods reduction of information is only
obtained when the number of measurements \( \geq 5 \).

7.2.3 Inverse Kalman filter

This method has been described by Drummond [53]. Consider the tracklet \( \vec{u}_n \) as
the unknown measurement which at time \( t_n \) is responsible for the extended
Kalman filter update from prediction \( \bar{s}_n \) to estimated state vector \( \hat{s}_n \), where
the prediction has been determined for the last updated state vector determined at
time \( t_k \). In the time interval \([t_k, t_n]\) more than one real measurement has been
processed. The tracklet measurement model at time \( t_n \) is given by

\[
\vec{u}_n = \vec{s}_n + \vec{v}_n
\]  

(7.8)

where \( \vec{v}_n \) is the measurement error and \( \vec{s}_n \) represents the real state of the object
to be tracked. The measurement error is assumed to be a white sequence [14],
with \( E(\vec{v}_n) = 0 \) and covariance matrix

\[
E(\vec{v}_n \times \vec{v}_k^T) = R_n \quad n = k \\
= 0 \quad n \neq k
\]  

(7.9)

The second line in the equation expresses the assumption that the measurement
error \( \vec{v}_n \) at time \( t_n \) is not correlated with the measurement error \( \vec{v}_k \) at time \( t_k \)
(\( t_n \neq t_k \)). The nonlinear evolution law of the state vector is defined by

\[
\vec{s}_n = f_k(\vec{s}_k) + \vec{w}_k
\]  

(7.10)

The process noise \( \vec{w}_k \) is also assumed to be a white sequence with \( E(\vec{w}_k) = 0 \) and
covariance matrix

\[
E(\vec{w}_j \times \vec{w}_k^T) = Q_j \quad j = k \\
= 0 \quad j \neq k
\]  

(7.11)

Furthermore, it is assumed that the covariance matrix \( E(\vec{w}_k \vec{v}_i) = 0 \) for all \( k \) and
\( i \). The correlation between the tracklet errors due to process noise is neglected.

The unknown tracklet \( \vec{u}_n \) and covariance matrix \( U_n \) have to fulfil the extended
first order Kalman filter update equations (section 2.3)

\[
\dot{s}_n = \ddot{s}_n + K_n \times (\vec{u}_n - \ddot{s}_n) \quad (7.12) \\
\dot{P}_n = \bar{P}_n - \bar{P}_n \times (\bar{P}_n + U_n)^{-1} \times \bar{P}_n \quad (7.13) \\
K_n = \bar{P}_n U_n^{-1} = \bar{P}_n (\bar{P}_n + U_n)^{-1} \quad (7.14)
\]

with \( \ddot{s}_n = f_k(\ddot{s}_k) \). Here \( \dot{s}_n \) and \( \ddot{s}_n \) are the estimated and predicted track state
vector, and \( \bar{P}_n \) and \( \bar{P}_n \) are the estimated and predicted covariance matrix. In
case of tracklets, the Jacobian \( H_n \) given in eq. 2.6 and 2.7 is equal to the unity
where the tracklet is estimated with

\[
\bar{u}_n = \bar{s}_n + A_n \times (\hat{s}_n - \bar{s}_n) \tag{7.15}
\]

\[
U_n = \hat{P}_n(\hat{P}_n - \hat{P}_n)^{-1} \hat{P}_n - \hat{P}_n \tag{7.16}
\]

\[
A_n = K_n^{-1} = (\hat{P}_n + U_n)\hat{P}_n^{-1} \tag{7.17}
\]

The inverse \(K_n^{-1}\) does only exist if \(K_n\) is square and non-singular. When the tracklet is based on only one measurement, the tracklet reduces to a measurement and the introduced measurement model, given by eq. 7.8, for the tracklet is not valid anymore and changes to \(\tilde{u}_k = h_k(\bar{s}_k) + \bar{v}_k\), where \(h_k\) represents the ideal connection between measurement and state vector. The consequence is that in that case equation 7.12 is not correct anymore and has to be changed to

\[
\hat{s}_n = \bar{s}_n + K_n \times (\bar{u}_n - h_k(\bar{s}_n)) \tag{7.18}
\]

Furthermore, in equation 7.18 the gain-matrix \(K_n\) is rectangular which means that the inverse \(K_n^{-1}\) does not exist. Due to the fact that in that case the matrix \(A_n\) (eq. 7.17) does not exist, the conclusion is that the tracklet equations cannot be applied after each track update, but only when \(K_n\) is square and non-singular.

### 7.2.4 Decorrelated state

An alternative method to determine a tracklet is proposed in [65]. Assume that \(\bar{s}_k\) (dim(\(\bar{s}_k\)) = \(n\)) is the state of the target estimated at measurement time \(t_k\) and that \(\hat{s}_n\) is the state of the same target estimated at measurement time \(t_n\) (\(t_n > t_k\)). The corresponding error covariance matrices are given by \(\hat{P}_k\) and \(\hat{P}_n\). The prediction of the state \(\hat{s}_k\) to time \(t_n\) is given by \(\hat{s}_n\) and the predicted error covariance matrix is given by \(\hat{P}_n\). Assuming the use of an extended first order Kalman filter and neglecting process noise, Frenkel [65] proves that

\[
\text{cov}(\hat{s}_n, \bar{s}_n) = E(\hat{s}_n - \bar{s}_n)(\hat{s}_n - \bar{s}_n)^T = \hat{P}_n \tag{7.19}
\]

Assume that a tracklet has to be determined at time \(t_n\). The measurement model for the tracklet to be calculated is defined by

\[
\bar{u}_n = B_n \times \hat{s}_n + \bar{g}_n \tag{7.20}
\]

where \(B_n = I - \hat{P}_n \times \hat{P}_n^{-1}\). The measurement error is assumed to be a white sequence [14], with \(E(\bar{g}_n) = 0\) and covariance matrix

\[
E(\bar{g}_n \times \bar{g}_k^T) = U_n \quad \text{for } n = k
\]

\[
= 0 \quad \text{for } n \neq k \tag{7.21}
\]

The tracklet is estimated with

\[
\bar{u}_n = \hat{s}_n - \hat{P}_n \times \hat{P}_n^{-1} \times \hat{s}_n \tag{7.22}
\]

where \(I\) is the \(n \times n\) unity matrix. Substitution of eq. 7.20 and eq. 7.22 in \(E(\bar{g}_n \times (\hat{s} - \bar{s})^T)\) produces

\[
E(\bar{g}_n \times (\hat{s} - \bar{s})^T) = E[(\hat{s}_n - \hat{P}_n \times \hat{P}_n^{-1} \times \hat{s}_n) - B_n \times \hat{s}_n] \times (\hat{s}_n - \bar{s})^T
\]

\[
= E[(\hat{s}_n - \hat{s}_n) + \hat{P}_n \times \hat{P}_n^{-1} \times (\hat{s}_n - \hat{s}_n)] \times (\hat{s}_n - \bar{s})^T
\]

\[
= -\hat{P}_n + \hat{P}_n \times \hat{P}_n^{-1} \times E(\hat{s}_n - \hat{s}_n) \times (\hat{s}_n - \bar{s})^T = 0 \tag{7.23}
\]
Using eq. 7.19, the tracklet covariance matrix, corresponding with eq. 7.22, can be written as

\[
U_n = E(\tilde{y}_n \times \tilde{y}_n^T) = \\
E([B_n \times \tilde{s}_n - (\tilde{s}_n - \tilde{P}_n \times P_n^{-1} \times \tilde{s}_n)] \times [B_n \times \tilde{s}_n - (\tilde{s}_n - \tilde{P}_n \times P_n^{-1} \times \tilde{s}_n)]^T = \\
E([\tilde{s}_n - \tilde{s}_n] - P_n \times P_n^{-1} \times (\tilde{s}_n - \tilde{s}_n)) \times [(\tilde{s}_n - \tilde{s}_n) - P_n \times P_n^{-1} \times (\tilde{s}_n - \tilde{s}_n)]^T = \\
\tilde{P}_n - E(\tilde{s}_n - \tilde{s}_n) \times (\tilde{s}_n - \tilde{s}_n)^T \times P_n^{-1} \times \tilde{P}_n = \\
\tilde{P}_n - \tilde{P}_n \times P_n^{-1} \times \tilde{P}_n =
\] (7.24)

Both equations 7.22 and 7.24 are only valid when \( P_n^{-1} \) is non-singular. Drummond [52] has shown that it is possible to derive the inverse Kalman filter approach from the decorrelated state approach resulting in the measurement model and the tracklet equations given in section 7.2.3. This implies that the tracklet equations derived in this section cannot be applied after each track update either.

### 7.2.5 Inverse information filter

Griffiths and Covino [53, 70] have proposed the inverse information filter approach. Due to the fact that the original report is propriety and has not been disclosed, here an alternative derivation is given. The starting point is the extended first order Kalman filter, defined in section 2.3. The error covariance matrix update equation is specified by eq. 2.6 and given by

\[
\hat{P}_j = \hat{P}_j - \hat{P}_j \times H_{s_j}^T \times (H_{s_j} \times \hat{P}_j \times H_{s_j}^T + R_j)^{-1} H_{s_j} \hat{P}_j
\] (7.25)

where \( H_{s_j} \) is the Jacobian of \( h(\tilde{s}_j) \), which represents the ideal (noiseless) connection between the measurement and the state vector at the measurement time \( t_j \). Furthermore, \( \tilde{s}_j \) is the state vector, predicted to time \( t_j \). It is assumed that the time interval, used for prediction, is determined by \( t_j - t_{j-1} \), where \( t_{j-1} \) is the time of the previously processed measurement. The equations 2.3 and 2.4 (section 2.3) specify the prediction step. Before proceeding, first the following theorem, taken from Anderson [6], is introduced.

**Theorem 7.1 (Matrix inversion theorem):**

In terms of an \( n \times n \) matrix \( \hat{P}_j \), a \( p \times p \) matrix \( R_j \), a \( p \times n \) Jacobian matrix \( H_{s_j} \), and the unity \( n \times n \) matrix \( I \), the following equalities hold on the assumption that the various inverses exist:

\[
(I + \hat{P}_j \times H_{s_j}^T \times R_j^{-1} \times H_{s_j})^{-1} \times \hat{P}_j =
\]

\[
(\hat{P}_j^{-1} + H_{s_j}^T \times R_j^{-1} \times H_{s_j})^{-1} =
\]

\[
\hat{P}_j - \hat{P}_j \times H_{s_j}^T \times (H_{s_j} \times \hat{P}_j \times H_{s_j}^T + R_j)^{-1} \times H_{s_j} \times \hat{P}_j
\]

Application of theorem 7.1 on eq. 7.25 produces the result

\[
\hat{P}_j = (\hat{P}_j^{-1} + H_{s_j}^T \times R_j^{-1} \times H_{s_j})^{-1}
\] (7.26)
Taking the inverse produces finally
\[ \hat{P}_j^{-1} = \tilde{P}_j^{-1} + H_{s_j}^T R_j^{-1} \times H_{s_j} \]  
(7.27)

It is possible to write the state update equation 2.5 as [30]
\[ \hat{s}_j = \tilde{s}_j + \hat{P}_j^{-1} \times H_{s_j}^T \bar{s}_j \times (\vec{z}_j - h(\vec{s}_j)) \]  
(7.28)
\[ \hat{P}_j^{-1} \times \hat{s}_j = \tilde{P}_j^{-1} \times \tilde{s}_j + \hat{P}_j^{-1} \times R_j^{-1} \times (\vec{z}_j - h(\tilde{s}_j)) \]  
(7.29)

Substituting eq. 7.27 in eq. 7.29 the following version of the state update equation is obtained
\[ \hat{P}_j^{-1} \times \hat{s}_j = \tilde{P}_j^{-1} \times \tilde{s}_j + \hat{P}_j^{-1} \times R_j^{-1} \times (\vec{z}_j - (h(\tilde{s}_j) - H_{s_j} \times \tilde{s}_j)) \]  
(7.30)

Equation 7.30 and 7.27 (and the corresponding prediction step) form the extended version of the information filter, which has been derived in [6] assuming a linear measurement model and a linear process evolution model.

Our objective is to determine a tracklet for the measurements collected in the time interval \([t_k, t_n]\). The tracklet measurement model is defined in the same way as for the inverse Kalman filter (section 7.2.3). The measurement error is assumed to be white noise. Sufficient measurements are collected to guarantee that the measurement model, specified by eq. 7.8, is true. Due to the fact that the Jacobian of the function representing the ideal connection with the tracklet and the state vector is \(H_k \equiv I\), the update equations of the extended information filter reduce to
\[ \hat{P}_n^{-1} = \tilde{P}_n^{-1} + U_n^{-1} \]  
(7.31)
\[ \hat{P}_n^{-1} \times \hat{s}_n = \tilde{P}_n^{-1} \times \tilde{s}_n + U_n^{-1} \times \vec{u}_n \]  
(7.32)

Some rewriting produces the inverse information filter equations, given by
\[ U_n^{-1} = \hat{P}_n^{-1} - \tilde{P}_n^{-1} \]  
(7.33)
\[ U_n^{-1} \times \vec{u}_n = \hat{P}_n^{-1} \times \hat{s}_n - \tilde{P}_n^{-1} \times \hat{s}_n \]  
(7.34)

Both equations 7.33 and 7.34 are only valid when the covariance matrices \(\hat{P}_n^{-1}\) and \(\tilde{P}_n^{-1}\) are non-singular. Furthermore, enough measurements have to be collected to guarantee that the measurement model (eq. 7.8) is valid. Consequently the inverse information filter equations can neither be applied after each track update. Using eq. 7.25, eq. 7.28, the measurement model given by eq. 7.8 and substituting \(H_{s_j} = I\), it is easily shown that the inverse information filter is equivalent to the inverse Kalman filter, described in section 7.2.3.

### 7.2.6 Periodic calculation

In this section a fourth decorrelation approach is proposed, which is based on a literal interpretation of the tracklet equivalence description given in section

\footnote{Drummond [52] has shown that the measurement model, specified by 7.20, is equivalent to the measurement model specified by eq. 7.8.}
7.2.2. In that description a tracklet is considered equivalent with a track for a target which is based only on the most recent measurements, received after the last time a tracklet has been calculated [54]. It is necessary to guarantee that there are no other tracklets calculated, which share one or more measurements. To guarantee this, each participant uses the measurements collected by the on-board sensors to produce sets of measurements for each target of interest. Using the measurements received after the last time a calculation has been carried out, tracklets are calculated independently for each platform from the different sets of measurements by applying a standard filtering technique like extended first order Kalman filtering (section 2.3). The calculated tracklets are distributed. In the example, shown in Fig. 7.2, a tracklet is periodically calculated from $k = 3$ collected sensor measurements, where $k$ is the tracklet reduction factor. The first of the three measurements is used to initialize the tracklet calculation and the other two measurements are used to refine the tracklet estimates. The characteristics of the periodically calculated tracklets approach are:

- The first collected measurement in each set generates a separate tracklet;
- Each time $k$ new measurements have been collected a tracklet is generated, where $k$ is the tracklet reduction factor. The first of the $k$ measurements initiates the tracklet;
- The further $k - 1$ measurements refine the tracklet estimates;
- In contrast with the earlier discussed methods, $k \geq 1$.

Using the extended first order Kalman filter equations, given in section 2.3, for the different reduction factors tracklet equations can be derived, which are valid at a certain time $t_n$. For $k = 3$ the equations are given by:

\[
\begin{align*}
\bar{u}_n &= f_2(f_1(\hat{s}_1) + K_2 \times (\bar{z}_2 - h(f_1(\hat{s}_1)))) \\
&\quad + K_3 \times (\bar{z}_3 - h(f_2(f_1(\hat{s}_1)) + K_2 \times (\bar{z}_2 - h(f_1(\hat{s}_1)))) \\
\hat{U}_n &= (I - K_3 \times H_3) \times [F_{\hat{s}_2} \times ((I - K_2 \times H_2) \times (F_{\hat{s}_1} \times \hat{P}_{\hat{s}_1} \times F_{\hat{s}_1}^T + Q_2)) \times F_{\hat{s}_2}^T + Q_3]
\end{align*}
\]

where $z_i$ is measurement $i$, $f_i(\cdots)$ is the nonlinear transition function and $K_i$ is the Kalman gain (see also section 2.3), both valid at the time of measurement $t_n$. The tracklet methods, introduced in section 7.2.2-7.2.5, require at least two measurements to calculate a tracklet.
7.3. TRACK-TO-TRACK FUSION EVALUATION

Furthermore, \( h(\vec{s}_k) \) represent the ideal (noiseless) connection between the measurement and the state vector at measurement time \( t_k \). The process noise at measurement time \( t_k \) is given by \( Q_k \).

Using the the range and angular information contained in the first measurement \( \vec{z}_1 = (R_1, \beta_1, \epsilon_1) \), the variable \( \hat{s}_1 \) is defined by

\[
\begin{pmatrix}
R_1 \times \cos(\epsilon_1) \\
R_1 \times \cos(\epsilon_1) \times \sin(\beta_1) \\
R_1 \times \sin(\epsilon_1)
\end{pmatrix}
\]

(7.36)

To determine a tracklet for the first measurement, the following calculation is carried out. To define the variable \( \hat{P}_1 \), first \( \hat{P}_{x,y,z} = G_{\vec{z}_1} \times R \times G_{\vec{z}_1}^T \) is defined, where

\[
G_{\vec{z}_1} = \begin{pmatrix}
\cos(\epsilon_1) \times \cos(\beta_1) & \cos(\epsilon_1) \times \sin(\beta_1) & \sin(\epsilon_1) \\
-R_1 \times \cos(\epsilon_1) \times \sin(\beta_1) & R_1 \times \cos(\epsilon_1) \times \cos(\beta_1) & 0 \\
-R_1 \times \sin(\epsilon_1) \times \cos(\beta_1) & -R_1 \times \sin(\epsilon_1) \times \sin(\beta_1) & R_1 \times \cos(\epsilon_1)
\end{pmatrix}
\]

(7.37)

and where \( R \) is the standard measurement noise matrix for measurement \( \vec{z}_1 \). Finally, \( \hat{P}_1 \) is defined as

\[
\hat{P}_1 = \begin{pmatrix}
\hat{P}_{x,y,z} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
\]

(7.38)

where \( v_{max} \) is the expected maximum target velocity in the horizontal plane and \( v_{max}^z \) is the expected maximum target velocity along the z direction. The tracklet corresponding with the first measurement is defined by \( \vec{u}_1 = \vec{s}_1 \) and \( \vec{U}_1 = \hat{P}_1 \).

As a consequence of using the standard measurement model, specified by eq. 2.2, the described method to derive tracklets can also be applied after each track update.

7.3 Track-to-track fusion evaluation

The various methods introduced in section 7.2 will be evaluated against the criteria introduced in 7.1: the amount of data to be communicated and the processing time required to solve the data association problem. In section 7.3.1 the measures of evaluation are introduced. Section 7.3.2 will deal with the track-to-track fusion approach and 7.3.3 will evaluate the track decorrelation methods.

7.3.1 Measures of evaluation

It is assumed that a scenario is considered for which the computation time to create a set of track hypotheses for cluster \( i \) is significantly higher than the computation time to determine one or more good solutions for the corresponding
 CHAPTER 7. ARTIFICIAL MEASUREMENTS

MDA problem (see also section 5.2 and 6.3.3) \(^4\). To analyze the effect on the computation time to create the set of track hypotheses, it is assumed that for each cluster the tasks defined in section 6.5.1 have to be carried out. The estimated computation time to predict the cluster ellipsoid and the cluster mean tracks on processor \(j\) is given by eq. 6.10 and the estimated computation time necessary for correlation, prediction and track maintenance is given by eq. 6.13. The estimated computation time necessary for prediction is given by

\[
f_{j1} = \bar{N}_i \times f(\bar{T}_i, c_p) \tag{7.39}
\]

where

\[
f(\bar{T}_i, c_p) = [\bar{T}_i + 1] \times c_p \tag{7.40}
\]

Furthermore, \(\bar{N}_i\) is the average number of measurements in a new data set which is assigned to cluster \(i\), \(\bar{T}_i\) is the average number of target trees for cluster \(i\) and \(c_p\) is the computation time to predict a state vector and covariance matrix. The estimated time necessary for correlation, prediction and track maintenance is written as (eq. 6.13)

\[
f_{j2} = \bar{N}_i \times g(\bar{T}_i, \bar{T}_{tr}^i, T_m, c_s, c_m) \tag{7.41}
\]

where

\[
g(\bar{T}_i, \bar{T}_{tr}^i, T_m, c_s, c_m) = [\bar{T}_i \times c_s + T_m \times \bar{T}_{tr}^i \times (c_s + a \times c_m)] \tag{7.42}
\]

\(\bar{T}_{tr}^i\) is the average number of track hypotheses in the \(\bar{T}_i\) target trees and \(T_m\) is the estimated number of mean tracks a measurement correlates with. \(\bar{T}_{tr}^i\) is defined as

\[
\bar{T}_{tr}^i = \frac{\sum_{l=1}^{T_i} \bar{n}_h^l}{T_i}, \text{ where } \bar{n}_h^l \text{ is the average number of track hypotheses in target tree } l.
\]

Furthermore, \(c_s\) is the computation time to correlate with a mean track or track hypothesis and \(c_m\) is the computation time to carry out track maintenance. The estimated total computation time for processor \(j\) to carry out the significant operations for cluster \(i\) during the problem initiation phase is given by

\[
\bar{T}_{ij}^p = f_{j1} + f_{j2} = \bar{N}_i \times (f(\bar{T}_i, c_p) + g(\bar{T}_i, \bar{T}_{tr}^i, T_m, c_s, c_m)) \tag{7.43}
\]

Normally the number of track maintenance operations is very low which means that it is reasonable to assume that the computation time related to track maintenance does not contribute much to the total computation time and eq. 7.43 can be approximated by

\[
\bar{T}_{ij}^p = f_{j1} + f_{j2} = \bar{N}_i \times (f(\bar{T}_i, c_p) + g^*(\bar{T}_i, \bar{T}_{tr}^i, T_m, c_s)) \tag{7.44}
\]

where

\[
g^*(\bar{T}_i, \bar{T}_{tr}^i, T_m, c_s) = [\bar{T}_i \times c_s + T_m \times \bar{T}_{tr}^i \times c_s] \tag{7.45}
\]

The communication time to distribute a set of \(T\) measurements or track hypotheses in a single data package over a communication link with bandwidth \(d_{bw}\) can be calculated with eq. 6.28 and is given by

\[
T_{comm} = \frac{T \times N_{bytes} + N_h}{d_{bw}} \tag{7.46}
\]

\(^4\)The time to decompress the received data is not taken into account.
where $N_{\text{bytes}}$ is the number of bytes necessary to distribute a state vector and a (partial) residual error covariance matrix or the Kalman gain for a track hypothesis. The data package is labelled with a standard header with a length of $N_h$ bytes.

### 7.3.2 Track fusion

Earlier, in section 7.2.1 it has been mentioned that it can easily occur that a platform composite track hypothesis is partially based on the same information as the received (composite) track hypothesis for the same object. Assume that the computation time on processor $j$ to remove the possible data dependency is given by $f_{\text{dep}}^j$. Applying eq. 7.44, the ratio of the resulting computation time to create the set of track hypotheses using $\bar{N}_i$ track hypotheses or $\bar{N}_i$ measurements is given by

$$\frac{T_{\text{dep}}^j(\text{tracks})}{T_{\text{dep}}^j(\text{measurements})} = \frac{f_{\text{dep}}^j + \bar{N}_i \times (f(T_i, c_p) + g^*(T_i, T_{tr}, T_m, c_{tr})\}}{N_i \times (f(T_i, c_p) + g^*(T_i, T_{tr}, T_m, c_{tr}|)\}}$$

(7.47)

After removing the data dependency (section 7.2.1) the ellipsoidal gate (eq. 7.4, section 7.2.1) can be used to correlate tracks with tracks (see e.g. tracklets) and $c_{ts}$ is the computation time to carry out track-to-track correlation. Furthermore, $c_{ms}$ is the computation time to correlate a measurement with a track hypothesis. Making the reasonable assumption that $c_{ts} \geq c_{ms}$, the ratio 7.47 shows that the computation time increases when track hypotheses instead of measurements are distributed.

Assume that for each local track update, a state vector with dimension $n = 6$ and a partial error covariance matrix containing 21 elements have to be distributed. It is assumed that an average compression factor $\alpha$ is obtained, using an appropriate compression technique (section 7.1). If 8 bytes (double precision) is required for an element of the state vector or the covariance matrix, a total number of $216 \times \alpha$ bytes is necessary to distribute the information related to a local track update. For a sensor measurement, only a vector containing for example three elements (range, bearing and elevation) and the diagonal of the measurement covariance matrix has to be distributed. In that case only $N_{\text{bytes}}^m = 48 \times \alpha$ bytes have to be distributed for each sensor measurement.

If we analyze the update equations of the extended first order Kalman filter (eq. 2.6), it is possible to reduce the amount of data which has to be distributed. Instead of distribution of the relevant 21 elements of the error covariance matrix, it is sufficient to distribute only the 18 elements of the Kalman gain, which is defined by eq. 2.7. This means that it is sufficient to distribute $N_{\text{bytes}}^t = 192 \times \alpha$ bytes after each local track update. Using eq. 7.46, the ratio of the communication time distributing $T$ track hypotheses or measurements over a certain communication link is given by

$$\frac{T_{\text{comm}}(\text{tracks})}{T_{\text{comm}}(\text{measurements})} = \frac{T \times N_{\text{bytes}}^t + N_h}{T \times N_{\text{bytes}}^m + N_h}$$

(7.48)

where $N_h$ is the number of bytes necessary for a standard header for a data package. Due to the fact that $N_{\text{bytes}}^t > N_{\text{bytes}}^m$ the communication time for transferring track data is significantly larger than for measurement data. For
example, if \( T = 100 \), \( N_h = 1.5 \) and \( \alpha = 0.5 \), the ratio is approximately 4. Compared with the distribution of correlated measurements, the track fusion method described in section 7.2.1 is not really an improvement.

### 7.3.3 Track decorrelation

Compared with the track fusion method, the different track decorrelation methods identified in section 7.2.2 have the advantage that only one tracklet is calculated for \( k \) sensor measurements (\( k \geq 1 \)), collected in the defined time interval \( \Delta t = [t_1, t_n] \). Assume that the computation time necessary to calculate a tracklet on processor \( j \) is given by \( f_j^{\text{tracklet}} \). Using eq. 7.44, the ratio, defined similar as in eq. 7.47, of the resulting computation time to create the set of track hypotheses using \( \bar{N}_i^{\text{tracklets}} \) or \( \bar{N}_i^{\text{measurements}} \) is given by

\[
R_{tr} = \frac{\bar{T}_p^{i} / \bar{T}_j^{\text{tracklets}}}{\bar{T}_p^{i} / \bar{T}_j^{\text{measurements}}} = \frac{\bar{N}_i \times (f_j^{\text{tracklet}} + g^*(\bar{T}_i, \bar{T}_m, c_s^{tr}))}{\bar{N}_i \times (f_j(\bar{T}_i, c_p) + g^*(\bar{T}_i, \bar{T}_m, c_s^{m})}) \tag{7.49}
\]

where \( c_s^{tr} \) is the computation time to correlate a tracklet and \( c_s^{m} \) is the computation time to correlate a measurement. Normally, \( c_s^{tr} \geq c_s^{m} \). To determine a lower bound for eq. 7.49, we set \( c_s^{tr} = c_s^{m} \) and assume that \( f_j^{\text{tracklet}} \ll f_j(\bar{T}_i, c_p) + g(\bar{T}_i, \bar{T}_m, c_s^{tr}) \). The result is that

\[
R_{tr} \geq \frac{1}{k} \tag{7.50}
\]

which means that the maximally obtainable reduction in computation time is \( k \).

Assume that for each tracklet a state vector with dimension \( n = 6 \) and a partial error covariance matrix containing 21 elements have to be distributed. Following section 7.3.2, this means that a total number of \( N_{\text{bytes}}^{\text{tracklets}} = 27 \times \alpha \) bytes are distributed. For a sensor measurement, only a vector containing for example three elements (range, bearing and elevation) and the diagonal of the measurement covariance matrix have to be distributed. In that case only \( N_{\text{bytes}}^{\text{measurements}} = 6 \times \alpha \) bytes have to be distributed. Assume that we want to know if the distribution of \( \frac{T}{k} \) tracklets instead of \( T \) measurements reduces the necessary communication time over a certain communication link. Using eq. 7.48, reduction in communication time is only obtained when

\[
\frac{T}{k} \times N_{\text{bytes}}^{\text{tracklets}} + N_h < T \times N_{\text{bytes}}^{\text{measurements}} + N_h \tag{7.51}
\]

The conclusion is that reduction is obtained if \(^5\)

\[
k \geq \left\lceil \frac{N_{\text{bytes}}^{\text{tracklets}}}{N_{\text{bytes}}^{\text{measurements}}} \right\rceil \tag{7.52}
\]

For the given values a reduction of information is only obtained when \( k \geq 5 \). This means that a significant delay \( \Delta t \) may be introduced before new tracklet information is available to update a composite track. Such large delays are

\(^5\left\lceil x \right\rceil \) rounds \( x \) off to the nearest integer towards \( \infty \).
not acceptable for realtime applications. Only when the number of sensors is sufficiently large, will the update rate for a composite track become high enough to guarantee a sufficiently small $\Delta t$.

In the sections 7.3.3.1, 7.3.3.2 and 7.3.3.3 it is shown that the first three decorrelation methods, enumerated in section 7.2.2, have major problems with the numerical stability of the proposed calculation.

7.3.3.1 Inverse Kalman filter

For the inverse Kalman filter approach (section 7.2.3), the tracklet covariance matrix $U_n$ is determined by eq. 7.16. In this equation, the inverse of the difference matrix $D = \bar{P} - \hat{P}$ is used. Two experiments have been carried out to determine the condition number (appendix A) and the determinant of the matrix $D$. In the first experiment, a target flies with a bearing angle of $45^o$ and a velocity of $500\,\text{m/s}$ directly in the direction of the radar, which is observing the airspace. The radar detects the target at a range of 400 km. The accuracy of the radar in range is $\sigma_r = 30\,\text{m}$., in bearing is $\sigma_\beta = 0.3^o$ and in elevation is $\sigma_\epsilon = 0.3^o$. The time, necessary for a $360^o$ scan, is $\Delta t = 5\,\text{sec}$. In the second experiment, the target is detected at the same maximum range and the target passes the radar at a cross distance of 30 km. In both experiments, the target flies at a height of 10 km. The results of the first experiment are given in Fig. 7.3. The results of the second experiment are reported in appendix H.

The condition number is defined by $\text{cond}(D) = \left| \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}} \right|$, where $\lambda_{\text{max}}$ is the maximum eigenvalue and $\lambda_{\text{min}}$ is the minimum eigenvalue of the matrix $D$ (appendix A). In the different figures shown in Fig. 7.3 a logarithmic scale is used. The upper three figures show the value of the condition number ($\text{cond}(D)$) for the matrix $D$ for three different tracklet reduction factors $k$, given as function of time. The lower three figures show the corresponding determinant value $|D|$ of this matrix. An upwards pointing triangle indicates a positive determinant value and a downwards pointing triangle a negative value. Furthermore, $\text{Sing}$ indicates the number of times the matrix $D$ is singular ($\text{det}(D) = 0$), $\text{Cond}$ indicates the number of times the condition number has been determined and $\text{Total}$ represents the total number of trajectory points considered. Due to the fact that the tracklet reduction factor $k \geq 1$, $\text{Cond} \leq \text{Total}$. Fig. 7.3 shows that $\text{cond}(D)$ increases significantly, when the distance of the target decreases or when the tracklet reduction factor increases. For condition numbers $\text{cond}(D) \geq 10^9$ the matrix $D$ is considered ill-conditioned (appendix A). For the results, given in Fig. 7.3, this means that the matrix $D$ is continuously ill-conditioned during a significant part of the target trajectory. This part increases, when the tracklet reduction factor is increased. For larger condition numbers values, we see also that the matrix $D$ is not only ill-conditioned but very close to becoming singular. This means that $|D|$ is very close to 0.
Figure 7.3: The condition number and determinant value are shown as function of simulation time.
7.3. TRACK-TO-TRACK FUSION EVALUATION

7.3.3.2 Decorrelated state

For the decorrelated state approach, discussed in section 7.2.4, the tracklet covariance matrix (eq. 7.24)

\[ U_n = \hat{P}_n - \hat{P}_n \times \hat{P}_n^{-1} \times \hat{P}_n \]  

(7.53)

is considered. If we assume that \( |U_n| \neq 0 \), using theorem 7.1 it is possible to rewrite the Kalman update equation (eq. 7.13) as

\[ \hat{P}_n = \hat{P}_n - \hat{P}_n \times (\hat{P}_n + U_n)^{-1} \times \hat{P}_n = (\hat{P}_n + U_n)^{-1} \]  

(7.54)

To determine the estimated error covariance matrix \( \hat{P}_n \), it is necessary to determine the inverse of the matrix \( \hat{P}_n + U_n \) or the inverse of the matrix \( U_n \). Now, the matrix \( D \) is defined by

\[ D \text{=} U_n \]

and the same experiments as described in section 7.3.3.1 are carried out. The results for the first experiment are given in Fig. 7.4. The results for the second experiment are reported in appendix H.2. Comparing the results in Fig. 7.4 with Fig. 7.3 very similar results are observed. Again the condition number \( \text{cond}(D) \) increases significantly when the distance of the target decreases or when the tracklet reduction factor increases.

Furthermore, the alternative formulation \( D = \hat{P}_n + U_n \) has also been tried. For this case, \( \text{cond}(D) \) and \( |D| \) showed the same behavior as shown in Fig. 7.4.

Further analysis and inspection of Fig. 7.4 will reveal another numerical problem. The tracklet density distribution function is assumed to be Gaussian. At time \( t_n \), this function is specified by \( N(\tilde{u}_n, U_n) \). The dimension of \( \tilde{u}_n \) is \( m \). The likely state vectors \( \tilde{\mathbf{s}}_n \) of the real target, corresponding with the tracklet, fall within the hyperellipsoid, defined by

\[ (\tilde{\mathbf{s}}_n - \tilde{u}_n)^T \times U_n^{-1} \times (\tilde{\mathbf{s}}_n - \tilde{u}_n) \leq \sigma^2 \]  

(7.55)

where the threshold \( \sigma^2 \), which determines the risk that the real state vector falls outside the defined hyperellipsoid, can be read from the \( \chi^2 \)-table. Taking \( \tilde{u}_n \) as the origin of the state space and redefining \( \tilde{\mathbf{x}} = \tilde{\mathbf{s}}_n - \tilde{u}_n \) produces the quadratic form

\[ Q(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}}^T \times U_n^{-1} \times \tilde{\mathbf{x}} \]  

(7.56)

Due to the fact that \( U_n^{-1} \) is symmetric and \( Q(\tilde{\mathbf{x}}) \geq 0 \) for all \( \tilde{\mathbf{x}} \in \mathbb{R}^n \), the matrix \( U_n^{-1} \) is semi-definite and all eigenvalues are \( \geq 0 \) [18]. Furthermore

\[ Q(\tilde{\mathbf{y}}) = \tilde{\mathbf{y}}^T \times U_n^{-1} \times \tilde{\mathbf{y}} \]

\[ = \tilde{\mathbf{y}}^T \times (U_n^{-1})^T \times U_n \times U_n^{-1} \times \tilde{\mathbf{y}} \]

\[ = \tilde{\mathbf{y}}^T \times U_n \times \tilde{\mathbf{y}} \]

where \( \tilde{\mathbf{y}} = U_n^{-1} \times \tilde{\mathbf{x}} \). Following the same reasoning as earlier, this means that \( U_n \) is also semi-definite. With the assumption \( |U_n| \neq 0 \), all eigenvalues are larger than zero. This means that \( |D| = |U_n| > 0 \). From the three lower figures from Fig. 7.4, we have to conclude that this requirement is not fulfilled in our experiment because all values for \( |D| < 0 \). This is an indication that numerical problems have occurred during the experiment. The conclusion is that there easily may occur numerical problems with this method to determine tracklets.

\footnote{For \( D = \hat{P}_n + U_n \), the same derivation can be made and the same results are found.}
Figure 7.4: The condition number and determinant value are shown as function of simulation time.
7.3. TRACK-TO-TRACK FUSION EVALUATION

7.3.3.3 Inverse information filter

Using the inverse information equations, given in section 7.2.5, the quantities estimated in the inverse information filter approach are \( U_{n-1} \) and \( U_{n-1} \times \vec{u}_n \).

Applying extended first order Kalman filtering, the error covariance matrix \( \hat{P}_n \) has to be determined with one of the following equations:

\[
\hat{P}_n = \begin{cases} 
(\hat{P}_n^{-1} + U_n^{-1})^{-1} \\
\hat{P}_n - \hat{P}_n \times (\hat{P}_n + U_n)^{-1} \times \hat{P}_n
\end{cases} 
\tag{7.57}
\]

For the second expression for \( \hat{P}_n \) it is necessary to determine the covariance matrix \( U_n \) by taking the inverse of \( U_n^{-1} \). The matrix to be considered is defined by \( D = U_n^{-1} \). It is assumed that \( U_n \neq 0 \). The results of the first experiment are given in Fig. 7.5. The results of the second experiment are reported in appendix H. Assuming again that a matrix is considered ill-conditioned for \( \text{cond}(D) \geq 10^4 \), the conclusion is that the matrix \( D \) is continuously ill-conditioned for a significant part of the second half of the target trajectory. Again, this part increases when the tracklet reduction factor \( k \) is increased. We see now that the |\( D \)| increases steadily from very small to very large negative values. Following the same derivation as given in section 7.3.3.2, it is possible to show that \( U_n^{-1} \) is semi-definite. With the assumption that \( U_n \neq 0 \), this means all eigenvalues are larger than zero. The results for |\( D \)| show again continuously negative values.

This means that there occurred numerical problems during the experiment.

For the second expression for \( \hat{P}_n \), it is necessary to determine the inverse of \( D = \hat{P}_n + U_n \). In that case, \( \text{cond}(D) \) is continuously higher that \( 10^5 \). This means that the matrix \( D \) is continuously ill-conditioned. Furthermore, using the same approach as shown in section 7.3.3.2, it is possible to show that the eigenvalues are larger than zero. This means that |\( D \)| > 0. Again, the values found during the experiments where continuously negative. The conclusion is that there easily may occur numerical problems with this method to determine tracklets.
Figure 7.5: The condition number and determinant value are shown as function of simulation time.
7.4 Artificial measurements

An alternative tracklet calculation has been proposed in section 7.2.6, which does not have the numerical problems identified in the sections 7.3.3.1-7.3.3.3. This track decorrelation method has still the disadvantage, identified in section 7.3.3, that reduction of communication time is only obtained when \( k \) fulfills eq. 7.52. For realtime applications, the resulting delay time before new tracklet information will be available to maintain a composite track will normally not be acceptable.

In the original formulation a tracklet contains full state vector information which means that \( k \) must be sufficiently large to reduce the amount of information to be distributed. Here an alternative approach is taken. After calculating a tracklet using \( k \) measurements (\( k \geq 1 \)), the tracklet is first transformed to an artificial measurement to obtain even for very low values of \( k \) a significant reduction for the data which has to be distributed. Artificial measurements are produced from the tracklets, using the following set of equations 7:

\[
\begin{align*}
R &= \sqrt{x^2 + y^2 + z^2} \\
\beta &= \text{atan2}(y, x) \\
\varepsilon &= \text{atan2}(z, \sqrt{x^2 + y^2})
\end{align*}
\]  \hspace{1cm} (7.58)

where \( \vec{r} = (x, y, z) \) is the positional part of the tracklet state vector. Here the bearing \( \beta \) is defined with respect to the \( x \)-axis. The elevation is given by \( \varepsilon \). It has to be noted that the transformation does not use the velocity-information contained in the tracklet state vector, which means that information is lost due to the specific transformation. If necessary, the Doppler velocity \( \dot{R} \) is calculated with

\[
\dot{R} = \vec{v}^T \times \vec{r} / ||r||^2
\]  \hspace{1cm} (7.59)

where \( \vec{v} = (v_x, v_y, v_z) \) is the velocity part of the tracklet state vector. The corresponding covariance matrix is calculated by

\[
R_{am} = H \times P_{tracklet} \times H^T
\]

where \( H \) is the Jacobian of the ideal (noiseless) connection \( h(\vec{s}) \) between the measurement and the state vector at the time of measurement \( t \) (see section 2.3). To obtain a higher reduction in the amount of information which has to be distributed for each calculated tracklet, the matrix \( R_{am} \) is approximated. To represent the likely, but unknown target positions \( \vec{z} \) in polar space, the ellipsoid defined by

\[
[\vec{z} - \vec{z}]^T R_{am}^{-1} [\vec{z} - \vec{z}] \leq r^2
\]  \hspace{1cm} (7.60)

is used, where \( \vec{z} = [R, \beta, \varepsilon] \) is calculated using eq. 7.58. It is assumed that the value for the threshold \( r^2 \), chosen from the \( \chi^2 \)-distributed table, is so large, that the risk that likely positions fall outside the ellipsoid is almost negligible. In section 4.5.2.1 a method has been proposed to determine the smallest cuboid which encloses a defined hyperellipsoid, oriented along the axes of an arbitrary chosen coordinate system. Using this method, it is easily shown that the size of the cuboid along the different axes is defined by

\[
\begin{align*}
r_R &= \sqrt{r^2 \times \sigma_R^2} \\
r_\beta &= \sqrt{r^2 \times \sigma_\beta^2}
\end{align*}
\]

It is also possible to use the position given by the tracklet state vector and the corresponding covariance matrix.

\[\text{atan2}(y, x)\] determines counterclockwise the angle \( \alpha \) between the \( x \)-axis and the vector \( (x, y) \), with \(-\pi < \alpha \leq \pi\).
and \( r_\varepsilon = \sqrt{r^2 \times \sigma^2} \). Assuming that \( R, \beta \) and \( \varepsilon \) are independent variables which are normally distributed, reasonable estimates for the different standard deviations are given by \( \sigma_R = \frac{r_R}{3}, \sigma_\beta = \frac{r_\beta}{3} \) and \( \sigma_\varepsilon = \frac{r_\varepsilon}{3} \). The original error covariance matrix is approximated by

\[
R_{am} \approx \begin{pmatrix}
\sigma_R^2 & 0 & 0 \\
0 & \sigma_\beta^2 & 0 \\
0 & 0 & \sigma_\varepsilon^2 \\
\end{pmatrix}
\]

(7.61)

It is trivial to determine \( R_{am} \), when the Doppler velocity has to be included in the AM vector.

To illustrate that tracking using artificial measurements instead of the original tracklets produces accurate tracks, here an example is given. An extended first order Kalman filter is used to track a target. The target velocity is 800 m/s and the target flies a linear trajectory, which is parallel with the x-axis. The crossing distance with respect to the origin is 10 km and the target height is 1 km. The platform, fitted with a single sensor, is positioned in the origin.

Two situations are considered: measurement tracking and artificial measurement (AM) tracking. The receiving platform has to re-construct the track formed by the originating platform. The simulated radar sensor detects the target at a range of around 100 km. The measurement noise is set to 0.03 km in range and to 0.3 degrees both in bearing and elevation. The process noise is set to 30 m/s.

Fig. 7.6 shows the obtained position accuracy, given as function of time. The results are only shown at the comparison times when there are track updates both by an artificial measurement and a measurement, starting from the second AM update time. Initially, \( \sigma_x \approx 65 \text{ m}, \sigma_y \approx 525 \text{ m} \) and \( \sigma_z \approx 525 \text{ m} \). Worst case, there is a 20 percent difference between measurement and AM tracking. This example shows the potential of AM tracking to provide a very good approximation of measurement tracking. Fig. 7.7 shows the obtained velocity accuracy, given as function of time. Initially, \( \sigma_{v_x} \approx 500 \text{ m/s}, \sigma_{v_y} \approx 500 \text{ m/s} \) and \( \sigma_{v_z} \approx 500 \text{ m/s} \). The conclusion is that the velocity accuracy of AM tracking provides a good approximation of the measurement tracking results. Due to the high range accuracy, the highest accuracy is obtained in the \( x \)-direction.

The claim is that a track maintained with artificial measurements is operationally equivalent with the track maintained with the original measurements. The analysis, presented in the sections 7.5.3-7.5.5, supports this claim.
Figure 7.6: The position accuracy obtained for measurement (plot) tracking and AM tracking is shown as function of simulation time.
Figure 7.7: The velocity accuracy obtained for measurement tracking and AM tracking is shown as function of simulation time.
7.5 Dynamic reduction

The objective of dynamic reduction is to determine the tracklet reduction factor in such a way, that a required average composite track accuracy is guaranteed for the single integrated air picture (SIAP). Consider the example shown in Fig. 7.8.

![Diagram of tracklet reduction](image)

**Figure 7.8**: In the left part of the figure a constant reduction factor is used and in the right part of the figure the reduction factor is determined dynamically.

The left part of the figure shows the old situation with a constant reduction factor. A composite track is shown, which is maintained with tracklets. The black dots originate from local track $j$, which is maintained in the onboard local track data base, and the white circles indicate tracklets originating from other sources.

In the right part of the figure, the tracklet calculation factor $k$ for local track $j$ is dynamically determined in such a way that a required (average) composite track quality is maintained. If the reduction factor for local track $j$ is increased, the time interval between two composite track updates with tracklets originating from track $j$ increases. In this example, the reduction factor has been increased in such a way that the number of track updates from local track $j$ has been reduced by 2 over the duration of the example. Upon detecting that the target carries out a right turn, $k$ is reduced to maintain track quality. Without decreasing the reduction factor, the systematic error between predicted composite and real target trajectory may grow too large to maintain an accurate track. Worst case, tracklets originating from the target will fall outside the correlation gate with the result that break-track occurs.
7.5.1 First approach

Our objective is to define a dynamic reduction approach, which does also work for an asynchronous distributed system, where only a very limited amount of information can be exchanged. Initially, the dynamic reduction algorithm described in this section was used. Despite the fact that this algorithm was discarded, it is discussed because a number of important lessons were learned after it was implemented. The algorithm was based on the following measures:

- **Measure 1**
  
  Using the volume of the accuracy ellipsoid (see appendix G) for both the local track (loc) and the composite track (com), corresponding with the same target, the ratio between the accuracies of both tracks can be expressed by $R_{\text{stat}} = \left( \frac{|\hat{P}_{\text{loc}}|}{|\hat{P}_{\text{com}}|} \right)^{1/2}$, where $\hat{P}_{\text{loc}}$ is the error covariance matrix of the local track and $\hat{P}_{\text{com}}$ is the error covariance matrix of the composite track.

- **Measure 2**
  
  After starting a turn, the difference between the predicted and estimated composite track position will increase, and eventually the target tracking starts to degrade. The difference between the predicted position and the estimated position defines the absolute bias error $\delta_{\text{pos}}$.

  It is required that for each composite track in the SIAP the absolute value of the deviation in position between the predicted track and the estimated track position is smaller than $\Delta_{\text{pos}}$ and that the ratio in accuracy is smaller than $R_r$. Initially, the tracklet reduction factor is given by $k = k_0$, where $k_0$ is a pre-defined constant (e.g. $k_0 = 3$). When a composite track in the SIAP of a certain platform is updated, first a check is made if the tracklet originates from a track of the participant’s local track data base. In that case the absolute bias measure $\delta_{\text{pos}}$ and the accuracy measure $R_{\text{stat}}$ are evaluated for the composite track. The following dynamic reduction algorithm determines the tracklet reduction factor $k$:

  - **Increase $k$**
    
    If $\Delta_{\text{pos}} < \Delta_{\text{pos}_{\text{min}}}$ and $R_{\text{stat}} < R_r$, the tracklet reduction factor is increased to $k = k + 1$.

  - **Decrease $k$**
    
    If $\Delta_{\text{pos}} \geq \Delta_{\text{pos}_{\text{max}}}$ or $R_{\text{stat}} \geq R_r$, the tracklet reduction factor is decreased to $k = k - 1$.

  During simulation tests, the proposed algorithm did not work properly. A large number of values were used for $\Delta_{\text{pos}_{\text{min}}}$, $\Delta_{\text{pos}_{\text{max}}}$ and $R_{\text{stat}}$. The following problems were identified:
7.5. **DYNAMIC REDUCTION**

- **Problem 1**

To increase $k$, both stated conditions had to be fulfilled. The local track and the composite track, corresponding with a certain target, degrade in the same way when the target starts to manoeuvre. This means that measure 1 is not sensitive enough to detect target manoeuvres;

- **Problem 2**

After distributing an artificial measurement, the test was carried out to increase or decrease the tracklet reduction factor for the corresponding local track. When a certain platform increased its reduction factor for a certain local track, upon reception of the artificial measurement other platforms could not take the change in reduction factor into account before they changed their own reduction factor. This resulted in undesirable system behavior.

- **Problem 3**

Using measure 1, each platform compared the accuracy of its own local track with the composite track of interest. The consequence was that each platform applied a different criterion to determine the tracklet reduction factor for the local track. If a certain platform decided that the tracklet reduction factor had to be maintained or lowered, another platform, using its own criterion, could decide that it had to increase the tracklet reduction factor for its local track. The result was that each platform maintained a composite track for the same target with a different accuracy. This behavior clearly conflicted with the requirement that each platform has to produce the same CTP.

7.5.2 **Successful approach**

Based on the lessons learned with the first approach, in section 7.5.2.1 an alternative approach is proposed to determine the tracklet reduction factor dynamically. The approach is based on a test to determine if a target possibly manoeuvres. If the test produces a positive result, the reduction factor $k$ is decreased, otherwise it is increased. In section 7.5.2.2, a number of standard target trajectories is proposed, which are considered representative for target trajectories against which it is useful to apply target tracking based on artificial measurements. A separate experiment to determine the maximal tracklet reduction factor has been carried out and the results of the experiment are reported in section 7.5.2.3.

7.5.2.1 **Manoeuvrability test**

Based on the ideas presented in section 7.5, a statistical test derived from eq. 1.7 (appendix I) is used after updating a local track hypothesis with a measurement to determine if a target possibly manoeuvres. The measure is defined by

$$M_{test} = \left([\hat{s}_{xyz}^k - \bar{s}_{xyz}^k]^T (P_{xyz})^{-1} [\hat{s}_{xyz}^k - \bar{s}_{xyz}^k]\right) > r^2 \quad (7.62)$$
where $\hat{s}_{xyz}^k$ is the estimated 3D-state vector, $\check{s}_{xyz}^k$ is the predicted 3D-state vector and $P_{xyz}$ is defined by the spatial part of $P_{\Delta s}$ (eq. I.3)\(^9\). The threshold $r^2$ can be read from the $\chi^2$-distribution table (section 2.4). The tracklet reduction factor $k$ is restricted to the interval $[1, k_{\text{max}}]$, where $k_{\text{max}}$ is the maximum allowed reduction factor, and is initialized with $k = k_{\text{max}}$. If the threshold is exceeded, it is assumed that the difference between estimated and predicted state vector is due to a target manoeuvre. In that case the reduction factor will be determined by $k = k - 1$. Otherwise, the reduction factor will be determined by $k = k + 1$.

### 7.5.2.2 Scenarios

To test the proposed dynamic reduction approach, targets will be considered which fly a linear trajectory followed by a circular trajectory with a certain transversal acceleration $a_{\text{circ}}$. Given a proper local time-frame with a corresponding $(x, y)$-system in the center of the circle, the trajectory can be modeled by the equations

$$
\begin{align*}
x(t) &= r \times \cos(\omega \times t) \\
y(t) &= r \times \sin(\omega \times t)
\end{align*}
$$

where $r$ is the unknown radius of the circle, $\omega$ is the angular velocity with which the manoeuvre is carried out and $t = 0$ corresponds with the positive $x$-axis. Differentiation eq. 7.63 with respect to the time $t$ produces for the velocity

$$
\begin{align*}
v_x(t) &= \dot{x}(t) = -\omega \times r \times \sin(\omega \times t) \\
v_y(t) &= \dot{y}(t) = \omega \times r \times \cos(\omega \times t)
\end{align*}
$$

The length of the velocity vector is given by

$$v = \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2} = \omega \times r$$

---

\(^9\)In appendix I, the interval corresponding with the target does not manoeuvre is subdivided in two intervals, which are not considered here.
7.5. DYNAMIC REDUCTION

Differentiation of eq. 7.64 produces

\[ \begin{align*}
\dot{\nu}_x(t) &= -\omega^2 \times r \times \cos(\omega \times t) \\
\dot{\nu}_y(t) &= -\omega^2 \times r \times \sin(\omega \times t)
\end{align*} \]  
(7.66)

The length of the acceleration vector is now given by

\[ a = \sqrt{\dot{\nu}_x(t)^2 + \dot{\nu}_y(t)^2} = \omega^2 \times r = \omega \times v \]  
(7.67)

where eq. 7.65 has been used. Rewriting eq. 7.65 produces

\[ \omega = \frac{v}{r} \]  
(7.68)

Substitution of \( a_{\text{circ}} \) in eq. 7.67 and substitution of eq. 7.68 produces

\[ a_{\text{circ}} = \omega^2 \times r = \frac{v^2}{r} \]  
(7.69)

Solving this equation for \( r \) produces finally

\[ r = \frac{v^2}{a_{\text{circ}}} \]  
(7.70)

Assuming the same acceleration \( a_{\text{circ}} \), the radius of the circle is different for targets with a different velocity. Furthermore, the prediction error increases when the target velocity increases. For the thesis, targets with a velocity of \( v = 300 \text{ m/s} \) and \( v = 800 \text{ m/s} \) are considered representative for targets with \( v \in [300, 800] \text{ m/s} \).

7.5.2.3 Determining \( k_{\text{max}} \)

To determine \( k_{\text{max}} \), an experiment has been carried out in which the reduction factor was varied \( (k \in [1, 10]) \). A Monte Carlo approach has been taken, in which for each scenario instantiation 100 runs were carried out. To determine \( k_{\text{max}} \), a constant reduction factor was used within each run. The scenario was derived from the scenario, described in section 7.4. At the end of the linear trajectory, the target carries out a full right turn \((180^\circ)\) with an acceleration of \(10 \text{ m/s}^2\). Two situations are considered: measurement tracking and artificial measurement (AM) tracking. The originating platform is positioned in the origin and is fitted with a single radar sensor. The receiving platform has to re-construct the track formed by the originating platform. It is assumed that the simulated radar sensor has a maximum detection range of around 100 km. The measurement noise is set to 0.03 km in range and to 0.3 degrees in bearing and elevation. The process noise is set to 30 m/s. A Gaussian distribution has been used to generate the sensor measurements.

In the upper part of Fig. 7.10, the target trajectory is shown with an example of measurement tracking and AM (artificial measurement) tracking results. Both kinds of tracking show a good performance. The AM tracking used the dynamical reduction approach with \( k_{\text{max}} = 3 \). The lower part of the figure shows the difference between the (artificial) measurement position and the actual target position. From the scatter plot, the conclusion is drawn that the difference
Figure 7.10: In the top half of the figure, a tracking example is shown ($p_{\chi^2} = 0.999$). The lower half of the figure shows the difference between the measurement (plot) position (artificial measurement position) and the actual target position. The target velocity is $800 \frac{\text{m}}{\text{s}}$.

values in the $x$-direction and $y$-direction are strongly correlated, which could be due to the fact that the target carries out a right turn (circle) at the end of the trajectory. The scatter plot is also used to determine outliers. An outlier is a result, which falls clearly outside the given scatter pattern. The occurrence of a significant number of outliers is a clear indicator that the performance of the used filter is degrading.

To compare the AM tracking results with the measurement tracking results, an appropriate measure is necessary. An obvious choice is to use a comparison measure based on the actual error covariance matrix. A disadvantage of this approach is that possibly occurring bias errors are neglected. Furthermore, large fluctuations in the error covariance matrix may occur from update to update, which makes an comparison very difficult. Instead, the average accuracy is used, which does not have the identified disadvantages. The average accuracy is expressed by the covariance matrix $M_j$ ($j \in \{\text{meas}, \text{AM}\}$), which is estimated
7.5. **DYNAMIC REDUCTION**

by (see appendix E)

\[ M_j = \frac{1}{N} \sum_{i=1}^{N} \delta_j^i \times (\hat{s}_j^i)^T \]  

(7.71)

At update time \( t_i \) the measured difference vector is given by \( \vec{\delta}_j = \hat{s}_j^i - s_o^i \), where \( \hat{s}_j^i \) is the estimated track state vector and \( s_o^i = [x, y, z, v_x, v_y, v_z]^T \) is the actual state vector of the real object.\(^{10}\) To compare the obtained AM tracking accuracy with the original measurement tracking, the ratio between the obtained average tracklet accuracy volume and the average measurement accuracy volume is used, which can be written as

\[ V_r = \frac{V_M}{V_{meas}} = \sqrt{\frac{|M_{AM}|}{|M_{meas}|}} \]  

(7.72)

using the results of appendix G.

In Fig. 7.11 the results of the different simulation runs are shown. For

\[ k \in [9, 10], \]  

the AM tracking degraded in such a way that it was not possible to use the results.\(^{11}\) The conclusion is that the lowest relevant estimated \( V_r \)

\[ \text{Figure 7.11: A comparison is made between artificial measurement tracking, using a constant tracklet reduction factor, and measurement tracking.} \]

\(^{10}\)Due to the fact that it is a simulation experiment, at each update time the actual state vector for the real object is known. It is common practice to use this information to evaluate the tracking performance in a simulation experiment.\(^{15}\).

\(^{11}\)From a tracking perspective, the simulated trajectory is not very difficult. It is expected that for more complicated trajectories, AM tracking degradation may occur even sooner. This is also an argument against the use of high constant reduction factors. Future research is necessary to determine the AM tracking degradation behavior against more difficult trajectories. It is expected, that no problems will occur for reduction factors \( k \in [1, 3] \).
for the linear and the circular trajectory is obtained for \( k = 3 \). For the linear trajectory the obtained \( V_r = 1.7 \) and for the circular trajectory \( V_r = 2.9 \)\(^{12}\). For this value of \( k \), the reduction in information is 3. For the analysis, described in section 7.5.3-7.5.5, \( k_{\text{max}} \) is set to 3. Possibly, \( k_{\text{max}} = 2 \) or \( k_{\text{max}} = 4 \) provide interesting alternatives.

When the reduction factor increases, also the prediction time interval for the composite track, corresponding with a specific target, increases. This means that also the volume of the ellipsoidal gate increases (section 2.4). If targets operate closely together, the probability increases that measurements, produced by a different target, are erroneously assigned to the composite track. To lower this risk, the reduction factor has to be chosen large enough to obtain the required information reduction, but not larger than necessary.

### 7.5.3 Tracklet analysis

The objective of the analysis is to determine if it is possible to obtain dynamically a significant tracklet reduction factor \( k \) at the cost of a limited reduction in average track accuracy, which is expressed by the measure defined by eq. 7.72. In the proposed test to determine if a target manoeuvres (eq. 7.5), the \( \chi^2 \)-distributed variable \( M_{\text{test}} \) is used which is compared with the \( \chi^2 \)-distributed threshold \( r^2 \). In Fig. 7.12 the relation between \( p_{\chi^2} \) and \( r^2 \) is shown. If \( r^2 = 4.6 \),

\[ p_{\chi^2} = 0.2 \]

Figure 7.12: The relation between \( r^2 \) and \( p_{\chi^2} \) is shown in the figure.

20 percent of the \( M_{\text{test}} \) values will be larger than \( r^2 \). This means that there is a risk of 20 percent that a non-manoeuvring target is considered to manoeuvre. In the results presented in this section, the parameter \( p_{\chi^2} \) is used instead of \( r^2 \).

\(^{12}\)A larger value for \( k_{\text{max}} \) is not considered useful, because the obtained artificial measurement results are less accurate.
For the different experiments a number of scenarios has been defined (table 7.1), which are variations of the scenario introduced in section 7.5.2.3. The originating platform is positioned in the origin and is fitted with a single radar sensor. The receiving platform has to re-construct the track formed by the originating platform. It is assumed that the simulated radar sensor has a maximum detection range of around 100 km. The measurement noise is set to 0.03 km in range and to 0.3 degrees in bearing and elevation. The process noise is set to $30 \text{ m s}^{-1}$. A Gaussian distribution has been used to generate the sensor measurements. To vary the sensitivity of the statistical manoeuvre test for erroneous decisions, $p_{\chi^2}$ has been varied in the interval $[0.2, 0.99999]$. This corresponds with $\chi^2 \in [1.0, 25.90]$. A Monte Carlo approach is taken, in which for each scenario instantiation 100 runs are carried out. The tracklet reduction factor is determined dynamically, using the method described in section 7.5.2.1. The requirement for the analysis is that the average estimated volume ratio $V_r \leq 1.7$

For $p_{\chi^2} < 0.999$, the reduction factor may vary continually, which means that it is not possible to use the dynamical reduction factor $k$ to compare the results of the different experiments. To illustrate this point, for scenario 1 the variation of $k$ during a scenario run has been determined, using two different values for $p_{\chi^2}$: 0.5 and 0.999\textsuperscript{13}. The results are shown in Fig. 7.13. For $p_{\chi^2} = 0.5$, the tracklet reduction factor varies continuously, which means that another measure is required, which takes the variation in $k$ during a scenario run into account. For a scenario run the average reduction factor is determined, which is defined as

$$K_{eff} = \frac{N_m - N_b}{N_t} \quad (7.73)$$

where $N_m$ is the number of received measurements, $N_b$ is the number of measurements which does not fall in the positioned correlation gate and $N_t$ is the number of calculated artificial measurements.

### Table 7.1: Defined scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Velocity [m s(^{-1})]</th>
<th>Acceleration [m s(^{-2})]</th>
<th>Cross dist. [m]</th>
<th>$\delta t$ circ. [sec]</th>
<th>$R$ [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>800</td>
<td>10</td>
<td>10,000</td>
<td>251.32</td>
<td>64.000</td>
</tr>
<tr>
<td>2</td>
<td>800</td>
<td>10</td>
<td>50,000</td>
<td>251.32</td>
<td>64.000</td>
</tr>
<tr>
<td>3</td>
<td>300</td>
<td>10</td>
<td>10,000</td>
<td>94.25</td>
<td>9.000</td>
</tr>
<tr>
<td>4</td>
<td>300</td>
<td>10</td>
<td>50,000</td>
<td>94.25</td>
<td>9.000</td>
</tr>
</tbody>
</table>

\textsuperscript{13}Of course other values could have been chosen for the example.
For the linear part of the trajectory, the results are shown in the upper part of Fig. 7.14. A significant increase in $p_{\chi^2}$ results only in a limited increase in volume ratio. The resulting average reduction factor $K_{\text{eff}}$, defined by eq. 7.73, is shown in the upper part of Fig. 7.15. Due to the high average reduction factor and the limited sensitivity of $V_r$ for larger $p_{\chi^2}$ values, the conclusion is that the difference between predicted and estimated state vector is small enough most of the time to declare the target non-maneuvering for larger values of $p_{\chi^2}$.

For the circular trajectory, the results for $V_r$ are shown in the lower part of Fig. 7.14 and for $K_{\text{eff}}$ in the lower part of Fig. 7.15. For $p_{\chi^2} \leq 0.6$, the estimated volume ratio is rather insensitive for the variation in $p_{\chi^2}$ and there the average reduction reduction factor is equal to 1. When the manoeuvrability measure is applied the test is not fulfilled and it is assumed that the target manoeuvres. In the described experiment only a first order Kalman filter is used to track the target during the 180° manoeuvre. Due to the fact that the filter uses a linear target prediction model, significant systematic errors between the...
estimated and predicted state vector will probably occur. If each platform uses IMM filters with appropriate manoeuvre models to produce local and composite tracks, the systematic error will be minimized and higher reduction factors are probably possible\textsuperscript{14}. For $p_{\chi^2} > 0.6$, a trade-off between estimated volume ratio and average reduction factor is possible by varying the $p_{\chi^2}$ value. If for example a $p_{\chi^2}$ value of 0.9 is chosen, for the linear trajectory a volume ratio of 1.4 and an average reduction factor 2.9 is obtained and for the circular trajectory a volume ratio of 2.5 and an average reduction factor of 1.8. If the obtained accuracy is not enough, lower values for $p_{\chi^2}$ have to be considered.

\textsuperscript{14}Artificial measurements are derived from local tracks, which are maintained by measurements of the onboard sensors. Normally, the local tracks will be maintained by an IMM filter using a number of appropriate target (manoeuvre) models (section 2.3). When an artificial measurement is exchanged, it is not necessary to exchange manoeuvre model information.
Figure 7.15: For the linear and the circular trajectory, the estimated average reduction factor $\hat{K}_{\text{eff}}$ is given as function of $p_{\chi^2}$. The crossing distance is 10,000 m. and the target velocity is $800 \frac{m}{s}$.

### 7.5.5 Analysis results

If the tracklet reduction factor for a local track is increased, the accuracy of the derived artificial measurements decreases with respect to the accuracy of the original measurements. The volume of the accuracy hyperellipsoid is determined by the square root of the determinant of the artificial measurement or measurement accuracy covariance matrix $U_c$ [56] in Cartesian space. Given the measurement $\vec{u} = (R, \beta, \varepsilon)$ (see section 7.4) and the measurement accuracy $U$, the covariance matrix $U_c$ is defined by

$$U_c = G \times U \times G^T$$

(7.74)

where

$$G = \begin{pmatrix} 
\cos(\varepsilon) \times \cos(\beta) & \cos(\varepsilon) \times \sin(\beta) & \sin(\varepsilon) \\
-R \times \cos(\varepsilon) \times \sin(\beta) & R \times \cos(\varepsilon) \times \cos(\beta) & 0 \\
-R \times \sin(\varepsilon) \times \cos(\beta) & -R \times \sin(\varepsilon) \times \sin(\beta) & R \times \cos(\varepsilon) 
\end{pmatrix}^T$$

(7.75)
7.5. DYNAMIC REDUCTION

Furthermore, \( R \) is the measurement range, \( \beta \) is the measurement bearing angle and \( \varepsilon \) is the measurement elevation angle. It is easily shown that

\[
| U_c | = R^4 \times \cos(\varepsilon)^2 \times \sigma_R^2 \times \sigma_\beta^2 \times \sigma_\varepsilon^2
\]

(7.76)

for Cartesian space, where \( \sigma_R^2 \), \( \sigma_\beta^2 \) and \( \sigma_\varepsilon^2 \) are the measurement accuracies.

Assume that a measurement \( \mathbf{z} \) and an artificial measurement \( \mathbf{u} \) contain approximately the same range and elevation for the same target. Denote the measurement accuracy by \( \sigma_R^m \), \( \sigma_\beta^m \) and \( \sigma_\varepsilon^m \) and the artificial measurement accuracy by \( \sigma_R^{am} \), \( \sigma_\beta^{am} \) and \( \sigma_\varepsilon^{am} \). When the accuracy of an artificial measurement decreases with respect to the original measurement, it is reasonable to assume that the estimated \( \sigma \) values increase with the same constant of proportionality \( \lambda \) with respect to the original measurement \( \sigma \) values. This means that

\[
\begin{align*}
\sigma_R^m & \approx \lambda \times \sigma_R^m \\
\sigma_\beta^m & \approx \lambda \times \sigma_\beta^m \\
\sigma_\varepsilon^m & \approx \lambda \times \sigma_\varepsilon^m
\end{align*}
\]

(7.77)

Using eq. 7.72, it is easily shown that the ratio \( V_r \) changes as \( V_r = \lambda^3 \) when the tracklet reduction factor increases.

For the analysis, the requirement is that the average estimated volume ratio \( V_r \leq 1.7 \) (section 7.5.3). This means that the different standard deviations of an artificial measurement increase maximally with \( \lambda = 1.7^{\frac{1}{3}} \approx 1.2 \). The results for the different scenarios, specified in table 7.1, are shown in table 7.2. The accuracies for the different estimates are indicated between brackets and are expressed using standard deviations. Average reduction factors in the neighborhood of the initial reduction factor (\( k \approx k_{max} \)) are obtained for the linear part of the different target trajectories. Only low average reduction factors are obtained for the circular part of the different trajectories. The following conclusions are drawn:

- Linear trajectory

A slight increase in estimated volume ratio produces a large increase in average reduction factor. Fulfilling the stated requirement, high average reduction factors are obtained;

- Circular trajectory

Higher average reduction factors are only obtained at a significant increase in estimated volume ratio. Fulfilling the stated requirement, only

<table>
<thead>
<tr>
<th>Scenario</th>
<th>( k_{lin} ) ([0.001])</th>
<th>( k_{circ} ) ([0.1])</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.000 ([0.001])</td>
<td>1.3 ([0.1])</td>
</tr>
<tr>
<td>2</td>
<td>2.970 ([0.010])</td>
<td>1.5 ([0.1])</td>
</tr>
<tr>
<td>3</td>
<td>3.000 ([0.001])</td>
<td>1.5 ([0.1])</td>
</tr>
<tr>
<td>4</td>
<td>3.000 ([0.001])</td>
<td>1.6 ([0.1])</td>
</tr>
</tbody>
</table>
low average reduction factors are obtained. The application of an IMM-filter with appropriate manoeuvre models to derive artificial measurements from local tracks may result in a more promising trade-off\textsuperscript{15}.

To obtain accurate tracking for linear trajectories, a constant high reduction factor (e.g. \( k = 3 \)) is appropriate. When an artificial measurement is distributed, it is not necessary to distribute additional information. To obtain accurate tracking for manoeuvring trajectories, low reduction factors (e.g. \( k = 1 \)) have to be used. Due to the occurrence of manoeuvring targets, a manoeuvre detector is required. An example of a manoeuvre detector has been proposed in section 7.5.2.1.

\subsection*{7.6 Communication and processing limitations}

Data reduction is important in situations, where there are communication bandwidth or processing power limitations. The objective of the analysis was to determine if it is possible to maintain an accurate composite track with artificial measurements, which were derived from a local track maintained with measurements from a single sensor. For the results, given in table 7.2, the requirement was that the average estimated volume ratio \( V_r \leq 1.7 \). Using the approach, described in the sections 7.5.3-7.5.5, it is possible to determine fixed reduction factors which guarantee required average composite track accuracies against a set of defined targets. In this section it is assumed that tracklets may contain contributions from more than one sensor and that composite tracks are maintained using the tracklets produced by one or more participants. An approach is described how to reduce the communication load and platform processing load.

\subsection*{7.6.1 Useful tracklet reduction factors}

Assume that \( n \) sensors contribute measurements to a certain local track \( t \). It is assumed that each sensor collects a maximum of one measurement from each target during one scan. Our objective is to produce artificial measurements which are accurate enough, to guarantee that a composite track is maintained with at least the average accuracy of a composite track maintained only with the most accurate sensor on board of any one platform contributing to the task force\textsuperscript{16}. It has been shown in section 7.5.5, that it is possible to obtain average reduction factors \( \bar{k} \approx k_{\text{max}} \), \( k_{\text{max}} = 3 \), against linear target trajectories. Without special manoeuvre models, a reduction factor \( \bar{k} \approx 1 \) is more appropriate when a target executes manoeuvres. To determine if a target manoeuvres or not, a manoeuvre detector is required. An example of a manoeuvre detector is described in section 7.5.2.1.

The local track \( t \) is maintained with measurements from \( n \) sensors. In our case the most accurate sensor is sensor \( i \). Assume that the probability of target detection for sensor \( j \) is \( P_{d}^{j} (j \in [1, n]) \). The probability to detect a target,\textsuperscript{15}

\textsuperscript{15}It is assumed that in that case all platforms use an IMM-filter with appropriate manoeuvre models to maintain local and composite tracks, which means that it is not necessary to exchange artificial measurements with manoeuvre model information.

\textsuperscript{16}A possible alternative is to require that a minimum number of artificial measurements per time unit is determined to minimize the communication load. A disadvantage of this approach is that it is not possible to guarantee a minimum composite track accuracy.
represented by a certain local track, depends on both the sensor and the target type and varies with time [120]. If track \( t \) is only maintained with measurements from sensor \( i \), \( k_{\text{max}}^s \) measurements have to be collected to obtain the required accuracy \(^{17}\). The average number of scans to obtain \( k_{\text{max}}^s \) measurements for sensor \( i \) is determined by the equation

\[
k_{\text{max}}^s = P_d^i \times n_i
\]  

(7.78)

The average number of scans \( n_i \), which has to be made by sensor \( i \), is given by

\[
n_i = \frac{k_{\text{max}}^s}{P_d^i}
\]  

(7.79)

The scan time of sensor \( i \) is given by \( \Delta t_i \). The average time interval, corresponding with eq. 7.79, is \( n_i \times \Delta t_i \). This time interval is used as reference interval. Using eq. 7.79, the average number of measurements, collected by sensor \( j \), during this time interval is given by

\[
n_j = P_d^j \times \frac{n_i \times \Delta t_i}{\Delta t_j}
\]  

(7.80)

The total number of measurements in our reference interval, contributing to the same local track, is given by

\[
n_t = \sum_{l=1}^{n} n_l
\]  

(7.81)

The objective, stated at the beginning of this section, is certainly fulfilled when each time \( k_{\text{max}}^s \) measurements of sensor \( i \) are collected, a tracklet is generated from all collected measurements, transformed to an artificial measurement and distributed. This guarantees that at least the required accuracy is obtained. Due to the fact that during the reference time interval \( n_t \) measurements have been collected, for the multisensor case the maximum tracklet reduction factor \( k_n \) is given by

\[
\frac{k_n = \left\lfloor k_{\text{max}}^s \times \frac{n_t}{n_i} \right\rfloor}{P_d^i \times \Delta t_i}
\]  

(7.82)

where \( \left\lfloor \cdots \right\rfloor \) rounds off to the nearest integer \( \rightarrow -\infty \). Under the condition that the most accurate sensor determines the times a tracklet is calculated, it is guaranteed that the stated requirement is fulfilled for reduction factors \( k_{\text{max}} \in [k_{\text{max}}^s, k_n] \).

Each platform carries out the same calculation to determine the tracklet interval. For the task force, this approach guarantees that composite tracks are formed which are at least as accurate as a composite track maintained only with artificial measurements produced by the sensor which at a certain moment is the most accurate sensor.

\(^{17}\)The superscript \( s \) indicates that the track is assumed to be maintained with measurements from a single sensor.
7.6.2 Communication bandwidth limitations

Fig. 7.16 shows an example of a possible task force, consisting of 5 contributors. Each contributor uses a point-to-point channel to transmit information to each of the other contributors in its line-of-sight. Each one-way channel has a bandwidth of $N_b\text{ Bytes/second}$. As example we consider contributor A, which has $n_t$ local tracks stored in the local track database. The artificial measurements, produced during a certain time interval $\Delta t$, are collected together and transmitted in one batch with a standard header of $N_h$ bytes. The transmission process has to be finished before the new set of artificial measurements is collected and has to be transmitted. This means that the information has to be transmitted during the same time interval $\Delta t$. Assume that during the considered time interval $n_m$ tracks are corresponding with manoeuvring targets and $n_l - n_m$ tracks are corresponding with non-manoeuvring targets. Assuming the same constant reduction factor $k$ ($k > 1$) for each non-manoeuvring track, the number of artificial measurements constructed for non-manoeuvring local tracks during the time interval is given by

$$n_1 = \sum_{t=1}^{n_l - n_m} \frac{n_t}{k} = \frac{1}{k} \sum_{t=1}^{n_l - n_m} n_t = \frac{1}{k} \times N_{tot}$$

(7.83)

where $n_t$ is defined by eq. 7.81 and $N_{tot} = \sum_{t=1}^{n_l - n_m} n_t$. The total number of artificial measurements, which are produced for manoeuvring local tracks is now given by

$$n_2 = \sum_{t=1}^{n_m} n_t$$

(7.84)

For an artificial measurement, it suffices to distribute the measurement vector and the diagonal of the constructed covariance matrix (section 7.4). If 8
7.6. COMMUNICATION AND PROCESSING LIMITATIONS

bytes (double precision) are required for an element of the state vector or the covariance matrix, only $48 \times \alpha$ bytes are necessary to distribute an artificial measurement, where $\alpha$ is the average data compression factor (section 7.1). The total number number of artificial measurements in the batch, which has to be distributed in the time interval $\Delta t$, is given by

$$N_{ba} = n_1 + n_2 = \frac{1}{k} \times N_{tot} + n_2$$  \hspace{1cm} (7.85)

For the available communication channels, the expected load during the time interval $\Delta t$ does not exceed the available channel bandwidth if

$$\frac{N_{ba}}{\Delta t} \times 48 \times \alpha + N_h \leq N_b$$  \hspace{1cm} (7.86)

To prevent communication channel overload, the reduction factor $k$ is determined by the equation

$$\frac{1}{k} \times N_{tot} + n_2 \leq (N_b - N_h) \times \frac{\Delta t}{48 \times \alpha}$$  \hspace{1cm} (7.87)

Solving the equation, produces the real required minimal reduction factor

$$k \geq \frac{N_{tot}}{(N_b - N_h) \times \frac{\Delta t}{48 \times \alpha} - n_2}$$  \hspace{1cm} (7.88)

The reduction factor can only take integer values. The requirement that the bandwidth is not exceeded, means that the final solution is given by

$$k \geq \lceil \frac{N_{tot}}{(N_b - N_h) \times \frac{\Delta t}{48 \times \alpha} - n_2} \rceil$$  \hspace{1cm} (7.89)

where $\lceil \cdots \rceil$ rounds off to the nearest integer $\to \infty$. Using section 7.6.1, the following possibilities can be distinguished:

- $k < k_{max}^s$
  In this case the bandwidth of the different communication channels is sufficient to transmit the batch of artificial measurements in the required time interval $\Delta t$;

- $k \in [k_{max}^s, k_n]$
  When the reduction factor is increased to the calculated value, the calculated artificial measurements are accurate enough to maintain a composite track, with the required properties;

- $k > k_n$
  If the reduction factor is too high, additional measures are necessary. A possible solution is to distribute artificial measurements with a reduction factor $k \in [k_{max}^s, k_n]$ for the most threatening targets. For the less threatening targets a much higher reduction factor is appropriate in that case.

The information in the SIAP can be used to engage the hostile targets as effectively as possible (section 1.1). Using the importance of the different
high value units, the available weapons and dynamic characteristics of
the different hostile targets, it is possible to define a distributed threat
evaluation function to assign to each target a unique threat value. This
subject is outside the scope of this thesis.

7.6.3 Processing power limitations

A composite track in the single integrated air picture (SIAP) is maintained with
artificial measurements received from a significant number of contributors. It
is assumed that each contributor uses the method described in section 7.6.2 to
guarantee that the bandwidth of the available communication channels is not
exceeded. This means that received artificial measurements are accurate enough
to maintain a composite track with sufficient quality. For each communication
channel, new information contained in a batch is received each time interval \( \Delta t \) (section 7.6.2). Each batch has to be processed before the new batch arrives. In
section 6.2, it has been shown that even fast heuristics like SGTS have an upper
bound for the processing time, which is nonlinear in the number of artificial
measurements in a batch and the size of the sliding window. The processing
resources may not be capable of processing all received information within the
required time interval. This may result in unacceptable processing delays. The
remainder of this section serves to develop an approach for dealing with this
situation.

It is assumed that the (artificial) measurement probability density function
is Gaussian. The measurement vector \( \mathbf{z}_m \) of an (artificial) measurement and the
accuracy or error covariance matrix \( R_m \) (see section 7.4) specify a single cloud
of possible real target positions \( \mathbf{z} \), expressed in polar coordinates. The center
of the cloud is determined by the measurement vector \( \mathbf{z}_m \) and the shape of the
cloud is determined by the covariance matrix \( R_m \). The contours of constant
density are hyperellipsoids of constant Mahalanobis distance to \( \mathbf{z}_m \) [56]. The
Mahalanobis distance has been discussed in section 2.4. The hyperellipsoid is
defined by

\[
(\mathbf{z} - \mathbf{z}_m)^T R_m^{-1} (\mathbf{z} - \mathbf{z}_m) = r^2  
\]

(7.90)

The threshold \( r^2 \) determines the risk to make an erroneous decision and can be
read from the \( \chi^2 \) table. The volume of the hyperellipsoid is given by [56]

\[
V = V_d \times | R_m |^\frac{d}{2} \times r^d  
\]

(7.91)

where \( V_d \) is the volume of a \( d \)-dimensional unit hypersphere:

\[
V_d = \left\{ \begin{array}{ll}
\frac{\pi^{d/2}}{(2)\Gamma(d/2)} & \text{d even} \\
2^{d/2} \times \frac{d-1}{d} \times \Gamma\left(\frac{d+1}{2}\right) & \text{d odd}
\end{array} \right.  
\]

(7.92)

Thus, for a given dimensionality, the scatter of the samples varies directly with
\( | R_m |^\frac{d}{2} \).

Given a pre-defined \( r^2 \)-value, the volume of the hyperellipsoid is used as
a measure to express the accuracy of an artificial measurement. For Carte-
sian space, the hyperellipsoid volume increases proportionally with \( R^2 \) (section
7.5.5). This means that a very accurate artificial measurement from a local track
at a large distance may have a larger ellipsoid volume than a very inaccurate artificial measurement from a local track at a short distance.

For each of the $n_i$ sources, contributing to a certain composite track $i$, the measure

$$M_i = \sum_{j=1}^{n_i} \frac{1}{V_j}$$

(7.93)

is calculated, where $V_j$ is defined by eq. 7.91. If there are processing power problems, the approach is to determine for all composite tracks the relative contribution of the artificial measurements to the different measures, defined by eq. 7.93, and to discard the source with the lowest relative contribution, defined by

$$\min_{i=1}^{n_{ct}} \min_{j=1}^{n_i} \frac{1}{M_i}$$

(7.94)

where $n_{ct}$ is the number of composite tracks in the SIAP. This process is repeated until the the processing power problems have been resolved. The test is carried out for each incoming batch of measurements.

### 7.7 Conclusions

In this chapter a new method for reducing the load on communication channels has been introduced. Compared with the classical tracklet methods ($k \geq 5$), the new method guarantees information reduction already when the number of measurements taken into account is $\geq 2$.

In the sections 7.5.3-7.5.5 an analysis has been carried out to determine the effectiveness of the approach, using a number of targets considered representative for the application domain of interest. For the analysis, only the situation has been considered that a single sensor was observing a target and a local track was constructed using measurement tracking. The track was reconstructed on board of a second platform. It has been shown that tracks maintained with artificial measurements are operationally equivalent with tracks maintained with the original measurements.

When measurements from more than one sensor are used to maintain a local track, significant higher reduction factors than for the single sensor case are possible. In section 7.6.1 it was shown, that it is possible to determine reduction factors which guarantee that composite tracks are formed which are at least as accurate as a composite track maintained with only the artificial measurements produced by the sensor which at a certain moment is the most accurate sensor within the task force.

In section 7.6.2, the case has been considered that the communication channels have bandwidth limitations. To prevent communication overload, a reduction factor is calculated to reduce the amount of information to be distributed. To guarantee that composite tracks are formed with the required minimum accuracy, it is checked that the calculated reduction factor fulfills stated reduction requirements. The operational procedures for dealing with the situation that this is not the case are beyond the scope of this thesis.

Finally, the formation of composite tracks with limited processing resources has been considered, where each composite track contains contributions from more than one source. It has been shown that the processing time depends
strongly on the number of artificial measurements to be processed. The proposed solution is to reduce the processing load by discarding the contribution of the source with the lowest relative accuracy contribution to a composite track. This process is repeated until the processing power problems have been resolved.