Performance of PDAF-based Tracking Methods in Heavy-Tailed Clutter

Edmund F. Brekke
University Graduate Center
Kjeller, Norway.
edmund@unik.no

Oddvar Hallingstad
University Graduate Center
Kjeller, Norway.
oh@unik.no

John H. Glattetre
Kongsberg Maritime
Horten, Norway.
jhg@kongsberg.com

Abstract – Harbor surveillance above and below the sea surface depends on sensors such as surveillance radar and multibeam sonar. These sensors attempt to detect and track moderately observable targets such as small boats or human divers in environments which often are characterized by heavy-tailed backgrounds. This paper provides simulation results which quantify the inevitable performance loss encountered in heavy-tailed environments. The results show that the performance loss can be reduced by accounting for heavy-tailedness in the detection and tracking processes, and by the utilization of Amplitude Information (AI). Two new amplitude likelihoods developed in a preceding paper come favorably out of this comparison. Furthermore, the evaluation of the Modified Riccati Equation (MRE) is outlined for the combination of AI and heavy-tailed clutter. The MRE can be used to decide the false alarm rate for the detection process preceding target tracking.

Keywords: Data association, Detection, Feature Aided Tracking

1 Introduction

This paper is a direct continuation of [4], in which tracking methods utilizing Amplitude Information (AI) in heavy-tailed clutter were proposed. In this paper we investigate the performance of these methods using two approaches:

1. Evaluation of expected performance using the Modified Riccati Equation,

2. Implementation on simulated data.

The Modified Riccati Equation (MRE) [5] is a generalization of the well known Riccati equation [2]. While the Riccati equation governs the covariance evolution of the Kalman filter, the MRE predicts the expected performance of the Probabilistic Data Association Filter (PDAF) [1]. In this paper we extend this kind of analysis to the Probabilistic Data Association Filter with Amplitude Information (PDAFAI).

The main tool in the analysis of tracking methods is implementation on simulated data. The actual performance of a tracking method depends on several factors which cannot possibly be reduced to a single equation. Ideally the tracking method should be tested in the real world. However, performance measures such as the occurrence of track-loss can hardly be investigated through real-world experiments. Since (hopefully) track-loss only occurs in a small fraction of all experiments, one must carry out thousands of experiments to obtain statistically valid information.

In this paper we apply a simulator recently developed in [10] to investigate the impact of heavy-tailed clutter on target tracking. The tracking methods to be tested are various versions of the PDAF and the PDAFAI which differ in the treatment of the background noise. To the best of our knowledge, such an analysis does not yet exist in the open literature.

The paper is organized as follows. Section 2 provides a description of the tracking problem and the methods used to solve it. It is mostly a cursorial summary of [4], and the reader should therefore read [4] before reading this paper. In Section 3 we discuss the choice of design false alarm rate by means of the MRE. In Section 4 a simulation scenario for testing the tracking methods is described. Test results show substantial improvements due to the new amplitude likelihoods developed in [4]. A conclusion is given in Section 5.

2 Summary of problem setting

We study the single-target tracking problem as posed by the Bayes equations

\[
p(x_k | Z^{k-1}) = \int p(x_k | x_{k-1}) p(x_{k-1} | Z^{k-1}) \, dx_{k-1},
\]

\[
p(x_k | Z^k) \propto p(Z_k | x_k) p(x_k | Z^{k-1}).
\]  

(1)

The kinematic state \( x_k \) is assumed to be propagated according to a linear model

\[
x_k = F x_{k-1} + v_k, \quad v_k \sim \mathcal{N}(0,Q),
\]  

(2)
where

\[ F = \begin{bmatrix}
1 & T & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & T \\
0 & 0 & 0 & 1 \\
\end{bmatrix}, \]

\[ Q = \begin{bmatrix}
\frac{\sigma^2}{4} T^3 & \frac{\sigma^2}{2} T^2 & 0 & 0 \\
\frac{\sigma^2}{2} T^2 & \sigma_v T & 0 & 0 \\
0 & 0 & \frac{\sigma^2}{2} T^3 & \frac{\sigma^2}{2} T^2 \\
0 & 0 & \frac{\sigma^2}{2} T^2 & \sigma_v T \\
\end{bmatrix}. \]

The measurement set \( Z_k = \{ \zeta_k(1), \ldots, \zeta_k(m_k) \} \) contains measurement vectors extracted from a radar or sonar image with parametrization

\[ \zeta_k(i) = \left[ z^T_k(i), a_k(i), q^T_k(i) \right]^T. \]

Its kinematic part \( z_k(i) \) is for the target-originating measurement related to the state by

\[ z_k = H x_k + w_k, \quad w_k \sim \mathcal{N}(0, R), \]

where \( H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \), \( R \) is given by a conversion from polar to cartesian coordinates as described in [1]. The amplitude \( a_k(i) \) has the probability density function (PDF) \( p_1(a_k(i)|d, q_k(i)) \) if \( \zeta_k(i) \) originates from the target, otherwise it has the PDF \( p_0(a_k(i)|q_k(i)) \). These densities are elaborated in [4]. Under standard assumptions the likelihood in (1) can be written explicitly as

\[
p(Z_k|x_k) = \frac{\gamma_0}{(VP_{FA})^{m_k}} \prod_{i=1}^{m_k} p_0(a_k(i)|q_k(i)) \\
+ \frac{\gamma_1}{P_D P_G (VP_{FA})^{m_k-1}} \\
\cdot \sum_{i=1}^{m_k} \mathcal{N}(z_k(i); H x_k, R) \\
\cdot p_1(a_k(i)|d, q_k(i)) \prod_{j \neq i}^{m_k} p_0(a_k(j)|q_k(i)).
\]

The PDAF and the PDAFAI [1] approximate the predicted density by a single Gaussian,

\[ p(x_k|Z^{k-1}) \approx \mathcal{N}(\hat{x}_{k|k-1}, P_{k|k-1}). \]

By combining (8) and (9) in (1) one can express the posterior \( p(x_k|Z^k) \) as a Gaussian mixture, which is then collapsed into a single Gaussian before propagating it to the next time step. This leads to the Kalman Filter-like equations

\[ \hat{x}_{k|k-1} = F \hat{x}_{k-1|k-1}, \]

\[ P_{k|k-1} = FP_{k-1|k-1}F^T + Q \]

\[ \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \sum_{i=1}^{m_k} \beta_k(i) \nu_k(i) \]

\[ P_{k|k} = P_{k|k-1} - (1 - \beta_k(0)) K_k S_k K^T_k + \tilde{P}_k \]

The association weights \( \beta_k(i) \) are given by

\[ \beta_k(i) = \left\{ \begin{array}{ll}
\frac{e(i)}{b + \sum_{j=1}^{m_k} e(j)} & i = 1, \ldots, m_k \\
0 & i = 0
\end{array} \right. \]

where

\[ K_k = P_k|k-1 H^T S_k^{-1}, \]

\[ S_k = HP_k|k-1 H^T + R_k \]

\[ \nu_k(i) = z_k(i) - H \hat{x}_{k|k-1} = z_k(i) - \hat{z}_{k|k-1} \]

\[ \tilde{P}_k = K_k \sum_{i=1}^{m_k} \beta_k(i) \nu_k(i) \nu_k(i)^T - \nu_k(\nu_k)^T K_k^T \]

The zero’th weight \( b \) is for the case of a Poisson clutter cardinality distribution and a correspondingly parametric PDAF given by

\[ \gamma_0 = P_D P_G [P_D P_G m + (1 - P_D P_G) \lambda V]^{-1} \]

\[ \gamma_1 = (1 - P_D P_G) [P_D P_G m + (1 - P_D P_G) \lambda V]^{-1} \]

\[ \Rightarrow b = \frac{2 \pi / \gamma}{\lambda V (1 - P_D P_G) / (c_n P_b)}. \]

For the more practical non-parametric PDAF the clutter intensity \( \lambda V \) is replaced by the number of validated measurements \( m_k \).

For the conventional PDAF without AI, the likelihood ratio \( l(a|\ldots) \) is just unity. For the PDAFAI we discuss three likelihood ratios of which the latter two were proposed in [4]. These are the conventional Rayleigh likelihood ratio,

\[
l(a|d, \eta) = \frac{P_{FA}}{P_b} \cdot \frac{\eta}{\eta + d} \cdot \exp \left( \frac{a^2 d}{2(\eta + d)} \right),
\]

the conservative Rayleigh likelihood ratio,

\[
l(a|d, \eta) = \frac{P_{FA}}{P_b} \cdot \frac{\Gamma(M)}{\eta^M (\eta + d)^M} \cdot \int_0^\infty \frac{1}{(\eta + d)^M} \exp \left( \frac{-\eta M}{\eta} - \frac{a^2}{2(\eta + d)} \right) d\eta,
\]

and the \( K \)-distribution likelihood ratio,

\[
l(a|d, \nu, b) = \frac{P_{FA}}{P_b} \cdot \frac{(a \sqrt{b})^{-\nu}}{4 K_{\nu-1} \left( \frac{2a^2}{\nu b} \right)} \cdot \int_0^{\infty} \frac{\eta^{\nu-1}}{\eta + d} \exp \left( \frac{-\eta}{b} - \frac{a^2}{2(\eta + d)} \right) d\eta.
\]
3 The MRE

The MRE [5] is a generalization of the Riccati equation [2] to single-target data association problems. The key idea of the MRE is a so-called information reduction factor

\[ q_2(S_k : P_D, P_{FA}) \]

which quantifies how much information is lost due to non-uniform detection probability and clutter.

The MRE is an important tool because it provides a connection between the detection process and the tracking process. For any detection scenario the probability of detection \( P_D \) is a function of the false alarm rate \( P_{FA} \). This function is known as the Receiving Operating Characteristics (ROC) and traces a curve in the plane spanned by \( P_{FA} \) and \( P_D \). An optimality criterion for the determination of the optimal false alarm is given by minimizing the steady-state output

\[ e(P_{FA}) = \sqrt{p_{21,\infty}^2 + p_{22,\infty}^2} \]

of the MRE along the ROC curve. In this section we carry out such an analysis for the PDAF and the PDAFAI in heavy-tailed clutter. Following [5] we write the MRE as

\[
P_{k|k-1} = FP_{k-1|k-1}^T + Q \]

\[
P_{k|k} = P_{k|k-1} - q_2(S_k : P_D, P_{FA})K_k S_k K_k^T \]

\[ S_k = HP_{k|k-1}H^T + R. \]  

(16)

The information reduction factor \( q_2 \) is given by

\[ q_2 = \sum_{m=1}^{\infty} \eta_m(S_k) P\{m\} \]

\[ \eta_m(S_k) S_k = E \left[ \sum_{i=1}^{m} \beta(i)^2 \nu_k(i) \nu_k(i)^T \right] m, Z^{k-1}. \]

We find \( \eta_m \) by averaging over the probability density

\[ p(\nu_k(1), \ldots, \nu_k(m_k), a_k(1), \ldots, a_k(m_k) | m_k, Z^{k-1}) \].

A rather tedious calculation along the lines of [6] leads to the following expression,

\[ \eta_m(S_k) = \left( \frac{m^2 \gamma_1 c_{n_k}}{P_D P_{FA} m^{\nu_1 - 1} (2\pi)^{n_x/2}} \right)^{m-1} \]

\[ \cdot \int_0^g \cdots \int_0^g \int_0^\infty \cdots \int_0^\infty r_1^{n_x-1} \exp \left( -\frac{1}{2} r_1^2 \right) \]

\[ \cdot \prod_{i=2}^{m} p_1(a_i) \prod_{i=2}^{m} p_0(a_i) \]

\[ \cdot \exp \left( -\frac{1}{2} r_2^2 \right) l(a_1) \]

\[ b + \sum_{m=1}^{\infty} \exp \left( -\frac{1}{2} r_2^2 \right) l(a_i) \]

(17)

Notice that \( q_2 \) depends on \( S \) through \( \gamma_1 \) and \( b \) which, as can be seen from (12), depend on \( \lambda \) and the gate volume \( V = c_{n_k} g^{n_x} \sqrt{|S|} \). The only way to evaluate (17) is by importance sampling. This can be done by averaging the fraction on the last line over distributions proportional to the preceding factors. The details are omitted for brevity.

In order to use this machinery we first evaluate \( q_2 \) over a suitable grid of values for the Signal-to-Noise Ratio (SNR, equivalent to \( P_D \)), \( \lambda V \) and \( P_{FA} \). In the non-AI case we can evaluate \( q_2 \) as a function of \( P_D \) and \( \lambda V \) only, but in the AI case \( P_{FA} \) must be included as well. Then (16) is iterated with the exact value of \( q_2 \) obtained by interpolation for 100 time steps in order to approximate the stationary covariance \( P_\infty \). In some cases the MRE diverges, implying that tracking cannot be carried out with this false alarm rate for the given SNR. The iteration is carried out with matrices as given in Section 2. The concrete values used can be found in Table 2. As an approximation to the scenario described in Section 4, we set the measurement noise matrix to

\[ R = \begin{bmatrix} (0.8m)^2 & 0 & 0 \\ 0 & (1m)^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \]

The initial covariance matrix is set to

\[ P_0 = 2Q + \begin{bmatrix} R & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \]

Figure 1 shows the expected RMSE obtained from (16) for a target in \( K \)-distributed clutter, where \( p_0(\ldots) \) and \( p_1(\ldots) \) are given by (6) and (8) in [4]. It can be seen that \( e(P_{FA}) \) of trackers not utilizing AI diverges for very high false alarm rates, while the AI-trackers are more robust. If the false alarm rate is low, there is little improvement to gain from the use of AI. The advantage of AI is to make the tracker capable of exploiting a higher \( P_{FA} \), with correspondingly higher \( P_D \), than the conventional PDAF would be able to handle.

In all cases it is indicated that a rather high false alarm rate will yield the best performance, and this is most critical for the more heavy-tailed scenario (\( \nu = 1 \)). To make sure that we stay out of the divergent regions, it seems reasonable to choose a design false alarm rate \( P_{FA} = 0.005 \). This value will be used throughout Section 4. The reader should be aware that this analysis is quite optimistic since it ignores sensor blurring and estimation uncertainties.
4 Test design and simulations

In this section we test several PDAF-based trackers on more or less heavy-tailed data. The trackers to be tested are:

T1: The standard PDAF with Rayleigh detector,

T2: The standard PDAF with $K$-based detector,

T3: The PDAFAI with Rayleigh detector and standard Rayleigh likelihood,

T4: The PDAFAI with Rayleigh detector and conservative Rayleigh likelihood,

T5: The PDAFAI with $K$-based detector and $K$-based likelihood,

Here Rayleigh detector means that detections are provided by a conventional cell averaging CFAR as in Section 3.1 of [4]. The $K$-based detector determines its threshold according to local $K$-parameter estimates as described in Section 3.2 of [4]. For standard Rayleigh, conservative Rayleigh and $K$-based likelihoods the AI is processed using (13), (14) and (15) respectively. Notice that the Rayleigh-based trackers do not have access to any information conveyed by the $K$-distribution framework.

4.1 Background simulation

In order to provide our simulated data with correlations similar to those encountered in the real world we use the compound $K$-model:

$$ p(\eta; \nu, b) = Ga(\eta; \nu, b/2) \propto \eta^{\nu-1} \exp\left(-\frac{2\eta}{b}\right) \frac{(b/2)^\nu}{\Gamma(\nu)} $$

$$ p(a|\eta) = Ra(a; \eta) \Rightarrow p(a; \nu, b) = KPDF(a; \nu, b). $$

The “texture” $\eta$ is supposed to have a “global” Autocorrelation Function (ACF) $R_G(z)$ which accounts for inhomogeneities in the medium or scattering surface such as wave patterns [10]. In order to also account for correlations induced by signal processing we let $a$ be correlated according to a “local” ACF $R_L(z)$. For notational brevity we let $z$ both refer to 3-tuples $[\rho, \phi, k]^T$ and to the “lags” between them. The coordinates $\rho$ and $\phi$ refer to range and bearing measured in resolution cells.

The correlated Gamma process $\eta(z)$ is simulated by generating a correlated Gaussian process $\xi(z)$ and then mapping it into a Gamma process using a so-called Memoryless Non-Linear Transform (MNLT). In order to do this one must determine what ACF $\xi$ should follow in order to provide $\eta$ with the desired ACF. The MNLT is given by solving the following equation with respect to $\eta$:

$$ \int_{-\infty}^{\xi} \mathcal{N}(u; 0, 1)du = \int_0^\eta Ga(\nu; \nu, b/2)dv. $$

This amounts to solving

$$ 1 - \frac{1}{\Gamma(\nu)} \gamma\left(\frac{2\eta}{b}, \nu\right) - \text{erfc}\left(\frac{\xi}{\sqrt{2}}\right)/2 = 0, $$

where $\gamma(\cdot, \cdot)$ is the lower incomplete Gamma function and $\text{erfc}(\cdot)$ is the complementary error function as defined in [7].

The MNLT is in practice carried out using linear interpolation between tabulated values of $\xi$ and $\nu$. In Figure 2 the MNLT is plotted for various values of $\nu$. The treatment of correlations is done using an expansion in Hermite polynomials $H_n(\xi)$. Denoting the ACF of $\xi$ by $R_G(z)$, it is shown in [10] that

$$ R_G(z) = \frac{1}{\pi} \sum_{n=0}^{\infty} \frac{R_G(z)^n}{2^n n!} \left( \int_{-\infty}^{\infty} \exp(-z^2)H_n(\xi) \right) \text{GammaCDF}^{-1}\left(1 - \frac{\text{erfc}(\xi)}{2}; \nu, b/2\right) d\xi $$

$$ \Delta = \sum_{n=0}^{\infty} \alpha_n R_G(z)^n. $$

The integrals in (20) depend on $\nu$, and must be evaluated off-line for a suitable grid of values of $\nu$ before the MNLT can be constructed. When this is done we have the Gamma correlation function $R_G(z)$ expressed as a polynomial in the Gaussian correlation function $R_G(z)$. Inverting this polynomial for all lags $z$ yields a mapping from the desired Gamma ACF to the Gaussian ACF used in simulation. The simulation of correlated Gaussian noise is done using standard techniques, and will not be elaborated.

To summarize, we generate correlated $K$-distributed noise with a given shape parameter $\nu$ and ACF $R_G(z)$ using the following recipe:

1. Store the mappings $\xi \rightarrow \eta$ from (19) in an interpolation table.

2. Store the coefficients from (20) in another table.

3. Convert the desired Gamma ACF $R_G(z)$ into the corresponding Gaussian ACF $R_G(z)$ using the output from step 2.

4. Draw Gaussian noise $\xi(z)$ with ACF $R_G(z)$.

5. Convert $\xi(z)$ into a Gamma process $\eta(z)$ with shape parameter $\nu$ using the interpolation table from step 1.

6. Draw complex unit-variance Gaussian noise $w^*(z)$ with ACF $R_L(z)$ according to (21).

7. Modulate the complex Gaussian noise with the Gamma process: $w(z) = w^*(z) \cdot \sqrt{\eta(z)}$.

The amplitude $a(z) = |w(z)|$ is $K$-distributed with the desired correlation properties. A simple Markov model is used as global ACF:

$$ R_G(z) = \exp\left(-\frac{|\phi|}{L_{\Gamma}}\right) \exp\left(-\frac{|\phi|}{L_{\Gamma}}\right) \exp\left(-\frac{|k|}{4}\right). $$
The local ACF \( R_L(z) \) is related to the sensor’s point spread function (PSF), which can be deduced from the beamforming and the reception processes. To keep things simple we approximate the range PSF as a sinc function and the bearing ACF as a Gaussian,

\[
h(r, \phi) = h_p(r) h_\phi(\phi) = \text{sinc} \left( \frac{\rho}{A} \right) \exp \left( - \frac{\phi^2}{2B^2} \right).
\]

This implies that the local ACF in the above recipe is

\[
R_L(r, \phi) = \text{sinc} \left( \frac{\rho}{A} \right) \exp \left( - \frac{\phi^2}{2B^2} \right).
\]

(21)

The coordinates \( \rho \) and \( \phi \) represent the sensor’s coordinate system normalized to the sensor’s resolution.

To mimic jamming and other sudden disturbances we have multiplied the background noise \( w(z) \) by two in time steps 5, 9, 15, 32, 37, 38 and 45. Thus the methods to be tested are not allowed to rely on temporal stationarity or correlation of the background.

### 4.2 Simulation of target signal

When a target is present it is mapped into the data by the PSF \( h(r, \phi) \). For any cell \( j \) in the vicinity of \( z_k = h(x_k) \) its complex value \( z_k^j \) is related to the state \( x_k \) by

\[
z_k^j = h^j(x_k)s_k + w_k^j
\]

\[
= h(\rho - \rho(x_k), \phi - \phi(x_k))s_k + w_k^j
\]

(22)

where \( s_k \sim N(0, \sigma) \) under the Swerling I model and \( w_k^j \) is the complex noise from Section 4.1. The notations \( \rho^j \) and \( \phi^j \) refer to the centroids of cell \( j \).

### 4.3 Simulation of target kinematics

The PDAF and PDAFAI are implemented using the linear kinematics model given by (3) and (4). Nevertheless, as our simulation model we use a more complicated curvilinear model developed in [3]. The curvilinear model is arguably more realistic; modeling colored process noise, maneuvers and sudden accelerations. In the curvilinear model the state is parameterized as \( x_k = [x_k, y_k, \dot{x}_k, \dot{y}_k, a_{tk}, a_{nk}] \) where \( a_{tk} \) is acceleration tangential to the target trajectory and \( a_{nk} \) is acceleration perpendicular to the trajectory. The kinematic process model can then, subject to some mild regularity conditions elaborated in [3], be written as

\[
x_k^{t+1} = \begin{bmatrix}
F & G_t(x_k^t) & G_n(x_k^t) \\
0 & \beta_t & 0 \\
0 & 0 & \beta_n
\end{bmatrix}
\begin{bmatrix}
x_k^t + v_k^t
\end{bmatrix}.
\]

(23)

The matrices \( G_t(x_k^t) \) and \( G_n(x_k^t) \) are explicitly given by

\[
G_t(x_k^t) = \begin{bmatrix}
- \frac{1}{\omega_k} \cos \varphi_k + 1 & \frac{1}{\omega_k} \cos \phi_k - \frac{1}{\omega_k} T \cos \phi_k \\
\frac{1}{\omega_k} \sin \varphi_k + 1 & \frac{1}{\omega_k} \sin \phi_k \\
\frac{1}{\omega_k} \sin \varphi_k + 1 & \frac{1}{\omega_k} \sin \phi_k + \frac{1}{\omega_k} T \cos \phi_k \\
\frac{1}{\omega_k} \cos \varphi_k + 1 & \frac{1}{\omega_k} \cos \phi_k
\end{bmatrix}
\]

\[
G_n(x_k^t) = \begin{bmatrix}
- \frac{1}{\omega_k} \sin \varphi_k + 1 & \frac{1}{\omega_k} \sin \phi_k - \frac{1}{\omega_k} T \sin \phi_k \\
\frac{1}{\omega_k} \cos \varphi_k + 1 & \frac{1}{\omega_k} \cos \phi_k \\
\frac{1}{\omega_k} \cos \varphi_k + 1 & \frac{1}{\omega_k} \cos \phi_k + \frac{1}{\omega_k} T \sin \phi_k \\
- \frac{1}{\omega_k} \cos \varphi_k + 1 & \frac{1}{\omega_k} \cos \phi_k
\end{bmatrix}
\]

\[
G_n(x_k^t) = \begin{bmatrix}
- \frac{1}{\omega_k} \sin \varphi_k + 1 & \frac{1}{\omega_k} \sin \phi_k - \frac{1}{\omega_k} T \sin \phi_k \\
\frac{1}{\omega_k} \cos \varphi_k + 1 & \frac{1}{\omega_k} \cos \phi_k \\
\frac{1}{\omega_k} \cos \varphi_k + 1 & \frac{1}{\omega_k} \cos \phi_k + \frac{1}{\omega_k} T \sin \phi_k \\
- \frac{1}{\omega_k} \cos \varphi_k + 1 & \frac{1}{\omega_k} \cos \phi_k
\end{bmatrix}
\]

We draw the accelerations according to first order Markov models, so that \( v^*_k \sim N(0, Q^x) \) where

\[
Q^x = \begin{bmatrix}
0 & \sigma_1^2 & 0 \\
\sigma_1^2 & 0 & \sigma_2^2 \\
0 & \sigma_2^2 & 0
\end{bmatrix}
\]

(24)

The noise matrix in (24) is more sparse than the matrix in (4) since the mapping of noise (i.e. acceleration) onto \( x_k, \dot{x}_k \) etc. is taken care of by (23) in the simulation model. The four tuning constants \( \sigma_1^2, \beta_{tm}, \sigma_2^2 \) and \( \tau_{tm} \) can be chosen to accommodate for a wide range of scenarios corresponding to various types of targets.

### 4.4 Resolution issues

The measurement noise is related to the sensor resolution. For a sensor working in polar coordinates we may as a first approximation define \( R \) by

\[
R_{\text{polar}} = \begin{bmatrix}
(\Delta r)^2/12 & 0 & 0 \\
0 & (\Delta \theta)^2/12 & 0 \\
0 & 0 & \sigma_0^2
\end{bmatrix}
\]

(25)

where \( \Delta r \) and \( \Delta \theta \) are the sizes of a resolution cell in range and bearing respectively. The uniform distribution over a resolution cell is approximated by a Gaussian distribution with the same variance. Thus the number 12 appears. The cartesian measurement noise variance matrix \( R \) is obtained from \( R_{\text{polar}} \) using the coordinate transform of (1).
### Table 1: Simulation model parameters

<table>
<thead>
<tr>
<th>Param.</th>
<th>Value</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>1.0 s</td>
<td>Sampling period</td>
</tr>
<tr>
<td>$N_{\rho} \times N_{\phi}$</td>
<td>140 x 80</td>
<td>Resolution cells</td>
</tr>
<tr>
<td>$r_{\text{min}}$</td>
<td>48.4 m</td>
<td>Innermost sensor range</td>
</tr>
<tr>
<td>$r_{\text{max}}$</td>
<td>158.6 m</td>
<td>Outermost sensor range</td>
</tr>
<tr>
<td>$\theta_{\text{max}}$</td>
<td>55.5 $^\circ$</td>
<td>Sensor bearing coverage</td>
</tr>
<tr>
<td>$\sigma^2_t$</td>
<td>(0.1 m$^3$/s$^3$)$^2$</td>
<td>Tan. acceleration power</td>
</tr>
<tr>
<td>$\tau_{\text{trm}}$</td>
<td>2.0 s</td>
<td>Tan. acc. time constant</td>
</tr>
<tr>
<td>$\sigma^2_n$</td>
<td>(0.05 m$^3$/s$^3$)$^2$</td>
<td>Perp. acceleration power</td>
</tr>
<tr>
<td>$\tau_{\text{rm}}$</td>
<td>1.0 s</td>
<td>Perp. acc. time constant</td>
</tr>
<tr>
<td>$A$</td>
<td>1.0 cells</td>
<td>Range ACF param.</td>
</tr>
<tr>
<td>$B$</td>
<td>1.3 cells</td>
<td>Bearing ACF param.</td>
</tr>
</tbody>
</table>

### Table 2: Filter model parameters

<table>
<thead>
<tr>
<th>Param.</th>
<th>Value</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g$</td>
<td>6</td>
<td>Gate size</td>
</tr>
<tr>
<td>$P_{\text{FA}}$</td>
<td>0.005</td>
<td>Design false alarm rate</td>
</tr>
<tr>
<td>$\sigma^2_r$</td>
<td>(0.23 m)$^2$</td>
<td>Mea. noise (range)</td>
</tr>
<tr>
<td>$\sigma^2_{\theta}$</td>
<td>(0.20 $^\circ$)$^2$</td>
<td>Mea. noise (bearing)</td>
</tr>
<tr>
<td>$\sigma^2_{\ell}$</td>
<td>(0.0125 m$^3$/s$^3$)$^2$</td>
<td>Process noise</td>
</tr>
<tr>
<td>$L$</td>
<td>20</td>
<td>Power estimation lag</td>
</tr>
<tr>
<td>$c_1$</td>
<td>5</td>
<td>Track-loss threshold</td>
</tr>
<tr>
<td>$c_2$</td>
<td>15</td>
<td>Track-loss threshold</td>
</tr>
</tbody>
</table>

The target appears at time $k = 0$ with initial kinematic state $x_0 = [64m, 82m, 0m/s, -0.75m/s]$. It then moves according to the model given by (23) and (24) for the next 50 seconds. The sensor observes the region given by $48.36m < r < 158.57m$ and $0^\circ < \theta < 56^\circ$ with a resolution of 140 range cells and 80 bearing cells. This is supposed to mimic a cutout of the images provided by the sonar used in [8]. This scenario is illustrated in Figure 3. Parameters used in the simulation of the scenario are summarized in Table 1. In order to distinguish the filter model from the simulation model we summarize parameters related to the filter model (i.e. governing the detection and tracking methods) in a separate Table 2.

In order to test the performance of our tracking methods this scenario is run 5000 times for various SNR and shape parameters $\nu$. The SNR is measured in decibel (dB), and can for a Swerling 1 target in $K$-distributed noise be related to the target power $d$ as follows,

$$\text{SNR} = 10 \log_{10} \left( \frac{E[\max(|s|^2)]}{E[|w|^2]} \right) = 10 \log_{10} \frac{2d}{\nu b}, \quad (26)$$

Following the convention established by [9] we measure the SNR according to the value of the strongest target-affected cell, and not according to the total energy since the former has the most direct impact on tracking performance. The tracking methods do not know the SNR in advance and must therefore estimate it over the last $L = 20$ time steps as explained in Section 3.5 of [4].

### 4.6 Performance measures

The most important measure of the performance of a tracking algorithm is whether or not it manages to maintain a track on the target. Other measures, such as RMSE, are primarily of interest subject to the condition that the tracker is “on target”. However, track-loss is a rather subjective phenomenon which cannot be defined rigorously.
Figure 4: Example $K$-distributed data generated with the recipe of [10] for different correlation lengths $L_\Gamma$. A target is embedded in the background at $\rho = 9$. Shorter correlation length makes it more difficult to spot.

### Table 3: Lost tracks for SNR 12dB and $L_\Gamma = 1$

<table>
<thead>
<tr>
<th>$\nu$</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>4018</td>
<td>4642</td>
<td>4568</td>
<td>4980</td>
<td>4883</td>
</tr>
<tr>
<td>0.5</td>
<td>3672</td>
<td>3232</td>
<td>4140</td>
<td>4665</td>
<td>3221</td>
</tr>
<tr>
<td>1</td>
<td>2812</td>
<td>2000</td>
<td>3175</td>
<td>3434</td>
<td>1501</td>
</tr>
<tr>
<td>2</td>
<td>1819</td>
<td>1152</td>
<td>1908</td>
<td>1625</td>
<td>622</td>
</tr>
<tr>
<td>4</td>
<td>1155</td>
<td>650</td>
<td>935</td>
<td>579</td>
<td>268</td>
</tr>
<tr>
<td>8</td>
<td>737</td>
<td>493</td>
<td>454</td>
<td>224</td>
<td>133</td>
</tr>
</tbody>
</table>

### Table 4: Lost tracks for SNR 12dB and $L_\Gamma = 8$

<table>
<thead>
<tr>
<th>$\nu$</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>2285</td>
<td>2585</td>
<td>2914</td>
<td>3844</td>
<td>2357</td>
</tr>
<tr>
<td>0.5</td>
<td>1793</td>
<td>1567</td>
<td>1905</td>
<td>1775</td>
<td>994</td>
</tr>
<tr>
<td>1</td>
<td>1190</td>
<td>1082</td>
<td>1023</td>
<td>637</td>
<td>533</td>
</tr>
<tr>
<td>2</td>
<td>797</td>
<td>671</td>
<td>538</td>
<td>228</td>
<td>254</td>
</tr>
<tr>
<td>4</td>
<td>556</td>
<td>431</td>
<td>294</td>
<td>151</td>
<td>140</td>
</tr>
<tr>
<td>8</td>
<td>465</td>
<td>311</td>
<td>209</td>
<td>116</td>
<td>78</td>
</tr>
</tbody>
</table>

### Table 5: Lost tracks for SNR 15dB and $L_\Gamma = 1$

<table>
<thead>
<tr>
<th>$\nu$</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>2399</td>
<td>2939</td>
<td>3383</td>
<td>4867</td>
<td>3404</td>
</tr>
<tr>
<td>0.5</td>
<td>1564</td>
<td>926</td>
<td>2123</td>
<td>3018</td>
<td>536</td>
</tr>
<tr>
<td>1</td>
<td>872</td>
<td>452</td>
<td>1047</td>
<td>1046</td>
<td>154</td>
</tr>
<tr>
<td>2</td>
<td>444</td>
<td>225</td>
<td>423</td>
<td>242</td>
<td>62</td>
</tr>
<tr>
<td>4</td>
<td>248</td>
<td>151</td>
<td>144</td>
<td>63</td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td>132</td>
<td>100</td>
<td>48</td>
<td>21</td>
<td>23</td>
</tr>
</tbody>
</table>

### Table 6: Lost tracks for SNR 15dB and $L_\Gamma = 8$

<table>
<thead>
<tr>
<th>$\nu$</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1224</td>
<td>1207</td>
<td>1666</td>
<td>2618</td>
<td>881</td>
</tr>
<tr>
<td>0.5</td>
<td>603</td>
<td>399</td>
<td>554</td>
<td>476</td>
<td>173</td>
</tr>
<tr>
<td>1</td>
<td>272</td>
<td>240</td>
<td>211</td>
<td>83</td>
<td>69</td>
</tr>
<tr>
<td>2</td>
<td>169</td>
<td>143</td>
<td>80</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>114</td>
<td>93</td>
<td>40</td>
<td>17</td>
<td>22</td>
</tr>
<tr>
<td>8</td>
<td>85</td>
<td>63</td>
<td>25</td>
<td>13</td>
<td>17</td>
</tr>
</tbody>
</table>

In this paper we treat track-loss as a two-stage process. A track is considered tentatively lost at time $k$ if the position error $\sqrt{(x_k - x_k^1)^2 + (y_k - y_k^1)^2}$ exceeds a threshold $c_1$. If the error later goes below $c_1$ the lost label is removed. On the other hand, if the error never manages to go below $c_1$ again, we consider it lost at time $k$. If the error exceeds the higher threshold $c_2 > c_1$ we immediately consider it lost at time $k$, irrespective of whether the error later goes below $c_1$. In our analysis we have used the values $c_1 = 5m$ and $c_2 = 15m$. For a different scenario different values may be more suitable.

### 4.7 Interpretation of the results

Tables 3 - 8 provide a summary of our results. These tables should not be interpreted as a definitive truth of our tracking methods’ performances, but as evidence for the following observations.

First, it can be seen that heavy-tailed clutter (as represented by $\nu$) makes the tracking scenario more challenging. The $\nu = 0.1$ scenario ranges from 5 times as difficult as the $\nu = 8$ scenario (T1 in Table 3) to 58 times as difficult as the $\nu = 8$ scenario (T5 in Table 7). A non-maneuvering 12dB target can easily be tracked in Rayleigh noise. A maneuvering target in a heavy-tailed background must be much stronger if we shall have any hope of tracking it.

Second, the difficulties encountered depend on the correlation length $L_\Gamma$. The conventional trackers T1 and T2 loose track roughly twice as often for $L_\Gamma = 1$ as for $L_\Gamma = 8$. The impact is more severe for the Rayleigh-based trackers T3 and T4 which lose track 5-8 times more often in uncorrelated clutter than in correlated clutter.

Third, performance is improved by accounting for heaviness in the detection and tracking processes. In most cases we see that T2 performs better than T1. The improvement is most noticeable for medium heavy-tailed clutter with $1 \leq \nu \leq 4$. T5 yields further improvements.

Fourth, the utilization of AI improves tracking performance. In the near-Rayleigh case both T3 and T4 give fewer
lost tracks than T1. In all cases when these outperform T1 it can also be seen that T4 outperforms T3. For \( \nu \approx 2 \) and SNR = 15 we see that T3 gives twice as many lost tracks as T4. Overall, T5 has the best performance. Compared to T1 we see that T5 can reduce the number of lost tracks by up to 90\% (\( \nu = 1 \) in Table 7). Also notice that the improvement of T5 compared to T2 is stronger than the improvement of T2 compared to T1.

Based on these observations we may conclude that T5 is the preferred tracker in moderately heavy-tailed clutter, while T4 is the preferred tracker in near-Rayleigh clutter. Even when T5 works equally well as T4 we may prefer T4 since T4 uses fewer auxiliary cells and thus is less susceptible to interference from other targets. For very heavy-tailed clutter there is little we can do, and all our trackers fail more or less.

The recommendation of the trackers T4 and T5 is subject to computational resources. While T1 and T3 are hardly more expensive than a simple Kalman filter, significant additional costs are incurred by the other trackers. This cost increase is of a linear nature, and may be considered acceptable if the alternative is more expensive tracking methods such as the Multiple Hypotheses Tracker, whose computational complexity is exponential.

### 5 Conclusion

This paper has presented a treatment of heavy-tailed clutter from a target tracking perspective. Simulation results show that heavy-tailed clutter inevitably leads to deteriorated performance. The performance loss can be substantially mitigated by accounting for heavy-tailedness in the detection and tracking processes, and by usage of amplitude information as explained in [4].

Future work may investigate the performance of other tracking methods than the PDAF-based methods discussed here. It may also be of interest to investigate other background models than the \( K \)-distribution. In particular it would be very interesting to find a heavy-tailed background model which allows closed-form evaluation of the amplitude likelihood, thereby avoiding the numerical integration of (15). The sensitivity to model mismatch for such developments should be checked by generating test data from a variety of background distributions.

### References


---

**Table 7:** Lost tracks for SNR 18dB and \( L_\Gamma = 1 \)

<table>
<thead>
<tr>
<th>( \nu ) = 0.1</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1045</td>
<td>900</td>
<td>1787</td>
<td>4309</td>
<td>930</td>
<td></td>
</tr>
<tr>
<td>( \nu ) = 0.5</td>
<td>453</td>
<td>223</td>
<td>623</td>
<td>1105</td>
<td>930</td>
</tr>
<tr>
<td>( \nu ) = 1</td>
<td>215</td>
<td>121</td>
<td>220</td>
<td>184</td>
<td>19</td>
</tr>
<tr>
<td>( \nu ) = 2</td>
<td>113</td>
<td>57</td>
<td>79</td>
<td>29</td>
<td>16</td>
</tr>
<tr>
<td>( \nu ) = 4</td>
<td>47</td>
<td>42</td>
<td>33</td>
<td>8</td>
<td>19</td>
</tr>
<tr>
<td>( \nu ) = 8</td>
<td>30</td>
<td>33</td>
<td>17</td>
<td>8</td>
<td>16</td>
</tr>
</tbody>
</table>

**Table 8:** Lost tracks for SNR 18dB and \( L_\Gamma = 8 \)

<table>
<thead>
<tr>
<th>( \nu ) = 0.1</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
</tr>
</thead>
<tbody>
<tr>
<td>502</td>
<td>361</td>
<td>717</td>
<td>1360</td>
<td>224</td>
<td></td>
</tr>
<tr>
<td>( \nu ) = 0.5</td>
<td>144</td>
<td>85</td>
<td>117</td>
<td>76</td>
<td>31</td>
</tr>
<tr>
<td>( \nu ) = 1</td>
<td>51</td>
<td>53</td>
<td>28</td>
<td>8</td>
<td>15</td>
</tr>
<tr>
<td>( \nu ) = 2</td>
<td>36</td>
<td>40</td>
<td>12</td>
<td>6</td>
<td>13</td>
</tr>
<tr>
<td>( \nu ) = 4</td>
<td>21</td>
<td>32</td>
<td>6</td>
<td>5</td>
<td>14</td>
</tr>
<tr>
<td>( \nu ) = 8</td>
<td>16</td>
<td>19</td>
<td>10</td>
<td>7</td>
<td>13</td>
</tr>
</tbody>
</table>