Abstract—This paper first proposes a method to estimate the altitude of an aircraft using two 2-D radars for Air Traffic Control (ATC) applications. Assuming that the aircraft flies with a nearly constant velocity and nearly constant height, we first estimate its velocity, initial position in 3-D coordinates from the measurements of two 2-D radars using a maximum likelihood estimator. Next, we incorporate the estimates as initial values for 3-D flight tracking that uses an extended Kalman filter (EKF). Furthermore, a multiple model approach is adopted to further improve the tracking performance. The simulation results are shown to demonstrate the effectiveness of the proposed method.

Keywords: Altitude estimation, parameter estimation, 3-D target tracking, extended Kalman filtering.

I. INTRODUCTION

Information on the altitude of an aircraft is necessary for Air Traffic Control (ATC) to manage the air traffic safety and effectively. Generally, for an airplane flying above 10000 ft, the aircraft is required to report the altitude to the ATC center [1] through Mode C transponder replies to the secondary surveillance radar interrogations. If the airplane is not equipped with a Mode C transponder, its altitude cannot be observed directly by the ATC center and the radars can provide only 2-D measurements, i.e. target’s range and azimuth. This results in the need for estimation of the aircraft altitude in such situations.

To perform 3-D tracking using measurements from 2-D radars, it naturally requires a method of altitude estimation to initialize the tracking algorithm, owing to the dimensional incompleteness of measurements. To estimate the initial altitude from a set of measurements, we usually have to impose an assumption to be able to find the solution. A common and practical assumption in ATC applications is that the airplane is cruising with constant altitude, velocity and heading. In [2], a method to estimate the altitude using a single 2-D radar by fitting the range curve from a few range measurement data is proposed, but the altitude estimation errors are large [3]. This is caused by the low observability of the target states from a single radar, which suggests that we use more radars. In [4], a 3-D tracking approach using measurements from two or more 2-D radars in WGS-84 coordinates is proposed. However, the RMSE of three dimensional position therein makes it not very attractive for practical applications. In [5], an algorithm referred to as the height-parameterized extended Kalman filtering (HPEKF), which incorporates the autonomous multiple model (AMM) method [6], is presented. In particular, the altitude interval of interest is first partitioned into subintervals, an independent extended Kalman filter (EKF) is run for each of the intervals, and finally the overall target’s state estimate is obtained by combining the EKF models’ state estimates. In [7], another altitude estimation method using a 2-D radar network is proposed. This method considers the earth surface’s curvature and employs the maximum likelihood (ML) estimator to find the target’s altitude using the measurements obtained from synchronous multiple radars. Recently, an approach for altitude estimation and mitigation of slant range errors on 3-D target tracking with 2-D radars has been proposed in [8], where it uses measurements in polar coordinates and state vectors in geodetic coordinates.

In this paper, we propose an approach for altitude estimation and 3-D tracking using two 2-D radars with the assumption of constant velocity and altitude. Even though the method proposed herein does not depend on the number of radars, we consider the case of two radars because in reality many airspaces are still covered by just two radars. Also, if the target is covered by more than two radars (see [9] and [10] for example), multilateration methods can be used to estimate the altitude. We do not consider the case of non-primary radars that can provide other kinds of measurements in addition to range and azimuth. To further improve the altitude estimation accuracy, we also propose the use of the EKF based AMM method, where each model has a different initial altitude value.

The paper is organized as follows. Section II presents the problem formulation of altitude estimation and discusses an iterative algorithm to solve for the altitude estimate. Section II shows the estimation performance for several scenarios and altitude levels. A Cramér-Rao lower bound (CRLB) of the altitude estimator’s variance is also provided. As an application of the proposed altitude estimation, Section IV uses the estimated altitude and optimized parameters to initialize the 3-D tracking algorithm with two 2-D radars using the EKF-based multiple approach method. The tracking results are also displayed and compared to the existing method. Finally, some concluding remarks are presented in Section V.
II. ALTITUDE ESTIMATION

This section aims at presenting the formulation of height estimation and how to estimate the height by using the maximum likelihood criterion.

A. Problem Formulation

To begin with, let $\rho_{m,i}$ and $\theta_{m,i}$ be the range and azimuth measurements obtained from the radar system at time $t_i$ ($i = 0, 1, \ldots, N - 1$), respectively. Assume the initial time $t_0 = 0$. Let the velocity and initial position of the aircraft at $t_0$, $v_x$ and $x_0$ in the X direction and $v_y$ and $y_0$ in the Y direction, and $z_0$ be the aircraft’s altitude. Therefore, the true position of the aircraft in the system plane at time $t$, $(x(t), y(t), z(t))$, is given by

$$x(t) = v_x t + x_0, \quad y(t) = v_y t + y_0, \quad z(t) = z_0.$$  

The true range and the true azimuth at time $t$ is given by

$$\rho(t) = \sqrt{(x(t) - x_R)^2 + (y(t) - y_R)^2 + (z(t) - z_R)^2},$$

$$\theta(t) = \arctan \left( \frac{x(t) - x_R}{y(t) - y_R} \right).$$  

(1)

where $(x_R, y_R, z_R)$ is the corresponding radar’s position in the system plane. Note that it is not necessary that the two radars detect the target simultaneously. The relationship between the true data and the measurement data from a radar is obtained by

$$\rho_{m,i} = \rho_m(t_i) + \tilde{\rho}_i$$

$$\theta_{m,i} = \theta_m(t_i) + \tilde{\theta}_i,$$

(2)

where $(\rho_m(t_i), \theta_m(t_i))$ can be computed from (1), and the measurement errors $\tilde{\rho}_i$ and $\tilde{\theta}_i$ are assumed to be independent Gaussian with zero mean and variance $\sigma^2_\rho$ and $\sigma^2_\theta$, respectively. Both $\tilde{\rho}_i$‘s and $\tilde{\theta}_i$‘s are also assumed to be independent.

Under the assumption of constant velocity and altitude, the aim is to estimate the target altitude. Here, we present the ML estimator to find the estimates of $v_x, x_0, v_y, y_0$ and $z_0$. Specifically, the log-likelihood function corresponding to a set of measurement data is given by

$$LL(p) = \log \left( \prod_{i=0}^{N-1} f(\rho_{m,i}, \theta_{m,i};p) \right) = \sum_{i=0}^{N-1} \log f(\rho_{m,i}, \theta_{m,i};p),$$

where $p = [v_x, x_0, v_y, y_0, z_0]^T$ and $f(\rho_{m,i}, \theta_{m,i};p)$ is the joint probability density function (pdf) of $\rho_{m,i}$ and $\theta_{m,i}$. Since $\tilde{\rho}_i$ and $\tilde{\theta}_i$ are independent, $f(\rho_{m,i}, \theta_{m,i};p)$ can be written as

$$f(\rho_{m,i}, \theta_{m,i};p) = f(\rho_{m,i};p)f(\theta_{m,i};p),$$

where

$$f(\rho_{m,i};p) = \frac{1}{\sigma_\rho \sqrt{2\pi}} \exp \left( -\frac{(\rho_{m,i} - \rho(t_i))^2}{2\sigma^2_\rho} \right),$$

$$f(\theta_{m,i};p) = \frac{1}{\sigma_\theta \sqrt{2\pi}} \exp \left( -\frac{(\theta_{m,i} - \theta(t_i))^2}{2\sigma^2_\theta} \right),$$

$\rho_{m,i}$, $\rho(t_i)$, $\theta_{m,i}$ and $\theta(t_i)$ are defined as in (2). As a consequence, maximizing $LL(p)$ is equivalent to minimizing $F(p)$, which is defined by

$$F(p) = \sum_{i=0}^{N-1} \left( G_i(p) + H_i(p) \right),$$

(3)

where $G_i(p) = \frac{(\rho_{m,i} - \rho(t_i))^2}{\sigma^2_\rho}$ and $H_i(p) = \frac{(\theta_{m,i} - \theta(t_i))^2}{\sigma^2_\theta}$. This optimization problem can be solved by using an iterative algorithm. Note that this ML estimator corresponding to (3) is also identical to the nonlinear least square (NLS) estimator.

B. Minimizing the Objective Function

To find the value of $p$ that minimizes $F(p)$ in (3), we need to solve the minimization problem described by

$$\hat{p} = \arg\min_p F(p).$$

It is rather straightforward to show that this is not a convex optimization problem. However, if you pick a good initial value, a sufficiently large value of $\epsilon$ in this case, and use the Newton-Raphson algorithm [11] with varying step size to guarantee that $F(p)$ decreases in every iteration $k$, we empirically found that this iterative algorithm yields satisfactory results. Specifically, we use this algorithm:

$$p^{[k+1]} = p^{[k]} - \lambda^k [\nabla^2 F(p^{[k]})]^{-1}[\nabla F(p^{[k]})]^T,$$

(4)

where $\lambda^k = \lambda^k$ for $0 < \lambda < 1$. The index $j_k$ is chosen at each step as the smallest non-negative integer that would make the objective function $F(p^{[k+1]}) < F(p^{[k]})$, i.e. the index $j_k$ is present to make sure that the objective function decreases at every iteration. The notations $\nabla F$ and $\nabla^2 F$ are the Jacobian and the Hessian of $F$, respectively. See Appendix A for their expressions.

In this paper, the initial value for the altitude is set to be a positive constant. The remaining initial values $v_x^0, x_0^0, v_y^0, y_0^0, z_0^0$ are chosen by minimizing

$$\sum_{i=0}^{N-1} \left[ (\rho_{m,i} \sin \theta_{m,i} - (v_x^0 t_i + x_0^0 - x_R))^2 + (\rho_{m,i} \cos \theta_{m,i} - (v_y^0 t_i + y_0^0 - y_R))^2 \right].$$

(5)

This minimization yields a closed form solution for these parameters as shown in Appendix B. Hence, the initial value $p^{[0]}$ is given by $p^{[0]} = [v_x^0, x_0^0, v_y^0, y_0^0, z_0^0]^T$. The stopping criterion is that the iteration stops when $\|p^{[k+1]} - p^{[k]}\| < \epsilon$, for some small positive constant $\epsilon$. It is worth noting that this problem can be also solved using a recursive algorithm such as an iterated Kalman filter (IKF) discussed in [12].

III. NUMERICAL RESULTS OF ALTITUDE ESTIMATION

To validate the method proposed in the previous section, we provide examples of estimating the altitude of a cruising aircraft with two 2-D radars in this section.

A. Scenario Setting

For simplicity, assume that the two radars A and B (Figure 1), are at the same level and located at the position (0, 0, 0) and (20, 0, 0), all in nautical miles (nmi), respectively. For both radars, the antenna rotating period are assumed to be equal to 4.61 s. The standard deviations of the measurement errors are assumed to be $\sigma_\rho = 0.06$ nmi, and $\sigma_\theta = 2^\circ = 3.4906 \times 10^{-3}$ rad. The simulation duration is 450 s. We test three flight scenarios with details as follows.
Scenario 1: The aircraft’s flight trajectory starts at (10, 1, \(z_0\)) with the heading \(0^\circ\) with a constant speed 300 knot = 8.3333 \times 10^{-2} \text{ nmi/s}, and constant altitude \(z_0\), where we try three cases of the true altitude value \(z_0 = 5000, 7500\) and 10000 ft.

Scenario 2: The flight trajectory starts at (-10, 5, \(z_0\)) with the heading \(90^\circ\). Other parameters are the same as in Scenario 1.

Scenario 3: The flight trajectory starts at (0, 10, \(z_0\)) with the heading \(45^\circ\). Other parameters are the same as in Scenarios 1 and 2.

The three scenarios are illustrated in Figure 1. The stopping criterion error is chosen as \(\varepsilon = 10^{-6}\), while \(\lambda = 0.9\) and the initial value of the height for the iterative method in (4) is chosen to be \(z_0^{[0]} = 15000 \text{ ft} = 2.4687 \text{ nmi}\), which is empirically found to be large enough for the general aviation case where the altitude is usually not more than 10000 ft.

### B. Simulation Results

We use the first \(N\) number of samples of the measurement data to estimate the altitude where \(N = 10, 20, ..., 100\). The simulation results are based on 100 Monte Carlo runs. Figures 2(a)-2(c) show the estimation performance in terms of the root mean square error (over 100 trials) for each \(N\), which is defined by

\[
\text{RMSE} = \sqrt{\frac{1}{100} \sum_{n=1}^{100} (\hat{z}_n[z]_n - z_0)^2}.
\]

As we expect, the RMSE value tends to decrease when there are more samples used in the ML estimation. Moreover, for each scenario, the estimation accuracy degrades when the value of the true altitude is smaller. This can be explained by considering the Cramér-Rao lower bound (CRLB) as follows.

For the estimation in this work, the CRLB of all five estimates \(\hat{v}_x, \hat{x}_0, \hat{v}_y, \hat{y}_0, \text{ and } \hat{z}_0\) define the lower limit of their estimation error covariance matrix. Nevertheless, let us assume in this analysis that the other four quantities, \(v_x, x_0, v_y, \text{ and } y_0\) are known because we want to focus only on the altitude estimate in order to see how the variance of the altitude estimator changes with the true altitude value. The Fisher information of the ML altitude estimator from Section II-A is given by [13]

\[
I(z_0) = -E \left[ \frac{\partial^2}{\partial z_0^2} \log f(\rho_m, 0, ..., \rho_m, N-1; z_0) \right]z_0,
\]

where \(\rho_{m,i}\)’s are defined as in Section II-A, and

\[
\log f(\rho_m, 0, ..., \rho_m, N-1; z_0) = -N \log(\sigma_\rho \sqrt{2\pi}) - \frac{1}{2} \sum_{i=0}^{N-1} \rho_{m,i} \cdot \frac{(z_0 - z_{R,i})^2}{\sigma_{\rho}^2}.
\]

From the CRLB, the variance of any unbiased estimator is bounded below by the inverse of Fisher information:

\[
\text{Var}(\hat{z}_0) \geq \frac{\sigma_{\rho}^2}{\sum_{i=0}^{N-1} \left(\frac{z_0 - z_{R,i}}{\sigma_{\rho}}\right)^2}.
\]

It is straightforward to show that this CRLB on the right-hand side of (6) is a decreasing function of \(z_0\) with \(z_0 \geq 0\). Therefore for a good ML estimator, it is reasonable to expect the RMSE to become smaller when the true value of the altitude becomes larger, given the other factors are fixed. Note that, for simplicity, we set \(z_{R,i} = 0\) for all \(i\) in our simulation.

The altitude estimation results for Scenario 2 and 3 in Figures 2(b) and 2(c) follow the same trend as in Scenario 1 but with larger RMSEs. For example, with the true altitude 10000 ft, the RMSE is around 440 ft for \(N = 20\) in Scenario 1, 1340 ft in Scenario 2, and around 1270 ft in Scenario 3. It is because this altitude estimation from two 2-D radars depends on the geometrical configurations of the flight scenario where the target states become unobservable due to dimensional incompleteness. From Figure 1, we can see that the ground ranges for the first few points of Scenarios 2 and 3 are larger than those of Scenario 1, which results in larger value of the CRLB in (6) for a fixed value of the altitude. Next, we will use this altitude estimate to initialize 3-D tracking algorithms.

### IV. AN APPLICATION IN 3-D TRAJECTORY TRACKING

The goal of this section is to use the height estimate obtained in the previous section along with the other four estimated parameters as initial values for 3-D flight tracking with two 2-D radars using the EKF.

#### A. 3-D Tracking with 2-D Radars using the EKF

As examples of using the altitude estimate for 3-D tracking, we perform the tracking algorithm for Scenarios 1, 2 and 3 as defined in Section II. We use the first 20 points to calculate the height estimate using the ML method as described in Section
II-A, and then start the tracking algorithm using the EKF from the point \( t = 20 \) onward. This is chosen as a trade-off between the computational time and the estimation accuracy. Assume the aircraft dynamic model is given by

\[
\mathbf{x}(i) = \mathbf{F}(i)\mathbf{x}(i-1) + \mathbf{Γ}(i)\mathbf{v}(i-1),
\]

where

\[
\mathbf{F}(i) = \begin{bmatrix}
1 & 0 & 0 & T_i & 0 & 0 \\
0 & 1 & 0 & 0 & T_i & 0 \\
0 & 0 & 1 & 0 & 0 & T_i \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix},
\]

\[
\mathbf{Γ}(i) = \begin{bmatrix}
\frac{1}{2}T_i^2 & 0 & 0 & 0 \\
0 & \frac{1}{2}T_i^2 & 0 & 0 \\
0 & 0 & \frac{1}{2}T_i^2 & 0 \\
0 & T_i & 0 & 0 \\
0 & 0 & T_i & 0 \\
0 & 0 & 0 & T_i \\
\end{bmatrix},
\]

and \( \mathbf{x}(i) = [x(t_i), y(t_i), z(t_i), \dot{x}(t_i), \dot{y}(t_i), \dot{z}(t_i)]^T \) is the state vector, \( T_i \) is the time interval. The process noise vector \( \mathbf{v}(i) \) is Gaussian with zero-mean and covariance matrix \( \mathbf{Q} = \text{diag}([q_{11}, q_{22}, q_{33}]) \), where \( q_{11} = q_{22} = 2 \times 10^{-9}, q_{33} = 10^{-11} \). The radar measurement is

\[
\mathbf{z}(i) = \mathbf{h}(\mathbf{x}(i)) + \mathbf{w}(i),
\]

where \( \mathbf{h}(\mathbf{x}(i)) = [\rho(t_i), \theta(t_i)]^T \), the measurement noise vector \( \mathbf{w}(i) \) is zero-mean Gaussian with covariance matrix \( \mathbf{R} = \text{diag}([\sigma^2_\rho, \sigma^2_\theta]) \). The values of \( \sigma_\rho \) and \( \sigma_\theta \) are the same as in Section II.

In this paper, the initialization is taken as follows. For the state estimate vector, we set \( \hat{\mathbf{x}}(19|19) = [\hat{x}_{19}, \hat{y}_{19}, \hat{z}_0, \hat{v}_x, \hat{v}_y, 0]^T \), which can be computed straightforwardly from the ML estimation in Section II-A. On the other hand, for the state estimate covariance matrix, we set \( \mathbf{P}(19|19) = \text{diag}([p_{11}, p_{22}, p_{33}, p_{44}, p_{55}, p_{66}]) \), where \( p_{11} = p_{22} = 4 \times 10^{-3}, p_{44} = p_{55} = 1 \times 10^{-4}, p_{66} = 0, \) and \( p_{33} \) is chosen such that the algorithm performs satisfactorily. We call this method the AE1 method since it uses an altitude estimation to initialize one single EKF. In addition to target tracking with a single EKF, we also propose a multiple model EKF approach to improve the tracking performance in the Z-axis. This approach is explained in the next section.

B. The Multiple EKF Models

The idea behind using multiple models is that more models would increase the chance that the initial altitude value of one of the models is closer to the true altitude value, which will result in tracking performance improvement because of the sensitivity of an EKF to the initial value [14]. In particular, we use the EKF-based multiple model approach where each model differs only by the initial value of the altitude. Since we assume the target flies with a constant height, the autonomous multiple model (AMM) algorithm [6] is a good fit in that it assumes the system mode does not change. The key idea of the AMM is that each EKF models a different altitude trajectory and the overall altitude estimate is obtained by combining estimates from individual EKFs. Specifically, the overall state estimate is computed from the sums of the state estimate of each filter weighted by its mode probability, which results from its model likelihood. We refer to [6] for the details related to the AMM algorithm.

In this subsection, we compare the tracking results of the proposed AE1 and AE3 methods with those of using the HPEKF method as in [5], which provides another method for initializing the altitude for 3-D tracking with 2-D radars. For the HPEKF method, we follow the method in [15] to design HPEKFs, where we use seven filters to cover the altitude range from 100-15000 ft. This number of filters (seven filters) is selected such that the coefficient of variation [15] is 20%. Particularly, the initial altitude values for the filters are 152, 312, 637, 1304, 2668, 5458 and 11166 ft. For the purpose of
where in each scenario, tracking with three values of the true altitude: \(z_0 = 5000, 7500\) and \(10000\) ft, is tested. For each figure, the top panel shows the comparison of the XY-position RMSE while the bottom one illustrates the Z-position RMSE. Comparing the two proposed AE1 and AE3 methods, it can be seen from the bottom panel of each figure that the AE3 method improves the tracking performance in terms of the altitude RMSE from the AE1 method for all three altitude
values and three scenarios. Specifically, the altitude RMSE value is reduced by using the AE3 method up to 200, 400 and 300 ft in Scenarios 1, 2 and 3, respectively, depending on the true values of the altitude. Note that the AE1 and AE3 methods perform better when the true altitude is higher because when the value of the true altitude decreases, the performance of the ML estimation degrades, as explained in Section II.

Considering the performance of using the HP3 and HP7 methods in terms of the altitude RMSE, we can observe that the HP3 and HP7 methods yield larger altitude RMSE values than the AE1 method (and much larger than the AE3 method) for all cases except in Scenario 2, where the HP3 and HP7 methods improve the altitude RMSE over the AE3 method up to 200 ft. This is because the HP3 and HP7 methods use three and seven fixed initial values of the altitude while the AE1 and AE3 methods use one and three initial values, varying with the ML estimate of the altitude. Nevertheless, in these two cases, if we check the performance in terms of the XY-position RMSE, the AE1 and the AE3 methods perform approximately the same while the HP3 and HP7 methods yield very large RMSE values at the first few data points (around 20-30 data points). This behavior is also observed in the corresponding altitude RMSE plots. This is one of the advantages of using the ML estimation for initializing EKFs from which we obtain not only the estimate of the altitude but also the estimates of the position and the velocity in X and Y axes when tracking with the AE1 and the AE3 methods.

In terms of computational complexity, the average values of computational time for the four methods in Scenario 1 with \( z_0 = 7500 \) ft are tabulated in Table I. The computational time is shown as a relative value compared to the computational time of the AE1 method. As shown, the AE3 method requires about three times of computational time used in the AE1 method. Comparing the two methods which use the same number of EKFs, the AE3 method requires about 10% more in terms of computational time than the HP3 method but yields much better results in general. This extra computational time comes from the ML estimation to estimate the altitude. In addition to producing worse results than those from the AE3 method, the HP7 method, which uses seven filters, requires approximately twice the computational time of the AE3 method, which uses three filters.

### V. Conclusion

In this paper, a method of altitude estimation and tracking of a cruising aircraft using two 2-D radars is proposed. Specifically, an estimation method using the maximum likelihood criterion is utilized to estimate the altitude with the assumption of constant velocity and altitude. The CRLB of the altitude estimator’s variance when other parameters are assumed to be known is also provided. The simulation results show that the proposed method performs satisfactorily in terms of the altitude RMSE. We then use the estimated altitude as an initial value for the 3-D flight tracking algorithm using the EKF. To further improve the tracking performance, especially in terms of altitude, the paper proposes three parallel EKFs, each with a different initial altitude estimate: one with the ML estimated altitude and two with estimated altitude perturbed by a value proportional to the CRLB of the estimator in both directions.

### Appendix A

The expressions for \( \nabla F \) and \( \nabla^2 F \) are given by

\[
\nabla F = [F_1, F_2, F_3, F_4, F_5], \quad \nabla^2 F = [F_{kl}]_{5 \times 5},
\]

where \( \{F_i; i = 1, 2, ..., 5\} \) and \( \{F_{kl}; 1 \leq l \leq 5, 1 \leq k \leq 5\} \) are given by

\[
F_1 = \frac{1}{\sigma^2} \sum_{i=0}^{N-1} 2t_i \left( -\frac{\rho_{m,i}}{a_i} + 1 \right) c_i + \frac{1}{\sigma^2} \sum_{i=0}^{N-1} 2t_i b_i \left( c_i^2 + d_i^2 \right),
\]

\[
F_2 = \frac{1}{\sigma^2} \sum_{i=0}^{N-1} 2 \left( -\frac{\rho_{m,i}}{a_i} + 1 \right) c_i + \frac{1}{\sigma^2} \sum_{i=0}^{N-1} 2b_i \left( \frac{d_i}{c_i^2 + d_i^2} \right),
\]

\[
F_3 = \frac{1}{\sigma^2} \sum_{i=0}^{N-1} 2 \left( -\frac{\rho_{m,i}}{a_i} + 1 \right) d_i - \frac{1}{\sigma^2} \sum_{i=0}^{N-1} 2b_i \left( \frac{c_i}{c_i^2 + d_i^2} \right),
\]

\[
F_4 = \frac{1}{\sigma^2} \sum_{i=0}^{N-1} 2 \left( -\frac{\rho_{m,i}}{a_i} + 1 \right) d_i - \frac{1}{\sigma^2} \sum_{i=0}^{N-1} 2b_i \left( \frac{c_i}{c_i^2 + d_i^2} \right),
\]

\[
F_5 = \frac{1}{\sigma^2} \sum_{i=0}^{N-1} 2 \left( -\frac{\rho_{m,i}}{a_i} + 1 \right) c_i,
\]

\[
F_{11} = \frac{1}{\sigma^2} \sum_{i=0}^{N-1} 2t_i^2 \left( -\frac{\rho_{m,i}}{a_i} + 1 + \frac{\rho_{m,c}c_i^2}{a_i^3} \right) + \frac{1}{\sigma^2} \sum_{i=0}^{N-1} 2t_i^2 \left( -\frac{2c_i b_i + d_i}{c_i^2 + d_i^2} \right),
\]

\[
F_{12} = \frac{1}{\sigma^2} \sum_{i=0}^{N-1} 2t_i \left( -\frac{\rho_{m,i}}{a_i} + 1 + \frac{\rho_{m,c}c_i^2}{a_i^3} \right) + \frac{1}{\sigma^2} \sum_{i=0}^{N-1} 2t_i \left( -\frac{2c_i b_i + d_i}{c_i^2 + d_i^2} \right),
\]

\[
F_{13} = \frac{1}{\sigma^2} \sum_{i=0}^{N-1} 2t_i \left( \frac{\rho_{m,c}c_i d_i}{a_i^3} \right) + \frac{1}{\sigma^2} \sum_{i=0}^{N-1} 2t_i \left( \frac{b_i(c_i^2 - d_i^2) - c_i d_i}{c_i^2 + d_i^2} \right),
\]

\[
F_{14} = \frac{1}{\sigma^2} \sum_{i=0}^{N-1} 2t_i \left( \frac{\rho_{m,c}c_i d_i}{a_i^3} \right) + \frac{1}{\sigma^2} \sum_{i=0}^{N-1} 2t_i \left( \frac{b_i(c_i^2 - d_i^2) - c_i d_i}{c_i^2 + d_i^2} \right),
\]
Figure 5. Scenario 3: The tracking performances of AE1, AE3, HP3 and HP7 methods in terms of the XY-position and the Z-position RMSE’s

\[ F_{15} = \frac{1}{\sigma_{\rho}^2} \sum_{i=0}^{N-1} 2t_i \left( \frac{\rho_{m,i}c_i e_i}{a_i^3} \right), \]
\[ F_{22} = \frac{1}{\sigma_{\rho}^2} \sum_{i=0}^{N-1} 2 \left( \frac{-\rho_{m,i}}{a_i} + 1 + \frac{\rho_{m,i}c_i^2}{a_i^3} \right) \]
\[ + \frac{1}{\sigma_{\theta}^2} \sum_{i=0}^{N-1} 2 \left( \frac{c_i (c_i^2 + d_i^2)}{(c_i^2 + d_i^2)^2} \right), \]
\[ F_{23} = F_{14}, \]
\[ F_{24} = \frac{1}{\sigma_{\rho}^2} \sum_{i=0}^{N-1} 2 \left( \frac{\rho_{m,i}c_i d_i}{a_i^3} \right) \]
\[ + \frac{1}{\sigma_{\theta}^2} \sum_{i=0}^{N-1} 2 \left( \frac{b_i (c_i^2 + d_i^2) - c_i d_i}{(c_i^2 + d_i^2)^2} \right), \]
\[ F_{25} = \frac{1}{\sigma_{\rho}^2} \sum_{i=0}^{N-1} 2 \left( \frac{\rho_{m,i}c_i^3}{a_i^3} \right), \]
\[ F_{33} = \frac{1}{\sigma_{\rho}^2} \sum_{i=0}^{N-1} 2t_i \left( \frac{-\rho_{m,i}}{a_i} + 1 + \frac{\rho_{m,i}d_i^2}{a_i^3} \right) \]
\[ + \frac{1}{\sigma_{\theta}^2} \sum_{i=0}^{N-1} 2t_i \left( \frac{(2d_i b_i + c_i) c_i}{(c_i^2 + d_i^2)^2} \right), \]
\[ F_{34} = \frac{1}{\sigma_{\rho}^2} \sum_{i=0}^{N-1} 2t_i \left( \frac{-\rho_{m,i}}{a_i} + 1 + \frac{\rho_{m,i}d_i^2}{a_i^3} \right) \]
\[ + \frac{1}{\sigma_{\theta}^2} \sum_{i=0}^{N-1} 2t_i \left( \frac{(2d_i b_i + c_i) c_i}{(c_i^2 + d_i^2)^2} \right), \]
\[ F_{35} = \frac{1}{\sigma_{\rho}^2} \sum_{i=0}^{N-1} 2t_i \left( \frac{\rho_{m,i}d_i e_i}{a_i^3} \right), \]
\[ F_{44} = \frac{1}{\sigma_{\rho}^2} \sum_{i=0}^{N-1} 2 \left( \frac{-\rho_{m,i}}{a_i} + 1 + \frac{\rho_{m,i}d_i^2}{a_i^3} \right) \]
\[ + \frac{1}{\sigma_{\theta}^2} \sum_{i=0}^{N-1} 2 \left( \frac{(2d_i b_i + c_i) c_i}{(c_i^2 + d_i^2)^2} \right), \]
\[ F_{45} = \frac{1}{\sigma_{\rho}^2} \sum_{i=0}^{N-1} 2 \left( \frac{\rho_{m,i}d_i c_i}{a_i^3} \right), \]
\[ F_{45} = \frac{1}{\sigma_{\rho}^2} \sum_{i=0}^{N-1} 2 \left( \frac{-\rho_{m,i}}{a_i} + 1 + \frac{\rho_{m,i}d_i^2}{a_i^3} \right), \]

where \( a_i = \sqrt{c_i^2 + d_i^2 + e_i^2}, \)
\( b_i = -\theta_{m,i} + \arctan \left( \frac{c_i}{d_i} \right), \) and \( c_i = v_x t_i + x_0 - x_{R,i}, d_i = v_y t_i + y_0 - y_{R,i}, \) and \( e_i = z_0 - z_{R,i}. \)

APPENDIX B

The solution for (5) is given by

\[ v_x^{[0]} = \frac{N \sum_{i=0}^{N-1} t_i A_i - \left( \sum_{i=0}^{N-1} t_i \right) \left( \sum_{i=0}^{N-1} A_i \right)}{N \sum_{i=0}^{N-1} t_i^2 - \left( \sum_{i=0}^{N-1} t_i \right)^2}, \]
\[ x_0^{[0]} = \frac{-\left( \sum_{i=0}^{N-1} t_i \right) \left( \sum_{i=0}^{N-1} t_i A_i \right) + \left( \sum_{i=0}^{N-1} t_i \right) \left( \sum_{i=0}^{N-1} A_i \right)}{N \sum_{i=0}^{N-1} t_i^2 - \left( \sum_{i=0}^{N-1} t_i \right)^2}, \]
\[ v_y^{[0]} = \frac{N \sum_{i=0}^{N-1} t_i B_i - \left( \sum_{i=0}^{N-1} t_i \right) \left( \sum_{i=0}^{N-1} B_i \right)}{N \sum_{i=0}^{N-1} t_i^2 - \left( \sum_{i=0}^{N-1} t_i \right)^2}, \]
\[ y_0^{[0]} = \frac{-\left( \sum_{i=0}^{N-1} t_i \right) \left( \sum_{i=0}^{N-1} t_i B_i \right) + \left( \sum_{i=0}^{N-1} t_i \right) \left( \sum_{i=0}^{N-1} B_i \right)}{N \sum_{i=0}^{N-1} t_i^2 - \left( \sum_{i=0}^{N-1} t_i \right)^2}, \]

where \( A_i = \rho_{m,i} \sin \theta_{m,i} + x_{R,i}, \) and \( B_i = \rho_{m,i} \cos \theta_{m,i} + y_{R,i}. \)
REFERENCES