An active beacon-based leader vehicle tracking system

Stanislaw Goll¹,2, Elena Zakharova ¹,2

¹ Department of Information-Measuring and Biomedical Engineering, Ryazan State Radio Engineering University (RSREU), Gagarin Street, 59/1, Ryazan, Russia
² LLC KB Avrora, Skomoroshinskaja Street 9, of 3, Ryazan, Russia

ABSTRACT
This article focuses on mobile robot convoying along a path travelled by a certain leader carrying the active ultrasonic beacon. The robot is equipped with the three-dimensional receiver array in order to receive both the ultrasonic wave and the RF wave marking the beginning of the measurement cycle. To increase measurement reliability, each receiver contains two independent measurement channels with automatic gain control. The distance measurements are pre-processed in order to identify the artefacts and then either remove them or replace them with the interpolated value. To estimate the position of the beacon in the robot’s local coordinate system, several methods are used, including the least squares method with subsequent exponential smoothing, the linear Kalman filter, the Rauch-Tung-Striebel smoother, the extended Kalman Filter, the unscented Kalman filter, and the particle filter. The experiments were undertaken in order to estimate the estimation method preferable for following the leader’s path.

1. INTRODUCTION
The convoying scenario is one of the most important tasks in modern robotics. A mobile robot autonomously following a leader can be widely used in such areas as agriculture, transportation, and the military. Many well-known international trials, such as ELROB, include this scenario either as an independent one or as a part of a more complex scenario.

The scenario is focused on the mobile robot autonomously following some leader, which can be either a human operator or another vehicle – either autonomous or remote-controlled. To detect the leader, different systems can be used, including GNSS systems [1], [2], video and infrared cameras [3]-[7], LiDARs [8]-[10], directional antennas [11], radars, ultrasonic rangefinders [12]-[14], and their combinations [15], [16]. The nature of these systems, however, can impose various restrictions on the conditions of their use. GNSS systems perform poorly in urban areas and inside buildings. LiDARs and radars can be used to measure the distance between the vehicle and the surrounding objects, but it is not an easy task to detect the leader based on range data. Cameras, both video and infrared, can help with leader detection but depend greatly on environmental conditions. Moreover, if the obstacle appears between the leader and the robot, the accuracy of their relative positions’ estimate can be severely decreased.

The mathematical methods used for localisation, obstacle detection, and occupancy grid mapping are not reliable enough to build a robust convoying system. Such methods usually require a redundant sensor data of a different nature. The leader detection suffers from the same constraints and, to make things even more difficult, the rough weather conditions, dense vegetation, smoke, and other factors should also be considered.

In section 2, we discuss the implementation of an ultrasonic-based leader detection system that consists of an active beacon carried by the leader and a set of several ultrasonic receivers mounted on the convoyed robot. In section 3, different methods of computing the estimate of the beacon’s position are described. Conclusions of this study, experimental results, and dynamic errors for all methods are illustrated in section 4. Finally, recommendations for future work are given in the last section.

2. THE APPROACH
In this article, we propose an ultrasonic-based leader detection system that includes an active beacon carried by the
leader and a set of $N$ ultrasonic receivers mounted on the convoyed robot (Figure 1).

The active beacon transmits the ultrasonic waves with constant time intervals between the waves. At the same time that the ultrasonic wave is sent, the radio frequency wave marking the beginning of the measurement cycle is also transmitted. For each of the $N$ receivers, the time interval between the moment of the radio frequency wave and ultrasonic wave arrivals is measured according to the time-of-flight principle. These intervals are proportional to the distances between the beacon and the respective receivers and can be calculated as $r_u = c \tau_u$, $n = 1, \ldots, N$, where $c$ is the speed of the ultrasound in the air, and $\tau_u$ is the time interval measured for the $n^{th}$ receiver.

To estimate the beacon’s coordinates in the robot’s local coordinate system, the system of $N$ nonlinear equations is solved.

$$r_u^2 = (x^n-x^n)^2+(y^n-y^n)^2+(z^n-z^n)^2,$$

where $n = 1, \ldots, N$, $\mathbf{r} = \begin{bmatrix} x^n & y^n & z^n \end{bmatrix}^T$ is a vector containing the coordinates of the beacon, and $\mathbf{r} = \begin{bmatrix} x^n & y^n & z^n \end{bmatrix}$ are the coordinates of the $n^{th}$ receiver in the convoyed robot’s local coordinate system.

During the system operation, an obstacle can cause line-of-sight loss between one or several receivers and the beacon. Moreover, the ultrasonic wave can be reflected by the surrounding objects, causing the problem of multipath propagation. For these reasons, acquired measurements can contain artefact distances $r_u$ (Figure 2(a)), which are identified using the threshold constant $\rho$. The distance measurement $r_{u,k}$ is considered an artefact if the following condition is met:

$$r_{u,k} - r_{u,k+1} > \rho,$$

where $\rho$ is an adjustable threshold constant, and $k$ is the consequent measurement number.

If the artefact is detected, the linear least squares extrapolation is used to calculate the substitute estimate based on the last $V$ estimates, the corresponding equations are excluded from the system (1). When the reliable measurements (i.e. condition (2) is not met for the two consecutive measurements) arrive, the receiver is included back in the system (1), and the artefact removal procedure becomes applicable again. The system must contain at least three reliable measurements from different receivers to calculate the beacon’s position estimate $\hat{x}$. If there are not enough reliable measurements available, the robot stops until the beacon’s position can be calculated again.

To make more robust measurements, each receiver contains two independent measurement channels with partly overlapping beam patterns and automatic gain control. If both channels succeeded at the distance measurement at some moment of time $k$, then the average value is used as a resulting measurement. If one of the channels failed to provide a measurement, then the over channel’s measurement is used as a resulting measurement.

3. POSITION ESTIMATION

Many of the effective estimation methods are based on the linear models and use normally distributed values. Figure 2b and 2c show the histograms of centred values for the measured distances $r_u$ between the beacon and the $n^{th}$ receiver with the...
beacon positioned along the axis of the receiver’s beam pattern at a distance of 11 and 3 metres respectively. Clearly, the histograms are multimodal with the constant distances \( c \cdot \Delta r \) between the modes, where \( \Delta r \) is the period of ultrasound. Since the envelope’s shape can be approximated with the Gaussian function, we can assume that the distances \( r_k \) are distributed normally, with the variance depending on the beacon’s position.

We can use a linear combination of the equations to transform the nonlinear equations of (1) into the linear ones [17]. For each pair of receivers, the difference of the squared distances to the beacon is calculated as follows:

\[
\begin{align*}
(r_i^2 - r_j^2) &= (x_i^2 - x_j^2) + (y_i^2 - y_j^2) + (z_i^2 - z_j^2) \\
&= (x_i^2 - x_j^2) - (y_i^2 - y_j^2) - (z_i^2 - z_j^2),
\end{align*}
\]

where \( i, j = 1, \ldots, N \), \( i \neq j \), \( C_N^2 \) – binomial factor, \( L = C_N^2 \) – the number of receiver combinations.

As a result, system (1) can be written in the form of the linear system

\[
\begin{align*}
\begin{bmatrix} \mathbf{v} \\
\end{bmatrix} &= \mathbf{B} \begin{bmatrix} \mathbf{x} \\
\end{bmatrix},
\end{align*}
\]

where \( \mathbf{B} \) is the system matrix, with \( L \) rows containing

\[
\begin{align*}
2(x_j - x_i), \quad 2(y_j - y_i), \quad 2(z_j - z_i),
\end{align*}
\]

and \( \mathbf{v} \) is the column vector of \( L \) constant terms

\[
(r_i^2 - r_j^2) = (x_i^2 - x_j^2) + (y_i^2 - y_j^2) + (z_i^2 - z_j^2),
\]

\( i, j = 1, \ldots, N \), \( i \neq j \).

Generally, the system (3) is inconsistent due to the measurement noise. Hence, to estimate the position of the beacon, the least squares method can be used:

\[
\begin{align*}
\begin{bmatrix} \mathbf{x} \\
\end{bmatrix} &= \mathbf{B}^\dagger \begin{bmatrix} \mathbf{v} \\
\end{bmatrix}.
\end{align*}
\]

To compute the estimate \( \hat{\mathbf{x}} \) of the beacon’s position, different methods can be used; therefore, several estimates were computed in the course of this research. During the research, the receiver array of \( N = 4 \) receivers was used; therefore, the system (3) contains \( L = C_N^2 = 6 \) equations.

### 3.1. Least squares method with exponential smoothing

\[
\begin{align*}
\hat{x}_k = \alpha \hat{x}_{k-1} + (1 - \alpha) x_k, \quad \alpha \in (0; 1),
\end{align*}
\]

where \( \alpha \) is the adjustable smoothing factor.

### 3.2. Kalman filter with the non-stationary measurement noise matrix

The overall filter design is undertaken according to [18]. The stochastic system model can be described as follows:

\[
\begin{align*}
\begin{bmatrix} x_k \n y_k \n \zeta_k \n v_k \\
\end{bmatrix} &= \begin{bmatrix} x_k \n y_k \n \zeta_k \n v_k \\
\end{bmatrix} + \mathbf{A} \begin{bmatrix} x_k \n y_k \n \zeta_k \n v_k \\
\end{bmatrix},
\end{align*}
\]

where \( \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\
0 & 1 & 0 & 0 & \Delta t & 0 \\
0 & 0 & 1 & 0 & 0 & \Delta t \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
\end{bmatrix} \) is the stationary transition matrix of the dynamic model, where \( \Delta t \) is the time interval between the consecutive measurements;

\[
\mathbf{z}_k = \begin{bmatrix} g_{I,k} \cdots g_{L,k} \end{bmatrix}^T
\]

is the measurement vector;

\( C \) is the stationary measurement model, with \( L \) rows

\[
\begin{align*}
2(x_j - x_i), \quad 2(y_j - y_i), \quad 2(z_j - z_i), \quad 0, \quad 0, \quad 0
\end{align*}
\]

\( i, j = 1, \ldots, N \), \( i \neq j \);

\[
\mathbf{z}_k = \begin{bmatrix} 0 & 0 & 0 & a_k^x & a_k^y & a_k^z \end{bmatrix} \]

is process noise,

\[
\mathbf{z}_k \sim \mathcal{N} \begin{bmatrix} \mathbf{0} \n \mathbf{0} \end{bmatrix}.
\]

\( Q = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2) \) is the stationary process noise covariance matrix;

\( \mathbf{R} \) is the observation noise matrix. To estimate the measurement noise covariance [19], we assume that

\[
\mathbf{R}_k = \begin{bmatrix} R_0, \quad k < D, \\
\frac{1}{D} \sum_{d=k-D}^k \mathbf{z}_d \mathbf{z}_d^T + CP_k C^T, \quad k \geq D,
\end{bmatrix}
\]

\( D \) is the estimation window size.

The initial state vector is assumed to be

\[
\mathbf{s}_0 = \begin{bmatrix} x_0 \n y_0 \n z_0 \n 0 \n 0 \n 0 \end{bmatrix}^T,
\]

where \( \sigma_{x_0}, \sigma_{y_0}, \sigma_{z_0} \) are the noise parameters.

**Algorithm:**

1. **input:** \( x_0, P_0, R_0, A, Q, C, D \)
2. **output:** \( \mathbf{s}_k, \mathbf{P}_k \)
3. **begin**
4. **for** \( k = 1 \rightarrow K \)
5. **Prediction**
6. \( \mathbf{S}_{k-1} = \mathbf{A} \mathbf{S}_{k-1} \)
7. \( \mathbf{P}_{k-1} = \mathbf{A} \mathbf{P}_{k-1} \mathbf{A}^T + \mathbf{Q} \)
8. **Update**
9. \( \mathbf{z}_{k-1} = \mathbf{z}_k - \mathbf{C} \mathbf{s}_{k-1} \)
10. \( \mathbf{G} = \mathbf{P}_{k-1} \mathbf{C}^T \mathbf{R}_k^{-1} \mathbf{C} \mathbf{P}_{k-1} \mathbf{C}^T \)
11. \( \mathbf{S}_k = \mathbf{S}_{k-1} + \mathbf{G} \mathbf{z}_{k-1} \)
12. \( \mathbf{P}_k = (\mathbf{I} - \mathbf{G} \mathbf{C}) \mathbf{P}_{k-1} \)
13. \( \mathbf{z}_k = \mathbf{z}_k - \mathbf{C} \mathbf{s}_k \)
14. **if** \( k < D \) **then**
15. \( \mathbf{R}_{k+1} = \mathbf{R}_0 \)
16. **end**
17. **else**
18. \( \mathbf{R}_{k+1} = \frac{1}{D} \sum_{d=k-D}^k \mathbf{z}_d \mathbf{z}_d^T + \mathbf{C} \mathbf{P}_k \mathbf{C}^T \)
19. **end**
20. **end**
where \[
\begin{bmatrix}
\mathbf{x}_0^0 & \mathbf{y}_0^0 & \mathbf{z}_0^0
\end{bmatrix}^T = (B^T B)^{-1} B^T \mathbf{z}_0.
\]

The posteriori error covariance matrix and the measurement noise covariance matrix are written as

\[
P_0 = \text{diag} \left( \sigma_{x_1}^2, \sigma_{x_2}^2, \sigma_{z_1}^2, \sigma_{z_2}^2, \sigma_{x_3}^2 \right)
\]

and

\[
R_0 = \text{diag} \left( \sigma_{x_1}^2, \ldots, \sigma_{x_n}^2 \right)
\]

respectively.

### 3.3. Rauch-Tung-Striebel smoother

The sequence of the beacon’s state estimates \{\hat{x}_k, \hat{x}_{k-1}, \ldots, \hat{x}_{T-1}\} can be used to set the desired path for the mobile robot. If the corresponding covariance matrices \{P_k, P_{k-1}, \ldots, P_{T-1}\} are also known, then this path-to-be can be smoothed using the Rauch-Tung-Striebel smoother. The estimate and covariance sequences are rewritten as \{\hat{x}_T, \hat{x}_{T-1}, \ldots, \hat{x}_0\} and \{P_T, P_{T-1}, \ldots, P_0\} accordingly. The initial smoothed estimate is \[ \hat{x}_T = \hat{x}_T \] with the covariance\[ P_T = P_T. \] Then the smoothed is applied from the last time step to the first (i.e., \( t = T-1, \ldots, 0 \)) according to the description given in [18].

```plaintext
1: input: \{\hat{x}_T, \hat{x}_{T-1}, \ldots, \hat{x}_0\}, \{P_T, P_{T-1}, \ldots, P_0\}, A, Q, T
2: output: \{\hat{x}'_T, \hat{x}'_{T-1}, \ldots, \hat{x}'_0\}, \{P'_T, P'_{T-1}, \ldots, P'_0\}
3: begin
4: for \( t = T \rightarrow 0 \)
5: \( \hat{x}_{t+1k} = A \hat{x}_t \)
6: \( P_{t+1k} = AP_t A^T + Q \)
7: \( \hat{x}'_t = \hat{x}_t + \left( P'_{t+1k} \right)^{-1} \left( \hat{x}_{t+1k} - \hat{x}_t \right) \)
8: \( P'_t = P_t + \left( P'_{t+1k} \right)^{-1} \left( \hat{x}_{t+1k} - \hat{x}_t \right) \left( \hat{x}_{t+1k} - \hat{x}_t \right)^T \left( P'_{t+1k} \right)^{-1} \)
9: end
```

### 3.4. Extended Kalman filter

According to [18], the stochastic system model can be described as follows:

\[
\begin{aligned}
\mathbf{x}_{t+1} &= A \mathbf{x}_t + \mathbf{w}_t, \\
\mathbf{y}_t &= \mathbf{h} (\mathbf{x}_t) + \mathbf{v}_t,
\end{aligned}
\]

\( \mathbf{h} (\mathbf{x}_t) \) is the vector-valued function containing \( N \) elements \( \sqrt{\left( \mathbf{x}_t^u - \mathbf{x}_t^u \right)^2 + \left( \mathbf{y}_t - \mathbf{y}_t \right)^2 + \left( \mathbf{z}_t - \mathbf{z}_t \right)^2} \), \( u = 1, \ldots, N \);

\[ \mathbf{R}_k = \begin{bmatrix} \mathbf{r}_1 & \ldots & \mathbf{r}_N \end{bmatrix}^T \] is the measurement vector;

\( \mathbf{P}_k \) is the covariance matrix of the state;

\( \mathbf{R}_k \) is the non-stationary measurement noise covariance matrix.

To estimate the measurement noise covariance [20], we assume that

\[
E \left[ \mathbf{z} \mathbf{z}^T \right] = E \left[ \mathbf{r} \mathbf{r}^T \right] - H_k P_k H_k^T, \quad \text{where} \quad \mathbf{r}_k = \mathbf{r}_k - \mathbf{h} (\mathbf{x}_k) \]
is residual, when

\[
\begin{align*}
R_k &= \begin{cases} R_0, & k < D, \\
\frac{I}{D} \sum_{d=1}^{K} \mathbf{j}_d \mathbf{j}_d^T + H_k P_k H_k^T, & k \geq D,
\end{cases}
\end{align*}
\]

\( D \) is the estimation window size,

\( H_k = \frac{d}{d \mathbf{x}_k} \) is the Jacobian matrix with \( N \) rows

\[
\begin{bmatrix}
\sqrt{\left( \mathbf{x}_k^u - \mathbf{x}_k^u \right)^2 + \left( \mathbf{y}_k - \mathbf{y}_k \right)^2 + \left( \mathbf{z}_k - \mathbf{z}_k \right)^2} \\
\sqrt{\left( \mathbf{x}_k^u - \mathbf{x}_k^u \right)^2 + \left( \mathbf{y}_k - \mathbf{y}_k \right)^2 + \left( \mathbf{z}_k - \mathbf{z}_k \right)^2} \\
\sqrt{\left( \mathbf{x}_k^u - \mathbf{x}_k^u \right)^2 + \left( \mathbf{y}_k - \mathbf{y}_k \right)^2 + \left( \mathbf{z}_k - \mathbf{z}_k \right)^2} \\
0 \\
0 \\
0
\end{bmatrix},
\]

where \[ \begin{bmatrix} \mathbf{x}_0^0 & \mathbf{y}_0^0 & \mathbf{z}_0^0 \end{bmatrix} = (B^T B)^{-1} B^T \mathbf{z}_0; \]

\( P_0 = \text{diag} \left( \sigma_{x_1}^2, \sigma_{x_2}^2, \sigma_{z_1}^2, \sigma_{z_2}^2, \sigma_{x_3}^2 \right) \);

\( R_0 = \text{diag} \left( \sigma_{x_1}^2, \ldots, \sigma_{x_n}^2 \right) \).

```plaintext
1: input: \mathbf{x}_0, \mathbf{P}_0, \mathbf{R}_0, A, Q, \mathbf{h}, \mathbf{D}
2: output: \mathbf{x}_k, \mathbf{P}_k
3: begin
4: for \( k = 1 \rightarrow K \)
5: Prediction
6: \( \mathbf{S}_k = A \mathbf{S}_{k-1} A^T + \mathbf{Q} \)
7: Update
8: \( \mathbf{X}_k = \mathbf{X}_{k-1} + \mathbf{P}_{k-1} \mathbf{h} \)
9: \( \mathbf{P}_k = \mathbf{P}_{k-1} H_k^T \left( \mathbf{R}_k + H_k \mathbf{P}_{k-1} H_k^T \right)^{-1} \)
10: \( \mathbf{S}_k = \mathbf{S}_{k-1} + \mathbf{G}_k \mathbf{S}_{k-1} \)
11: \( \mathbf{P}_k = \left( I - \mathbf{G}_k \mathbf{S}_{k-1} \right) \mathbf{P}_{k-1} \)
12: \( \mathbf{r}_k = \mathbf{r}_k - \mathbf{h} (\mathbf{x}_k) \)
13: if \( k < D \) then
14: \( \mathbf{R}_{k+1} = \mathbf{R}_0 \)
15: else
16: \( \mathbf{R}_{k+1} = \frac{I}{D} \sum_{d=1}^{K} \mathbf{j}_d \mathbf{j}_d^T + H_k \mathbf{P}_k H_k^T \)
17: end
end
```

}\end{document}
3.5. Unscented Kalman filter

In the initial step, the set of sigma points is computed as
\[
X_0 = \left[ \begin{bmatrix} x_0^w \mid \ldots \mid x_0^w \end{bmatrix} + \sqrt{U + \lambda} \begin{bmatrix} 0 \mid \ldots \mid 0 \end{bmatrix} \right] + \sqrt{P_0} \begin{bmatrix} 0 \mid \ldots \mid 0 \end{bmatrix},
\]
where
\[
S_0 = \left[ \begin{bmatrix} x_0^w \mid \ldots \mid x_0^w \end{bmatrix} + \begin{bmatrix} 0 \mid \ldots \mid 0 \end{bmatrix} \right]^T,
\]
\[
\left[ \begin{bmatrix} x_0^w \mid \ldots \mid x_0^w \end{bmatrix} \right]^T = (B^T B)^{-1} B^T g_0.
\]
Each sigma point has its own weight, calculated by the formulas
\[
W = \left( I - \begin{bmatrix} w^w \mid \ldots \mid w^w \end{bmatrix} \right) \cdot \text{diag}(w_0, \ldots, w_{2U}) \times
\left( I - \begin{bmatrix} w^w \mid \ldots \mid w^w \end{bmatrix} \right)^T,
\]
\[
\begin{array}{l}
\text{input: } X_0, P_0, R_0, A, Q, W, \lambda, U, D, \tilde{b} \\
\text{output: } \bar{x}_k, P_k \\
\text{begin} \\
\quad \text{for } k = 1 \rightarrow K \\
\quad \quad \text{Prediction} \\
\quad \quad X_{k|k-1} = AX_k^{k-1} \\
\quad \quad S_{4k-1} = X_k^{k-1} \cdot R^{k-1} \\
\quad \quad P_{4k-1} = X_k^{k-1} W X_k^{k-1} + Q \\
\quad \quad \text{Update} \\
\quad \quad X_{4k-1} = \left[ S_{4k-1} \mid \ldots \mid S_{4k-1} \right] + \\
\quad \quad \quad \sqrt{U + \lambda} \begin{bmatrix} 0 \mid \ldots \mid 0 \end{bmatrix} \begin{bmatrix} X_k^{k-1} \mid \ldots \mid X_k^{k-1} \end{bmatrix} + \\
\quad \quad \quad \sqrt{P_{4k-1}} \begin{bmatrix} 0 \mid \ldots \mid 0 \end{bmatrix}
\end{array}
\]
\[
Y_{4k-1} = \bar{b}(X_{4k-1})
\]
\[
\bar{y}_{4k-1} = \bar{r}_k - Y_{4k-1} \cdot w^w
\]
\[
G_k = X_{4k-1} W Y_{4k-1} \left( Y_{4k-1} W Y_{4k-1} + R_k \right)^{-1}
\]
\[
S_{k} = S_{4k-1} + G_k \bar{y}_{4k-1}
\]
\[
P_k = P_{4k-1} - G_k \left( Y_{4k-1} W Y_{4k-1} + R_k \right) G_k^T
\]
\[
X_k = \left[ \begin{bmatrix} S_k \mid \ldots \mid S_k \end{bmatrix} \right] + \\
\quad \quad \sqrt{U + \lambda} \begin{bmatrix} 0 \mid \ldots \mid 0 \end{bmatrix} \begin{bmatrix} P_k \mid \ldots \mid P_k \end{bmatrix} - \\
\quad \quad \sqrt{P_0} \begin{bmatrix} 0 \mid \ldots \mid 0 \end{bmatrix}
\end{array}
\]
\[
\bar{y}_k = \bar{r}_k - Y_k \cdot w^w
\]
\[
\text{if } k < D \text{ then} \\
\quad \quad R_{k+1} = R_{0} \\
\quad \quad \text{else} \\
\quad \quad \quad R_{k+1} = \frac{1}{D} \sum_{d=k-D}^{k} \bar{y}_d \bar{y}_d^T + Y_k W Y_k^T
\]
\end{array}
\]
where
\[
\begin{bmatrix} w^w \mid \ldots \mid w^w \end{bmatrix}^T, \ w_0^w = \frac{\lambda}{U + \lambda},
\]
\[
w_0 = \frac{\lambda}{U + \lambda} + 1 - \alpha^2 + \beta, \ w_0^w = \frac{t}{2(U + \lambda)}, \ i = 1, \ldots, 2U;
\]
\[
U = 6 \text{ refers to the number of components in the state vector } \bar{x}_k;
\]
\[
\lambda = \alpha^2(U + \kappa) - U \text{ is a scaling parameter;}
\]
\[
\alpha, \kappa, \beta \text{ are the parameters of the method;}
\]
\[
R_k \text{ is the non-stationary measurement noise covariance matrix. To estimate the measurement noise covariance [21], we assume that}
\]
\[
E\left[ \bar{r}_k \bar{r}_k^T \right] = E\left[ \bar{y}_k \bar{y}_k^T \right] - Y_k W Y_k^T, \text{ where } \bar{r}_k = \bar{r}_k - Y_k \cdot w^w \text{ is residual,}
\]
\[
R_k = \left[ \begin{array}{c} R_k, \\
\vdots \end{array} \right], \quad k < D,
\]
\[
R_k = \left[ \begin{array}{c} R_k, \\
\vdots \end{array} \right], \quad k \geq D,
\]
\[
D \text{ is the estimation window size,}
\]
\[
Y_k = \bar{b}(X_k), \text{ } X_k \text{ is the array of sigma points.}
\]
The initial value of \[ P_0 = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_v^2, \sigma_v^2), \]
\[
R_0 = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_v^2, \sigma_v^2).
\]

The filter implementation complies with [18].

3.6. Particle filter

To describe the beacon’s position, the following state vector
\[
\bar{x} = \begin{bmatrix} x^w \mid y^w \mid z^w \mid v^w \mid v^w \end{bmatrix}^T
\]
is used, where
\[
\bar{x} \sim \mathcal{N}(\bar{x}_0, \Sigma).
\]

The initial estimate is \[ \bar{x}_0 = \begin{bmatrix} x_0^w \mid y_0^w \mid z_0^w \mid 0 \mid 0 \end{bmatrix}^T, \]
where \[ \bar{x}_0 = \begin{bmatrix} x_0^w \mid y_0^w \mid z_0^w \mid 0 \mid 0 \end{bmatrix}^T, \]
\[
\Sigma = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_v^2, \sigma_v^2).
\]

Based on the distribution \[ \mathcal{N}(\bar{x}_0, \Sigma), \]
the set of particles \[ X_0 \]
is generated. For each particle \[ S_{j,k}, \]
the weight \[ w_{j,0} = \frac{1}{f} \]
is set, with the sum of the weights being \[ w_X = \sum_{j=0}^f w_{j,0} = 1, \]
\[ j = 1, \ldots, f. \]
input: $X_0, A, \beta, N, J, \sigma, \{x_1, \ldots, x_N\}, \{y_1, \ldots, y_N\}, \{z_1, \ldots, z_N\}$

output: $S_k$

begin
for $k = 1 \rightarrow K$

\[X_k = AX_{k-1}\]

for $n = 1 \rightarrow N$

\[\hat{r}_{n,j,k} = \sqrt{(x_{j,k}^n - x_s)^2 + (y_{j,k}^n - y_s)^2 + (z_{j,k}^n - z_s)^2}\]

end

for $j = 1 \rightarrow J$

\[w_{j,k} = \mathbb{N}(\hat{r}_{j,k}, \sigma) \]

end

The particles with the weights less than $\beta$ are removed

\[X_{\text{old}} = \{S_{j,k} \mid w_{j,k} > \beta\}, j = 1, \ldots, J\]

end

\[\bar{S}_k = \frac{\sum_{j=1}^{J} w_{j,k} S_{j,k}}{\sum_{j=1}^{J} w_{j,k}}\]

end

\[X_k = X_{\text{old}} + X_{\text{avg}}\]

end

end

4. RESULTS

During the experiments, an array of $N = 4$ receivers was statically placed, while a mobile robot carrying the active beacon was moved in front of it following a rectangular path (Figure 3) with a width of 4 m and a length of 7 m (1st experiment) or 10.5 m (2nd experiment). The robot moved clockwise, starting and finishing at the coordinates $(−2, 1)$ in the array’s local coordinate system.

For each beacon’s position, we computed a total of six estimates using the methods described above. The number of particles in the particle filter is $J = 5000$. To compare the performance of these methods, we calculated the average of the Root Mean Square Errors (RMSE).

Dynamic errors in the form of distances between the estimate of the active beacon trajectory and the exemplary trajectory in the two-dimensional $|\Delta_s|$ and three-dimensional coordinate systems $|\Delta_s|$ in time are presented in Figure 3.

The grey area on the dynamic error graphs marks the parts of the active beacon path corresponding to the smallest sides of the rectangle. On the left-hand side, the path estimates and their dynamic errors are shown, which are calculated based on the shifted differences of the square distances between the beacon and each of the receivers $r_{e_k}$. On the right-hand side, the path estimates and their dynamic errors are shown, which are calculated based on the distances between the beacon and each of the receivers $r_{e_k}$.

Table 1 shows the RMSEs of the beacon’s position estimates during the first experiment. Errors RMSE($|\Delta_s|$) and RMSE($|\Delta_s|$) correspond to the estimates made in the two-dimensional and three-dimensional coordinate systems, respectively.

The Rauch-Tung-Striebel smoother outperforms other considered methods in terms of RMSEs and dynamic errors. However, the path estimates shown in Figure 3 appear to be more parallelogrammatic than rectangular, altering the shape of the original path. These alterations were caused by the side wind during the experiment.

During the second experiment, no wind was present. Furthermore, the length of the ground truth rectangle was increased to 10.5 m. The path estimates and the corresponding dynamic errors are shown in Figure 4. The RMSEs are shown in Table 2.

Clearly, the Rauch-Tung-Striebel smoother demonstrates the best estimate again, despite changes in the experimental conditions.

Table 1. The root mean square errors of the beacon’s position estimates during the first experiment

| Method | RMSE($|\Delta_s|$), m | RMSE($|\Delta_s|$), m |
|--------|------------------|------------------|
| ES     | 0.7365           | 0.8948           |
| RTS    | 0.6161           | 1.0029           |
| KF     | 0.4776           | 0.7410           |
| EKF    | 0.5569           | 0.9255           |
| UKF    | 0.4932           | 0.8875           |
| PF     | 0.4830           | 1.0422           |
The main goal of our future research in this area is to eliminate biases caused by the wind and measurement errors during the estimation of the system’s structural parameters. To estimate these parameters, we plan to perform a calibration procedure involving the 3D LiDAR mounted on the robot. Given the 3 cm accuracy of the LiDAR measurements and usage of the machine learning algorithms, such calibration should outperform the current manual measurements made with the slide rule. Moreover, it should also allow for estimating the size and the shape of the receivers.

Currently, the effective measurement range is up to 20 m; however, it can be extended to 100 m. This can be achieved by combining the ultrasonic measurements with the GNSS-based measurements. In this case, it is possible to mitigate the relative displacement between the robot’s and beacon’s GNSS receivers by mixing the ultrasonic and the GNSS data, while the distance between the robot and the beacon is relatively small.

REFERENCES


[18] J. Hartikainen, S. Sumo, Optimal filtering with Kalman filters and smoothers, a manual for the Matlab toolbox EKF/UKF, Department of Biomedical Engineering and Computational Science, Aalto University School of Science, Espoo, Finland, August 2011, p.129.

