Dynamic Ultrasonic Hybrid Localization System for Indoor Mobile Robots

Seong Jin Kim and Byung Kook Kim, *Member, IEEE*

***Abstract*—An accurate dynamic ultrasonic hybrid localization system is presented for autonomous navigation of indoor mobile robots using multiple ultrasonic distance measurements and an extended Kalman filter (EKF). The ultrasonic sensor subsystem is composed of several ultrasonic transmitters (Txs) attached to the ceiling at known positions and several ultrasonic receivers equilaterally located on the top of the mobile robot, which has a moving speed that is not negligible. An EKF-based algorithm with a state/observation vector composed of the robot pose (or the position and the orientation) is presented using odometric and ultrasonic distance measurements. A dynamic distance estimation method is proposed to track the estimates of ultrasonic distance information from available Txs of interest using both odometric information from the robot and actual ultrasonic distance measurements. This continuous dynamic distance estimation allows persistent use of the hybrid self-localization algorithm to accurately determine the pose of the robot. The experimental results with various trajectories clearly show that the proposed method is much more accurate than only the hybrid self-localization algorithm (without the dynamic distance estimation method).**

***Index Terms*—Distance measurements, extended Kalman filter (EKF), localization, mobile robots, ultrasonic sensor.**

I. Introduction

LOCALIZATION for autonomous mobile robots is the process of determining and tracking the position and the orientation (i.e., the pose) of the robots in any given environment. Localization is a fundamental technique that allows robots to navigate, explore, or perform their intended tasks successfully without human intervention. The topic of localization has received considerable attention over the past few decades [1], [2].

Suitable sensory systems, including proprioceptive and exteroceptive types, are vital for the task of localization for robots. Using proprioceptive sensors, such as an inertial navigation system (sensing acceleration and angular velocity components) [3], [4] or odometry (sensing rotation angles of wheels) [5], most mobile robots estimate their internal information. Odometry is a widely used method due to its ease of use, low cost, and high update rate, and it is also a good estimator of localization if given a reasonably short time interval. However, this method results in an unbounded accumulation of errors depending on the distance traveled by the robot. Several efforts to control the errors produced by odometry, such as propagation of odometry errors [6], [7], calibration of systematic errors [8]–[10], and modeling of nonsystematic errors (or random errors) [11], have been presented.

Exteroceptive sensory systems, such as ultrasonic sensors, laser range finders (LRFs) (or light detection and ranging devices), or vision sensors, have given rise to alternative techniques to address the aforementioned limitation by providing external information about the environment around the robot. Utilizing these sensory systems, the pose of the robot can be determined using external measurements (distances, angles, or both) to the distinct features, landmarks, or beacons in the environment. Self-localization methods can be applied when the positions of the features are known and all external measurements correspond to the same robot pose. *Trilateration* using the distances to three or more other known features [12], [13] or *triangulation* using angles along with known landmarks [14], [15] are widely used as representative examples of self-localization methods. Other cases using both distances and angles are presented in studies by Se *et al.* [16] and Penne *et al.* [17].

When the robot moves continuously, and external measurements are obtained at different times, hence, these external measurements do not precisely coincide with identical robot poses. Thus, the self-localization method can result in large errors in the calculation of the robot pose. To solve this problem, most approaches combine odometry values (*prediction phase*) with measurements from the external environment (*correction phase*) to obtain optimal estimations of the robot pose. The extended Kalman filter (KF) (EKF) is the most extensively used tool to perform this fusion technique [18]. Some remarkable work using the EKF or its variants for localization can be found in the literature [19], [20]. Another tool to fuse odometry and external measurements is the particle filter, which is one of the so-called probabilistic approaches [18], [21].

This study considers how self-localization algorithms can be used to provide an accurate estimation of the pose of a moving robot that can be incorporated into the EKF process. These algorithms can only be used to determine the static poses of a robot. Font and Batlle proposed a two-step estimator, which is composed of triangulation and the angular-state EKF, given odometry and angular measurements from a laser sensor on a robot and three retroreflectors [22]. Lee *et al.* presented the two KFs (an EKF with distance and position state and a linear KF

Manuscript received October 19, 2011; revised May 6, 2012 and July 18, 2012; accepted July 30, 2012. Date of publication September 4, 2012; date of current version May 16, 2013. This work was supported in part by the Ministry of Knowledge Economy through the Human Resources Development Program for Convergence Robot Specialists, funded by the Korean Government.

S. J. Kim is with the Systems Solution Research Department, Research Institute of Industrial Science and Technology, Pohang 790-600, Korea (e-mail: sjkim@rtcl.kaist.ac.kr).

B. K. Kim is with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea (e-mail: bkkim@kaist.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2012.2216235

with the position state/observation) and a combination method using several trilateration methods [23]. They used odometry and distance measurements between three beacons and a listener on the robot. These previous works attempted to use self-localization algorithms and the EKF at the same time, but it is possible to localize a moving robot, given three landmarks only. However, in a real environment, such as hallways and rooms of an office building, more than three landmarks would be placed.

In this paper, the problem of tracking mobile robots with a given initial pose (*a priori*) is considered, where a group of ultrasonic transmitters (Txs) is fixed at known reference positions and there is an equilateral ultrasonic receiver (Rx) array on the top of the robot. The proposed algorithm is based on an EKF with a robot pose state/observation vector using odometry and ultrasonic distance measurements between Txs and Rxs on the robot. The state/observation vectors simplify the observation equation of the EKF, and the linearity of this equation improves the convergence of the filter. In a previous work, a hybrid self-localization algorithm in a static case, called ultrasonic hybrid localization (UHL), was developed to obtain the best possible accuracy over an entire region of interest by using ultrasonic distance measurements from available Txs [24].

This paper presents an accurate dynamic UHL (DUHL) algorithm that combines a dynamic distance estimation method for moving robots and the UHL algorithm to generate the pose of the robot within the framework of an EKF. This estimation method can predict ultrasonic distances from the previous robot pose to the current robot pose using odometric information. This method can also manage the number of available Txs and discard the predicted distances that have a large uncertainty. Hence, this DUHL algorithm can then determine the robot pose accurately because of all of the estimates of distances at the same time instant (i.e., at the same robot pose).

This paper is organized as follows. Section II presents an overview of the localization subsystem using ultrasonic sensors and the UHL algorithm. In Section III, an EKF-based algorithm with a state/observation vector composed of the robot pose is proposed. Section IV presents a dynamic ultrasonic distance estimation method to complete the DUHL algorithm. The experimental setup and result are given in Section V, and Section VI presents concluding remarks.

II. Localization Subsystem Using Ultrasonic Sensors and the UHL Algorithm

Ultrasonic sensors are widely used for perception of the environment and obstacle detection in many robotic applications. These sensors are quite successful in terms of cost efficiency, processing time, and accuracy. Localization subsystems that use ultrasonic sensors as active beacons and sensor arrays on robots also have been widely studied [24]–[33]. Although artificial landmark and active beacon approaches are not flexible compared with the ones using natural landmarks [34]–[36], the proficiency to identify landmarks and obtain active readings of landmarks can be enhanced, and the mapping process can be simplified.

The localization performance of these systems depends on the correct location of beacons. An ineffective configuration of beacons results in some areas that are either not covered at all (e.g., a robot does not receive ultrasonic signals from any beacons) or when the position cannot be calculated (e.g., when beacons and a robot are placed in a collinear manner). Hence, these systems can use relatively simple layouts of beacons (e.g., regular polygonal layouts) and then can cover a given area with cost efficiency as long as a sufficient number of beacons (two or more beacons in this paper) are available. These layouts can produce high localization accuracy around the center of polygons and are easy to deploy.

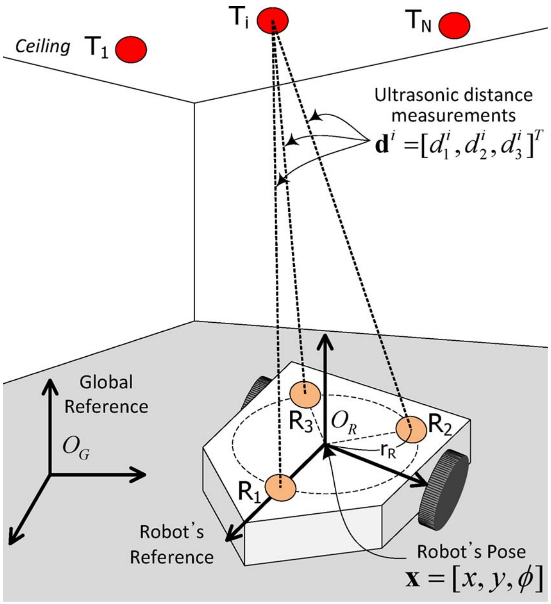
Ceiling
T_1
T_i
T_N
Ultrasonic distance measurements
$\mathbf{d}^i = [d_1^i, d_2^i, d_3^i]^T$
Global Reference
O_G
R_3
O_R
R_2
r_R
R_1
Robot's Reference
Robot's Pose
$\mathbf{x} = [x, y, \phi]$

Fig. 1. Localization subsystem using ultrasonic sensors. The origin of the global coordinate system O_G, the origin of the robot coordinate system O_R, the ultrasonic Tx module T_i, and the ultrasonic Rxs (R_1, R_2, and R_3) on the mobile robot.

This section describes the localization subsystem, which uses ultrasonic sensors and the UHL algorithm.

A. Localization Subsystem Using Ultrasonic Sensors

In this localization subsystem using ultrasonic sensors, several ultrasonic Txs are fixed to the ceiling with known positions in the global coordinate system. Three ultrasonic Rxs are equilaterally placed on the top of a mobile robot, as shown in Fig. 1. It is assumed that the robot moves on an indoor plane, which is common for most indoor applications and that the z-value of the robot and the Rxs in the global coordinate system (with the origin O_G) is zero. Hence, the pose of the robot can be described by a vector $\mathbf{x} = [x, y, \phi]^{\mathrm{T}}$, which is composed of the position (x, y) and the orientation (or the heading angle) ϕ in the global coordinate system. Let the robot coordinate be located at the center of the robot O_R and the x-axis be aligned to the forward direction. Then, the positions of the Rxs, ${}^{L}R_\ell = [{}^{L}x_{R_\ell}, {}^{L}y_{R_\ell}, 0]^{\mathrm{T}}$, in the robot coordinate system are given by

$$ {}^{L}R_\ell = r_R \cdot [\cos\gamma_\ell, \sin\gamma_\ell, 0]^{\mathrm{T}}, \quad \ell = 1, 2, 3 \tag{1} $$

where r_R is the distance from the center of the robot to each Rx and $\gamma_\ell = 2\pi(\ell - 1)/3$ is the angle of the ℓth Rx to the

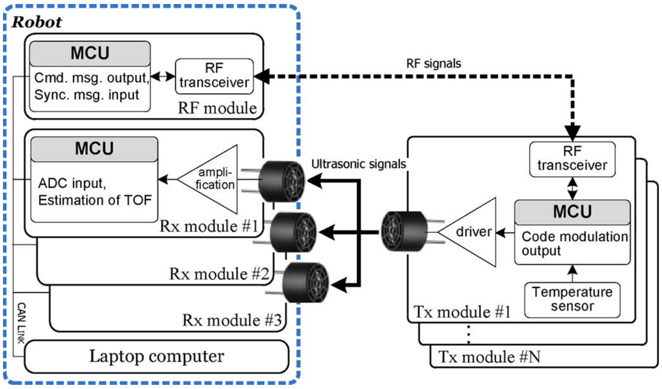

Fig. 2. Block diagram of the ultrasonic localization subsystem.

robot longitudinal axis, as shown in Fig. 1. The positions of the Rxs, $R_\ell = [x_{R_\ell}, y_{R_\ell}, 0]^{\mathrm{T}}$, in the global coordinate system are as follows:

$$\begin{bmatrix} x_{R_\ell} \\ y_{R_\ell} \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} {}^{L}x_{R_\ell} \\ {}^{L}y_{R_\ell} \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}, \qquad \ell = 1, 2, 3. \tag{2}$$

To measure distances between the Rx modules on the robot and the ultrasonic Tx modules, the robot sequentially activates one of the Tx modules by a command message from a radio-frequency (RF) module on the robot at a sampling instant, as shown in Fig. 2. A Tx module is composed of a microcontroller (MCU), an ultrasonic Tx, an RF transceiver, and a temperature sensor. After receiving the aforementioned activation message, the Tx module responds with a radio message including its identification (Id) number, position, and temperature information and simultaneously transmits a modulated ultrasonic signal. Using this radio message, the Rx module can measure the time difference between the arrival of the RF signal and the ultrasonic signal.

To improve accuracy and reduce complexity, a signal emitting method with code modulation in a Tx module and a matched filter with reduced calculation in an Rx module are used to measure time of flight (TOF). Both modules are implemented using MCUs for size, cost, and power efficiency. Although Barker codes [37] have some constraints that have been previously enhanced by other codes and sequences [38], [39], these codes enable efficient two-level encoding using biphase modulation, with less hardware requirements to frequency modulation [40]. Hence, an MCU in a Tx module can easily generate signals containing low and high amplitudes as well as modulate these codes onto a square wave of ultrasonic firing signals. A 7-bit length of Barker code is used, and each bit of the code consists of three cycles of the ultrasonic main frequency. The main frequency and the bandwidth of the emitted ultrasonic signal are 40 and approximately 10 kHz, respectively. Due to the restrictions of the MCU such as memory size and clock speed, the type and the bit length of code and a number of cycles in a bit are carefully chosen to achieve real-time performance in this system.

For fast analog-to-digital conversion, a ring buffer with four sectors and a direct memory access function are used to sample the ultrasonic signal received, and two sectors are used to calculate TOF in the matched filtering. The matched filter algorithm for accurate ultrasonic signal reception is implemented using the nesC language [41] and ported to an ultrasonic board with MCU boards. The three Rx modules send all TOF measurements to the robot via a controller area network link. The robot then converts the TOFs into distances by multiplying by the speed of ultrasonic waves, which is calculated by $20.05\sqrt{T_c + 273.16}$ (in meters per second), where T_c is the temperature in degrees Celsius.

B. UHL Algorithm

The UHL algorithm can determine the pose of a robot using only ultrasonic distance measurements from Txs to Rxs [24]. Given a set of Txs whose positions are $T_i = [x_i, y_i, z_i]^{\mathrm{T}}$, of which a subset is visible at each robot location, the problem is to determine the position and orientation of the robot from the measured ultrasonic distances $\mathbf{d}^i = [d_1^i, d_2^i, d_3^i]^{\mathrm{T}}$ between the available Txs and three Rxs ($i = 1, \ldots, N$ where N is the number of the available Txs). It is assumed that all Txs are placed on the ceiling with the same height $z_i = z_T$ and the height of the ceiling is constant. Using these assumptions, the problem of finding the pose of a robot is simplified in a 2-D environment. Because the plane of the ceiling where the Txs are placed is parallel with a plane of the floor where the robot moves, it is not required to calculate the z-value of the robot. If the z-value of the robot was required, Txs would be installed at different heights. The robot pose could then be determined in a 3-D environment with additional calculations.

The pose of the robot can be estimated by solving the following:

$$\begin{aligned} d_1^i &= \sqrt{(x_i - x_{R_1})^2 + (y_i - y_{R_1})^2 + z_T^2} \\ d_2^i &= \sqrt{(x_i - x_{R_2})^2 + (y_i - y_{R_2})^2 + z_T^2} \\ d_3^i &= \sqrt{(x_i - x_{R_3})^2 + (y_i - y_{R_3})^2 + z_T^2}, \qquad i = 1, \ldots, N. \end{aligned} \tag{3}$$

A minimum of two available Txs is required to solve (3) for pose determination. When more than two Txs are used, the problem is overdetermined, and a least squares method can be effectively used to find the solution of (3) while minimizing the effect of measurement errors.

In a previous work, the UHL algorithm was developed to obtain the best possible accuracy for the position and the orientation in the entire area of interest [24]. This algorithm is a hybrid type to select the best self-localization algorithm between the direct method (DM) and the indirect method using ranges (IMR), as shown in Table I. Line 3 of Table I refers to whether the distance between a position p_{DM}, obtained by the DM, and a line l_T, determined by the available Txs, is less than a prespecified threshold dth. The threshold value is experimentally determined by the distances between Txs and z_T.

The remainder of this section will describe the DM and the IMR in the UHL algorithm. The DM uses the measured ultrasonic distances directly, without additional processes such

TABLE I
UHL ALGORITHM

```
Estimate the robot orientation.
Estimate the position, $p_{\mathrm{DM}}$, by the **Direct Method (DM)**.
**if** available Txs are collinear and $dist(l_T, p_{\mathrm{DM}})$ is under threshold $d_{th}$ **then**
    Robot position $\leftarrow p_{\mathrm{DM}}$.
**else**
    Estimate the position, $p_{\mathrm{IMR}}$, by the **Indirect Method using Ranges (IMR)**.
    Robot position $\leftarrow p_{\mathrm{IMR}}$.
**end if**
```

as feature extraction, which is used for multirobots [29]. After performing some calculations of (3), the orientation vector $\Phi = [\cos\phi, \sin\phi]^{\mathrm{T}}$ can be obtained as follows:

$$\mathbf{A}\Phi = \mathbf{b}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{A}^{1,2} \\ \mathbf{A}^{1,3} \\ \vdots \\ \mathbf{A}^{(N-1),N} \end{bmatrix}; \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}^{1,2} \\ \mathbf{b}^{1,3} \\ \vdots \\ \mathbf{b}^{(N-1),N} \end{bmatrix} \tag{4}$$

where

$$\mathbf{A}^{i,j} = \begin{bmatrix} \sqrt{3}(x_j - x_i) & \sqrt{3}(y_j - y_i) \\ -(y_j - y_i) & (x_j - x_i) \end{bmatrix}$$

$$\mathbf{b}^{i,j} = \frac{1}{2\sqrt{3}r_R} \times \begin{bmatrix} 2\left(d_1^i\right)^2 - \left(d_2^i\right)^2 - \left(d_3^i\right)^2 - 2\left(d_1^j\right)^2 + \left(d_2^j\right)^2 + \left(d_3^j\right)^2 \\ -\left(d_2^i\right)^2 + \left(d_3^i\right)^2 + \left(d_2^j\right)^2 - \left(d_3^j\right)^2 \end{bmatrix}.$$

The orientation vector can be estimated using a pseudoinverse [28]. To achieve better performance in orientation estimation, a Lagrangian equation from (4) can be constructed, satisfying the trigonometric identity $\Phi^{\mathrm{T}}\Phi = 1$, which is given by

$$\mathcal{L}(\Phi, \lambda) = (\mathbf{A}\Phi - \mathbf{b})^{\mathrm{T}}(\mathbf{A}\Phi - \mathbf{b}) + \lambda(\Phi^{\mathrm{T}}\Phi - 1) \tag{5}$$

where λ is a Lagrange multiplier. The minimum of (5) can be obtained by differentiating with respect to Φ and then setting it equal to zero

$$\frac{\partial \mathcal{L}(\Phi, \lambda)}{\partial \Phi} = (\mathbf{A}^{\mathrm{T}}\mathbf{A} + \lambda\mathbf{I})\Phi - \mathbf{A}^{\mathrm{T}}\mathbf{b} = 0. \tag{6}$$

After substituting $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ and $\mathbf{A}^{\mathrm{T}}\mathbf{b}$, λ in (6) can be eliminated, which yields the following equation of $\sin\phi$ and $\cos\phi$:

$$a_3(\cos^2\phi - \sin^2\phi) + (a_2 - a_1)\cos\phi\sin\phi + b_2\cos\phi - b_1\sin\phi = 0 \tag{7}$$

where

$$\mathbf{A}^{\mathrm{T}}\mathbf{A} = \begin{bmatrix} a_1 & a_3 \\ a_3 & a_2 \end{bmatrix} \quad \mathbf{A}^{\mathrm{T}}\mathbf{b} = -\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

Expanding (7) by the identity constraint, a quartic polynomial of $\sin\phi$ can be obtained as follows:

$$\kappa_4 \sin^4\phi + \kappa_3 \sin^3\phi + \kappa_2 \sin^2\phi + \kappa_1 \sin\phi + \kappa_0 = 0 \tag{8}$$

where $\kappa_4 = (a_1 - a_2)^2 + 4a_3^2$, $\kappa_3 = -2(a_1 - a_2)b_2 + 4a_3b_1$, $\kappa_2 = b_1^2 + b_2^2 - \kappa_4$, $\kappa_1 = 2(a_1 - a_2)b_2 - 2a_3b_1$, and $\kappa_0 = a_3^2 - b_2^2$.

Multiple solutions of ϕ must be resolved. Up to four real roots (or estimates) of $\sin\phi$ can be obtained from the quartic polynomial of (8). The corresponding two values of $\cos\phi$ can then be calculated by using $\cos\phi = \pm\sqrt{1 - \sin^2\phi}$. Hence, a maximum of eight estimates of ϕ can be obtained using the following:

$$\phi = \operatorname{atan2}(\sin\phi, \cos\phi). \tag{9}$$

Among multiple orientation estimates, the estimate that is closest to the pseudoinverse solution of (4) in the sense of Euclidean distance on a unit circle, in which the x-axis is $\cos\phi$ and the y-axis is $\sin\phi$, is taken.

After putting the orientation estimation result into (3), the position estimate of the DM can be linearized into the following matrix form by the spherical interpolation technique [42]:

$$\mathbf{G}_{\mathrm{DM}} \begin{bmatrix} x \\ y \\ x^2 + y^2 \end{bmatrix} = \mathbf{h}_{\mathrm{DM}} \tag{10}$$

where the matrix $\mathbf{G}_{\mathrm{DM}}$ is a $3N \times 3$ matrix and the vector $\mathbf{h}_{\mathrm{DM}}$ is a $3N$-dimensional vector

$$\mathbf{G}_{\mathrm{DM}} = \begin{bmatrix} x_1 - g_1^x & y_1 - g_1^y & -0.5 \\ x_1 - g_2^x & y_1 - g_2^y & -0.5 \\ x_1 - g_3^x & y_1 - g_3^y & -0.5 \\ \vdots & \vdots & \vdots \\ x_N - g_1^x & y_N - g_1^y & -0.5 \\ x_N - g_2^x & y_N - g_2^y & -0.5 \\ x_N - g_3^x & y_N - g_3^y & -0.5 \end{bmatrix}$$

$$\mathbf{h}_{\mathrm{DM}} = \frac{1}{2} \begin{bmatrix} -\left(d_1^1\right)^2 + (x_1 - g_1^x)^2 + (y_1 - g_1^y)^2 + z_T^2 \\ -\left(d_2^1\right)^2 + (x_1 - g_2^x)^2 + (y_1 - g_2^y)^2 + z_T^2 \\ -\left(d_3^1\right)^2 + (x_1 - g_3^x)^2 + (y_1 - g_3^y)^2 + z_T^2 \\ \vdots \\ -\left(d_1^N\right)^2 + (x_N - g_1^x)^2 + (y_N - g_1^y)^2 + z_T^2 \\ -\left(d_2^N\right)^2 + (x_N - g_2^x)^2 + (y_N - g_2^y)^2 + z_T^2 \\ -\left(d_3^N\right)^2 + (x_N - g_3^x)^2 + (y_N - g_3^y)^2 + z_T^2 \end{bmatrix}$$

where $g_\ell^x = {}^L x_{R_\ell}\cos\phi - {}^L y_{R_\ell}\sin\phi$ and $g_\ell^y = {}^L x_{R_\ell}\sin\phi + {}^L y_{R_\ell}\cos\phi$, with $\ell = 1,\ 2,\ 3$.

The IMR is based on the principle that a *range* between the center of the robot and a Tx constrains the robot position such that it must be on a sphere. Since all Txs are installed at the same known height, the robot position can be simplified to the intersection of circles rather than spheres. A circle whose center is the position of the Tx and whose radius is equal to the range is given by

$$\rho^i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + z_T^2}, \quad i = 1, \ldots, N \tag{11}$$

where $\rho^i = \{((d_1^i)^2 + (d_2^i)^2 + (d_3^i)^2 - 3r_R^2)/3\}^{1/2}$ is calculated from three measured distances using the properties of the isotropic array [43]. This principle enables the use of the well-known multilateration method (except for singular cases).

On the contrary, when the positions of the Txs are placed collinearly, the singularities of the multilateration method arise because two solutions may exist and it is not possible to find the unique solution. In this case, by solving the quadratic equation for two Txs T_i and T_j $(i \neq j)$, a set $\mathcal{C}$ of 0, 1, or 2 intersections of two circles around the positions of two Txs with radii ρ_i and ρ_j can be obtained as follows:

$$\mathcal{C} = \emptyset \quad \text{or} \quad \mathcal{C} = \left\{C_1^{i,j}\right\} \quad \text{or} \quad \mathcal{C} = \left\{C_1^{i,j}, C_2^{i,j}\right\}. \tag{12}$$

Using other Txs and ranges, up to $N(N-1)$ solutions can be computed for $C_k^{i,j}$, where $i = 1, \ldots, N$, $j = 1, \ldots, N$, and $k = 1,\ 2$. All possible solutions can be divided into two sets, denoted by $\mathcal{I}_1$ and $\mathcal{I}_2$, by the sign of $\Delta_k^{i,j}$ as follows:

$$\begin{cases} \mathcal{I}_1 = \cup_{i=1}^{N} \cup_{j=1}^{N} C_k^{i,j}, & \text{if } \Delta_k^{i,j} \geqslant 0 \\ \mathcal{I}_2 = \cup_{i=1}^{N} \cup_{j=1}^{N} C_k^{i,j}, & \text{if } \Delta_k^{i,j} \leqslant 0 \end{cases} \tag{13}$$

where $\Delta_k^{i,j} = \xi_a x_{C_k}^{i,j} + \xi_b y_{C_k}^{i,j} + \xi_c$ and $|\Delta_k^{i,j}|$ denotes the distance between $C_k^{i,j} = [x_{C_k}^{i,j}, y_{C_k}^{i,j}]^{\mathrm{T}}$ and a line l_T through the positions of all collinear Txs. ξ_a, ξ_b, and ξ_c are the normalized coefficients of the line l_T, where $\xi_a^2 + \xi_b^2 = 1$. For two sets of intersection points, two subestimates of the robot position can be calculated by averaging elements in $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively. To determine the correct solution between two position subestimates, two subestimates are compared with a position estimate of the DM. Only one position estimate, which is the closest Euclidean distance to that of the DM, will remain.

Experiments for the UHL algorithm were conducted in two layouts of Txs, such as a collinear arrangement with three Txs and a nearly equilateral triangle using three Txs. First, in a collinear arrangement, experiments for four different positions were executed over 12 orientations (from 0° to 330° at intervals of 30°). The localization results were calculated over 500 runs for each orientation. For the second nearly equilateral triangle, the robot was placed at two positions: One was inside the triangle, and the other was outside of the triangle. Likewise, we conducted 500 localization procedures over 12 different orientations. We also incrementally moved the robot across and down the x–y plane at steps of 0.3 m (35 locations) in the first layout.

The position and orientation errors are calculated by the root mean square (RMS) of position and orientation differences between an actual robot pose and a pose calculated by the UHL algorithm, respectively. Through the aforementioned experiments, the accuracy of the UHL algorithm was found to be capable of having a position mean error of 0.02 m (0.018 and 0.009 m in x- and y-axes, respectively) and an orientation error of 1°. Since the positions of all Txs and the actual robot pose were given, the actual distances were calculated between the Txs used and the three Rxs on the robot, and then, measurement errors were also calculated. Hence, the variances of the errors in the measured ultrasonic distances were calculated as 0.01^2 m. The time to compute a pose using the UHL algorithm was 0.83 μs on a 2.66-GHz laptop computer (excluding the time required to obtain ultrasonic distance measurements for all available Txs).

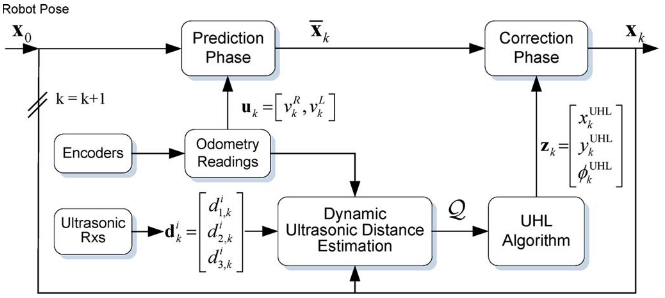
Robot Pose
$\mathbf{x}_0$
Prediction Phase
$\bar{\mathbf{x}}_k$
Correction Phase
$\mathbf{x}_k$
k = k+1
$\mathbf{u}_k = [v_k^R, v_k^L]$
Encoders
Odometry Readings
$\mathbf{z}_k = [x_k^{\mathrm{UHL}}, y_k^{\mathrm{UHL}}, \phi_k^{\mathrm{UHL}}]$
Ultrasonic Rxs
$\mathbf{d}_k^i = [d_{1,k}^i, d_{2,k}^i, d_{3,k}^i]$
Dynamic Ultrasonic Distance Estimation
$\mathcal{Q}$
UHL Algorithm

Fig. 3. Overview of the DUHL algorithm. The state $\mathbf{x}_k$, the input $\mathbf{u}_k$, and the observation $\mathbf{z}_k$ in the framework of the EKF. $\mathcal{Q} = \{\mathbf{q}_k^i | T_i \in \mathcal{T}_A\}$ is referred to as a set of estimates of the ultrasonic information between a set of available Txs $(\mathcal{T}_A)$ and Rxs, which is estimated by the dynamic distance estimation method (in Section IV), where $\mathbf{q}_k^i$ is the estimate of distance information for T_i at time k.

III. EKF WITH THE UHL ALGORITHM

An EKF was added to the UHL algorithm, given measurements of odometry and ultrasonic distances from stationary Txs to Rxs. Odometry is used in the prediction phase of the EKF, while the resultant pose obtained by the UHL algorithm is incorporated in the correction phase of the EKF. Fig. 3 shows an overview of the DUHL algorithm using the EKF and the UHL algorithm. The function $\mathbf{f}(\cdot)$ that models the transition of the state $\mathbf{x}_{k-1}$ in the presence of a control input vector $\mathbf{u}_k$ and a noise $\mathbf{w}_k$ can be described by

$$\begin{aligned} \bar{\mathbf{x}}_k &= \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) \\ &= \mathbf{x}_{k-1} + \begin{bmatrix} \Delta D_k \cos\left(\phi_{k-1} + \frac{\Delta\phi_k}{2}\right) \\ \Delta D_k \sin\left(\phi_{k-1} + \frac{\Delta\phi_k}{2}\right) \\ \Delta\phi_k \end{bmatrix} \end{aligned} \tag{14}$$

where $\mathbf{u}_k = [v_k^R, v_k^L]^{\mathrm{T}}$ is composed of the velocities for the right and left wheels from time t_{k-1} to t_k, $\Delta D_k = \Delta T_s(v_k^R + v_k^L)/2$ is the distance traveled from time t_{k-1} to t_k, and $\Delta\phi_k = \Delta T_s(v_k^R - v_k^L)/B_w$ is the orientation change traveled from time t_{k-1} to t_k, where B_w is the distance between the two wheels and ΔT_s denotes the sampling interval of odometry (which is much smaller than the period of the ultrasonic transmission and reception). The process noise $\mathbf{w}_k$ denotes the error in the odometric measurements, and it can usually be considered as an uncorrelated white Gaussian process with zero mean and a covariance matrix $\mathbf{U}_k = \mathrm{diag}\{\sigma_{v_k^R}^2, \sigma_{v_k^L}^2\}$, i.e., $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{U}_k)$, which must be experimentally determined.

The function $\mathbf{h}(\cdot)$ that incorporates the observation vector $\mathbf{z}_k$ with the state vector $\mathbf{x}_k$ is given by

$$\begin{aligned} \mathbf{z}_k &= \mathbf{h}(\mathbf{x}_k, \delta\mathbf{x}^{\mathrm{UHL}}) \\ &= \left[x_k^{\mathrm{UHL}}, y_k^{\mathrm{UHL}}, \phi_k^{\mathrm{UHL}}\right]^{\mathrm{T}} + \delta\mathbf{x}^{\mathrm{UHL}} \end{aligned} \tag{15}$$

where $[x_k^{\mathrm{UHL}}, y_k^{\mathrm{UHL}}, \phi_k^{\mathrm{UHL}}]^{\mathrm{T}}$ is the resultant pose of the UHL algorithm described in the previous section and $\delta\mathbf{x}^{\mathrm{UHL}}$ denotes the error associated to the observation vector $\mathbf{z}_k$. This error can be computed by error propagation using the uncertainties in the estimates of distances between the Txs that are used and Rxs. The objective of (15) is to match the robot pose computed by the UHL algorithm to the pose that is computed by the state vector in the EKF. The identity of this equation improves the convergence of the filter compared to other EKF approaches with a nonlinear observation function [25]–[27], [30]. This is because the observation function has no approximation errors due to linear relations in the correction phase.

A. Prediction Phase

The prediction phase, or the time update, of the EKF is recursively applied at each time step with an odometry measurement, even when there are no observations. In this phase, the transition function of (14) is used to predict the state at time t_k with the knowledge of $\mathbf{u}_k$ from the odometry measurement. The state error covariance matrix $\bar{\mathbf{P}}_k$ associated with this prediction is calculated by

$$\bar{\mathbf{P}}_k = \mathbf{J}_{\mathbf{f},\mathbf{x}}\mathbf{P}_{k-1}\mathbf{J}_{\mathbf{f},\mathbf{x}}^{\mathrm{T}} + \mathbf{J}_{\mathbf{f},\mathbf{u}}\mathbf{U}_k\mathbf{J}_{\mathbf{f},\mathbf{u}}^{\mathrm{T}} \tag{16}$$

where $\mathbf{P}_{k-1}$ is the state error covariance matrix at time t_{k-1} and $\mathbf{U}_k$ is the covariance matrix associated with the error vector $\mathbf{w}_k$. $\mathbf{J}_{\mathbf{f},\mathbf{x}}$ and $\mathbf{J}_{\mathbf{f},\mathbf{u}}$ are the Jacobians of (14) with respect to $\mathbf{x}_k$ and $\mathbf{u}_k$, respectively, given by

$$\begin{aligned}\mathbf{J}_{\mathbf{f},\mathbf{x}} &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\\ &= \begin{bmatrix} 1 & 0 & -\Delta D_k \sin\left(\phi_{k-1} + \frac{\Delta\phi_k}{2}\right) \\ 0 & 1 & \Delta D_k \cos\left(\phi_{k-1} + \frac{\Delta\phi_k}{2}\right) \\ 0 & 0 & 1 \end{bmatrix}\\ \mathbf{J}_{\mathbf{f},\mathbf{u}} &= \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\\ &= \begin{bmatrix} \cos\left(\phi_{k-1} + \frac{\Delta\phi_k}{2}\right) & -\frac{\Delta D_k}{2}\sin\left(\phi_{k-1} + \frac{\Delta\phi_k}{2}\right) \\ \sin\left(\phi_{k-1} + \frac{\Delta\phi_k}{2}\right) & \frac{\Delta D_k}{2}\cos\left(\phi_{k-1} + \frac{\Delta\phi_k}{2}\right) \\ 0 & 1 \end{bmatrix}\\ &\quad \times \begin{bmatrix} \frac{\Delta T_s}{2} & \frac{\Delta T_s}{2} \\ \frac{\Delta T_s}{B_w} & -\frac{\Delta T_s}{B_w} \end{bmatrix}.\end{aligned}$$

B. Correction Phase

The correction phase of the EKF, also known as the measurement update, corrects the prediction using the noisy observations. This correction is based on the deviation of the actual observation from the predicted observation, called *innovation*, and the covariance matrix of the innovation vector is given by

$$\mathbf{S}_k = \mathbf{J}_{\mathbf{h},\mathbf{x}}\bar{\mathbf{P}}_k\mathbf{J}_{\mathbf{h},\mathbf{x}}^{\mathrm{T}} + \mathbf{R}_k \tag{17}$$

where $\mathbf{R}_k$ is the covariance matrix in the actual observation vector and $\mathbf{J}_{\mathbf{h},\mathbf{x}} = \partial\mathbf{h}/\partial\mathbf{x}$ is the Jacobian of (15) evaluated at $\bar{\mathbf{x}}_k$, which is revealed to be the identity matrix $\mathbf{I}_{3\times 3}$. This innovation then is used to evaluate the pose state-vector estimation, and the covariance matrix of the estimation error is given by

$$\begin{aligned}\mathbf{x}_k &= \bar{\mathbf{x}}_k + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{h}(\bar{\mathbf{x}}_k, 0)\right)\\ &= \bar{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{z}_k - \bar{\mathbf{x}}_k) \qquad (18)\\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k\mathbf{J}_{\mathbf{h},\mathbf{x}})\bar{\mathbf{P}}_k \qquad (19)\end{aligned}$$

where the Kalman gain $\mathbf{K}_k$ is calculated as

$$\mathbf{K}_k = \bar{\mathbf{P}}_k\mathbf{J}_{\mathbf{h},\mathbf{x}}^{\mathrm{T}}\mathbf{S}_k^{-1}. \tag{20}$$

This phase is executed only when a new observation is available. Using the dynamic distance estimation method, which will be described in Section IV, the estimates of ultrasonic distance information for two or more Txs are always available. Thus, the UHL algorithm, which is described in Section II-B, can globally and accurately construct the new pose observation using the estimates of ultrasonic distance information for the available Txs at every sample step. In this paper, this observation will be generated at 5 Hz, which is equal to the sampling time of sequential activation for the Txs. Thus, the correction phase can be executed once for multiple prediction phases.

For the observation uncertainty, it is assumed that the errors in $\mathbf{q}_k^i$ are zero-mean Gaussian with the covariance matrix $\mathbf{Q}_k^i$. The errors in the estimates of ultrasonic information are propagated to observation $\mathbf{z}$ by (3). These systems are nonlinear; however, it is assumed that they can be adequately approximated by the first two terms of Taylor series expansion around the estimated values of $\mathbf{q}_k^i$. This assumption leads to the following expression for the covariance matrix of error in robot observation:

$$\mathbf{R}_k = \sum_{\forall T_i \in \mathcal{T}_A} \mathbf{J}_{\mathbf{z},\mathbf{q}^i}\mathbf{Q}_k^i\mathbf{J}_{\mathbf{z},\mathbf{q}^i}^{\mathrm{T}} \tag{21}$$

where $\mathbf{J}_{\mathbf{z},\mathbf{q}^i}$ is the Jacobian of $\mathbf{z}_k$ with respect to $\mathbf{q}_k^i$ and $\mathcal{T}_A$ is a set of available Txs.

IV. Dynamic Distance Estimation Method

This section presents a dynamic distance estimation method, which enables the UHL algorithm (Section II) to accurately generate observations of a robot pose for the EKF process and hence completes the DUHL algorithm (Section III). When the robot moves continuously, ultrasonic distance measurements from different Txs are obtained at different poses of the robot. If these ultrasonic distance data are directly used in the UHL algorithm, the robot pose output inevitably contains errors. When the robot is moving slowly, the pose errors by the UHL algorithm could be small and within an acceptable range. However, when the robot moves quickly, the pose errors by the UHL algorithm would be too large and therefore unacceptable. Our DUHL algorithm has small errors even at high robot speeds, in contrast to the UHL algorithm [24] which has less error at low robot speeds.

The key concept of the dynamic distance estimation method is estimating new ultrasonic data from a Tx to Rxs at the current robot pose using the ultrasonic distance measurements from the Tx to Rxs at the previous robot pose. Odometry is used to compensate the robot movements, and ultrasonic data from other Txs can be estimated in the same way. Hence, ultrasonic distance data from Txs of interest to Rxs can be obtained at the same robot pose. For each Tx, the most recent transmission data are used, unless it has a large uncertainty in this estimate.

A. Robot Motion and Distance Prediction

The distances between actual ultrasonic measurements can be calculated by the time integration of their time derivatives according to the kinematics of the robot. This gradual integration can be obtained from the values of $\dot{\mathbf{d}}_\ell^i(t) = [\dot{d}_1^i(t), \dot{d}_2^i(t), \dot{d}_3^i(t)]^{\mathrm{T}}$, which are simply obtained by the time derivative of (3), as follows:

$$\begin{bmatrix} \frac{\partial d_1^i}{\partial t} \\ \frac{\partial d_2^i}{\partial t} \\ \frac{\partial d_3^i}{\partial t} \end{bmatrix} = \begin{bmatrix} \frac{\partial d_1^i}{\partial x} & \frac{\partial d_1^i}{\partial y} & \frac{\partial d_1^i}{\partial \phi} \\ \frac{\partial d_2^i}{\partial x} & \frac{\partial d_2^i}{\partial y} & \frac{\partial d_2^i}{\partial \phi} \\ \frac{\partial d_3^i}{\partial x} & \frac{\partial d_3^i}{\partial y} & \frac{\partial d_3^i}{\partial \phi} \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial t} \\ \frac{\partial y}{\partial t} \\ \frac{\partial \phi}{\partial t} \end{bmatrix} \tag{22}$$

where

$$\begin{aligned} \frac{\partial d_\ell^i}{\partial x} &= \frac{1}{d_\ell^i}\left(x - x_i + {}^L x_{R_\ell}\cos\phi - {}^L y_{R_\ell}\sin\phi\right) \\ \frac{\partial d_\ell^i}{\partial y} &= \frac{1}{d_\ell^i}\left(y - y_i + {}^L x_{R_\ell}\sin\phi + {}^L y_{R_\ell}\cos\phi\right) \\ \frac{\partial d_\ell^i}{\partial \phi} &= \frac{1}{d_\ell^i}\left\{-(x - x_i)\left({}^L x_{R_\ell}\sin\phi + {}^L y_{R_\ell}\cos\phi\right)\right. \\ &\quad \left. + (y - y_i)\left({}^L x_{R_\ell}\cos\phi - {}^L y_{R_\ell}\sin\phi\right)\right\}, \quad \ell = 1, 2, 3. \end{aligned}$$

Let $\mathbf{q}_k^i = [q_{1,k}^i, q_{2,k}^i, q_{3,k}^i]^{\mathrm{T}}$ denote the discrete-time expression of $\mathbf{d}^i(t)$. From the discrete expression in (22) and the incremental traveled changes in (14), the discrete-time evolution of the ultrasonic information for T_i can be calculated by internal odometric measurements of the robot, and geometric variables to the corresponding Tx, as shown in Fig. 4, as follows:

$$\begin{aligned} \bar{\mathbf{q}}_k^i &= \mathbf{g}^i\left(\mathbf{q}_{k-1}^i, \mathbf{u}_k, \mathbf{w}_k\right) \\ &= \mathbf{q}_{k-1}^i + \begin{bmatrix} \frac{\left(\alpha_{1,k}^i - \beta_{1,k}^i\right)}{q_{1,k-1}^i} \\ \frac{\left(\alpha_{2,k}^i - \beta_{2,k}^i\right)}{q_{2,k-1}^i} \\ \frac{\left(\alpha_{3,k}^i - \beta_{3,k}^i\right)}{q_{3,k-1}^i} \end{bmatrix} \end{aligned} \tag{23}$$

where

$$\begin{aligned} \alpha_{\ell,k}^i &= \Delta D_k\left({}^L x_{R_\ell}\cos\frac{\Delta\phi_k}{2} + {}^L y_{R_\ell}\sin\frac{\Delta\phi_k}{2}\right) \\ \beta_{\ell,k}^i &= \sqrt{\left(\rho_{k-1}^i\right)^2 - z_T^2}\left\{\Delta D_k\cos\left(\theta_{k-1}^i - \frac{\Delta\phi_k}{2}\right)\right. \\ &\quad \left. + \Delta\phi_k\left({}^L x_{R_\ell}\sin\theta_{k-1}^i - {}^L y_{R_\ell}\cos\theta_{k-1}^i\right)\right\}, \quad \ell = 1, 2, 3 \end{aligned}$$

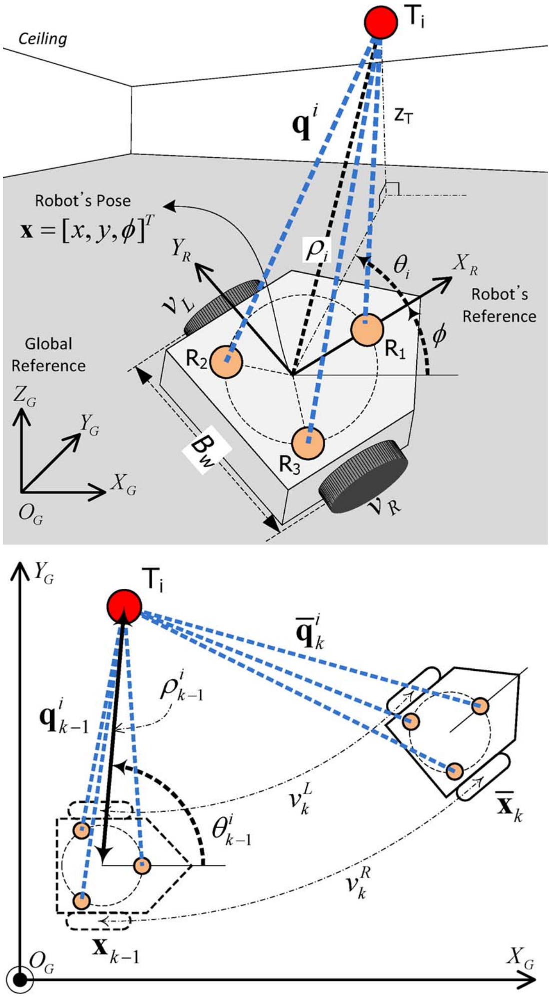

Fig. 4. Velocities of the robot (i.e., v^R and v^L) and geometric variables (i.e., ρ^i and θ^i) associated with a Tx (T_i). (Top) Perspective view. (Bottom) Top view.

and $\rho_{k-1}^i = \|[x_{k-1}, y_{k-1}, 0]^{\mathrm{T}} - T_i\|$ denotes the distance between the robot center and T_i, and $\theta_{k-1}^i = \operatorname{atan2}(y_i - y_{k-1}, x_i - x_{k-1}) - \phi_{k-1}$ represents the relative angle with respect to the longitudinal axis of the robot.

This prediction step using (23) enables the previously measured distances to be available at any time in the present or future using odometry measurements. The covariance matrix $\bar{\mathbf{Q}}_k^i$ of the error related to this prediction is calculated by

$$\bar{\mathbf{Q}}_k^i = \mathbf{J}_{\mathbf{g}^i,\mathbf{q}^i}\mathbf{Q}_{k-1}^i\mathbf{J}_{\mathbf{g}^i,\mathbf{q}^i}^{\mathrm{T}} + \mathbf{J}_{\mathbf{g}^i,\mathbf{u}}\mathbf{U}_k\mathbf{J}_{\mathbf{g}^i,\mathbf{u}}^{\mathrm{T}} \tag{24}$$

where $\mathbf{J}_{\mathbf{g}^i,\mathbf{q}^i}$ and $\mathbf{J}_{\mathbf{g}^i,\mathbf{u}}$ are the Jacobians of (23) with respect to $\mathbf{q}^i$ and $\mathbf{u}$, respectively

$$\mathbf{J}_{\mathbf{g}^i,\mathbf{q}^i} = \frac{\partial \mathbf{g}^i}{\partial \mathbf{q}^i} = \begin{bmatrix} \frac{\partial \mathbf{g}_1^i}{\partial q_1^i} & \frac{\partial \mathbf{g}_1^i}{\partial q_2^i} & \frac{\partial \mathbf{g}_1^i}{\partial q_3^i} \\ \frac{\partial \mathbf{g}_2^i}{\partial q_1^i} & \frac{\partial \mathbf{g}_2^i}{\partial q_2^i} & \frac{\partial \mathbf{g}_2^i}{\partial q_3^i} \\ \frac{\partial \mathbf{g}_3^i}{\partial q_1^i} & \frac{\partial \mathbf{g}_3^i}{\partial q_2^i} & \frac{\partial \mathbf{g}_3^i}{\partial q_3^i} \end{bmatrix}$$

$$\mathbf{J}_{\mathbf{g}^i,\mathbf{u}} = \frac{\partial \mathbf{g}^i}{\partial \mathbf{u}} = \begin{bmatrix} \frac{\partial \mathbf{g}_1^i}{\partial v^R} & \frac{\partial \mathbf{g}_1^i}{\partial v^L} \\ \frac{\partial \mathbf{g}_2^i}{\partial v^R} & \frac{\partial \mathbf{g}_2^i}{\partial v^L} \\ \frac{\partial \mathbf{g}_3^i}{\partial v^R} & \frac{\partial \mathbf{g}_3^i}{\partial v^L} \end{bmatrix}$$

where

$$\frac{\partial \mathbf{g}_\ell^i}{\partial q_m^i} = 1 - \frac{\left(\alpha_{\ell,k}^i - \beta_{\ell,k}^i\right)}{\left(q_{\ell,k-1}^i\right)^2} - \frac{1}{q_{\ell,k-1}^i}\frac{\partial \beta_\ell^i}{\partial q_\ell^i}, \quad \text{if } \ell = m$$

$$\frac{\partial \mathbf{g}_\ell^i}{\partial q_m^i} = -\frac{1}{q_{\ell,k-1}^i}\frac{\partial \beta_\ell^i}{\partial q_m^i}, \quad \text{if } \ell \neq m$$

$$\frac{\partial \beta_\ell^i}{\partial q_m^i} = \begin{bmatrix} \frac{\partial \beta_\ell^i}{\partial \rho^i} & \frac{\partial \beta_\ell^i}{\partial \theta^i} \end{bmatrix} \begin{bmatrix} \frac{\partial \rho^i}{\partial x} & \frac{\partial \rho^i}{\partial y} & \frac{\partial \rho^i}{\partial \phi} \\ \frac{\partial \theta^i}{\partial x} & \frac{\partial \theta^i}{\partial y} & \frac{\partial \theta_i}{\partial \phi} \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial q_m^i} \\ \frac{\partial y}{\partial q_m^i} \\ \frac{\partial \phi}{\partial q_m^i} \end{bmatrix}.$$

The partial derivatives in the equation earlier can be easily obtained.

B. Distance Observation and Correction

When actual distance measurements $\mathbf{d}_k^i$ are acquired from a Tx (i.e., T_i) already in the set of available Txs, the estimates of ultrasonic distance information are updated as the correction phase of the EKF as follows:

$$\mathbf{q}_k^i = \bar{\mathbf{q}}_k^i + \mathbf{K}_k^{\mathbf{q}^i}\left(\mathbf{d}_k^i - \bar{\mathbf{q}}_k^i\right) \tag{25}$$

$$\mathbf{Q}_k^i = \left(\mathbf{I}_{3\times3} - \mathbf{K}_k^{\mathbf{q}^i}\right)\bar{\mathbf{Q}}_k^i \tag{26}$$

where $\mathbf{K}_k^{\mathbf{q}^i} = \bar{\mathbf{Q}}_k^i(\bar{\mathbf{Q}}_k^i + \mathbf{R}_k^{\mathbf{q}^i})^{-1}$ is the Kalman gain for $\mathbf{q}^i$. The constant covariance matrix $\mathbf{R}_k^{\mathbf{q}^i}$ of actual distance observation is taken to be diagonal as follows:

$$\mathbf{R}_k^{\mathbf{q}^i} = \text{diag}\left\{\sigma_{d_1^i}^2, \sigma_{d_2^i}^2, \sigma_{d_3^i}^2\right\} = \sigma_{dd}^2\mathbf{I}_{3\times3} \tag{27}$$

where $\sigma_{d_\ell^i}^2$ is a variance value of actual ultrasonic distance measurement d_ℓ^i ($\ell = 1,\ 2,\ 3$), which is assumed to have a variance identical to that of σ_{dd}^2. Since the predicted and measured distances have a linear relationship, the Jacobian matrix in this correction process is the identity matrix and omitted in (26).

C. Initialization of Distance Information for a New Tx

When distances are measured from a new Tx, which is not an element within the set of available Txs, for the first time, the performance of this method can be improved by appropriately initializing the estimates of new ultrasonic information for the Tx and the uncertainty in this estimate. Using the estimate $\mathbf{x}_k$ of the robot pose in the EKF, which is described in Section III and the position T_i of the new Tx, ultrasonic distances for T_i from (2) and (3) are expected to be

$$\mathbf{q}_k^i = \begin{bmatrix} q_{1,k}^i \\ q_{2,k}^i \\ q_{3,k}^i \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - x_{R_1,k})^2 + (y_i - y_{R_1,k})^2 + z_T^2} \\ \sqrt{(x_i - x_{R_2,k})^2 + (y_i - y_{R_2,k})^2 + z_T^2} \\ \sqrt{(x_i - x_{R_3,k})^2 + (y_i - y_{R_3,k})^2 + z_T^2} \end{bmatrix}. \tag{28}$$

The uncertainty in this prediction is calculated by

$$\mathbf{Q}_k^i = \mathbf{J}_{\mathbf{q}^i,\mathbf{x}}\mathbf{P}_k\mathbf{J}_{\mathbf{q}^i,\mathbf{x}}^{\mathrm{T}} \tag{29}$$

where $\mathbf{J}_{\mathbf{q}^i,\mathbf{x}}$ is the Jacobian of (28) with respect to the robot pose $\mathbf{x}$ and $\mathbf{P}_k$ denotes the uncertainty in the robot pose. After this initialization step is performed, a set of available Txs ($\mathcal{T}_A$) and a set of estimates of the ultrasonic distance information between $\mathcal{T}_A$ and Rxs (i.e., $\mathcal{Q} = \{\mathbf{q}_k^i | T_i \in \mathcal{T}_A\}$) are updated as follows:

$$\mathcal{Q} \leftarrow \mathcal{Q} \cup \mathbf{q}_k^i, \quad \mathcal{T}_A \leftarrow \mathcal{T}_A \cup T_i. \tag{30}$$

Then, using the actual distance measurements, the correction step for $\mathbf{q}_k^i$ will follow.

D. Management of Txs

An important feature of the method is sensible management of the number of Txs in the set of available Txs ($\mathcal{T}_A$), and it might be necessary to delete distance information and the corresponding Tx. The maintenance criterion for Txs aims to constrain the number of Txs with reliable distance information to typically two or more Txs while satisfying the required localization accuracy. Txs are added to the set of available Txs, regardless of the number of the available Txs. The UHL algorithm can determine the robot pose more accurately using ultrasonic distance information from not only a number of available Txs but also recently observed Txs. Increasing the number of Txs could increase the computational complexity of the method only slightly.

As the robot moves and the distance prediction step of (23) continues, the uncertainty in the estimates of the ultrasonic information in (24) will increase. This uncertainty is very harmful to the UHL algorithm, yielding inaccurate results for the robot pose. To provide an accurate observation in the EKF process, this problem must be addressed. To detect the distance information with large uncertainty, an error function for $\mathbf{q}_k^i$ can be used. In this paper, the error function $\varepsilon(\mathbf{Q}^i)$ is defined as the maximum diagonal element of the covariance matrix of $\mathbf{q}_k^i$ and is given by

$$\varepsilon(\mathbf{Q}^i) = \max_{1\leqslant\ell\leqslant3} Q_{\ell\ell} \tag{31}$$

where $Q_{\ell\ell}$ is the (ℓ,ℓ)th entry of the matrix $\mathbf{Q}^i$, which is associated with $\mathbf{q}_k^i$. When the value of the error function $\varepsilon(\mathbf{Q}^i)$ is larger than the prespecified threshold, $\mathbf{q}_k^i$ and the corresponding Tx are discarded as follows:

$$\mathcal{Q} \leftarrow \mathcal{Q} \setminus \mathbf{q}_k^i, \quad \mathcal{T}_A \leftarrow \mathcal{T}_A \setminus T_i. \tag{32}$$

The threshold value is chosen to detect large uncertainty of distances and is determined experimentally. This discard step is executed before performing the UHL algorithm using the estimate of the ultrasonic information from the set of available Txs.

The management of Txs in the DUHL algorithm is illustrated with an example shown in Fig. 5, which shows the Id numbers of Txs at each instant of time during a moving robot experiment. Until t_5, $\mathbf{q}_k^1$ and $\mathbf{q}_k^2$ in $\mathcal{Q}$ were predicted by (23) using odometry. When $\mathbf{d}_6^1$ at t_6 and $\mathbf{d}_7^2$ at t_7 were measured, respectively, the distance correction for each estimate was executed by (25). The same correction processes were performed at t_{12} and at t_{13}, and

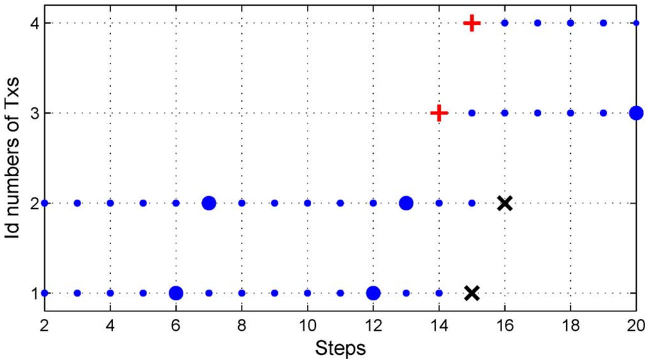

Fig. 5. Example of Id numbers of Txs at each time instant used in the UHL algorithm. For each Tx, (*dots*) the prediction, (*filled circles*) the correction only, (*red crosses*) the initialization (with the correction), and (*x-marks*) the discard processes are shown.

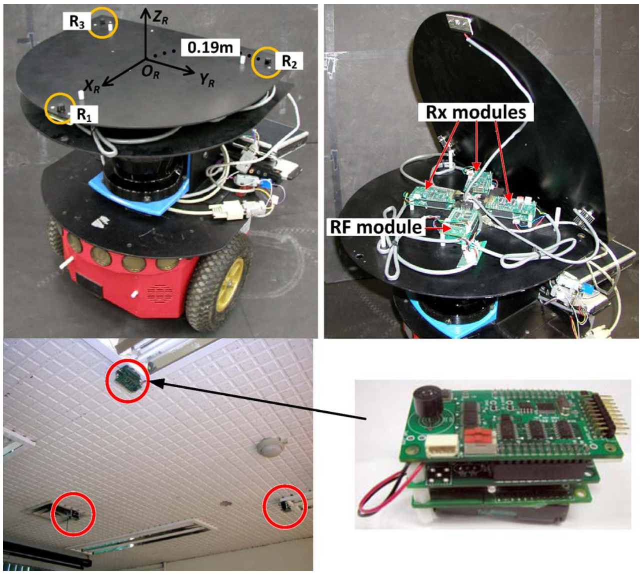

Fig. 6. Experimental systems. Pioneer P3-DX mobile robot equipped with three ultrasonic Rxs, three Rx modules, and the RF module on the robot, an example of several Tx modules attached to the ceiling, and the Tx module.

$\mathbf{q}_{12}^1$ and $\mathbf{q}_{13}^2$ were calculated. At t_{14}, $\mathbf{d}_{14}^3$ was measured, but $\mathbf{q}_{14}^3$ was not an element in $\mathcal{Q}$. Hence, after the initialization in (28) using the robot pose $\mathbf{x}_{14}$, $\mathbf{q}_{14}^3$ was included in $\mathcal{Q}$ and then, the correction step using $\mathbf{d}_{14}^3$ followed. $\mathbf{q}_{15}^4$ at t_{15} could also be constructed using the initialization process. At t_{15}, however, the value of the error function for $\mathbf{q}_{15}^1$ was larger than the threshold, and thus, $\mathbf{q}_{15}^1$ was discarded. Likewise, the discard for $\mathbf{q}_{16}^2$ was performed at t_{16}.

V. Experiments

The proposed method was examined on various experimental setups for a mobile robot, which traveled paths in an indoor environment, along with several ultrasonic Txs (MA40S4R from Murata) attached to the ceiling. A Pioneer P3-DX mobile robot equipped with three ultrasonic Rxs (MA40S4R from Murata) and the RF module was used, as shown in Fig. 6. The distance from the center of the robot to each Rx was 0.19 m. Three Rx modules, the RF module, and the Tx modules were implemented using MSP430 MCUs from Texas Instruments. A laptop computer with a 2.66-GHz CPU was used to execute the proposed method.

In the experiments, the robot moved while recording ultrasonic distances to the Txs and odometry information. The odometry was sampled at 100 Hz ($\Delta T_s = 10$ ms), and the ultrasonic distances were sampled at 5 Hz ($= 200$ ms). The ultrasonic distance measurement error of (27) was modeled as a zero-mean Gaussian with a variance $\sigma_{dd}^2 = 0.01^2$ m, which was experimentally measured. Interferences from different sources or signal reflections were not considered in the experiments. The variance of the error in the odometric measurements was modeled to be proportional to the absolute value of the velocities and is denoted by $\sigma_{v_k^{R/L}}^2 = k_v |v_k^{R/L}|/\Delta T_s$ with a value of $k_v \approx 4 \times 10^{-4}$ m, which was experimentally measured [11].

To evaluate the accuracy of the proposed method, an LRF (LMS-200 from SICK) was used to compute a highly accurate estimate of the robot pose, which will be considered as the reference robot pose, i.e., the *ground truth* (GT). The Mobile Robot Programming Toolkit version 0.8.1 [44] was used to perform an LRF-based localization, which used laser scans sampled at 38 Hz ($\simeq 26$ ms).

For tracking performances, the position error $E_p(k)$ and the orientation error $E_\phi(k)$ were used as follows:

$$E_p(k) = \left\| \left[x_k^{\mathrm{GT}}, y_k^{\mathrm{GT}}\right]^{\mathrm{T}} - [x_k, y_k]^{\mathrm{T}} \right\| \tag{33}$$

$$E_\phi(k) = \left| \phi_k^{\mathrm{GT}} - \phi_k \right| \tag{34}$$

where $\mathbf{x}_k^{\mathrm{GT}} = [x_k^{\mathrm{GT}}, y_k^{\mathrm{GT}}, \phi_k^{\mathrm{GT}}]^T$ is determined from the LRF information and $\mathbf{x}_k = [x_k, y_k, \phi_k]^{\mathrm{T}}$ represents the estimates of the proposed algorithm using the ultrasonic distance measurements and odometry. An interesting Euclidean metric was also introduced to evaluate the overall pose error (containing both the position and the orientation errors) using significant points, which are representative points of interest in the robot geometry [24]. This metric is defined as the RMS of the Euclidean distances between the ground truth ($S_{i,k}^{\mathrm{GT}}$) and the estimated ($S_{i,k}$) significant points as follows:

$$E_S(k) = \sqrt{\frac{1}{M} \sum_{k=1}^{M} \left\| S_{i,k}^{\mathrm{GT}} - S_{i,k} \right\|^2} \tag{35}$$

where M is the number of significant points and, given $\mathbf{x} = [x, y, \phi]^{\mathrm{T}}$, the position of the ith significant point S_i in the global coordinate system is given by

$$S_i = \begin{bmatrix} x + {}^L x_{S_i} \cos\phi - {}^L y_{S_i} \sin\phi \\ y + {}^L x_{S_i} \sin\phi + {}^L y_{S_i} \cos\phi \end{bmatrix}$$

where ${}^L S_i = [{}^L x_{S_i}, {}^L y_{S_i}]^{\mathrm{T}}$ is the position of the ith significant point, referred from the local coordinate system (i.e., the robot pose). In this paper, five significant points in the real robot geometry (in Fig. 6) were employed, namely, $[0.2, 0]^{\mathrm{T}}$, $[0, 0.2]^{\mathrm{T}}$, $[0, -0.2]^{\mathrm{T}}$, $[-0.24, 0.2]^{\mathrm{T}}$, and $[-0.24, -0.2]^{\mathrm{T}}$.

First, the tracking performances for three different trajectories at a nominal velocity of 0.37 m/s were examined, as shown in Fig. 7. Six Txs in trajectories 1 and 2 ($z_T = 2.011$ m) and ten Txs in trajectory 3 ($z_T = 1.887$ m) were installed on

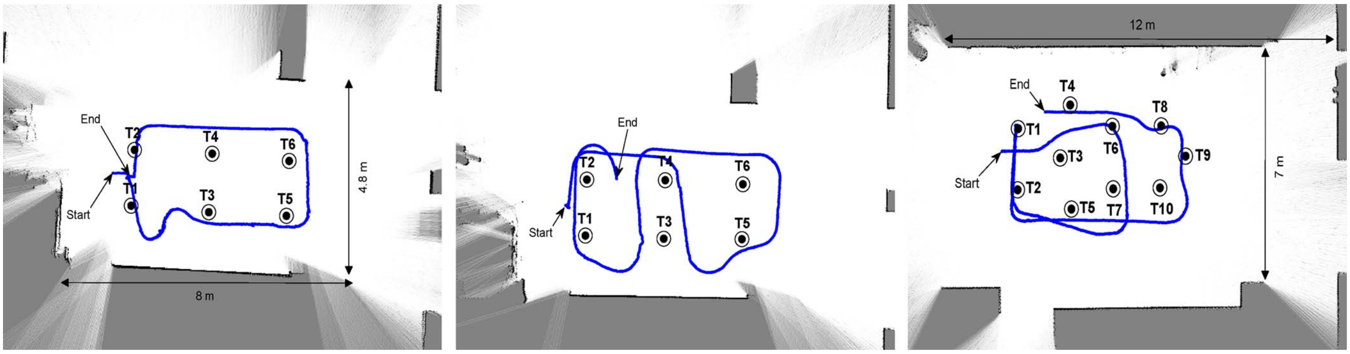

Fig. 7. Experiment I. For each trajectory, the nominal velocity of the robot was 0.34 m/s. (*Solid lines*) Reference trajectories traveled by the robot and the grid maps from the LRF information in the experiments are shown. (Left) Trajectory 1. (Middle) Trajectory 2. (Right) Trajectory 3. Six Txs in trajectories 1 and 2 and ten Txs in trajectory 3 were installed.

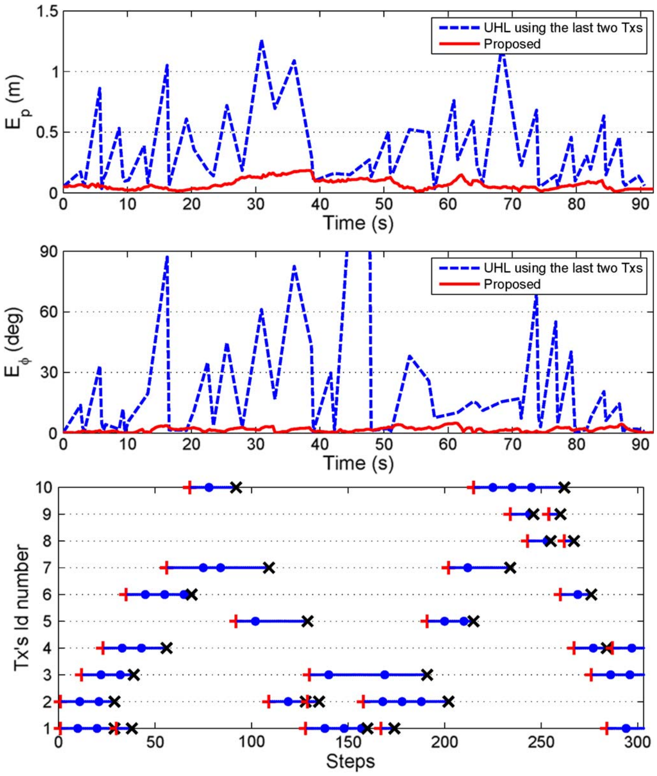

Fig. 8. Results for trajectory 3 in Experiment I. The position and orientation errors and Id numbers of Txs used in the UHL algorithm at each time instant.

the ceiling. The positions of the Txs used for each trajectory and the starting and ending positions are shown. The reference trajectories and the grid maps were also drawn from the LRF information. For each trajectory, the proposed algorithm estimated the pose of the robot over time. The UHL algorithm, which is described in Section II, was also executed only using the distances from the last two Txs. Because the time and the pose differences between distance measurements from the last two Txs are small, the aforementioned setup (using the last two Txs) was used.

The comparisons of the position and the orientation errors between the proposed method and the UHL algorithm using the last two Txs for trajectory 3 are shown in Fig. 8. The UHL algorithm using all available Txs revealed worse results than those using the last two Txs. For trajectories 1 and 2, the results were similar. Using the proposed method, the tracking results of the robot over the entire path were far accurate than the UHL algorithm using the last two Txs. When the robot was near the starting and the ending positions, the errors of the UHL algorithm were small due to the low speed of the robot. However, when the robot was moving at the nominal speed, the maximum position and orientation errors of the UHL algorithm were 2.166 m and 164° for trajectory 1, 1.789 m and 169° for trajectory 2, and 1.260 m and 158° for trajectory 3.

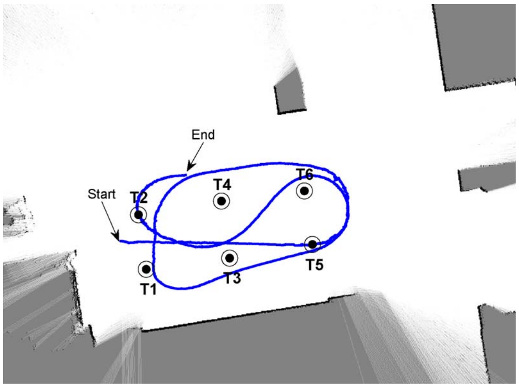

Fig. 9. Experiment II. The nominal velocity of the robot was 1.02 m/s (three times faster than Experiment I). The reference trajectory traveled by the robot and the grid map from the LRF information are also shown. Six Txs were installed as for trajectories 1 and 2.

For the second experiment, another trajectory was tested at a nominal velocity of 1.02 m/s, which is three times faster than the speed in Experiment I. Fig. 9 shows the positions of six Txs, the reference trajectories, and the grid maps from the LRF information. As shown in Fig. 10, when the robot was moving faster, the DUHL algorithm showed moderate accuracy degradation compared to the UHL algorithm, which showed severe accuracy degradation.

Table II summarizes the RMS values of the position and the orientation errors and the mean values for the overall pose errors using significant points in all four trajectories. The maximum position and orientation errors of the DUHL algorithm were only 0.257 m and 6° for trajectory 4 in Experiment II. The average execution times of each step on a 2.66-GHz laptop computer are also presented for each trajectory in Table III. The execution times of the proposed method were nearly constant for all trajectories, but trajectory 3 required more execution

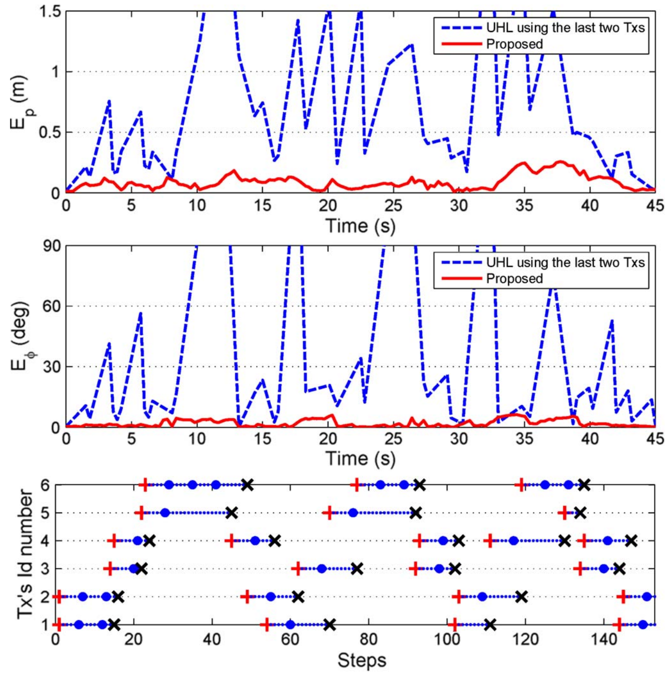

Fig. 10. Results for trajectory 4 in Experiment II. The position and orientation errors and Id numbers of Txs used in the UHL algorithm at each time instant.

TABLE II
RMS VALUES OF THE POSITION AND THE ORIENTATION ERRORS AND THE MEAN OF THE OVERALL POSE ERRORS USING SIGNIFICANT POINTS

Trajectory	Methods	E_p (m)	E_ϕ (deg.)	E_S (m)
1	**Proposed**	**0.070**	**1.61**	**0.061**
	UHL using the last 2 Txs	0.599	40.36	0.386
2	**Proposed**	**0.063**	**2.20**	**0.054**
	UHL using the last 2 Txs	0.606	43.21	0.447
3	**Proposed**	**0.081**	**1.98**	**0.069**
	UHL using the last 2 Txs	0.435	34.99	0.327
4	**Proposed**	**0.104**	**2.59**	**0.086**
	UHL using the last 2 Txs	0.878	52.29	0.637

TABLE III
AVERAGE EXECUTION TIME IN EACH STEP FOR EACH TRAJECTORY (UNIT: IN MILLISECONDS)

	Trajectory			
Step	1	2	3	4
EKF (including the UHL)	0.24	0.26	0.38	0.27
Dynamic Distance Estimation	0.31	0.37	0.59	0.36
Total	0.55	0.63	0.97	0.63

time because the number of Txs in trajectory 3 was greater than the others and greater distances to the Txs were required to perform the dynamic distance estimation method described in Section IV. However, the proposed method required less than 1 ms to calculate the entire algorithm on the laptop computer.

VI. CONCLUSION

An efficient DUHL system based on an EKF with a robot pose state/observation vector and a UHL algorithm using dynamic estimates of ultrasonic distance information has been presented. The proposed algorithm used odometric and ultrasonic distance measurements in the localization subsystem using ultrasonic sensors. The localization subsystem using ultrasonic sensors consisted of several ultrasonic Txs fixed at reference positions in global coordinates and an equilateral ultrasonic Rx array with three Rxs on the top of the robot. It was demonstrated how the self-localization algorithm (i.e., the UHL algorithm) could be used in conjunction with the EKF. To enable the persistent use of the UHL algorithm in the framework of the EKF, a dynamic distance estimation method was presented to track the estimates of ultrasonic distance information from available Txs of interest and the uncertainty in these estimates. The estimates of the aforementioned ultrasonic distance information are updated during robot motion and distance observation from the set of available Txs. These estimates and the corresponding Tx can also be discarded due to a large uncertainty, if necessary. The proposed DUHL algorithm was compared against the UHL algorithm using distances from the last two Txs for four different trajectories. Using this algorithm, the tracking results of the robot were more accurate than those of the UHL algorithm using the last two Txs.

The operation mode based on a sequential RF synchronization with successive Txs allowed the Barker code to work suitably, as described in Section II-A. Nevertheless, orthogonal codes, such as complementary sets of sequences, could be used for faster localization process, because only one RF synchronization signal would be necessary to measure distances between available Txs and the Rxs on a robot. Additionally, some of these codes present efficient correlation schemes that significantly reduce the computational complexity [45], [46].

In further work, the problem of simultaneous localization and mapping (SLAM) [47], [48] will be considered. A variant of the SLAM problem applicable to this case is to find both the pose of the robot and the positions of the Txs simultaneously when the positions of the Txs are unknown. This problem, which is related to SLAM using only distance measurements, is an interesting issue in mobile robotics [49].

REFERENCES

[1] J. Borenstein, H. R. Everett, and L. Feng, "'Where Am I?' Sensors and methods for mobile robot positioning," Univ. Michigan, Ann Arbor, MI, 1996, Tech. Rep.

[2] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Berling, Germany: Springer-Verlag, 2008.

[3] B. Barshan and H. Durrant-Whyte, "Inertial navigation systems for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 11, no. 3, pp. 328–342, Jun. 1995.

[4] S.-H. P. Won, W. Melek, and F. Golnaraghi, "A Kalman/particle filter-based position and orientation estimation method using a position sensor/inertial measurement unit hybrid system," *IEEE Trans. Ind. Electron.*, vol. 57, no. 5, pp. 1787–1798, May 2010.

[5] C. Wang, "Location estimation and uncertainty analysis for mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1988, pp. 1231–1235.

[6] A. Kelly, "Linearized error propagation in odometry," *Int. J. Robot. Res.*, vol. 23, no. 2, pp. 179–218, Feb. 2004.

[7] C. Borja, J. Tur, and J. Gordillo, "State your position: Accurate position estimation and propagation in autonomous vehicles," *IEEE Robot. Autom. Mag.*, vol. 16, no. 2, pp. 82–90, Jun. 2009.

[8] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Trans. Robot. Autom.*, vol. 12, no. 6, pp. 869–880, Dec. 1996.

[9] G. Antonelli and S. Chiaverini, “Linear estimation of the physical odometric parameters for differential-drive mobile robots,” *Auton. Robots*, vol. 23, no. 1, pp. 59–68, Jul. 2007.

[10] A. Martinelli, N. Tomatis, and R. Siegwart, “Simultaneous localization and odometry self calibration for mobile robot,” *Auton. Robots*, vol. 22, no. 1, pp. 75–85, Jan. 2007.

[11] K. Chong and L. Kleeman, “Accurate odometry and error modelling for a mobile robot,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 1997, pp. 2783–2788.

[12] F. Thomas and L. Ros, “Revisiting trilateration for robot localization,” *IEEE Trans. Robot.*, vol. 21, no. 1, pp. 93–101, Feb. 2005.

[13] H. Cho and S. W. Kim, “Mobile robot localization using biased chirp-spread-spectrum ranging,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 8, pp. 2826–2835, Aug. 2010.

[14] I. Shimshoni, “On mobile robot localization from landmark bearings,” *IEEE Trans. Robot. Autom.*, vol. 18, no. 6, pp. 971–976, Dec. 2002.

[15] K. Briechle and U. D. Hanebeck, “Localization of a mobile robot using relative bearing measurements,” *IEEE Trans. Robot. Autom.*, vol. 20, no. 1, pp. 36–44, Feb. 2004.

[16] S. Se, D. G. Lowe, and J. J. Little, “Vision-based global localization and mapping for mobile robots,” *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 364–375, Jun. 2005.

[17] R. Penne, L. Mertens, and J. Veraart, “Mobile camera localization using apollonius circles and virtual landmarks,” *J. Intell. Robot. Syst.*, vol. 58, no. 3/4, pp. 287–308, Jun. 2010.

[18] S. Thrun, W. Burgard, and D. Fox, Eds., *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2008.

[19] J. J. Leonard and H. F. Durrant-Whyte, “Mobile robot localization by tracking geometric beacons,” *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 376–382, Jun. 1991.

[20] P. Jensfelt and H. Christensen, “Pose tracking using laser scanning and minimalistic environmental models,” *IEEE Trans. Robot. Autom.*, vol. 17, no. 2, pp. 138–147, Apr. 2001.

[21] J. González, J. Blanco, C. Galindoa, A. O. de Galisteoa, J. Fernández-Madrigala, F. Morenoa, and J. Martíneza, “Mobile robot localization based on ultra-wide-band ranging: A particle filter approach,” *Robot. Auton. Syst.*, vol. 57, no. 5, pp. 496–507, May 2009.

[22] J. M. Font and J. A. Batlle, “Consistent triangulation for mobile robot localization using discontinuous angular measurements,” *Robot. Auton. Syst.*, vol. 57, no. 9, pp. 931–942, Sep. 2009.

[23] K.-W. Lee, J.-B. Park, and B.-H. Lee, “Dynamic localization with hybrid trilateration for mobile robots in intelligent space,” *Intell. Serv. Robot.*, vol. 1, no. 3, pp. 221–235, Jul. 2008.

[24] S. J. Kim and B. K. Kim, “Accurate hybrid global self-localization algorithm for indoor mobile robots with two-dimensional isotropic ultrasonic receivers,” *IEEE Trans. Instrum. Meas.*, vol. 60, no. 10, pp. 3391–3404, Oct. 2011.

[25] C.-C. Tsai, “A localization system of a mobile robot by fusing dead-reckoning and ultrasonic measurements,” *IEEE Trans. Instrum. Meas.*, vol. 47, no. 5, pp. 1399–1404, Oct. 1998.

[26] S.-Y. Yi and B.-W. Choi, “Autonomous navigation of indoor mobile robots using a global ultrasonic system,” *Robotica*, vol. 22, no. 4, pp. 369–374, Aug. 2004.

[27] C.-C. Tsai, H.-H. Lin, and S.-W. Lai, “Multisensor 3-D posture determination of a mobile robot using inertial and ultrasonic sensors,” *J. Intell. Robot. Syst.*, vol. 42, no. 4, pp. 317–335, Apr. 2005.

[28] H.-H. Lin, C.-C. Tsai, and J.-C. Hsu, “Ultrasonic localization and pose tracking of an autonomous mobile robot via fuzzy adaptive extended information filtering,” *IEEE Trans. Instrum. Meas.*, vol. 57, no. 9, pp. 2024–2034, Sep. 2008.

[29] F. Rivard, J. Bisson, F. Michaud, and D. Létourneay, “Ultrasonic relative positioning for multi-robot systems,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 323–328.

[30] S. Y. Yi, “Global ultrasonic system with selective activation for autonomous navigation of an indoor mobile robot,” *Robotica*, vol. 26, no. 3, pp. 277–283, May 2008.

[31] J. Park, M. Choi, Y. Zu, and J. Lee, “Indoor localization system in a multi-block workspace,” *Robotica*, vol. 28, no. 3, pp. 397–403, May 2010.

[32] S. J. Kim and B. K. Kim, “Dynamic localization based on EKF for indoor mobile robots using discontinuous ultrasonic distance measurements,” in *Proc. Int. Conf. Control, Autom. Syst.*, Oct. 2010, pp. 1912–1917.

[33] H. Song, V. Shin, and M. Jeon, “Mobile node localization using fusion prediction based interacting multiple model in cricket sensor network,” *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4349–4359, Nov. 2011.

[34] D. Fox, W. Burgard, and S. Thrun, “Markov localization for mobile robots in dynamic environments,” *J. Artif. Intell. Res.*, vol. 11, pp. 391–427, 1999.

[35] J. Nieto, J. Guivant, E. Nebot, and S. Thrun, “Real time data association for FastSLAM,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Sep. 2003, pp. 412–418.

[36] S.-Y. Hwang and J.-B. Song, “Monocular vision-based SLAM in indoor environment using corner, lamp, and door features from upward-looking camera,” *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4804–4812, Oct. 2011.

[37] H. Peremans, K. Audenaert, and J. V. Campenhout, “A high-resolution sensor based on tri-aural perception,” *IEEE Trans. Robot. Autom.*, vol. 9, no. 1, pp. 36–48, Feb. 1993.

[38] K.-W. Jorg and M. Berg, “First results in eliminating crosstalk and noise by applying pseudo-random sequences to mobile robot sonar sensing,” in *Proc. Int. Conf. Intell. Robots Syst.*, 1996, pp. 292–297.

[39] Á. Hernandez, J. Urena, J. Garcia, M. Mazo, D. Hernanz, J.-P. Derutin, and J. Serot, “Ultrasonic ranging sensor using simultaneous emissions from different transducers,” *IEEE Trans. Ultrason., Ferroelect., Freq. Control*, vol. 51, no. 12, pp. 1660–1670, Dec. 2004.

[40] H. Zhao, L. Mo, and S. Gao, “Barker-coded ultrasound color flow imaging: Theoretical and practical design considerations,” *IEEE Trans. Ultrason., Ferroelect., Freq. Control*, vol. 54, no. 2, pp. 319–331, Feb. 2007.

[41] D. Gay, P. Levis, D. Culler, and E. Brewer, *nesC 1.2 Language Reference Manual*: TinyOS Working Group, Aug. 2005. [Online]. Available: http://www.tinyos.net/dist-2.0.0/tinyos-2.0.0beta1/doc/nesc/ref.pdf

[42] K. W. Cheung, H. C. So, W.-K. Ma, and Y. T. Chan, “A constrained least squares approach to mobile positioning: Algorithms and optimality,” *EURASIP J. Appl. Signal Process.*, vol. 2006, pp. 1–23, 2006.

[43] Ü. Baysal and R. L. Moses, “On the geometry of isotropic arrays,” *IEEE Trans. Signal Process.*, vol. 51, no. 6, pp. 1469–1478, Jun. 2003.

[44] J. L. Blanco, *The Mobile Robot Programming Toolkit (MRPT)*: MRPT, Apr. 2009. [Online]. Available: http://www.mrpt.org/downloads/mrpt-book.pdf

[45] B. M. Popovic, “Efficient golay correlator,” *Electron. Lett.*, vol. 35, no. 17, pp. 1427–1428, Aug. 1999.

[46] C. D. Marziani, J. Ureña, A. Hernández, M. Mazo, F. Alvarez, J. Garcia, and P. Donato, “Modular architecture for efficient generation and correlation of complementary set of sequences,” *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 2323–2337, May 2007.

[47] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, “A solution to the simultaneous localization and map building (SLAM) problem,” *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 229–241, Jun. 2001.

[48] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “FastSLAM: A factored solution to the simultaneous localization and mapping problem,” in *Proc. Nat. Conf. AAAI*, Jul. 2002, pp. 593–598.

[49] J. Djugash and S. Singh, “Motion-aided network SLAM with range,” *Int. J. Robot. Res.*, vol. 31, no. 5, pp. 604–625, Apr. 2012.

Seong Jin Kim received the B.S. degree from Yonsei University, Seoul, Korea, in 2001, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 2003 and 2012, respectively.

He is currently a Senior Researcher with the System Solution Research Department, Research Institute of Industrial Science and Technology, Pohang, Korea. His current research interests include the mobile robot navigation, localization, and mapping.

Byung Kook Kim (S’75–M’81) received the B.S. degree from Seoul National University, Seoul, Korea, in 1975, and the M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1977 and 1981, respectively.

Since 1986, he has been with the Department of Electrical Engineering, KAIST, where he is currently a Professor. His research interests include real-time systems, reliable process control, mobile robot sensing and navigation, and manipulator control.

Dr. Kim is a member of The Institute of Electronics Engineers of Korea, The Korean Institute of Electrical Engineers, and The Korean Institute of Information Scientists and Engineers.