IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 22, 2019, accepted September 15, 2019, date of publication September 27, 2019,
date of current version November 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2944222

A Novel Simple Two-Robot Precise Self-Localization Method

DANA EREZ, SHAI AROGETI, (Member, IEEE), AND DAVID ZARROUK
Mechanical Engineering Department, Ben-Gurion University of the Negev, Be'er Sheva 84105, Israel
Corresponding author: Dana Erez (erezdan@post.bgu.ac.il)

This work was supported in part by the Pearlstone Center for Aeronautical Studies, in part by the Helmsley Charitable Trust through the Agricultural, Biological and Cognitive Robotics Initiative, Ben-Gurion University of the Negev, and in part by the Marcus Endowment Fund, Ben-Gurion University of the Negev.

ABSTRACT This paper presents a novel two-robot collaboration method for precise 2D self-localization using relatively simple sensors. The main advantage of this method lies in its ability to precisely measure the orientations of the robots, therefore reducing cumulative errors. Each robot is fitted with a rotating turret carrying a camera to track the moving robot and calculate the relative distance and position, and an encoder to provide the orientation of the turret. At each step, a single robot advances while the other remains stationary and measures the position of the moving robot (continuously or at the end of the step), using the angular orientation of the turret and the distance measured using the camera. The orientation of the moving robot is obtained by turning its own turret towards the static robot and measuring its turret orientation. By fusing the data from the two robots, the precise location and orientation of the moving robot are obtained. We also present an analytical model of the position of the robots as a function of the sensor data and then proceed to present a statistical estimate using Monte Carlo simulations of the location of the robots while assuming that the sensor data includes random errors. Additionally, lab experiments are presented and compared to simulation results.

INDEX TERMS Localization, multi-robot systems.

I. INTRODUCTION

Many robotic applications such as search and rescue, surveillance and others require simultaneous localization and mapping (SLAM) of unknown unstructured locations. SLAM techniques become more crucial where GPS and other localization techniques are unavailable such as indoors, inside caves or in tunnels. Many solutions for self-localization rely on measuring the relative position of the robot with respect to known features in space, also known as landmarks. However, the complexity grows in cases where there is no prior knowledge of the explored area. In 1994, Kurazume et al. first suggested cooperative positioning for multi-robot systems as a solution to the SLAM problem [1]. By advancing the robots in alternating steps, such that at each point in time some robots remain stationary and the others travel to new positions, the stationary robots whose absolute locations are known serve as landmarks for the traveling robots.

This cooperative positioning method has been further developed by other groups [2]–[6] to suggest the use of different kinds of sensors to determine relative positioning with different advancing algorithms. The advantage of this method is that a unified map of the robots' trajectories is created using all available relative measurements. However, to implement this method, a centralized communication system is required. Centralized approaches, though theoretically effective, require ideal communication and high computational cost, thus making them vulnerable to single-point failures especially as the number of robots increases.

The main challenge in using relative measurements is determining the absolute locations of the robots, since the locations are obtained with regard to a local coordinate system. Some solutions address this issue by combining both external measurements such as GPS [7] or an affixed IR range detector [8], which return inaccurate yet absolute locations and relative measurements between the robots to enhance accuracy and obtain the orientation of the robots as well. The practicality of these methods is limited since they require either GPS reception which is not available in many cases such as indoor or underground areas or prior placing of sensing tools. Similarly, many solutions use filtering techniques, most commonly the Extended Kalman Filter (EKF) [9]–[12],

The associate editor coordinating the review of this manuscript and approving it for publication was Ludovico Minati.

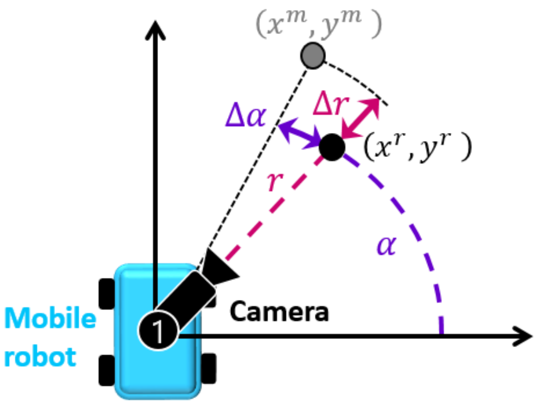

FIGURE 1. The mobile robot and its sensors. Each robot incorporates a camera fitted on a rotating turret and a bearing sensor.

where the robots' locations are predicted by odometry data (such as linear and angular velocities) and corrected by relative measurements between neighboring robots. Recently, the use of Ultra-Wideband (UWB) range-sensors has become popular for relative distance measurements in multi-robot systems, because they make it possible to perform the localization process in a fully decentralized manner [13]–[15].

While the relative locations of a multi-robot system can be calculated by using any of the aforementioned methods, obtaining the accurate relative orientation of the robots is much more challenging. Besides visual methods [16]–[18], many attempts to find the orientation of the robots have been made using range-only measurements [6], [13]–[15], [19] and angle-only measurements [11] or a combination of both [10], [20]–[22]. The accuracy of the orientation remains however very challenging at long distances.

Our goal in this work is to provide a simple low-cost high accuracy localization and orientation method for a multi-robot system, suitable for indoor areas where GPS signals are unavailable, and visibility is relatively low. We consider two robots each of which is equipped with one camera and one rotation/bearing sensor mounted on a rotating turret. The localization algorithm is described in Section II and the error evaluation using an analytical exact method and first order approximation method is presented in Section III. The two methods are used to statistically evaluate the location and orientation errors using Monte Carlo simulations in Section IV and lab experiments are described in Section V. Finally, conclusions and future work are discussed in Section VI.

II. LOCALIZATION METHOD

In this section, we present our robotic setup, its sensors and two localization methods. The first method is based on a two-point measurement approach to calculate the orientation, whereas the second method, which is our newly developed method, fuses the distance and relative orientation to yield more accurate results.

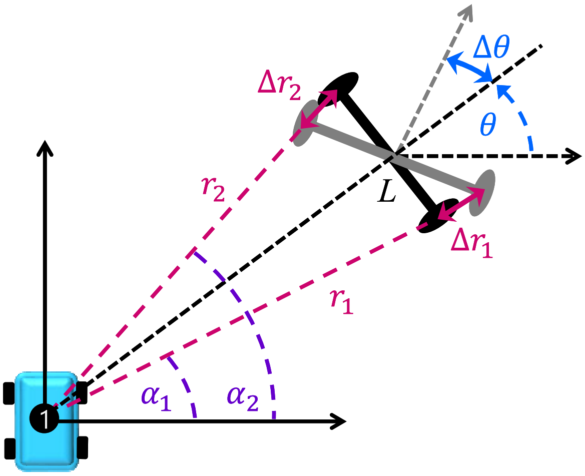

FIGURE 2. Distance measurement errors when estimating the orientation of a vehicle. In black: real position of vehicle, in grey: estimated position.

A. ROBOTIC SYSTEM

Consider a robot fitted with a rotating turret which carries a camera (see Fig. 1). The orientation of the turret relative to the robot is measured with a bearing sensor. The camera is used to detect the target and to aim the turret towards it. The distance is measured using the camera whereas the bearing sensor measures its angular coordinates. The polar coordinates can then be transformed into the real Cartesian location coordinates (x^r, y^r) using:

$$\begin{aligned} x^r &= r\cos(\alpha) \\ y^r &= r\sin(\alpha)\,, \end{aligned} \tag{1}$$

where r is the distance of the target and α is the orientation of the turret. Practically speaking, each of the sensor measurements contains a small error. We denote by Δr and $\Delta\alpha$, respectively the distance and orientation errors. Then the coordinates (x^m, y^m) based on the sensor measurement become:

$$\begin{aligned} x^m &= (r + \Delta r)\cos(\alpha + \Delta\alpha) \\ y^m &= (r + \Delta r)\sin(\alpha + \Delta\alpha)\,. \end{aligned} \tag{2}$$

The distance error range is often (according to many laser sensor catalogs and visual based sensing) proportional to the measured distance, whereas the angular error is dependent on the resolution of the camera and encoders and is constant for a long range of distances (as long as the target is detected by multiple camera pixels). Assuming small measurement errors Δr, $\Delta\alpha$ and using a first order Taylor series approximation:

$$\begin{aligned} \cos(\alpha + \Delta\alpha) &= \cos(\alpha) - \sin(\alpha)\,\Delta\alpha \\ \sin(\alpha + \Delta\alpha) &= \sin(\alpha) + \cos(\alpha)\,\Delta\alpha, \end{aligned} \tag{3}$$

neglecting the product of Δr times $\Delta\alpha$, Eq. (2) becomes:

$$\begin{aligned} x^m &\approx (r + \Delta r)\cos(\alpha) - \ r\sin(\alpha)\,\Delta\alpha \\ y^m &\approx (r + \Delta r)\sin(\alpha) + \ r\cos(\alpha)\,\Delta\alpha. \end{aligned} \tag{4}$$

B. TWO POINT MEASUREMENT APPROACH

A straightforward approach is to estimate the orientation of the robot by measuring the position of two specific points on its side. Assuming that the measured distance and relative orientation of two points 1 and 2 are respectively r_1, α_1, r_2 and α_2 (see Fig. 2), the position of the center of the robot (x, y) and its orientation θ can be calculated as follows:

$$\begin{aligned} x &= \frac{r_1 \cos\alpha_1 + r_2 \cos\alpha_2}{2} \\ y &= \frac{r_1 \sin\alpha_1 + r_2 \sin\alpha_2}{2}, \end{aligned} \tag{5}$$

and

$$\theta = -\arctan\left(\frac{r_2 \cos\alpha_2 - r_1 \cos\alpha_1}{r_2 \sin\alpha_2 - r_1 \sin\alpha_1}\right). \tag{6}$$

This method results in a relatively large error in the robot's orientation if the errors Δr_1, Δr_2 become significantly large relative to the distance L between the two measured points. Omitting the angle measurement errors, the maximal orientation error of the robot:

$$\Delta\theta \approx \frac{|\Delta r_1| + |\Delta r_2|}{L \cos(\alpha - \theta)}. \tag{7}$$

For example, assume a robot with a length of $L = 0.5$ m is measured from a distance of $r = 10$ m by a distance measurement with a resolution of 0.2%; hence $\Delta r = 2$ cm. Given $\alpha = 45\,^\circ$ and $\theta = 30^\circ$, the orientation error according to Eq. (7) is $\Delta\theta \approx 4.8\,^\circ$. Note that this orientation error for each single step is very large especially since the error is cumulative.

C. RELATIVE ORIENTATION METHOD (SUGGESTED METHOD)

Our method is based on the approach of two vehicles which advance in alternating steps. At any given time, one vehicle whose position $x_{i,s}, y_{i,s}$ and orientation $\theta_{i,s}$ are known remains stationary, while the other vehicle advances. The index i indicates the step number and $'s'$ stands for stationary. At the end of each step, the distance and bearing of the two vehicles are measured $(r_i, \alpha_{is}, \alpha_{it})$; These measurements are used to estimate the traveling vehicle's position $x_{i,t}, y_{i,t}$ and orientation $\theta_{i,t}$ (where $'t'$ stands for traveling).

The traveling vehicle's location and orientation at each step is determined with respect to the observing vehicle's position. The general form of the Cartesian location and orientation of the traveling vehicle at step i is:

$$X_{i,t} = X_{i,s} + F\left(\theta_{i,s}, r_i, \alpha_{is}, \alpha_{it}\right), \tag{8}$$

where the vector X_i includes both the position and orientation of the vehicle: $X_i = \left[x_i\ y_i\ \theta_i\right]^T$. For example, in step 1, assume that vehicle 1 is stationary and its position $x_{1,1}$, $y_{1,1}$ and orientation $\theta_{1,1}$ are known and vehicle 2 traveled to a new position. The measured distance between the vehicles is r_1 and the measured bearing angles are α_{11} and α_{12}, where the first index refers to the step number and the second index refers to the measuring vehicle (see Fig. 3, left). Therefore, the Cartesian position and orientation of vehicle 2 with respect to vehicle 1 is:

$$\begin{aligned} x_{1,2} &= r_1 \cos\alpha_{11} \\ y_{1,2} &= r_1 \sin\alpha_{11} \\ \theta_{1,2} &= \alpha_{11} + 180^\circ - \alpha_{12}. \end{aligned} \tag{9}$$

By setting the initial position and orientation of vehicle 1 as the origin of the global coordinate system, meaning $x_{1,1} = 0$, $y_{1,1} = 0$ and $\theta_{1,1} = 0$, Eq. (9) represents the global position of vehicle 2 at the end of the first step.

Note that the orientation θ is determined solely by bearing measurements and is hardly influenced at all by the distance measurement, unlike in the two-point approach, where the orientation accuracy is decreased by the distance. This is one of the key advantages of our method since distance errors tend to increase together with the distance while angle measurements remain almost unchanged.

In step 2, vehicle 1 travels to its next target point while vehicle 2 is stationary and its position is known ($x_{2,2} = x_{1,2}$, $y_{2,2} = y_{1,2}$ and $\theta_{2,2} = \theta_{1,2}$). At the end of the step, the distance and angle measurements are r_2, α_{22} and α_{21} (see Fig. 3, center). It should be noted that the measurements are obtained with respect to vehicle's 2 current position and its local coordinate system. In order to obtain the position of vehicle 1 in the global coordinate system, a transformation is needed.

D. MULTISTEP REPRESENTATION USING HOMOGENEOUS COORDINATES

The local transformation matrix at step n from the traveling vehicle's coordinate system (n) to the stationary vehicle's coordinate system (n-1) is:

$$A_{n-1}^{n} = \begin{bmatrix} \cos\psi_n & -\sin\psi_n & r_n \cos\alpha_{ns} \\ \sin\psi_n & \cos\psi_n & r_n \sin\alpha_{ns} \\ 0 & 0 & 1 \end{bmatrix}, \tag{10}$$

where ψ_n is the relative angle between the two coordinate systems, hence $\psi_n = \alpha_{ns} + 180^\circ - \alpha_{nt}$ (see Fig. 3). Obtaining the position of the traveling vehicle in the global coordinate system (0), can be achieved recursively as follows:

$$A_0^n = A_0^{n-1} \cdot A_{n-1}^n, \tag{11}$$

where A_0^{n-1} is the overall transformation matrix obtained in the last step (n-1), and A_{n-1}^n is the n step's local transformation matrix as shown in Eq. (10).

Since the transformation matrix is composed of a rotation matrix and a shifting vector, the first two expressions of the third column of the matrix A_0^n are the Cartesian location of the traveling vehicle in the global coordinate system at step n, and the angle of the rotation matrix is the vehicle's orientation in the global coordinate system.

III. ERROR EVALUATION

Since all sensor measurements contain precision errors, this section presents a statistical analysis to evaluate the influence of the cumulative errors on the overall location of the robot after a large number of steps.

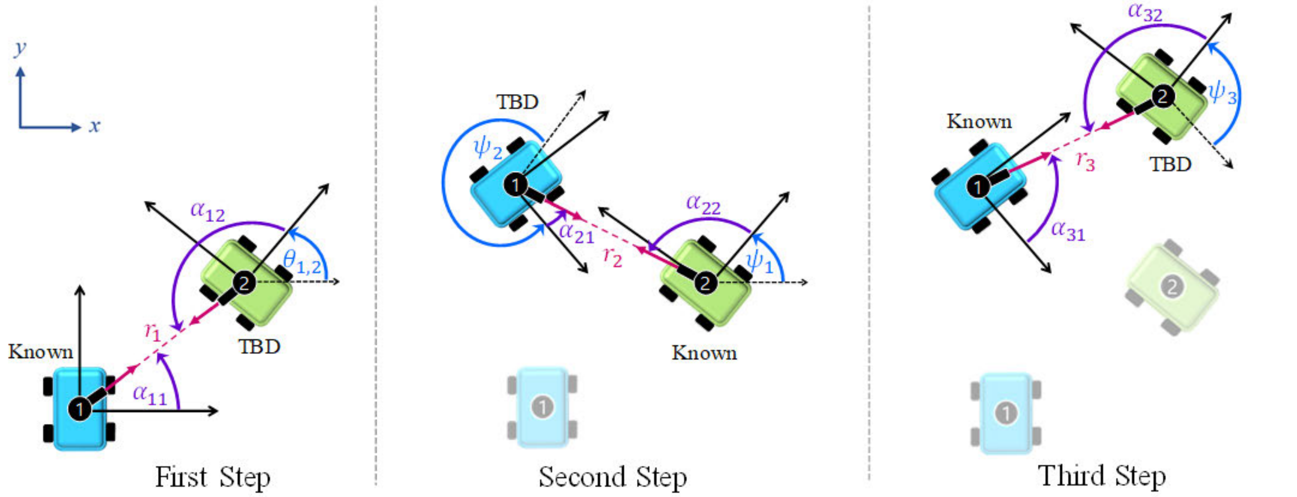

FIGURE 3. The first three steps and their measurements. At each step, one robot is static and tracks the motion of the advancing robot.

A. EXACT METHOD

The measured location of the traveling vehicle at step n is expressed as a function of measured distances $r_1, r_2, \ldots, r_n$ and angles $\alpha_{11}, \alpha_{12}, \ldots, \alpha_{n1}, \alpha_{n2}$ in the global coordinate system $f(r_1, \ldots, r_\mathrm{n}, \alpha_{11}, \ldots, \alpha_{n2})$; thus the real location including distance and bearing measurement errors, $\Delta r_1, \Delta r_2, \ldots, \Delta r_n$ and $\Delta\alpha_{11}, \Delta\alpha_{12}, \ldots, \Delta\alpha_{n1}, \Delta\alpha_{n2}$ respectively, is $f(r_1 + \Delta r_1, \ldots, r_n + \Delta r_n, \alpha_{11} + \Delta\alpha_{11}, \ldots, \alpha_{n2} + \Delta\alpha_{n2})$.

For example, if during the first step the distance and bearing were measured with an error of Δr_1, $\Delta\alpha_{11}$ and $\Delta\alpha_{12}$ respectively, the measured location of vehicle 2:

$$\begin{aligned} x_{1,2}^m &= (r_1 + \Delta r_1)\cos(\alpha_{11} + \Delta\alpha_{11}) \\ y_{1,2}^m &= (r_1 + \Delta r_1)\sin(\alpha_{11} + \Delta\alpha_{11}) \\ \theta_{1,2}^m &= \alpha_{11} + \Delta\alpha_{11} + 180^\circ - (\alpha_{12} + \Delta\alpha_{12}). \end{aligned} \tag{12}$$

This method uses the presented localization method directly (see Section II-C and II-D), and hence requires multiple matrix multiplications and a large number of trigonometric calculations which result in high numerical complexity.

B. FIRST ORDER APPROXIMATED METHOD

An approximated, yet computationally simpler method to evaluate the measured locations of the vehicles uses:

$$\begin{bmatrix} x^m \\ y^m \\ \theta^m \end{bmatrix} \approx \begin{bmatrix} x^r \\ y^r \\ \theta^r \end{bmatrix} + [J][\Delta], \tag{13}$$

where the index $'r'$ refers to the real location, $[\Delta]$ is the measurement errors vector and $[J]$ stands for the Jacobian matrix:

$$J_{ij} = \frac{\partial g_i}{\partial q_j}, \tag{14}$$

where g_i are the functions of Cartesian location and orientation and q_j are the variables of these functions, hence $r_1, \alpha_{11}, \alpha_{12}, \ldots, r_j, \alpha_{j1}, \alpha_{j2}$. For example, the estimated position of vehicle 2 after the first step is:

$$\begin{bmatrix} x_{1,2}^m \\ y_{1,2}^m \\ \theta_{1,2}^m \end{bmatrix} \approx \begin{bmatrix} r_1\cos\alpha_{11} \\ r_1\sin\alpha_{11} \\ \alpha_{11} + 180^\circ - \alpha_{12} \end{bmatrix} + \begin{bmatrix} \cos\alpha_{11} & -r_1\sin\alpha_{11} & 0 \\ \sin\alpha_{11} & r_1\cos\alpha_{11} & 0 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} \Delta r_1 \\ \Delta\alpha_{11} \\ \Delta\alpha_{12} \end{bmatrix}. \tag{15}$$

At the next step, the location is determined by 6 measurements; thus, the Jacobian becomes a 3 × 6 matrix and the measurement error is a 6 × 1 vector. At step n, the Jacobian is a $3 \times 3n$ matrix and the measurement error is a $3n \times 1$ vector. A general form of the location error for step n is:

$$E_n = [J][\Delta]_n = E_{n-1} + \begin{bmatrix} -r_n\sin\left(\theta_{n-1} + \alpha_{n,s}\right)\cdot\varepsilon + \Delta r_n\cos\left(\theta_{n-1} + \alpha_{n,s}\right) \\ r_n\cos\left(\theta_{n-1} + \alpha_{n,s}\right)\cdot\varepsilon + \Delta r_n\sin\left(\theta_{n-1} + \alpha_{n,s}\right) \\ \Delta\alpha_{n,s} - \Delta\alpha_{n,t} \end{bmatrix}, \tag{16}$$

where:

$$\varepsilon = \sum\nolimits_{i=1}^{n} \Delta\alpha_{i,s} - \sum\nolimits_{j=1}^{n-1} \Delta\alpha_{j,t}. \tag{17}$$

IV. MONTE CARLO SIMULATION

A Monte Carlo Simulation (MCS) was used in order to simulate a real-life scenario where the input of the sensors contains statistical errors. A natural random statistical error with a given standard deviation was inserted to the "measured values" and the statistical distribution of the position of the vehicles was calculated (using 10,000 simulations for each step). This section first presents a comparison between the first order statistical approximation to the exact method (IV-A), the statistical distribution along the path (IV-B), the influence of the sensor error on the accuracy of the measured location (IV-C) and finally a comparison between different paths and advancing (parallel, alternating and following) methods (IV-D).

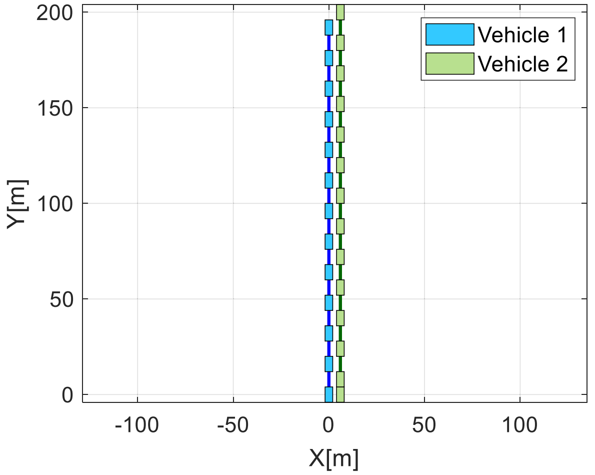

FIGURE 4. A 25 step straight line path. This simple path was chosen for our numerical MCS.

A. COMPARING THE FIRST ORDER APPROXIMATED METHOD TO THE EXACT METHOD

The MCS was first performed throughout a simple path composed of two straight lines as seen in Fig. 4. At each step, the traveling vehicle advances by 8 m and the final distance from the stationary vehicle is 10 m; i.e. the system overall advances 200 m throughout 25 steps.

The distance and angle measurement errors were simulated as normally distributed sets of N samples each (for each step), with zero mean. The standard deviation of the distance measurement error was set to $\sigma_d \cdot r_i$, where σ_d is the distance measurement resolution and r_i is the current step's measured relative distance. The standard deviation of the angle measurement error was set to σ_α, the bearing sensor's resolution.

The MCS comparison was performed using the exact method (III-A) and the approximated method (III-B). In both cases, $N = 10{,}000$; i.e., each step of the path was evaluated 10,000 times, for a set of 10,000 samples of random measurement errors, resulting in 10,000 possible locations for each step. Fig. 5 presents the relative difference between the final locations calculated by both methods, relative to the total traveled distance. Fig. 5 (top) shows that for $\sigma_\alpha \leq 0.5°$ (a reasonable assumption for a standard bearing sensor), the difference between the exact and approximated methods is less than 0.1% of the traveled distance (200 meters in 25 steps). The error increases to 1.5% for $\sigma_\alpha = 2°$. Fig. 5 (bottom) which presents the difference between the two methods as a function of distance standard deviation σ_d shows that the error is dominated by the angle error. In terms of computation time, the approximated method was found to be nearly 200 times faster than the exact method computation time. Therefore, the approximated method was used in the following MCS.

B. STATISTICAL DISTRIBUTION

This section presents a statistical analysis of the MCS location errors using $\sigma_d = 2\%$ and $\sigma_\alpha = 0.5°$. The distribution of possible locations for each step is presented as a two-dimensional histogram (see Fig. 6). The size and shape of the distribution can be described by three standard deviation values. The first is the total standard deviation according to the distance between the centroid and the

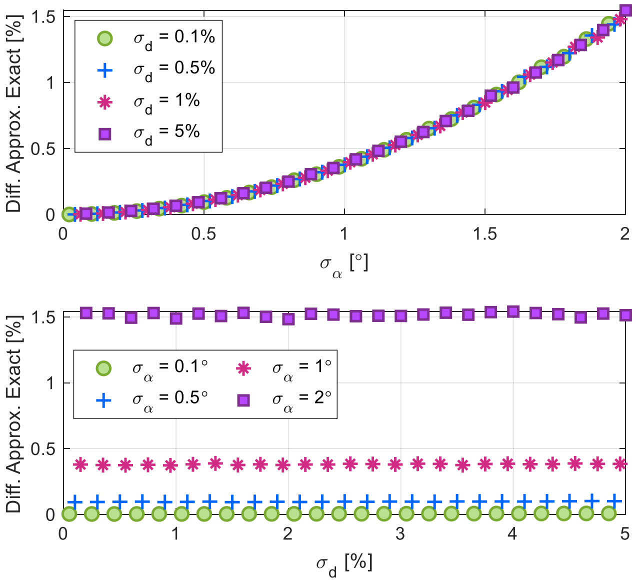

FIGURE 5. Relative difference between the last step's locations, calculated using the exact and approximated methods, relative to total distance traveled (200 m), as a function of bearing sensor's resolution (top), and as a function of range measurement's resolution (bottom). Each point is the average of 10,000 simulations.

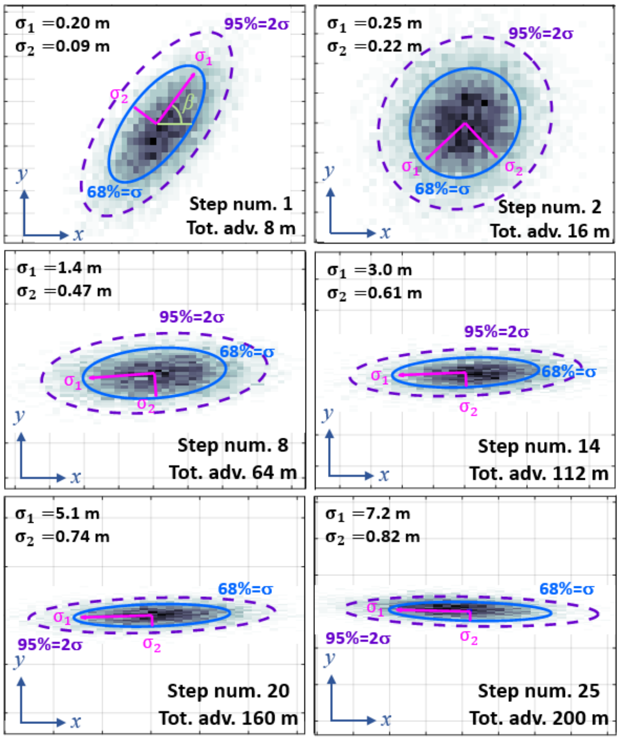

FIGURE 6. Histogram distribution of the measured locations using the MCS with 10,000 paths with $\sigma_d = 2\%$ and $\sigma_\alpha = 0.5°$ for steps 1, 2, 8, 14, 20, and 25 with the confidence distribution 68% (σ) and 95% (2σ).

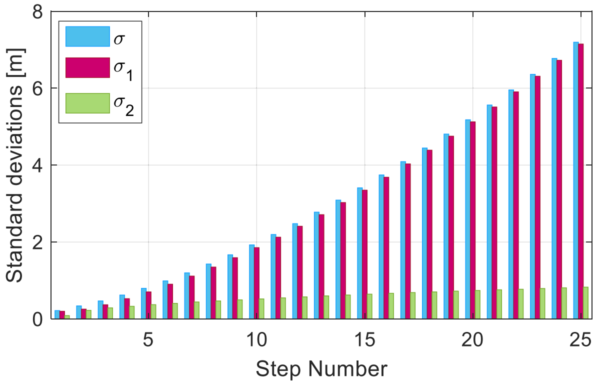

FIGURE 7. Standard deviations as a function of the number of steps for a 200 m straight path using $\sigma_d = 2\%$ and $\sigma_\alpha = 0.5°$.

different simulation results:

$$\sigma = \left(\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu_x)^2+(y_i-\mu_y)^2\right)^{0.5}, \tag{18}$$

where (μ_x, μ_y) are the coordinates of the approximated method's centroid and (x_i, y_i) are the coordinates of all possible locations, $i = 1, \ldots, N$.

Since the distribution pattern of possible locations tends to yield an ellipse, two other standard deviations were calculated according to the ellipse's axes. These values were obtained by calculating the covariance matrix of the N Cartesian locations:

$$\operatorname{cov}\begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_N & y_N \end{bmatrix}, \tag{19}$$

resulting in a 2 × 2 covariance matrix, with two 2 × 1 eigenvectors $\{V_1, V_2\}$ and two corresponding eigenvalues $\{\lambda_1, \lambda_2\}$. The eigenvectors of the covariance matrix represent the direction of the ellipse's axes, and the square root of their corresponding eigenvalues represent the standard deviations in their direction. Assuming $\lambda_1 > \lambda_2$, the two standard deviations values are:

$$\sigma_1 = \lambda_1^{0.5}, \quad \sigma_2 = \lambda_2^{0.5}, \tag{20}$$

where σ_1 is the standard deviation in the direction of the main axis of the ellipse and σ_2 is the standard deviation in the perpendicular direction. The angle between the ellipse's main axis and the global x positive axis (see Fig. 6 top left) is:

$$\beta = \arctan\frac{V_1(y)}{V_1(x)}. \tag{21}$$

Although the error distribution of the first step seemed to be affected mostly by the distance measurement's error, the error distribution of the next steps had a circular pattern. The pattern became elliptical in the next steps with σ_1 becoming larger relative to σ_2 (see Fig. 6 and Fig. 7). The overall standard deviation of the error σ grew almost linearly with the number of steps and distance traveled. The ratio of σ divided by the traveled distance is 0.036 which is in the same order of magnitude as the sensors' relative error.

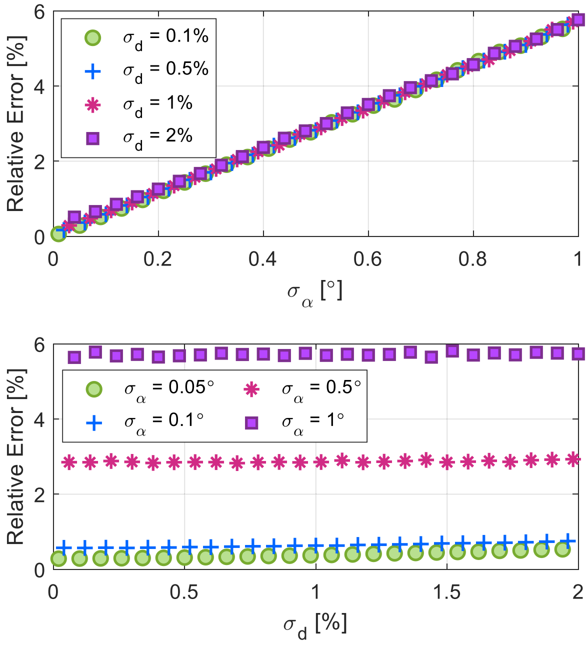

FIGURE 8. Relative error between the measured position of the robot and the real position for a travelled distance of 200 m, as a function of bearing sensor's resolution (top), and as a function of range sensor's resolution (bottom). Each point is the average of 10,000 simulations.

Using Eq. (16) the standard deviation of the orientation of the vehicle at step n can be evaluated directly as follows:

$$\sigma_\theta = \sqrt{2n}\cdot\sigma_\alpha, \tag{22}$$

implying that the orientation error depends solely on the number of steps and the bearing sensor's accuracy.

Additionally, the standard deviations in the 'x' and 'y' axes directions can be analytically derived from Eq. (16) (see Appendix):

$$\sigma_x = \left[\sum_{i=1}^{n} r_i^2\cos^2\left(\theta_{i-1}+\alpha_{i,s}\right)\sigma_d^2 + \sum_{j=1}^{n}\Omega(j)\left(\sum_{i=j}^{n} r_i\sin\left(\theta_{i-1}+\alpha_{i,s}\right)\right)^2\sigma_\alpha^2\right]^{0.5}, \tag{23}$$

$$\sigma_y = \left[\sum_{i=1}^{n} r_i^2\sin^2\left(\theta_{i-1}+\alpha_{i,s}\right)\sigma_d^2 + \sum_{j=1}^{n}\Omega(j)\left(\sum_{i=j}^{n} r_i\cos\left(\theta_{i-1}+\alpha_{i,s}\right)\right)^2\sigma_\alpha^2\right]^{0.5}, \tag{24}$$

where:

$$\Omega(j) = \begin{cases} 1, & j = 1 \\ 2, & j > 1. \end{cases} \tag{25}$$

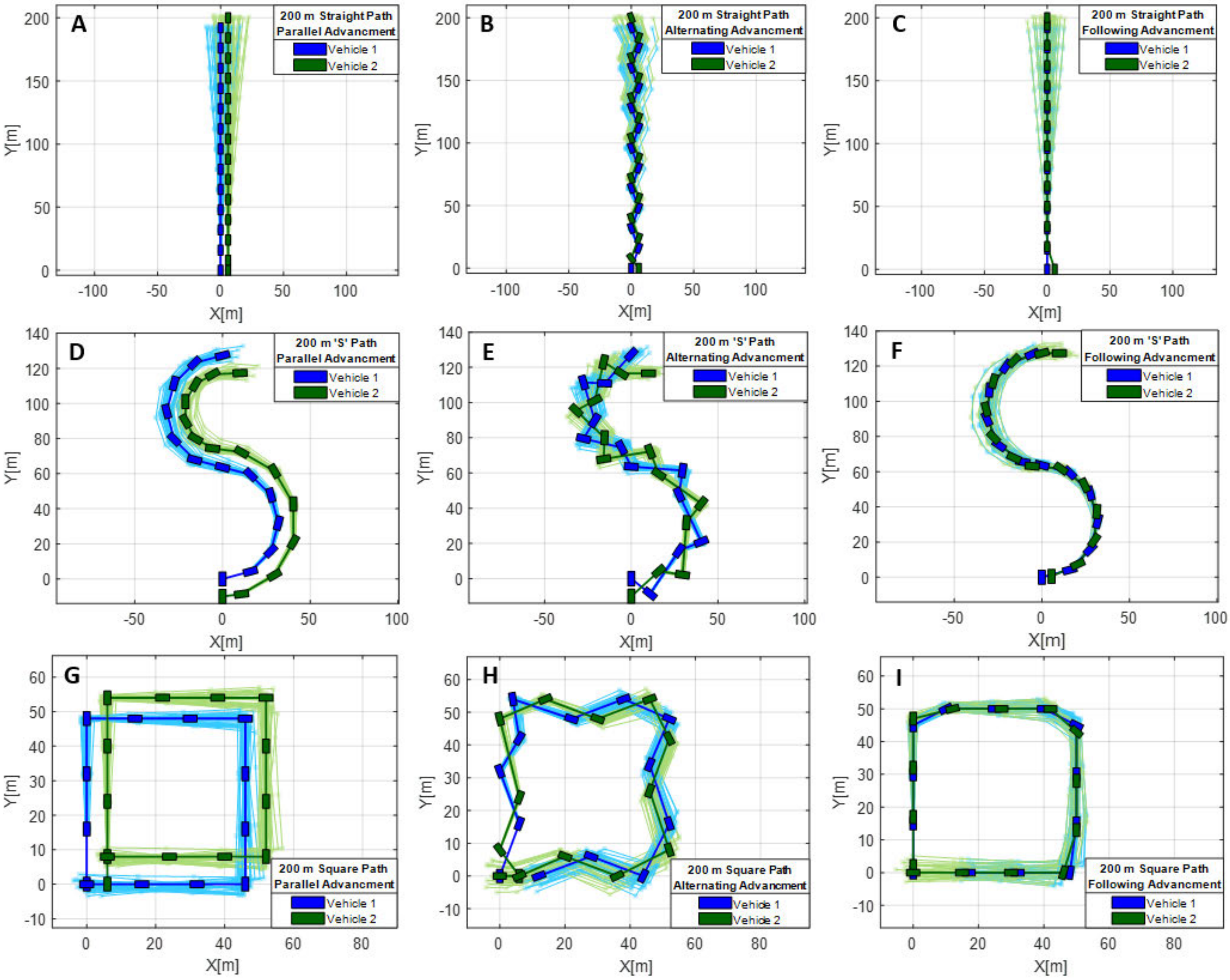

FIGURE 9. Nine scenarios of three different paths and three different advancing methods. From top to bottom: straight path, 'S' path and square path. From left to right: parallel advancing, alternating advancing and following advancing. Lighter colors present 30 optional locations due to random sensors errors with $\sigma_d = 2\%$ and $\sigma_\alpha = 0.5°$.

C. COMPARING THE INFLUENCE OF THE SENSOR ERROR ON THE ACCURACY OF THE MEASURED LOCATION

Fig. 8 (top) presents the relative error between the measured position of the robot and the real position as a function of the standard deviation of the bearing measurement error σ_α, while Fig. 8 (bottom) presents the same error as a function of the standard deviation of the distance measurement error σ_d. For each case, we ran 10,000 simulations, each composed of 25 steps and the total net advancement is 200 meters. The results presented in this figure show that the relative error is governed by the bearing error measurements.

D. PATH COMPARISON

In this section, MCS are used to statistically calculate the influence of the sensor accuracy on the location error for three different paths (straight, 'S' shape, and square) using three advancing methods (parallel, alternating and following).

TABLE 1. Values of sensor variables used in the simulation.

Sensor variables	STD #1	STD #2	STD #3	STD #4
σ_d=*error/distance*	1%	5%	2%	5%
σ_α [°]	0.1	0.1	0.5	1

In total, nine scenarios were examined for four different combinations of sensor errors (see Table 1).

Fig. 9 A-C presents the path distribution of 30 simulations in a straight path using three different advancing methods: A) parallel, B) alternating and C) following. Figure 9 D-F, and G-I present the same different advancing methods for 'S' shape and square paths respectively. We used $\sigma_d = 2\%$ and $\sigma_\alpha = 0.5°$.

The resulting total standard deviation, and its components along the direction of motion and in the vertical direction are summarized in Table 2 (a,b,c). Note that

TABLE 2. (a) 200 meters Straight path standard deviation values for different sensors resolution (using 10,000 simulations). (b) 200 meters 'S' path standard deviation values for different sensors resolution (using 10,000 simulations). (c) 200 meters Square path standard deviation values for different sensors resolution (using 10,000 simulations).

Sensor	Total standard deviation (σ)			Advancing direction (σ_y)			Perpendicular direction (σ_x)		
variables	Parallel	Alternating	Following	Parallel	Alternating	Following	Parallel	Alternating	Following
σ_d=1% σ_α=0.1°	1.51 m [0.75%]	**1.49 m [0.72%]**	1.51 m [0.75%]	**0.404 m [0.20%]**	**0.403 m [0.20%]**	0.631 m [0.31%]	1.46 m [0.72%]	1.43 m [0.72%]	**1.37 m [0.69%]**
σ_d=5% σ_α=0.1°	2.86 m [1.4%]	**2.66 m [1.3%]**	3.43 m [1.7%]	**1.97 m [1.0%]**	**1.99 m [1.0%]**	3.14 m [1.6%]	2.08 m [1.0%]	1.77 m [0.88%]	**1.38 m [0.69%]**
σ_d=2% σ_α=0.5°	7.23 m [3.6%]	7.16 m [3.6%]	**6.97 m [3.5%]**	**0.836 m [0.42%]**	**0.840 m [0.42%]**	1.26 m [0.63%]	7.18 m [3.6%]	7.11 m [3.6%]	**6.86 m [3.4%]**
σ_d=5% σ_α=1°	14.3 m [7.2%]	14.4 m [7.2%]	**14.1 m [7.1%]**	**2.06 m [1.0%]**	**2.08 m [1.0%]**	3.17 m [1.6%]	14.2 m [7.2%]	14.2 m [7.1%]	**13.8 m [6.9%]**

(a)

Sensor	Total standard deviation (σ)			Advancing direction (σ_x)			Perpendicular direction (σ_y)		
variables	Parallel	Alternating	Following	Parallel	Alternating	Following	Parallel	Alternating	Following
σ_d=1% σ_α=0.1°	**1.07 m [0.54%]**	1.14 m [0.57%]	1.20 m [0.60%]	**0.926 m [0.46%]**	0.960 m [0.48%]	1.02 m [0.51%]	**0.542 m [0.27%]**	0.609 m [0.30%]	0.629 m [0.31%]
σ_d=5% σ_α=0.1°	**3.35 m [1.7%]**	3.89 m [1.9%]	4.02 m [2.0%]	**2.43 m [1.2%]**	2.80 m [1.4%]	2.91 m [1.4%]	**2.31 m [1.2%]**	2.70 m [1.3%]	2.78 m [1.4%]
σ_d=2% σ_α=0.5°	**4.46 m [2.2%]**	**4.48 m [2.2%]**	4.86 m [2.4%]	**4.12 m [2.1%]**	**4.11 m [2.1%]**	4.49 m [2.2%]	**1.71 m [0.85%]**	1.79 m [0.90%]	1.87 m [0.94%]
σ_d=5% σ_α=1°	**9.17 m [4.6%]**	**9.23 m [4.6%]**	10.0 m [5.0%]	**8.40 m [4.2%]**	**8.36 m [4.2%]**	9.12 m [4.6%]	**3.68 m [1.8%]**	3.92 m [2.0%]	4.12 m [2.1%]

(b)

Sensor	Total standard deviation (σ)			Advancing direction (σ_x)			Perpendicular direction (σ_y)		
variables	Parallel	Alternating	Following	Parallel	Alternating	Following	Parallel	Alternating	Following
σ_d=1% σ_α=0.1°	0.877 m [0.44%]	0.973 m [0.49%]	**0.831 m [0.42%]**	0.607 m [0.30%]	0.734 m [0.37%]	**0.587 m [0.29%]**	0.634 m [0.32%]	0.638 m [0.32%]	**0.583 m [0.29%]**
σ_d=5% σ_α=0.1°	3.70 m [1.8%]	3.96 m [2.0%]	**3.03 m [1.5%]**	2.54 m [1.3%]	3.02 m [1.5%]	**2.14 m [1.1%]**	2.69 m [1.3%]	2.56 m [1.3%]	**2.15 m [1.1%]**
σ_d=2% σ_α=0.5°	**2.83 m [1.4%]**	3.24 m [1.6%]	3.15 m [1.6%]	**1.99 m [0.99%]**	2.41 m [1.2%]	2.25 m [1.1%]	**2.01 m [1.0%]**	2.19 m [1.1%]	2.20 m [1.1%]
σ_d=5% σ_α=1°	**6.07 m [3.0%]**	6.92 m [3.5%]	6.48 m [3.2%]	**4.27 m [2.1%]**	5.14 m [2.6%]	4.59 m [2.3%]	**4.32 m [2.2%]**	4.63 m [2.3%]	4.57 m [2.3%]

(c)

the standard deviation values were calculated twice; once using MCS (10,000 simulations) and then using the analytical expressions developed in the Appendix. The relative difference between both methods is always smaller than 1% (see Appendix).

For the straight path, (a), the overall standard deviation is the largest compared to the other paths and is nearly unaffected by the advancing method. However, for the other two paths ('S' shape (b) and square (c)), the parallel advancing method mostly generated smaller location errors, where the square path resulted with the smallest errors.

For the straight path, the standard deviation in the vertical direction is substantially larger than the standard deviation in the direction of motion. The square path on the other hand, resulted in nearly equal standard deviations both in the parallel and perpendicular direction, due to equal advancement in both directions.

Overall, the three different advancing methods (parallel, alternating and following) do not result in significant differences in the standard deviation values within a specific path. The size and the distribution pattern of the errors are influenced mainly by the overall advancing direction of the system and almost unaffected by the relative position of the vehicles within each step.

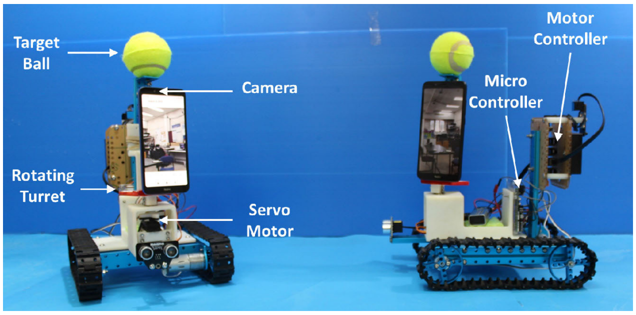

FIGURE 10. The robotic system used in the experiments.

V. EXPERIMENTS

This section presents an experimental system that was used to validate our algorithm, experimental results and comparison to the previously presented Monte Carlo simulation.

A. EXPERIMENTAL SYSTEM

To validate our algorithm and simulations, we built a two-robot experimental system fitted with rotating turrets and cameras (see Fig. 10). Each turret is equipped with a smartphone's video camera (1080 × 1920 pixels at 30fps). A green 6.5 cm tennis ball was placed at the top of the turret for

TABLE 3. Standard deviation and mean error values of experiments' last step result.

Paths	Total		Advancing direction		Perpendicular direction		Orientation	
	σ	Mean error	σ_y	Mean error y	σ_x	Mean error x	σ_θ	Mean error θ
Straight path [10 steps, 7.5 m]	0.0773 m [1.0%]	0.0797 m [1.1%]	0.0492 m [0.66%]	-0.0424 m [0.56%]	0.0597 m [0.79%]	-0.0278 m [0.37%]	3.00°	3.57°
Square path [16 steps, 12 m]	0.0751 m [0.63%]	0.0174 m [0.14%]	0.0594 m [0.49%]	-0.00780 m [0.065%]	0.0458 m [0.38%]	-0.00006 m [0.0005%]	1.06°	-0.011°
'S' path [8 steps, 6 m]	0.179 m [3.0%]	0.207 m [3.4%]	0.127 m [2.1%]	0.0529 m [0.88%]	0.0997 m [1.7%]	0.200 m [3.3%]	4.78°	-3.35°

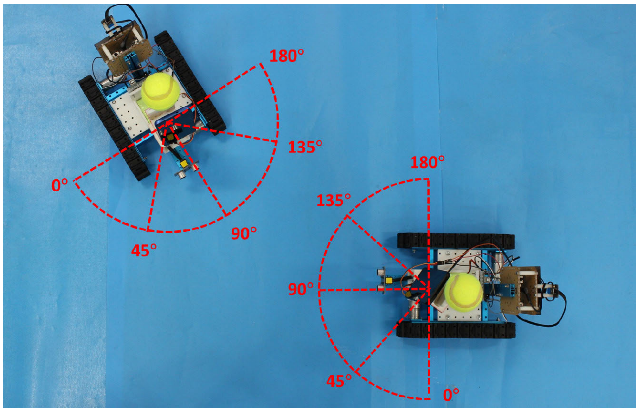

FIGURE 11. Top view of the robotics system. The turrets rotate in steps of 45 degrees.

visual identification. The turret is connected to a servo motor controlled by an Arduino microcontroller programmed to continuously rotate the turret by steps of 45 degrees, from zero till 180 and returning. At each stop, the turret pauses for one second (see Fig. 11). Given the camera's field of view in the horizontal direction γ_x is 40 degrees, the total field of view of each robot is 220 degrees.

We ran four different experiments in lab conditions that were each repeated five times. In each experiment, the robots (controlled by a human operator) advance in alternating steps while the turret rotates as the camera continuously records video. The localization of the robots is performed off-line at the end of the experiment. Each step is represented by two images (1080 × 1920), one from each camera and the orientation of the turrets α_{turret}.

The bearing angle α of each robot is the sum of the orientation of the turret α_{turret} plus the angular position of the tennis ball in the picture α_{image}:

$$\alpha = \alpha_{turret} + \alpha_{image}. \tag{26}$$

The angular position in the image is calculated using:

$$\alpha_{image} = \text{atan}\left(\frac{C_x}{N_x/2}\tan\left(\frac{\gamma_x}{2}\right)\right), \tag{27}$$

where C_x is the x coordinate of the center of the ball in pixels, with respect to the center of the frame (see Fig. 12, top right).

The distance r between the robots (see Fig. 12, bottom right) is calculated using:

$$r = \frac{L}{\tan^{-1}(\alpha_L)}, \tag{28}$$

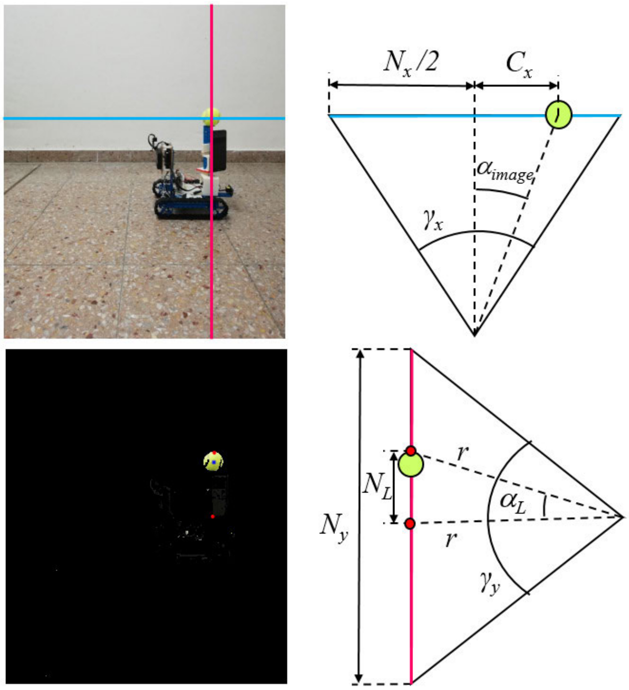

FIGURE 12. Schematic diagram of distance (bottom right) and bearing (top right) calculation from frame. Top left: original frame, bottom left: frame after image filtering, center of ball and top and bottom of turret detected.

where L is the length of the turret (21 cm) and α_L is the view angle of the height of the turret in the frame, calculated from the image:

$$\alpha_L = \frac{N_L}{N_y}\cdot\gamma_y, \tag{29}$$

where N_L is the size of the turret in pixels and N_y is the overall size of the image in the y direction. The camera's field of view γ_y in the vertical direction is 70 degrees. The distance r at each step is calculated from the average of both images (one from each robot).

B. RESULTS

The results of the experiments are presented in Fig. 13 and Table 3. All three experiments show high repeatability with a relatively small average error (1.1%, 0.14% and 3.4% respectively for the straight, square and 'S' experiments). The total standard deviation is also relatively small with respectively 1%, 0.63% and 3.0%. We believe that the most significant error in our experimental system is a systematic error in the

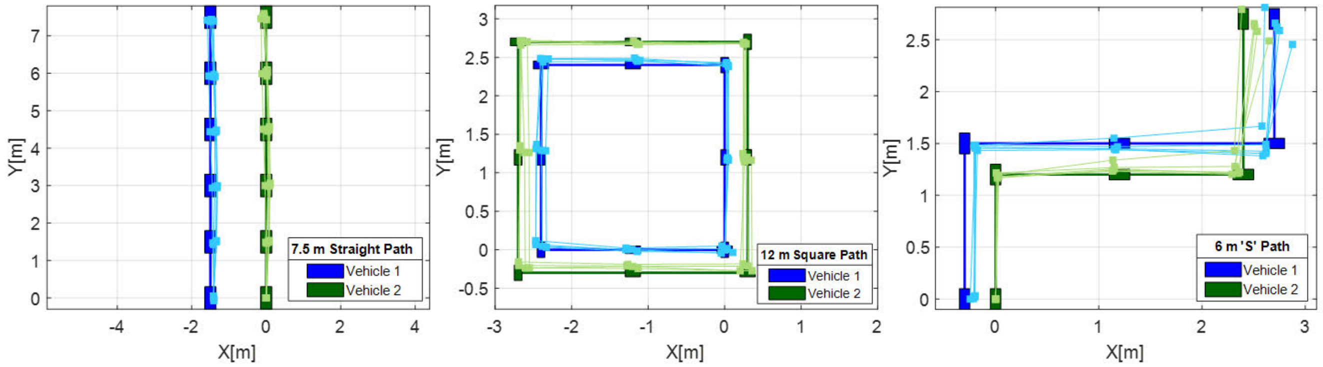

FIGURE 13. Experiment results of three different paths; from left to right: straight path, square path and 'S' path. Darker colors present real locations and lighter colors present calculated locations from five experiment results.

TABLE 4. Standard deviation values of simulations' last step with σ_d = 1% and $\sigma_\alpha = 0.3°$.

Paths	Total σ	Adv. dir. σ_y	Per. dir. σ_x	Orient. σ_θ
Straight path [10 steps, 7.5 m]	0.114 m [1.5%]	0.0423 m [0.56%]	0.106 m [1.4%]	0.0234°
Square path [16 steps, 12 m]	0.0778 m [0.65%]	0.0637 m [0.53%]	0.0446 m [0.37%]	0.0296°
'S' path [8 steps, 6 m]	0.0523 m [0.87%]	0.0389 m [0.65%]	0.0350 m [0.58%]	0.0209°

bearing measurements. As previously discussed, the bearing sensor's error has a high impact on the localization error.

C. COMPARISON TO SIMULATION

The system's repeatability error was evaluated by repeating the same measurement at least 10 times. The standard deviation of the calculated distances and bearing angles were $\sigma_d = 1\%$ (with respect to the real distance) and $\sigma_\alpha = 0.3°$. The values of the standard deviation were implemented in the MCS in order to compare the simulation to the experiments. The simulation results presented in Table 4 show that the experimental results are of the same order as expected by the simulation. Note that since the MCS uses normally distributed random errors, the mean error for each step is zero.

For the first two experiments, straight and square paths, the standard deviation is very similar to the simulation. In the 'S' shape path, the standard deviation of the experiment is of the same order as the simulation but is slightly more than 3 times larger.

VI. CONCLUSION AND FUTURE WORK

In this paper we presented a simple, low cost method for precise multi-robot self-localization that relies on distance and bearing measurements. The system can be deployed in indoor areas where GPS signals are unavailable, and visibility is relatively low. The key advantage of this method is that it reduces the errors resulting from the inaccuracies of evaluating the orientation of the robots. We developed an analytical solution for the position of the robots and a numerical simulation to account for the statistical sensors' errors. We show that the total relative error (cumulative error divided by travelled distance) is on the same order of magnitude as the sensors' relative errors (error divided by distance), and that the angular error has a larger impact on the location errors than the distance error, thus making it important to use a relatively accurate bearing sensor.

Given that the sensor measurement contains statistical errors, we ran a Monte Carlo Simulation (MCS) and determined the spatial distribution of the measured/estimated location of the robot with the given sensors' random errors (10,000 simulations for each case). To reduce the MCS computation time, we developed an approximated error evaluation method based on first order linear approximation. This method was 200 times faster than the direct method.

We then used the MCS to compare between different paths and advancing methods. We found that the chosen path governed the size of the location error, whereas the different advancing methods had little influence on the total error. For a given equal number of steps and total travelled distance, the smallest error is in the square path, followed by the 'S' shaped path and the largest error is with the straight path. Overall, using our localization algorithm, it is best to increase the size of the steps and decrease their number in order to reduce the bearing errors and increase the accuracy of the localization.

Next we present a two-robot experimental system used to further validate our algorithm. We performed experiments in three different paths, calculated the standard deviation and mean error values and compared the results to the Monte Carlo simulation. The results show that the method is very accurate with errors of about 1-3% of the total distance travelled.

Besides its advantages, the method presented in this paper does require a line of sight between the two cooperating robots and that one of the robots must remain fully static during each step. Another limitation is that the method is currently limited to 2D localization. However, we expect that it can be generalized to 3D problems.

Our future work will include further development of our experimental system using off-the-shelf sensors to

TABLE 5. Total standard deviation values using 10,000 simulations, analytic calculation and relative difference, 200 meters 'S' path.

Sensor variables	Parallel			Alternating			Following		
	Simulation	Analytic	Relative diff.	Simulation	Analytic	Relative diff.	Simulation	Analytic	Relative diff.
σ_d=1% σ_α=0.1°	1.073 m	1.075 m	0.23%	1.137 m	1.134 m	0.27%	1.198 m	1.204 m	0.55%
σ_d=5% σ_α=0.1°	3.353 m	3.372 m	0.58%	3.887 m	3.857 m	0.79%	4.023 m	4.010 m	0.32%
σ_d=2% σ_α=0.5°	4.465 m	4.468 m	0.080%	4.484 m	4.499 m	0.32%	4.864 m	4.843 m	0.45%
σ_d=5% σ_α=1°	9.173 m	9.148 m	0.27%	9.233 m	9.276 m	0.46%	10.01 m	9.965 m	0.43%

verify the extent to which this system is implementable on some of our previously developed crawling and flying robots [23],[24]. On the theoretical level, we plan to expand our algorithm to a multi-robot system (three or more robots) and develop path planning strategies to increase accuracy, control the shape of the error distribution pattern and adapt the system to more complex dynamic environments.

APPENDIX

In this appendix we develop an analytical approximation of the standard deviation of the location error as a function of the standard deviations of the sensor accuracy. The approximated location and orientation of the vehicles is cumulative, meaning that it is affected by all the previous error measurements of the distance and bearing. The measurement error of the first step (see Eq. (15)) is:

$$E_1 = \begin{bmatrix} \cos\alpha_{11} & -r_1 \sin\alpha_{11} & 0 \\ \sin\alpha_{11} & r_1 \cos\alpha_{11} & 0 \\ 0 & 1 & -1 \end{bmatrix} \Delta_1. \tag{30}$$

At the second step it becomes:

$$\begin{aligned} E_2 &= E_1 \\ &\quad + \begin{bmatrix} 0 & -r_2 \sin\left(\theta_1 + \alpha_{2,s}\right) & r_2 \sin\left(\theta_1 + \alpha_{2,s}\right) \\ 0 & r_2 \cos\left(\theta_1 + \alpha_{2,s}\right) & -r_2 \cos\left(\theta_1 + \alpha_{2,s}\right) \\ 0 & 0 & 0 \end{bmatrix} \Delta_1 \\ &\quad + \begin{bmatrix} \cos\left(\theta_1 + \alpha_{2,s}\right) & -r_2 \sin\left(\theta_1 + \alpha_{2,s}\right) & 0 \\ \sin\left(\theta_1 + \alpha_{2,s}\right) & r_2 \cos\left(\theta_1 + \alpha_{2,s}\right) & 0 \\ 0 & 1 & -1 \end{bmatrix} \Delta_2. \end{aligned} \tag{31}$$

And at the third step:

$$\begin{aligned} E_3 &= E_2 \\ &\quad + \begin{bmatrix} 0 & -r_3 \sin\left(\theta_2 + \alpha_{3,s}\right) & r_3 \sin\left(\theta_2 + \alpha_{3,s}\right) \\ 0 & r_3 \cos\left(\theta_2 + \alpha_{3,s}\right) & -r_3 \cos\left(\theta_2 + \alpha_{3,s}\right) \\ 0 & 0 & 0 \end{bmatrix} \Delta_1 \\ &\quad + \begin{bmatrix} 0 & -r_3 \sin\left(\theta_2 + \alpha_{3,s}\right) & r_3 \sin\left(\theta_2 + \alpha_{3,s}\right) \\ 0 & r_3 \cos\left(\theta_2 + \alpha_{3,s}\right) & -r_3 \cos\left(\theta_2 + \alpha_{3,s}\right) \\ 0 & 0 & 0 \end{bmatrix} \Delta_2 \\ &\quad + \begin{bmatrix} \cos\left(\theta_2 + \alpha_{3,s}\right) & -r_3 \sin\left(\theta_2 + \alpha_{3,s}\right) & 0 \\ \sin\left(\theta_2 + \alpha_{3,s}\right) & r_3 \cos\left(\theta_2 + \alpha_{3,s}\right) & 0 \\ 0 & 1 & -1 \end{bmatrix} \Delta_3. \end{aligned} \tag{32}$$

Generalizing the total error at step n, the error E_n in the directions x, y and the orientation θ is:

$$E_n = \begin{bmatrix} E_x \\ E_y \\ E_\theta \end{bmatrix}_n = \sum_{i=1}^{n} B_{i\_n} \Delta_i, \tag{33}$$

where the sensor error Δ_i at step i is:

$$\Delta_i = \begin{bmatrix} \Delta r_i & \Delta\alpha_{i,s} & \Delta\alpha_{i,t} \end{bmatrix}^T, \tag{34}$$

and the matrix B_{i_n} is calculated using:

$$B_{i\_n} = B_i + \sum_{j=i+1}^{n} C_j, \tag{35}$$

where:

$$B_i = \begin{bmatrix} \cos\left(\theta_{i-1} + \alpha_{i,s}\right) & -r_i \sin\left(\theta_{i-1} + \alpha_{i,s}\right) & 0 \\ \sin\left(\theta_{i-1} + \alpha_{i,s}\right) & r_i \cos\left(\theta_{i-1} + \alpha_{i,s}\right) & 0 \\ 0 & 1 & -1 \end{bmatrix}, \tag{36}$$

and:

$$C_j = \begin{bmatrix} 0 & -r_j \sin\left(\theta_{j-1} + \alpha_{j,s}\right) & r_j \sin\left(\theta_{j-1} + \alpha_{j,s}\right) \\ 0 & r_j \cos\left(\theta_{j-1} + \alpha_{j,s}\right) & -r_j \cos\left(\theta_{j-1} + \alpha_{j,s}\right) \\ 0 & 0 & 0 \end{bmatrix}. \tag{37}$$

Using Eq. (33), and under the assumption of random uncorrelated sensor measurement errors Δ_i (i.e. covariance of any two measurements is zero),

$$var\left(\Delta_i\right) = \begin{bmatrix} var\left(\Delta r_i\right) \\ var\left(\Delta\alpha_{i,s}\right) \\ var\left(\Delta\alpha_{i,t}\right) \end{bmatrix} = \begin{bmatrix} r_i^2 \sigma_d^2 \\ \sigma_\alpha^2 \\ \sigma_\alpha^2 \end{bmatrix}, \tag{38}$$

the variance of the total measurement error is:

$$var\left(E_n\right) = \sum_{i=1}^{n} var\left(B_{i\_n} \Delta_i\right). \tag{39}$$

The standard deviation in the x and y directions, respectively σ_x and σ_y are (Eq. (23)-(25)):

$$\sigma_x = \left[\sum_{i=1}^{n} r_i^2 \cos^2\left(\theta_{i-1} + \alpha_{i,s}\right) \sigma_d^2 + \sum_{j=1}^{n} \Omega\left(j\right) \left(\sum_{i=j}^{n} r_i \sin\left(\theta_{i-1} + \alpha_{i,s}\right) \right)^2 \sigma_\alpha^2 \right]^{0.5}, \tag{40}$$

$$\sigma_y = \left[\sum_{i=1}^{n} r_i^2 \sin^2\left(\theta_{i-1} + \alpha_{i,s}\right) \sigma_d^2 + \sum_{j=1}^{n} \Omega\left(j\right) \left(\sum_{i=j}^{n} r_i \cos\left(\theta_{i-1} + \alpha_{i,s}\right) \right)^2 \sigma_\alpha^2 \right]^{0.5}, \tag{41}$$

where:

$$\Omega(j) = \begin{cases} 1, & j = 1 \\ 2, & j > 1. \end{cases} \tag{42}$$

And the standard deviation of the orientation σ_θ is (Eq. (22)):

$$\sigma_\theta = \sqrt{2n} \cdot \sigma_\alpha, \tag{43}$$

The validation of Eq. (40)-(42) is performed in Table 5 where a comparison is made between the total standard deviation values σ calculated using the MCS (10,000 simulations) and the analytical expression. The results show that the largest relative difference between the two methods is 0.79%.

REFERENCES

[1] R. Kurazume, S. Nagata, and S. Hirose, "Cooperative positioning with multiple robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1994, pp. 1250–1257.

[2] R. Kurazume, S. Nagata, N. Sashida, and S. Hirose, "Study on cooperative positioning system (basic principle and measurement experiment)," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1996, pp. 1421–1426.

[3] I. M. Rekleitis, G. Dudek, and E. E. Milios, "Multi-robot exploration of an unknown environment, efficiently reducing the odometry Error," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, vol. 2, 1997, pp. 1340–1345.

[4] I. M. Rekleitis, G. Dudek, and E. E. Milios, "Experiments in free-space triangulation using cooperative localization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct. 2003, pp. 1777–1782.

[5] A. J. Davison and N. Kita, "Active visual localisation for cooperating inspection robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2000, pp. 1709–1715.

[6] M. Moors, F. Schneider, and D. Wildermuth, "Relative position estimation in a group of robots," in *Proc. Int. Conf. Climbing Walking Robots*, 2003, pp. 983–990.

[7] M. Todescato, A. Carron, R. Carli, A. Franchi, and L. Schenato, "Multi-robot localization via GPS and relative measurements in the presence of asynchronous and lossy communication," in *Proc. IEEE Eur. Control Conf.*, Jun./Jul. 2016, pp. 2527–2532.

[8] A. T. Rashid, M. Frasca, A. A. Ali, A. Rizzo, and L. Fortuna, "Multi-robot localization and orientation estimation using robotic cluster matching algorithm," *Robot. Auton. Syst.*, vol. 63, pp. 108–121, Jan. 2015.

[9] A. Martinelli, F. Pont, and R. Siegwart, "Multi-robot localization using relative observations," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 2797–2802.

[10] O. De Silva, G. K. I. Mann, and R. G. Gosine, "Efficient distributed multi-robot localization: A target tracking inspired design," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2015, pp. 434–439.

[11] C. Lin, Z. Lin, R. Zheng, G. Yan, and G. Mao, "Distributed source localization of multi-agent systems with bearing angle measurements," *IEEE Trans. Autom. Control*, vol. 61, no. 4, pp. 1105–1110, Apr. 2016.

[12] L. Luft, T. Schubert, S. I. Roumeliotis, and W. Burgard, "Recursive decentralized localization for multi-robot systems with asynchronous pairwise communication," *Int. J. Robot. Res.*, vol. 37, no. 10, pp. 1152–1167, 2018.

[13] J. Liu, J. Pu, L. Sun, and Y. Zhang, "Multi-robot cooperative localization with range-only measurement by UWB," in *Proc. IEEE Chin. Autom. Congr.*, Nov./Dec. 2018, pp. 2809–2813.

[14] C. Pierre, R. Chapuis, R. Aufrère, J. Laneurit, and C. Debain, "Range-only based cooperative localization for mobile robots," in *Proc. IEEE Int. Conf. Inf. Fusion*, Jul. 2018, pp. 1933–1939.

[15] Y. Cao, M. Li, I. Švogor, S. Wei, and G. Beltrame, "Dynamic range-only localization for multi-robot systems," *IEEE Access*, vol. 6, pp. 46527–46537, 2018.

[16] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *Int. J. Robot. Res.*, vol. 21, no. 8, pp. 735–758, 2002.

[17] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: A survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 55–81, 2015.

[18] M. Saska, T. Baca, J. Thomas, J. Chudoba, L. Preucil, T. Krajnik, J. Faigl, G. Loianno, and V. Kumar, "System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization," *Auton. Robots*, vol. 41, no. 4, pp. 919–944, 2017.

[19] A. Cornejo and R. Nagpal, "Distributed range-based relative localization of robot swarms," in *Algorithmic Foundations of Robotics XI*. Cham, Switzerland: Springer, 2015, pp. 91–107.

[20] A. Prorok, A. Bahr, and A. Martinoli, "Low-cost collaborative localization for large-scale multi-robot systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 4236–4241.

[21] X. S. Zhou and S. I. Roumeliotis, "Determining the robot-to-robot 3D relative pose using combinations of range and bearing measurements (Part II)," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 4736–4743.

[22] X. S. Zhou and S. I. Roumeliotis, "Determining 3-D relative transformations for any combination of range and bearing measurements," *IEEE Trans. Robot.*, vol. 29, no. 2, pp. 458–474, Apr. 2013.

[23] D. Zarrouk and L. Yehezkel, "Rising STAR: A highly reconfigurable sprawl tuned robot," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1888–1895, Jul. 2018.

[24] N. Meiri and D. Zarrouk, "Flying STAR, a hybrid crawling and flying sprawl tuned robot," in *Proc. IEEE Int. Conf. Robot. Autom*, May 2019, pp. 5302–5308.

DANA EREZ received the B.Sc. degree (Hons.) from the Ben-Gurion University of the Negev, Israel, in 2018, where she is currently pursuing the M.Sc. degree with the Department of Mechanical Engineering. Her current research interests include mobile robots and control systems. She received multiple prizes for excellence during her studies and research including being listed on the Dean's List, an Excellence Scholarship from the Mechanical Engineering Department and from the Agricultural, Biological and Cognitive Robotics Center, BGU, and the Special Excellence Prize from the President of the University.

SHAI AROGETI (M'09) received the B.Sc. degree in mechanical engineering and the M.Sc. and Ph.D. degrees from the Ben-Gurion University of the Negev, Be'er Sheva, Israel, in 1997, 2000, and 2006, respectively. In 2006, he was a Research Fellow with the School of Electrical and Electronic Engineering, Nanyang Technological University, where he worked on the project Performance Monitoring, Diagnosis, and Prognosis. Since 2009, he has been a Senior Lecturer with the Department of Mechanical Engineering, Ben-Gurion University of the Negev, Israel. His current research interests include nonlinear and optimal control, and control applications in robotics and automotive systems.

DAVID ZARROUK received the B.Sc., M.Sc., and Ph.D. degrees from Technion, Israel, in 2005, 2008, and 2011, respectively. From 2011 to 2013, he was a Postdoctoral Fellow with the Biomimetic Millisystems Laboratory, UC, Berkeley. He is currently an Assistant Professor with the Department of Mechanical Engineering, Ben-Gurion University of the Negev. His current research interests include medical robotics, robotics in flexible and slippery surfaces interactions, biomimetics, and minimally actuated mechanisms. He received multiple prizes for excellence in research and teaching which include the Fulbright Postdoctoral Fellowship, in 2011, the Fulbright-Ilan Ramon Postdoctoral Fellowship for most prominent Israeli Fulbright Fellow, in 2011, the Hershel Rich Innovation Award, the Aharon and Ovadia Barazani Prize for Excellent Ph.D. Thesis, and the Alfred and Yehuda Weisman Prize for consistent excellence in teaching.