Motion-based Single-Anchor Localization in a Two-Robot System

Suleyman Ali
Supervisors: Ranga Rao Venkatesha Prasad, Ashutosh Simha, Suryansh Sharma
EEMCS, Delft University of Technology, The Netherlands

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Motion-based Single-Anchor Localization in a Two-Robot System

Suleyman Ali
Responsible Professor: Ranga Rao Venkatesha Prasad
Supervisors: Ashutosh Simha, Suryansh Sharma
EEMCS, Delft University of Technology, The Netherlands

***Abstract*—A majority of existing single-anchor localization algorithms make use of antenna arrays or special antenna systems. However, the need for specialized antenna systems incurs higher costs, complexity and power consumption. This paper presents a novel single-anchor localization algorithm, which does not require antenna arrays or special antennas. The algorithm is implemented in a two-robot system, where one of the robots acts as the anchor and the other acts as the target. The localization algorithm uses velocity measurements of each robot and distance measurements between robots. Using the change in distance between the robots and the velocity of the target robot relative to the anchor robot, the target robot can be localized relative to the anchor robot. The localization algorithm uses a Kalman filter as a state estimator of the movement of the target robot and a Savitzky-Golay filter to filter the distance measurements. A simulation with Gaussian noise in the distance measurements is performed to display the impact of the filtering methods. Another simulation showed an average localization error of 1.38 meters when the distance measurements are acquired through RSSI measurements, which is comparable to existing RSSI-based localization methods.**

***Index Terms*—Localization, single-anchor, Kalman filter, RSSI**

I. Introduction

Precise localization, one of the many challenging sub-tasks of multi-robot navigation, is a key technology to achieve coordination and control of swarm robots [16]. Specifically, localization concerns itself with estimating the location of robots in an environment, given perceived sensor data [14]. To achieve reliable and efficient collaboration in a multi-robot system, robots must be able to sense each other and interact with each other locally.

Literature shows many localization methodologies for localization with a known infrastructure in a well-designed environment, e.g., landmarks in the environment or a given map [10]. These methodologies can use anchors: nodes placed in the environment with a known global position. These anchor-based localization methods can be categorized into multi-anchor and single-anchor localization methods.

A conventional multi-anchor setup for localization uses at least three anchors and ranging sensors on the robots. The positions of the robots are calculated based on the distinct anchor-robot distances, using triangulation or multilateration techniques. However, specific issues remain in multi-anchor solutions: there is a relatively high setup cost in materials and human involvement, as found by Lymberopoulos and Liu [7]. The anchors have to be carefully deployed over the entire environment where localization is needed. This deployment overhead remains too high. Furthermore, the anchors' costs, power usage, and required maintenance shouldn't be neglected.

Single-anchor localization methods have also been explored in recent literature. A common scheme for single-anchor localization uses both angle-of-arrival (AOA) and time-of-arrival (TOA) measurements, where the anchor is equipped with a special antenna or antenna array to perform angular measurements. A majority of recent works make use of (multiple) directional antennas, or electronically steerable or switchable antenna systems [5][11][4][18]. Some works enable localization based on the exploitation of multipath reflections in an indoor environment [5][9]. The common trend in these works is the use of specialized equipment in the form of an antenna array or special antenna system. Unfortunately, the need for specialized antenna arrays makes these systems less flexible and incurs higher costs, complexity and power consumption [17]. Moreover, not all works explore scenarios where both the anchor and targets are mobile. Thus, exploring a single-anchor localization method that does not require specialized antennas or antenna arrays is warranted.

The objective of this paper is to introduce a single-anchor localization algorithm that can be implemented on a system without using antenna arrays or specialized antennas. This is achieved by relatively localizing a two-robot system, where one robot acts as a mobile anchor and the other acts as the target. Furthermore, the algorithm uses only one distance measurement between anchor and target, which can be implemented with any type of ranging sensor, and one velocity measurement for each robot, which, depending on the robot used, can be measured through varying approaches. The feasibility of this algorithm is put to trial by simulating the algorithm when there is noise present in the two previously mentioned measurements. Two experiments are performed: the first experiment discusses the impact of filters used to reduce errors in the proposed algorithm, and the second experiment simulates the proposed algorithm under a realistic noise profile, using RSSI-based distance measurements, and analyzes the performance.

The remainder of this paper proceeds as follows: first, the problem of interest to this paper is introduced. Second, a detailed explanation of the developed motion-based single-

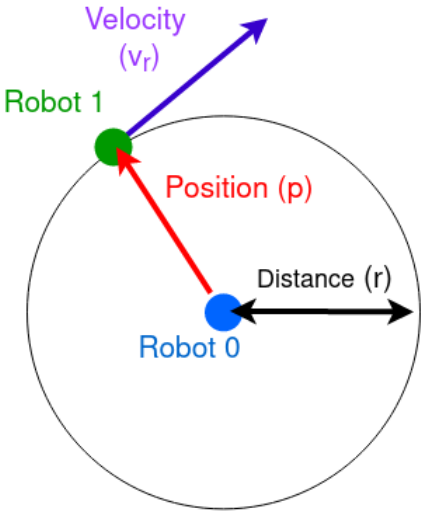

Fig. 1: A model of the problem setup. Given are two robots (R_0, R_1). Robot R_0 acts as a mobile anchor and R_1 is the target. Two given measurements are the velocity v_r of R_1 relative to R_0 and the distance r between the two robots. The desired result is the position p of R_1 relative to R_0 .

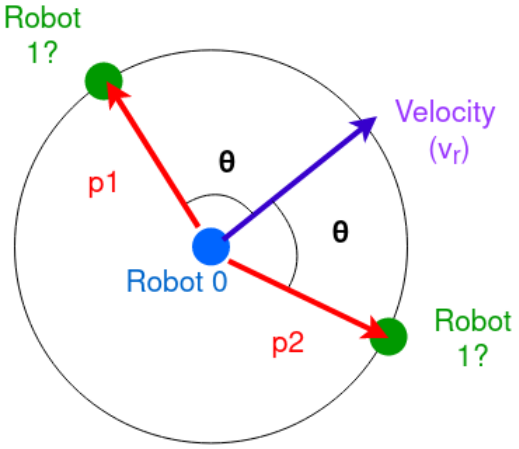

Fig. 2: A depiction of the position discrepancy when performing localization using the proposed localization algorithm. Note that the velocity $v_r(t)$ of R_1 relative to R_0 is drawn on the frame of R_0 for simplicity of drawing angles. Given only the angle $\theta(t)$ between the robots and the velocity $v_r(t)$, we find two possible locations for R_1. Thus, it is not possible to precisely localize R_1 given only the previously stated information.

anchor localization algorithm is given. Finally, the proposed algorithm is simulated, and the results are discussed.

II. Problem of Interest

Consider a system composed by two mobile robots $\{R_0, R_1\}$ in a two-dimensional Euclidian space. In this problem, R_0 will be referred to as the anchor robot and R_1 will be the target robot. The goal is to find the position of R_1 relative to R_0. We define this position as the vector $p(t)$ as depicted in Figure 1:

$$p(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix},$$

where $x(t)$ and $y(t)$ are the x and y coordinates respectively.

To achieve relative localization of R_1, it is assumed that the robots have certain equipment available to sense distance and velocity information in the system. The exact hardware implementations are considered out of the scope for this paper. First, assume R_0 has a practical method to measure the distance between R_0 and R_1. This distance is defined as $r(t)$, which is the norm of $p(t)$:

$$r(t) = ||p(t)|| = \sqrt{x(t)^2 + y(t)^2}$$

Second, R_1 can measure its own velocity and communicate this measurement to R_0. R_0 can measure its own velocity as well. The velocity of R_1 to relative to R_0 is defined as $v_r(t)$:

$$v_r(t) = v_1(t) - v_0(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix},$$

where $v_0(t)$ and $v_1(t)$ are the velocities of R_0 and R_1 respectively. A graphical depiction of the system can be found in Figure 1.

III. Motion-based Single-Anchor Localization Algorithm

This section describes the developed motion-based single-anchor localization algorithm under the circumstances discussed in the previous section. In order to localize R_1 relative to R_0, the angle $\theta(t)$ between $v_r(t)$ and $p(t)$ can be calculated. The equation for $\theta(t)$ is given below (see Appendix A for the full derivation):

$$\theta(t) = \arccos \frac{\dot{r}(t)}{||v_r(t)||} \tag{1}$$

Thus, to calculate the angle $\theta(t)$, the change in distance between the robots and the velocity of R_1 relative to R_0 is utilized.

However, given only the angle $\theta(t)$ and the velocity $v_r(t)$, it is not possible to precisely localize R_1 relative to R_0: the result is two possible positions for R_1. This is due to the fact that $p(t)$ could be on either side of $v_r(t)$ with an angle of $\theta(t)$. This issue will be called the position discrepancy in the remainder of this section and is demonstrated in Figure 2. The equation for finding the position $p(t)$ of R_1 relative to R_0 now becomes:

$$p(t) = \begin{bmatrix} r(t)\cos\left(\alpha(t) \pm \theta(t)\right) \\ r(t)\sin\left(\alpha(t) \pm \theta(t)\right) \end{bmatrix}, \tag{2}$$

where $\alpha(t)$ denotes the angle of $v_r(t)$:

$$\alpha(t) = \arctan \frac{\dot{y}(t)}{\dot{x}(t)},$$

and the $\pm$ signifies the discrepancy in the position $p(t)$ as discussed earlier.

Two methods of eliminating this position discrepancy are discussed in the following subsections. The first method as-

sumes a known initial position of R_1, and the second assumes the initial position is unknown.

Case 1 - Localization with a known initial position

After calculating the two possible positions using the localization algorithm, the simplest way to select one of the two positions is by picking the position closest to the previous position. This method is only achievable if the initial position is known. Moreover, it's the simplest way of eliminating the position discrepancy. However, the need for an initial position might reduce the use of the algorithm in practice.

Case 2 - Localization with an unknown initial position

The second scenario of the position discrepancy is one where the initial position is unknown. Due to the absence of an initial position, the simple algorithm described in the previous subsection cannot be utilized. In this case, to precisely localize R_1, we need to make an assumption about the velocity $v_r(t)$. We assume that at some point during the movement of the two robots, there is a significant change in the direction of $v_r(t)$. At the exact moment this change in direction happens, we can select one of the two possible positions and thus precisely localize R_1 and determine future positions. This process is demonstrated in Figure 3. In this figure, two consecutive time steps of the movement of the robots are shown.

In the first time step, the angle $\theta(0)$ between $v_r(0)$ and $p(0)$ has been calculated and the two possible positions $p_{0,true}$ and $p_{0,false}$ are shown, which correspond to the real and wrong position of R_1 in time step 0 respectively. From the information known in just this time step, it is not possible to localize R_1.

In the second time step, the direction of the velocity vector has changed. Again, we can calculate $\theta(1)$ and use it to acquire two possible positions for R_1: $p_1, true$ and $p_1, false$, which correspond to the real and wrong position of R_1 in time step 1 respectively. Now, if we calculate the change in position between the pairs of positions in the first time step and the second time step, we observe that the change in position between $p_0, true$ and $p_1, true$ is the smallest. The changes in position between all the other possible pairs are significantly larger than the change in position of the true positions. Thus, when a change in velocity direction occurs in consecutive time steps, by picking the pair of consecutive possible positions with the smallest change in position, R_1 can be precisely localized.

A cosine similarity metric quantifies the change in the direction of velocity $v_r(t)$. The cosine similarity score of two consequent velocity vectors $v_r(t)$ and $v_r(t-1)$ is computed and compared to a threshold. If the cosine similarity score is smaller than the threshold, we pick the pair of consecutive possible positions with the smallest change in position. The threshold depends on how big we want the change of direction to be, which should be decided by the amount of noise in the measurement of velocity $v_r(t)$.

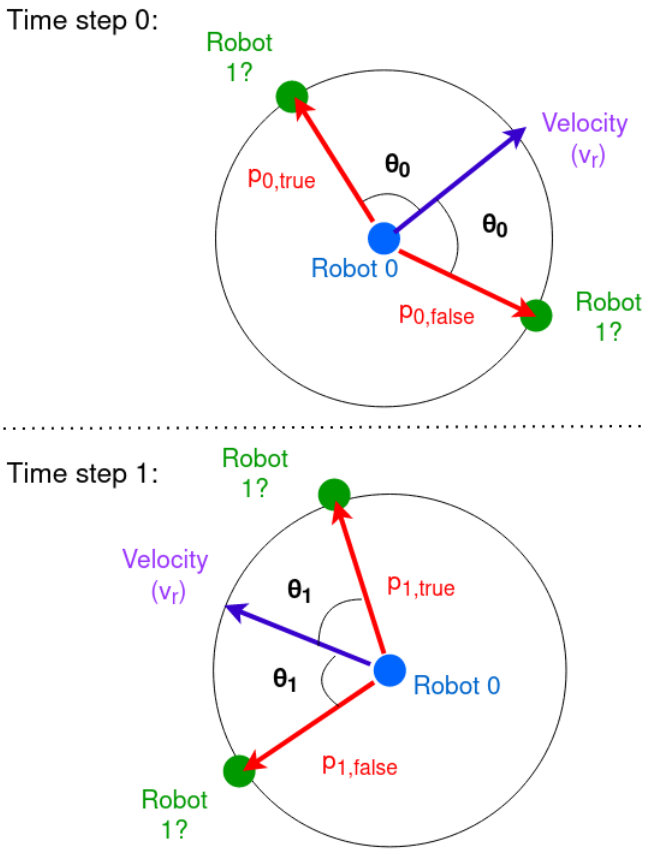

Fig. 3: Two consecutive time steps of the localization algorithm when the initial position is unknown. In each time step, the possible positions are drawn using $\theta(t)$. When there is a change in direction of velocity $v_r(t)$, R_1 can be localized by comparing pairs of possible positions between time steps. We can observe that the combination $p_{0,true}$ and $p_{1,true}$ has the lowest change in position between the time steps. The pair with the smallest change in position will be the pair of true positions of R_1.

In both cases, assumptions are made: the first case assumes the initial position is known, and the second case assumes there is some change of direction in $v_r(t)$. In the remainder of this paper, we will consider the second case. This case could be of more use in practice if the change of direction is relatively accurate.

IV. Reducing the Impact of Noise

The localization algorithm is based on mobile robots, which make measurements of physical data: the velocity $v_r(t)$ and distance $r(t)$. In the physical world, both these measurements will contain noise. We assume that these measurements contain zero-mean Gaussian noise. Further details on the exact measurement noise can be found in section V. The real challenge of the algorithm is to localize accurately despite having noisy measurements. The following subsections will introduce techniques which were utilized to maintain accurate localization with noisy measurements.

A. Kalman Filter

In this work, we have chosen to use the Kalman filter to track the target robot's position accurately. The Kalman filter is an adaptive filter that estimates the state of a linear system and has been widely used for many applications, of which mobile robot navigation is one [15].

1) Kinematic model: The state of the target robot is modeled according to the Constant Velocity (CV) model. This model has been used in many applications due to its relative simplicity, versatility and effectiveness [12].

The state vector is modeled in terms of position and velocity in a two-dimensional space:

$$X_k = \begin{bmatrix} x_k & y_k & \dot{x}_k & \dot{y}_k \end{bmatrix}^T$$

The general kinematic state model is given by:

$$X_{k+1} = FX_k + \omega_k,$$

with the transition matrix F:

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

where T represents the time interval for each step and ω_k is the process noise which is assumed to be zero-mean Gaussian noise with covariance matrix Q. The selected covariance matrix Q is the most commonly used random acceleration process noise for conventional tracking systems, given by:

$$Q = \begin{bmatrix} T^4/4 & 0 & T^3/2 & 0 \\ 0 & T^4/4 & 0 & T^3/2 \\ T^3/2 & 0 & T^2 & 0 \\ 0 & T^3/2 & 0 & T^2 \end{bmatrix} \sigma_q^2,$$

where σ_q is a tuning parameter that is chosen empirically based on the assumed motion of the target, as is often done in numerous studies [12]. The choice of σ_q is important, as it directly impacts the performance of the Kalman filter.

2) Measurement Model: The measurements are modeled as:

$$z_k = HX_k + n_k,$$

where z_k is the measurement vector, H is the measurement matrix and n_k is the measurement noise, which is assumed to be zero-mean Gaussian noise with covariance matrix R. The measurement vector z_k and measurement matrix H are given by:

$$z_k = \begin{bmatrix} \bar{x}_k & \bar{y}_k & \bar{v}_k^x & \bar{v}_k^y \end{bmatrix}^T \qquad H = I,$$

where $\bar{x}_k$ and $\bar{y}_k$ are coordinates calculated with equation 2, as explained in section III, and $\bar{v}_k^x$ and $\bar{v}_k^y$ are measurements of velocity $v_r(t)$.

The covariance matrix R of the measurements is formed from the error in $\bar{x}_k$ and $\bar{y}_k$ and the variance of noise in the measurements of velocity $v_r(t)$. The error in $\bar{x}_k$ and $\bar{y}_k$ is acquired statistically by performing over 1000 Monte Carlo runs of localization and calculating the mean squared error in each position measurement. The matrix contains both diagonal and non-diagonal elements, because the noise in the x and y measurements are correlated due to both of them being computed with the same $r(t)$ and $v_r(t)$. The measurement covariance matrix is given by:

$$R = \begin{bmatrix} E[n_x^2] & E[n_x n_y] & 0 & 0 \\ E[n_x n_y] & E[n_y^2] & 0 & 0 \\ 0 & 0 & \sigma_v^2 & 0 \\ 0 & 0 & 0 & \sigma_v^2 \end{bmatrix},$$

where n_x and n_y are the errors of the measured positions in the x and y axis respectively and σ_v^2 is the variance of the noise in the measurement of velocity $v_r(t)$.

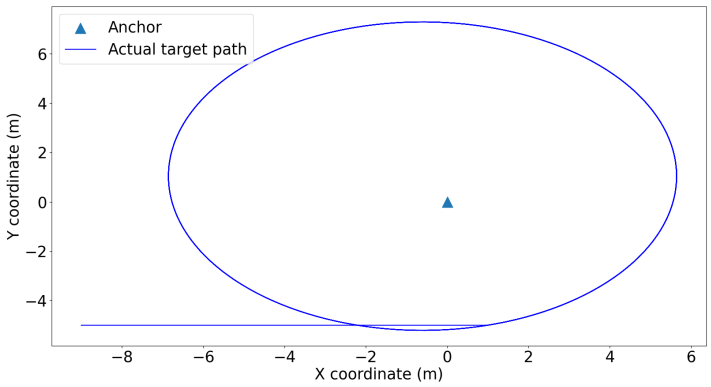

Fig. 4: Path of the target robot during the simulations. The target robot starts at the coordinate [-9, -5] and runs in a straight line until [1, -5], where it makes a change in direction, turning into a circular track and running for 5 laps.

B. Filtering $\dot{r}(t)$

The proposed algorithm makes use of $\dot{r}(t)$ when calculating $\theta(t)$, as seen in equation 1. In our system, there is noise in the measurement of $r(t)$. By differentiating the measurement $r(t)$, we amplify the noise greatly. If we want accurate localization, we need to filter this noise. For this filtering, we make use of a Savitzky-Golay filter [13]. This filter is a popular method of smoothing data and is based on fitting a polynomial to the data. Furthermore, it's commonly used to calculate derivatives of noisy data. Thus, this filter can be used to calculate a filtered $\dot{r}(t)$.

V. Performance Evaluation

The evaluation of the proposed algorithm is performed by simulations written in Python. The target robot is localized using the developed localization algorithm for a certain path, and accuracy is judged based on the root-mean-square error (RMSE) of the position estimate. The following subsections describe the simulation setup and results.

A. Simulation Setup

The experiments are split into two parts based on the type of noise added to the distance measurement $r(t)$. Both experiments assume a static anchor robot, an unknown initial position of the target robot and velocity $v_r(t)$ measurements with zero-mean Gaussian noise with $\sigma_v = 0.1$ meters. The robot runs at a speed of 1 m/s and takes measurements at every 0.5 meters traveled (T = 0.5). The path of the target robot is given in Figure 4. The target robot starts at the point [-4, -5] and runs in a straight line until around the point [0, -5], where it changes direction and begins turning for five laps around the track. This change in direction is needed for localization using the developed algorithm (with an unknown initial position).

The first part of the experiments is performed to evaluate how noisy measurements affect the localization algorithm and how the filtering techniques discussed in the previous section improve the position estimates. In this experiment, there is zero-mean Gaussian noise with $\sigma_r = 1.0$ meter added directly to the distance measurement $r(t)$.

For the second part of the experiments, the goal is to get a view of the performance of the proposed algorithm under realistic noise circumstances. To achieve this, an implementation of the localization algorithm using RSSI to measure the distance $r(t)$ is simulated. RSSI is a popular choice for distance measurement techniques due to its low cost, easy implementation, and low power consumption [2]. Furthermore, it fits the use case for our proposed algorithm since RSS measurements can be made without the use of antenna arrays or special antenna systems. Unfortunately, RSSI values are often extremely sensitive to environmental dynamics, such as multipath propagation, environmental noise and transmission conditions.

To express the relation between distance and RSSI, we use a lognormal shadowing path loss model, which is the most popular channel model for RSSI-based localization [2]. The model is given by:

$$RSSI = P_0 - 10\eta \log d + \sigma_{RSS},$$

where P_0 is the reference power at 1 meter from the anchor, η is the path loss exponent, d is the distance between anchor and target, and σ_{RSS} is a zero-mean Gaussian random variable, which denotes a measurement noise.

For the setup of our simulation, we have taken channel parameters from an estimation of channel parameters for an environment described in [1], containing obstacles such as tables and computers: $P_0 = -52$ [dBm], $\eta = 1.8$ and $\sigma_{RSS} = 5.8$ [dB].

Before the RSSI measurements are converted to distances, they are preprocessed with a Simple Moving Average filter, which positively impacts the filtering of raw RSSI data [6].

B. Results

1) Impact of noise reduction techniques: In this part of the results, we first show a simulation run with no noise in either the velocity or distance measurement. After that, we add Gaussian noise to the measurements and discuss the effectiveness of the filtering techniques covered in section IV.

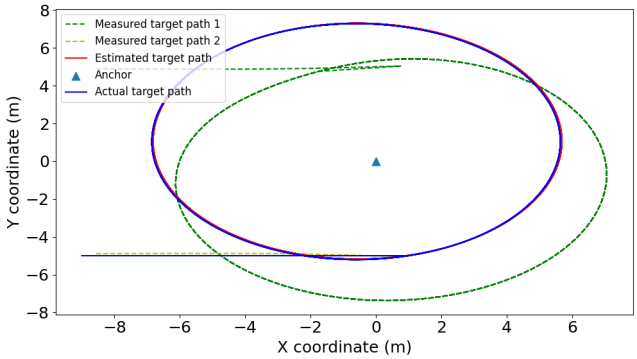

Fig. 5: Noiseless run of the simulation.

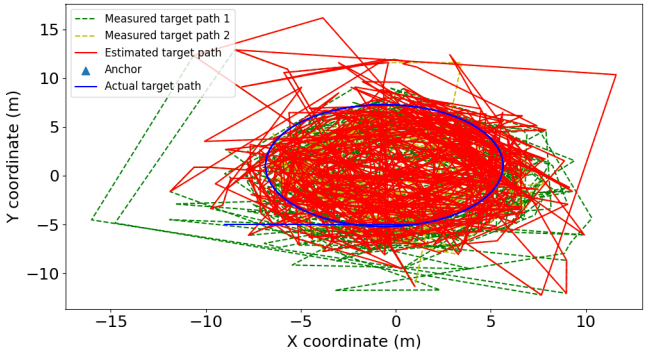

Fig. 6: Simulation with Gaussian noise added to measurements ($\sigma_r = 1.0$, $\sigma_v = 0.1$).

A noiseless run of the simulation is given in Figure 5. We can see that the estimated target path and actual target path overlap. As expected, there is no position estimate before the target robot changed direction onto the circular path. The figure also shows the two possible positions calculated with the measurements for each time step. Note that both of the measured positions for all time steps are drawn in the figures, though, in reality, only one measured position is selected (after a change in direction, as explained in section III). In the case of this path, one of these measured paths overlaps with the actual path, and the other is inaccurate. It should be noted that not in all paths the actual robot path is given by only one of the measured paths: the real path can be composed of a combination of both measured paths.

A simulation run with noise added to both measurements is given in Figure 6. In this case, no Kalman filter or filtering on $\dot{r}(t)$ has been done. The measurements are extremely chaotic, and no information can be extracted from the figure: the RMSE of the estimate is 8.51 meters.

Another run with noisy measurements is shown in Figure 7. However, in this case, the state estimation from the Kalman filter is added. We can observe that the localization

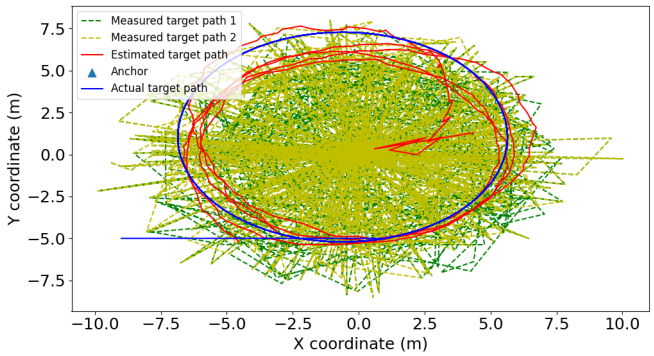

Fig. 7: Simulation with Gaussian noise added to measurements, using Kalman filter ($\sigma_r = 1.0$, $\sigma_v = 0.1$).

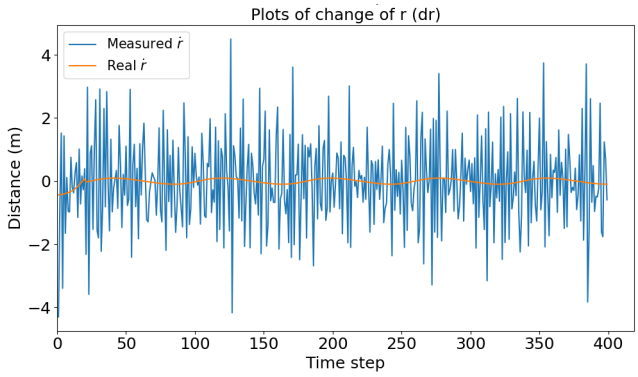

Fig. 8: The change in distance $\dot{r}(t)$ without filtering the distance measurements (same simulation as Figure 7).

is performed more accurately than in the previous case: the RMSE of the position estimation is 2.286 meters. In this case, the accuracy of the estimation is mostly due to the relatively accurate velocity measurements. This can be concluded due to the fact that the measurements can still be seen chaotically jumping around the space: the RMSE of the measured positions is roughly 7.94 meters.

The key to reducing the inaccuracy in the position measurements is by filtering the distance measurements.

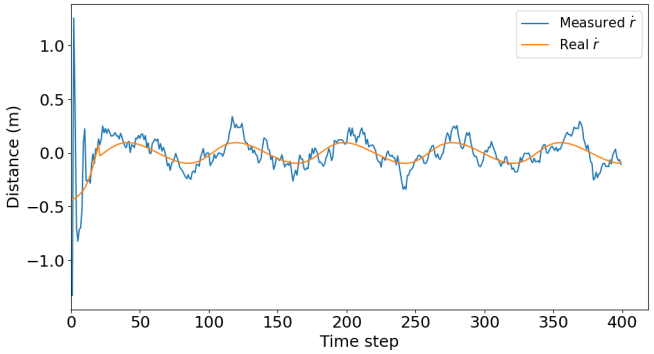

Fig. 9: The change in distance $\dot{r}(t)$ after filtering, using the Savitzky-Golay filter.

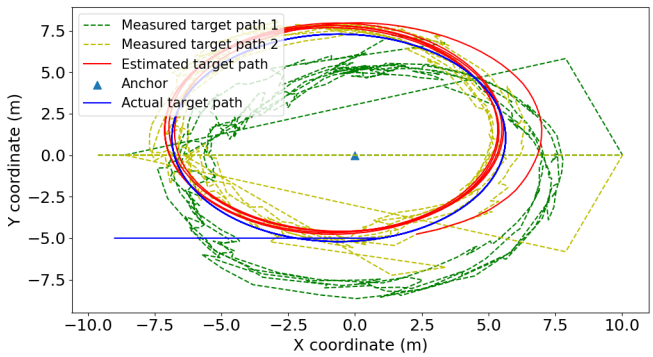

Fig. 10: Simulation with Gaussian noise added to the measurements, combining Kalman filter and Savitzky-Golay filter to filter the distance measurements ($\sigma_r = 1.0$, $\sigma_v = 0.1$).

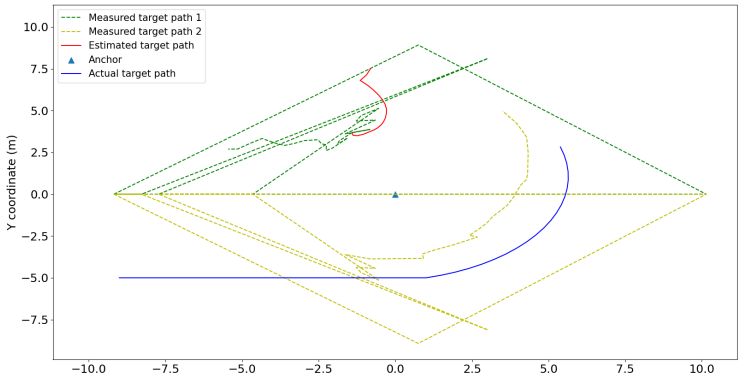

Fig. 11: A part of a noisy simulation run, using the Kalman filter and Savitzky-Golay filter. Even though $\dot{r}(t)$ has been filtered, we still initially localize incorrectly when a change of direction occurs.

Figure 8 plots $\dot{r}(t)$ over time in the simulation case of Figure 7, where we used a Kalman filter and unfiltered distance measurements. This figure showcases the high noise in the distance measurements. Using the Savitzky-Golay filter, we can obtain a much smoother $\dot{r}(t)$ signal, as shown in Figure 9. Combining all these techniques, we can use the developed localization algorithm to localize even more accurately. Figure 10 shows localization using a Kalman filter and a filtered $\dot{r}(t)$. Comparing this figure to 7, we can notice that the measured positions are a lot less chaotic: they are spaced around in a circular manner, as expected. Naturally, the filtered $\dot{r}(t)$ signal is not perfect, so there will still be some noise in the measured positions. In this case, the RMSE of the measured positions is 1.41 meters, and the RMSE of the estimated positions is 0.61 meters. Thus, the combination of a Kalman filter and Savitzky-Golay filter results in the best performance and the filtering of $\dot{r}(t)$ is key to obtaining relatively accurate measured positions.

However, there are still a few problems in this process. The first problem arises due to noise in the $\dot{r}(t)$ curve. As stated in the previous paragraphs, when the noise in $\dot{r}(t)$ is high, the measured positions become chaotic and jump around the space. This noise also causes the initial localization, when

Fig. 12: The error in estimated position over time. In this case, the algorithm initially localized to the incorrect measured position (same scenario as Figure 11). The error in estimated position gradually decreases over time and converges.

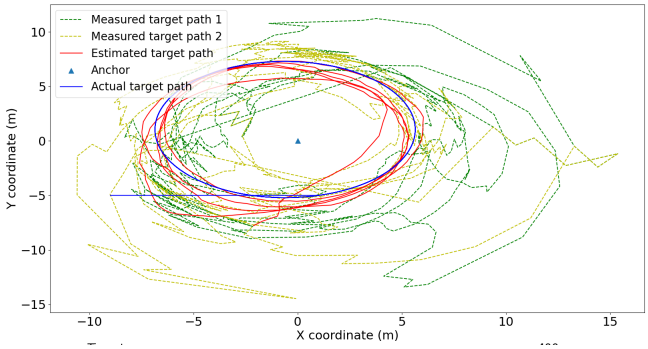

Fig. 13: Simulation using RSSI for distance measurements, according to the lognormal shadowing path loss model described in section V-A

there is a change in the direction of velocity, to be incorrect: this can be seen in Figure 7. The problem is that this incorrect initial localization can still happen even though we filter $\dot{r}(t)$. This can be seen in Figure 11. In the case of our robot path, this isn't a big problem since it is a track with many changes in the direction of velocity: the localization starts with a big error in position estimate, but gradually decreases this error and converges, as shown in Figure 12. However, this could lead to severe localization inaccuracies on more straight robot paths.

The second problem is related to issues with the algorithm itself. In equation 2, we calculate the position using an angle-based approach: $r(t)\cos\gamma$. When $r(t)$ increases and there is a constant error in γ, the error in the position gets amplified by the increasing $r(t)$. The same problem is present in our algorithm. Even if the noise in $\theta(t)$ is small and constant after filtering, the more $r(t)$ increases, the more noisy the measurements will be. The current Kalman filter implementation does not take this problem into account and will have to be improved.

2) RSSI-based distance measurements: This part of the experiment simulates distance measurements acquired through RSSI measurements. To model the RSSI channel, we used a lognormal shadowing path loss model and parameters described in section V-A. The simulation result is shown in Figure 13. Compared to the simulation in Figure 10, we can see the estimation is noisier, due to noisier $r(t)$ measurements.

The RMSE of the estimated position in Figure 13 is 1.51 meters, and the RMSE of the estimated position after running the simulation 1000 times is 2.3 meters. This increase in RMSE is due to the fact that the algorithm occasionally initially localizes to the incorrect measured position and then converges to the correct measurement, similar to Figure 12. If we consider only the RMSE after convergence, we get an error of 1.38 meters. This performance aligns with other RSSI-based localization systems, which typically have an error of 1-5m, depending on the environment and experiment parameters [3]. In [8], RSSI measurements from multiple anchors are fused with IMU measurements to localize a mobile unit. This approach is similar to the method described in our simulation, since it localizes a mobile target using RSSI measurements and the motion of the target. The result is an RMSE in an estimated position of 1.28 meters, which is similar to the result obtained in our simulation. Thus, the proposed algorithm seems like a promising technique for RSSI-based single-anchor localization.

VI. Conclusions

In this paper, we proposed a novel motion-based single-anchor localization algorithm. The proposed algorithm uses the distance measurement $r(t)$ between the anchor and target and the velocity measurement $v_r(t)$ of the target relative to the anchor. The algorithm is implemented in a two-robot system, where one of the robots acts as an anchor and the other acts as the target. To localize accurately when these two measurements are noisy, a Kalman filter was used as a state estimator, and a Savitzky-Golay filter was used to obtain more accurate $r(t)$ and $\dot{r}(t)$ measurements. Simulations showed the positive impact that the Kalman filter and Savitzky-Golay filter had. In particular, to get accurate measured positions from the proposed algorithm, it is essential to have precise $\dot{r}(t)$ measurements. Another experiment simulated distance measurements acquired through RSSI, using a lognormal shadowing path loss model. This experiment resulted in an RMSE of 1.38 meters, which could make the proposed algorithm promising for RSSI-based single-anchor localization. However, the results could vary in a real-life simulation, depending on the environment, the severity of multipath propagation, model inaccuracies or measurement errors.

Future works can expand upon this work in many ways. First of all, different target robot paths can be explored, including paths where both the anchor and target are moving. Second, the system can be expanded to a system with more than two robots. Third and most important, studies can be done on improving the filtering of $r(t)$ and $\dot{r}(t)$, since this is key to increasing the measured positions obtained from the proposed algorithm. Finally, the system can be implemented on hardware, and the performance can be analyzed.

VII. Reproducibility and Integrity

In this section we reflect upon the reproducibility and integrity of our work. Two factors were maintained to ensure the reproducibility and integrity of our work. First, in this paper, the proposed algorithm and all the necessary models were described with as clearly as possible in Sections III and IV. This allows readers to implement a similar system themselves, if they should wish so. Second, the code repository[1] for the simulations in this paper is available publicly. This code is made publicly available to allow future researchers to start from where we left off and evaluate the exact parameters used in the simulations. Due to the random variables involved in the experiments, the simulations won't be identical to the results in this paper in every case, however results will converge to the results found in this paper.

Appendix

A. Derivation of $\theta(t)$

Consider two robots $\{R_0, R_1\}$. Let $p(t)$ be the vector of the position of R_1 relative to R_0 and $v(t)$ the velocity of R_1 relative to R_0 (see Figure 1):

$$p(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} \quad v(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix}$$

We define $r(t)$ as the norm of $p(t)$, which is the distance between the the two robots:

$$r(t) = ||p(t)|| = \sqrt{x(t)^2 + y(t)^2}$$

Now we derive the time derivative of $r(t)$:

$$\begin{aligned} \dot{r}(t) &= \frac{x(t)\dot{x}(t) + y(t)\dot{y}(t)}{\sqrt{x(t)^2 + y(t)^2}} \\ &= \frac{p(t) \cdot v(t)}{\sqrt{x(t)^2 + y(t)^2}} \\ &= \frac{p(t) \cdot v(t)}{r(t)} \end{aligned}$$

Consider the two unit vectors $\hat{p}(t)$ and $\hat{v}(t)$:

$$\hat{p}(t) = \frac{p(t)}{||p(t)||} = \frac{p(t)}{r(t)} \quad \hat{v}(t) = \frac{v(t)}{||v(t)||}$$

Using these unit vectors, we can rewrite $\dot{r}(t)$ into a form containing $\theta(t)$, which is the angle between $v(t)$ and $p(t)$:

$$\begin{aligned} \dot{r}(t) &= \frac{||v(t)||r(t) * \hat{p}(t) \cdot \hat{v}(t)}{r(t)} \\ &= ||v(t)|| * \hat{p}(t) \cdot \hat{v}(t) \\ &= ||v(t)|| \cos(\theta(t)) \end{aligned}$$

This gives the formula for $\theta(t)$:

$$\theta(t) = \arccos \frac{\dot{r}(t)}{||v(t)||}$$

[1] https://github.com/suleyman222/motion-based-single-anchor-localization

References

[1] Melih Doğan. Indoor localization and tracking based on rssi and accelerometer measurements. Master's thesis, Middle East Technical University, 2015.

[2] Jinze Du, Chao Yuan, Min Yue, and Tao Ma. A novel localization algorithm based on rssi and multilateration for indoor environments. *Electronics*, 11(2):289, 2022.

[3] Md. Osman Gani, Casey OBrien, Sheikh I. Ahamed, and Roger O. Smith. Rssi based indoor localization for smartphone using fixed and mobile wireless node. In *2013 IEEE 37th Annual Computer Software and Applications Conference*, pages 110–117, 2013.

[4] Gianni Giorgetti, Alessandro Cidronali, Sandeep K.S. Gupta, and Gianfranco Manes. Single-anchor indoor localization using a switched-beam antenna. *IEEE Communications Letters*, 13(1):58–60, 2009.

[5] Bernhard Großwindhager, Michael Rath, Josef Kulmer, Mustafa S. Bakr, Carlo Alberto Boano, Klaus Witrisal, and Kay Römer. Salma: Uwb-based single-anchor localization system using multipath assistance. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, SenSys '18, page 132–144, New York, NY, USA, 2018. Association for Computing Machinery.

[6] Moses A. Koledoye, Daniele De Martini, Simone Rigoni, and Tullio Facchinetti. A comparison of rssi filtering techniques for range-based localization. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 761–767, 2018.

[7] Dimitrios Lymberopoulos and Jie Liu. The microsoft indoor localization competition: Experiences and lessons learned. *IEEE Signal Processing Magazine*, 34(5):125–140, 2017.

[8] Veerachai Malyavej and Prakasit Udomthanatheera. Rssi/imu sensor fusion-based localization using unscented kalman filter. In *The 20th Asia-Pacific Conference on Communication (APCC2014)*, pages 227–232, 2014.

[9] Paul Meissner and Klaus Witrisal. Multipath-assisted single-anchor indoor localization in an office environment. In *2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 22–25, 2012.

[10] Prabin Panigrahi and Sukant Bisoy. Localization strategies for autonomous mobile robots: A review. *Journal of King Saud University - Computer and Information Sciences*, 03 2021.

[11] M. Rzymowski, P. Woznica, and L. Kulas. Single-anchor indoor localization using espar antenna. *IEEE Antennas and Wireless Propagation Letters*, 15:1183–1186, 2016.

[12] Kenshi Saho. Kalman filter for moving object tracking: Performance analysis and filter design. In *Kalman Filters - Theory for Advanced Applications*. InTech, February 2018.

[13] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

[14] Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to Autonomous Mobile Robots*. Intelligent Robotics and Autonomous Agents series. MIT Press, London, England, 2 edition, February 2011.

[15] Caius Suliman, Cristina Cruceru, and Florin Moldoveanu. Mobile robot position estimation using the kalman filter. 2010.

[16] Han Wu, Shizhen Qu, Dongdong Xu, and Chunlin Chen. Precise localization and formation control of swarm robots via wireless sensor networks. *Mathematical Problems in Engineering*, 2014:1–12, 2014.

[17] S A Zekavat. *Handbook of position location - theory, practice, and advances, second edition*. IEEE Series on Digital & Mobile Communication. Wiley-Blackwell, Hoboken, NJ, 2 edition, April 2019.

[18] Yanbin Zou and Qun Wan. Emitter source localization using time-of-arrival measurements from single moving receiver. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3444–3448, 2017.