Range-based Collision Prediction for Dynamic Motion

Aarti Singh
Electrical and Systems Engineering
Washington University in St. Louis, USA
aartisingh@wustl.edu

Neal Patwari
McKelvey School of Engineering
Washington University in St. Louis, USA
npatwari@wustl.edu

***Abstract*—Real-time proximity and collision detection via radio distance measurements has application in smart-helmets, drones, autonomous vehicles, and social distancing. In this paper we present a range-based, infrastructure-free, distributed algorithm that utilizes inter-robot range data and intra-robot acceleration data to estimate each robot's recent positions. As the next step, a collision prediction scheme is proposed that uses the derived relative kinematics of each robot to predict an impending collision between any pair of robot. The algorithm is tested and validated with the help of "measurement trace-based simulation of motion involving acceleration.**

***Index Terms*—Collision prediction, multidimensional scaling, autonomous vehicles**

I. Introduction

Autonomous and real-time collision prediction and collision avoidance is crucial in a world filled with multiple mobile entities operating in the close vicinity of each other. Collisions, which do happen [1], are both life-threatening and expensive. GPS and lidar are insufficient to reliably predict the collision of small objects moving quickly towards each other, e.g., multiple drones, or a smart helmet and a baseball. In many cases, it will be possible to add a radio frequency (RF) tag to robots, vehicles, or objects that need to monitor to avoid collisions. However, RF tag localization systems are insufficient to predict collisions because they do not predict future positions, and they require a fixed, known-location infrastructure which may not be present or may be too inconvenient or expensive to deploy for the application. Finally, using coordinates as an intermediate step in the process of predicting collisions is sub-optimal [11]. We argue that, fundamentally, collision prediction from range measurements should be distributed. In this paper, we present a method to address this gap by enabling mobile agents of any size or speed to predict impending collisions without relying on a centralized infrastructure.

A known-location infrastructure (or anchors) is usually utilized for obtaining the absolute coordinate knowledge of the mobile objects as is done in [2], however, their availability may not be always possible. In addition of deployment and maintenance costs, the infrastructure can be a single point of failure. In cases where GPS is used to know the locations of the infrastructure, it will be unavailable indoors or in tunnels for example. Moreover, deploying a local positioning system infrastructure is inappropriate when robots must operate globally or in emergency situations e.g. first-responder drones or self-driving vehicles. Fortunately, collision between two objects does not require the coordinates of each object in a global coordinate system because the collision between two objects is a matter of their relative kinematics, such as their relative position, velocity, and acceleration. The perspective of this paper is that collision prediction should be local, distributed, and relative, freeing us from requiring an infrastructure or a global reference.

A popular approach to obtain relative positions is multidimensional scaling (MDS) [3]. In MDS, 'dissimilarities' between each pair of objects (e.g., ranges between robots in our case) are mapped into a low dimensional (2D or 3D) relative coordinate as an output. The distances between robots are preserved as much as possible and a relative position mapping of the robots is formed without the use of known-location infrastructure. However, MDS is a centralized algorithm as it requires a all dissimilarities to be known by one processing unit. For N robots, classical MDS has a computational complexity of $O(N^3)$. A distributed method of estimating location is proposed in [4], is but implemented with infrastructure. Based on the same work, another approach is presented to obtain a relative map of objects in motion, which although does not require known-location infrastructure, but is centralized in its implementation [5]. However,in order to achieve true autonomy and independence from a centralized decision maker, collision detection and prediction decision should be made in a decentralized manner by each device.

Another challenge with using MDS to generate kinematics over a time period is that since there is no fixed frame of reference, the generated map can undergo random translation, rotation, and flip. Therefore, without infrastructure, successive application of MDS over time is going to provide incorrect kinematics. A modification of classical MDS such that a common frame of reference is maintained for position and higher order kinematics (velocity and acceleration) are obtained is implemented in [6]. Using higher order derivatives of squared distance measurements, the relative kinematics are estimated. However, this method s highly sensitive to noise in range measurements. However, in order to predict collision we need relative kinematics which are tolerant to noisy measurements.

We present a robust, decentralized, infrastructure-less algorithm that produces high quality estimates of the relative kinematics of robots by using noisy inter-node range measurements and intra-node acceleration data. With the estimated kinematics, this distributed scheme predicts the future trajectory without requiring known-location infrastructure and any impending collision. In addition to the decentralized and infrastructure-free approach, our solution can be complementary to other widely adopted technologies for collision prediction. These technologies involving either active sensors such as lidar [7], radar [8], or passive sensors [9] such as cameras, are dependent on size, shape, reflective properties, luminescence of, and distance between the objects involved in the collision. The algorithm presented in this paper is free of such limitations involving physical properties of the objects.

II. Problem Statement

We take a network of N unknown-location nodes in a D-dimensional space. Over a period of time, node i collects pairwise range measurements between itself and its neighbors $\mathcal{N}_i$, and we use $\delta_{i,j}^t$ for $j \in \mathcal{N}_i$ at time t. A real-world scenario is considered in which nodes move with arbitrary motion. The problems we explore include:

1) estimation of the coordinates $\boldsymbol{x}_i^t$ for $i = 1, \dots, N$ and $t = 1, \dots, T$, where $\boldsymbol{x}_i \in \mathcal{R}^D$ given pairwise range measurements $\{\delta_{i,j}^t\}$ and individual acceleration measurements, $\boldsymbol{\alpha}_i^t$, both taken over the time window $t = 1, \dots, T$;
2) predicting an impending collision between i and any neighbor node in $\mathcal{N}_i$ at a time soon after $t = T$.

We assume that the primary goal of the system is to predict collisions, but in the case of an impending collision, the recent positions may be useful for the system reaction, for example, to know which direction to swerve.

III. Proposed Algorithm

To achieve the goals we articulate in the Introduction, in this section we formulate a new cost function based on errors between measured and calculated ranges and acceleration over time for all nodes. Our insight is that our formulation allows for a distributed, low computational complexity algorithm in which each device estimates its recent positions locally. It minimizes its local cost function and broadcasts its position estimates to its neighbors. Each node successively refines its recent position estimates based on its range and acceleration measurements in addition to the most recent received position estimates from its neighbors. This distributed nature ensures a decrease in the local cost functions, which contribute additively. Thus each sensor contributes to the minimization of the global cost function. Further, the local optimization step is low complexity because it is based on finding the minimum of a quadratic expression. Finally, the distributed optimization is guaranteed to converge, because it is using majorization approach which guarantees each round's global cost is non-increasing. Our approach follows the *scaling by majorizing a complicated function* (SMACOF) [10] approach, but expands it to enable simultaneous estimation of multiple recent positions, and for use of acceleration measurements in the cost function. After we present our algorithm for recent position estimation, we then present in Section III-D how these estimates are used to extrapolate and predict collisions.

A. Proposed Cost Function

Consider N mobile nodes with their position at time t represented as $X^t = [\boldsymbol{x}_1^t, .., \boldsymbol{x}_N^t]^T$. Our global cost function is:

$$\begin{aligned} S = &\sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \sum_{t=1}^{T} w_{i,j}^t \left[\delta_{i,j}^t - d_{i,j}(X^t)\right]^2 \\ &+ \sum_{t=1}^{T} r_i \left[\boldsymbol{\alpha}_i^t - \boldsymbol{a}_i(\boldsymbol{x}_i^t)\right]^2, \end{aligned} \tag{1}$$

where the first term represents the error between measured distances $\delta_{i,j}^t$ and the actual distances based on location coordinates $d_{i,j}(X^t)$, which are calculated as,

$$d_{i,j}(X^t) = \|\boldsymbol{x}_i^t - \boldsymbol{x}_j^t\| = \sqrt{(\boldsymbol{x}_i^t - \boldsymbol{x}_j^t)^T(\boldsymbol{x}_i^t - \boldsymbol{x}_j^t)}. \tag{2}$$

Whenever a node's velocity changes, its non-zero acceleration is measured by its accelerometer. We incorporate this extra information in the latter part of the sum, representing the error between the measured acceleration $\boldsymbol{\alpha}_i^t$ and acceleration $\boldsymbol{a}_i^t$ calculated from the coordinate path as follows:

$$\begin{aligned} \boldsymbol{a}_i(\boldsymbol{x_i^t}) &= \left(\boldsymbol{x}_i^{t+1} - \boldsymbol{x}_i^t\right) - \left(\boldsymbol{x}_i^t - \boldsymbol{x}_i^{t-1}\right) \\ &= \boldsymbol{x}_i^{t+1} + \boldsymbol{x}_i^{t-1} - 2\boldsymbol{x}_i^t. \end{aligned} \tag{3}$$

Our goal is to minimize this cost function in a distributed manner to provide optimal location estimates $\{\boldsymbol{x_i^t}$ for node.

Following is an equivalent expression for S:

$$S = \sum_i S_i. \tag{4}$$

Therefore, at each node i, we have a local cost function as:

$$\begin{aligned} S_i = &\sum_{t=1}^{T} \sum_{j \in \mathcal{N}_i} w_{i,j}^t \left[\delta_{i,j}^t - d_{i,j}(X^t)\right]^2 \\ &+ \sum_{t=1}^{T} r_i \left[\boldsymbol{\alpha}_i^t - \boldsymbol{a}_i(\boldsymbol{x_i^t})\right]^2. \end{aligned} \tag{5}$$

We note that S_i is local to i since it only depends on the measurements available at i and positions of neighbour nodes. Minimizing this local cost function will result in position estimates of this node. The majorization approach guarantees non-increasing local cost. Implementing our approach at each robot constructs the backbone of this distributed method. Each local cost distributes additively over the network, thus each sensor contributes to the minimization of the global cost function (1) by minimizing its own local cost function (5). This way, our algorithm produces a sequence of position estimates with non-increasing global cost.

B. Majorization

To minimize of our local cost function, we use a majorization approach inspired by SMACOF [10], but adding acceleration, and with multiple measurements over time. SMACOF is a gradient-decent algorithm which majorizes the cost function of the MDS problem. Iteratively minimizing the majorizing function guarantees a monotonously decreasing sequence of cost values. Our algorithm starts the minimization of its cost function 5 from an initial estimate of position and updates the estimates until the update in position is smaller than a certain value after a certain number of iterations. The final position estimates are the coordinates which minimize the summed weighted squared error. The majorizing function $T_i(\boldsymbol{x}_i, \boldsymbol{y}_i)$ of $S_i(\boldsymbol{x}_i)$ satisfies: (i) $T_i(\boldsymbol{x}_i, \boldsymbol{y}_i) \geq S_i(\boldsymbol{x}_i)$ for all y_i, and (ii) $T_i(\boldsymbol{x}_i, \boldsymbol{x}_i) = S_i(\boldsymbol{x}_i)$. The majorizing function $T_i(\boldsymbol{x}_i, \boldsymbol{y}_i)$ is thus defined as:

$$\begin{aligned} T_i(x_i, y_i) = \sum_{t=1}^{T} \sum_{j \in \mathcal{N}_i} & w_{i,j}^t [(\delta_{i,j}^t)^2 + (d_{i,j}(X^t))^2 \\ & - \frac{2\delta_{i,j}^t}{d_{i,j}^t(Y)} (\boldsymbol{x}_i^t - \boldsymbol{x}_j^t)^T (\boldsymbol{y}_i^t - \boldsymbol{y}_j^t)] \\ & + \sum_{t=1}^{T} r_i \left[(\boldsymbol{\alpha}_i^t - \boldsymbol{a}_i^t(\boldsymbol{x}_i^t)) \right]^2 . \end{aligned} \tag{6}$$

Here, the majorization function T_i is quadratic in nature, so we can find its minimum analytically. The analytical solution comes from differentiating it with respect to $\{\boldsymbol{x}_i^t\}$ and equating it zero. This minimization is done iteratively until convergence is reached to provide the values of $\{\boldsymbol{x}_i^t\}$ for which the majorized cost function S_i is low.

C. Proposed Algorithm

The proposed method is described in Algorithm 1. It should be noted that,

1) The algorithm requires initialized positions $X^{(0)}$. Generally, each round is initialized using the projected coordinates of positions from the previous round's estimates. The first time a neighbor j appears to node i, it must somehow provide the algorithm with $\{\boldsymbol{x}_j^t\}_t$. We leave a distributed initialization for this infrequent case to future work. Here, we use classical MDS to generate $\{\boldsymbol{x}_j^t\}_t$ when there is no estimate from a prior round.
2) The Euclidean distances (used here as the 'dissimilarities') do not have a frame of reference. They are measured locally – node i computes its range to each neighbor, and node i also measures its own acceleration vector $\boldsymbol{\alpha}_i^{(t)}$.
3) The weights applied to measured acceleration r_i should be chosen to account for accuracy of acceleration measurement for each node i. Lower values of r_i indicate more noisy measurements.
4) For all the measured $\delta_{i,j}$, the weights $w_{i,j}$ are simplistically set as equation 7, in essence making *all* the other j nodes as i's neighbors i.e. $j \in \mathcal{N}_i$.

$$w_{i,j} = \begin{cases} 1, & \text{if } \delta_{i,j} \text{ is measured.} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

However they can be adaptively set as a function of measurements $\delta_{i,j}$ such as to reflect its accuracy, such that less accurate measurements are down-weighted in the overall cost function and only the nodes with less noisy measurements are counted as neighbors.

Algorithm 1: Location Estimation with Ranges and acceleration data

```
Inputs : $\{\delta_{i,j}^t\}, \{w_{i,j}^t\}, \epsilon, \{r_i\}, X^{(0)}, \{\boldsymbol{\alpha}_i^t\}$
Initialize : $k = 0, S^{(0)}, a_i$ compute $q_i$ from
  Equation (8)
repeat
    k = k+1;
    for $i = 1$ to $N$ do
        for $t = 1$ to $T$ do
            compute $\boldsymbol{b}_i^{k-1}$;
            $\boldsymbol{x}_i^t \leftarrow q_i r_i (2\boldsymbol{x}_i^{t+1} + 2\boldsymbol{x}_i^{t-1} - \boldsymbol{x}_i^{t+2} - \boldsymbol{x}_i^{t-2} +$
              $\boldsymbol{\alpha}^{t+1} + \boldsymbol{\alpha}^{t-1} - \boldsymbol{\alpha}^t) + q_i X^{(k-1)} \boldsymbol{b}_i^{(k-1)}$;
            $S^{(k)} \leftarrow S^{(k)} - S_i^{(k-1)} + S_i^{(k)}$ ;
        send $\{\boldsymbol{x}_i^t\}_t$ to friend nodes;
        send $\boldsymbol{S}^{(k)}$ to node $(i+1)$ mod N;
until $S^{(k-1)} - S^{(k)} < \epsilon$;
```

The two helper variables used in the algorithm are q and b given by:

$$q_i^{-1} = \sum_{j \in \mathcal{N}_i} w_{i,j} + r_i \tag{8}$$

and $\boldsymbol{b}_i^{(k)} = [b_1, b_2, .., b_N]^T$ is a vector given by

$$\begin{aligned} b_j &= w_{i,j}[1 - \delta_{i,j}^t / d_{i,j}(X^{(k)})] \\ b_i &= \sum_{j \in \mathcal{N}_i} w_{i,j} \delta_{i,j}^t / d_{i,j}(X^{(k)}) \end{aligned} \tag{9}$$

D. Regression Based Collision Prediction Algorithm

Mobile agents in a network run a risk of colliding with each other and thus, need to have autonomous decisions making capabilities whether to pursue or move away from a trajectory. Predicting any future collisions between two agents involves predicting whether individual motions of those two moving agents will intersect, which depends on the knowledge of relative kinematics of those two. Using the relative locations estimated from previous stage, we can predict locations into the nearby future. Regression analysis is widely used for prediction and forecasting, as they reveal the causal relationships between a dependent variable and one or a collection of independent variables in a fixed dataset, which can later be used to estimate causal relationships using new observational data.

We choose polynomial regression since it works very well for non-linear interpolation problems. In our case, we seek to predict the future locations of the agent or robot which has a non-linear relationship with time due to the dynamic nature of the motion. The output of the polynomial regression in such a scenario can also be interpreted as higher order kinematics. Given T data points $(t_i, \boldsymbol{x}_i)$, where, dependent variable t_i is a time instance and $\boldsymbol{x}_i$ is the corresponding location of a robot i, for $i = (1, 2, ...T)$, we fit a 2^{nd} degree polynomial to approximate the robot's locations,

$$\boldsymbol{x}_i = \boldsymbol{p}_2 + \boldsymbol{p}_1 t_i + \boldsymbol{p}_0 t_i^2 \tag{10}$$

Here, $\boldsymbol{p}_0$, $\boldsymbol{p}_1$, and $\boldsymbol{p}_2$ make the coefficient of this polynomial that predicts the location $\boldsymbol{x}_i$ with estimation errors. We use least squares approximation, in which the polynomial coefficients can be obtained by minimizing the sum of the error squares. These coefficients are used to extrapolate values for the polynomial, giving us future relative locations, thereby, giving us future inter-object distances of two robots. We define *future* as a time-frame that is equal to or less than the reaction time of the robot. If the minimum inter-robot distance threshold between two robots is crossed within this time into future, a collision is predicted. We define collisions based on various distance thresholds and can detect collisions by comparing the extrapolated distances with these user-defined distance thresholds. In Section IV, we evaluate the performance of this regression-based collision prediction method for different distance thresholds, while also analyzing the location estimated from our algorithm.

IV. Results

In this section we test our proposed algorithm in terms of location estimation and collision prediction using simulated data. We demonstrate the performance of the proposed algorithm on a network of 4 robots with unknown location arranged on a uniform grid of 4x4 area units. Three robots are stationary and one robot is made to travel in a circular motion which contributes to it having acceleration. No known-location infrastructure is present and only inter-robot ranges and each robot's acceleration are calculated and used by our proposed algorithm. We trust that this simulation can be realized with an experiment conducted with irobots having UWB tags for ranging and inertial measurement units for acceleration measurements. In that case, the ranges obtained will be prone to error and noise. For this preliminary study, we introduce noise to ranging measurement, with standard deviation of 0.02 meters as per the findings in [11]. Using this range data, we present the results in the following section.

A. Location Estimation

We first demonstrate the quality of location estimates generated from our algorithm. For the $N = 4$ robots, 1 robot moving in a near-circular motion for multiple laps and 3 being stationary (thus, all having relative motion), we are able to track the curved trajectory that was followed by the mobile robot over time. We take a window of 20 instances at once ($T = 20$), and for all i in N, we estimate 20 locations $\{\boldsymbol{x}_i^t\}$, one for each time instance of the window. In a sliding window

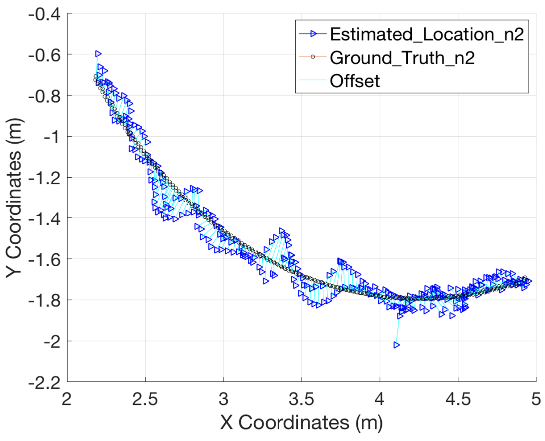

Fig. 1. Location estimates only for the moving robot: Blue triangles represents our algorithm's location estimates plotted on top of black circles representing the ground truth of the robot in motion. The measured distances have added noise with st. deviation of 0.02 and algorithm runs only when a user-defined proximity distance threshold is crossed.

manner, next window's 20 estimates are calculated. We keep the middle point $(T/2)$ as an 'average' solution from that window with respect to the center point of that window's ground truth. Plotting $T/2's$ estimates versus ground truth, in Figure 1 we see location estimates for only robot-2 (n2). The ground truth of each robot and its estimated positions are marked 'o' and '▷' respectively, where the lines represent the offset between the estimates and ground truth. In Figure 2, each window's such middle location estimates (in red) for robot-2 are plotted against its ground truth (black) for its circular trajectory. Also, the stationary robots (n $0, 1, 3$) are plotted for their estimated locations and ground truth, alongside robot-2's circular motion so as to provide a complete picture of the experiments. We see that our algorithm is able to estimate the locations of all the robots, more importantly of robot-2, and commendably able to follow the trajectory of robot-2's motion over time.

Next, we compare the performance of our algorithm with another MDS-based localization algorithm [6]. For one time window, this method provides joint higher order relative kinematics estimates (position, velocity, and acceleration), however, they are extremely sensitive to ranging noise. Figure 3 shows the root mean square error (RMSE) of location estimated for each robot by the two methods when compared to ground truth. We can see that our algorithm outperforms the other MDS-based implementation, which is credited to our algorithm's capability of providing better position estimates by incorporating the robots' acceleration information in cost function minimization. This makes the algorithm to trace the motion well and provide a lower RMSE by one order of magnitude when compared to the other MDS-based method.

We emphasizes that none of the robots' locations are known before the start of our algorithm, i.e. there is no known-location infrastructure (anchors) present in the system. Every node independently runs our proposed algorithm to estimate

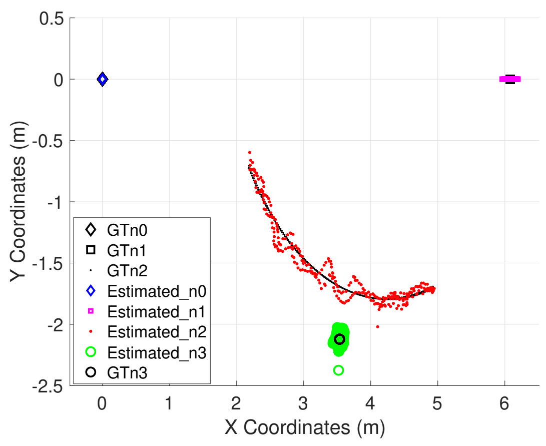

Fig. 2. Location estimates for all the robots in the system (4 here): Each robot's locations estimated by our algorithm are plotted on top of black markers representing the ground truth for that respective robot. The measured distances have added noise with st. deviation of 0.02 and algorithm runs only when a user-defined proximity distance threshold is crossed.

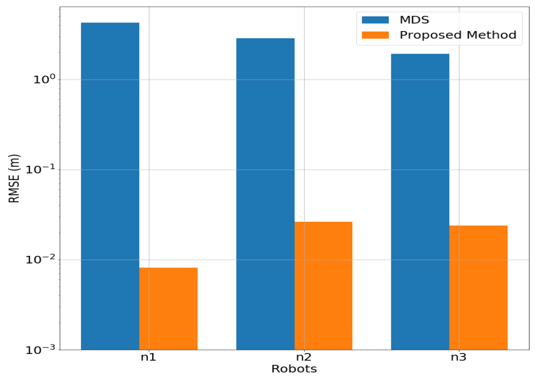

Fig. 3. RMSE comparison location estimates from MDS and our algorithm, with distances having added noise of standard deviation 0.02

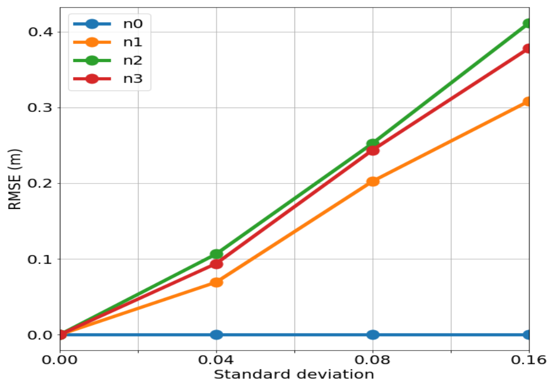

Fig. 4. RMSE of the position estimates for each node Versus various levels of noise in the range measurement

an optimal solution for its position. That is, after receiving it's neighbours optimal solution and using its own acceleration, each node conducts a successful minimization of its local cost function as explained in Section III-B. However, we translate one device to become the origin ($n0$) to constitute a consistent frame of reference, although it should be noted that this does not change the relative positions among the robots, and yet is helpful when we need to compare against a ground truth with similar translation. Additionally, the received estimates from one device's neighbours can be noisy and hence can impact the optimal solution for that node. We explore the performance of estimated locations when different amount of noise is present in measured inter-robot ranges. In Figure 4, we see that as the noise in the range measurements is increased, the error in the position estimated with our algorithm also increases for all the nodes n1,n2, and n3.

B. *Interpolated Distance-based Collision Prediction*

In order to avoid collision, a device or robot needs to make a prediction whether it is going to collide or not, before it happens. For collision prediction, only relative position between two robots is enough to predict any future distance threshold violation that counts as collision. Using relative location estimates from our algorithm, we extrapolate distances from the model explained in III-D. Let us say that the future separation i.e. distance $\hat{d}_{i,j}^t$ between two robots i and j goes lower than a threshold at some time into the nearby future. We define future as a time window equal to reaction time of a robot. If the predicted distances between two robots go under a set threshold within this window, robots can not swerve to avoid collision and collision in the future is predicted. The reaction time τ for each robot is taken as 0.09 *seconds* for all simulations. Within this reaction time into the future, if the predicted distances distance $\hat{d}_{i,j}^t$ is smaller than any predecided minimum-distance threshold d_{thd}, it is counted as a collision. Note that separation between the center of two robots is at least 2 times its radius r, which is equal to the diameter of one robot (0.34 meter is the diameter of one common irobot). Adding extra distance value ϵ_d to $2r$, we get minimum-distance threshold, given as

$$d_{thd} = \epsilon_d + 2r \tag{11}$$

If future inter-robot distances into the future are under this value it is counted as a collision. Figure 5 shows the receiver operating characteristic where the probability of detection P_D of collision is given as the function the probability of false alarm. We report that our extrapolation based collision prediction model is able to provide higher probability of detection for the same probability of false alarms when compared with the other kinematics method FACT [11]. Our algorithm gives 100% detection with a 2% False Alarm, where as FACT gives a detection of 80% at same false alarm rate. Another collision detection approach based on pairwise regression performs even worse. This is because our algorithm's localization provides

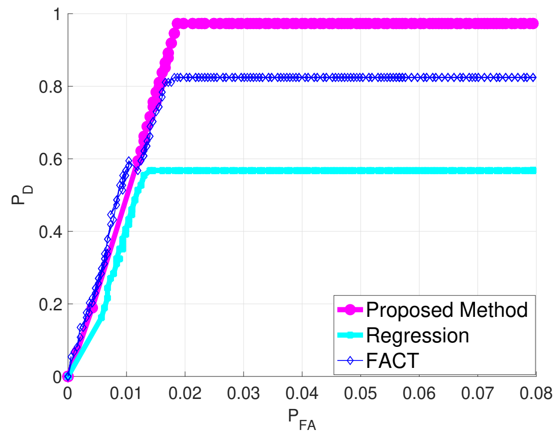

Fig. 5. Receiver Operating Characteristic plot for a comparison between our collision prediction method, FACT, and pairwise regression

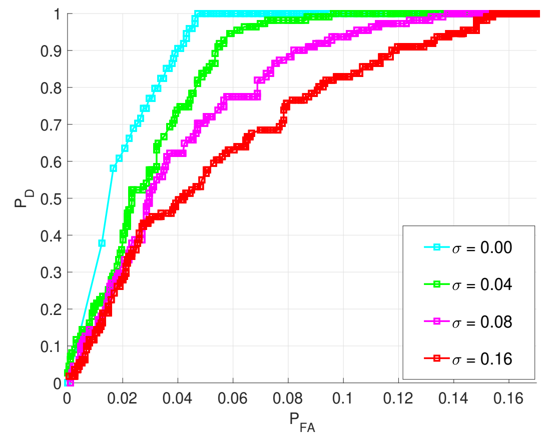

Fig. 6. Proposed Method's Receiver Operating Characteristic plot for various amount of noise added into ranges collected

accurate position estimates for any degree of complex motion, which lets us extrapolate complex trajectories very well, giving extremely accurate future inter-robot distances and kinematics to predict collisions. The 2^{nd} degree regression's coefficients are able to extrapolate the future locations while taking each robot's acceleration into account, a trait not achievable by FACT or pairwise regression.

Lastly, as explained in Section IV, robots equipped with UWB tags for ranging have inter-robot ranges prone to noise. Prediction of future locations, and thereafter collisions, by using these noisy range measurements can be studied as a function of noise. In order to investigate the effect of noise, we test our collision prediction methodology for noisy range measurements with varying standard deviation. The results corroborates the intuition that the detection accuracy deteriorates with increase in noise as is shown in Figure 6.

V. Conclusions

This paper presented a new approach to estimate locations and predict collisions for systems involving mobile devices that are capable of communicating their range measurements to each other. Our method uses each device's acceleration into estimating location coordinates for every device, achieving a commendable location estimation performance. The algorithm extends further by predicting collisions into future and it does not require a known-location infrastructure or a central decision maker to achieve a good collision prediction performance. We test its performance in simulation settings and provide a detailed analysis of results.

Acknowledgment

This work is supported by the US National Science Foundation under Grant No. 1622741. We thank Alemyaheyu Abrar for sharing his code with us. We also thank the our colleagues for their comments and suggestions to improve the quality of this paper.

References

[1] D. H. Daneshvar, C. J. Nowinski, A. C. McKee, and R. C. Cantu, "The epidemiology of sport-related concussion," Clinics in sports medicine, vol. 30, no. 1, pp. 1–17, 2011.

[2] Y. Shang and W. Ruml and Y. Zhang and M. Fromherz, "Localization from mere connectivity". In:Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking amp; Computing. MobiHoc '03. Annapolis, Maryland, USA: Association for Computing Machinery, 2003, pp. 201–212.

[3] W. S. Torgeson. Multidimensional scaling of similarity. Psychometrika, 30:379–393, 1965.

[4] J. A. Costa, N. Patwari, and A. O. Hero. "Distributed weighted-multidimensional scaling for node localization in sensor networks". In: ACM Trans. Sen. Netw. 2.1 (Feb. 2006), pp. 39–64.

[5] B. Beck, R. Baxley and J. Kim, "Real-time, anchor-free node tracking using ultrawideband range and odometry data," 2014 IEEE International Conference on Ultra-WideBand (ICUWB), Paris, 2014, pp. 286-291, doi: 10.1109/ICUWB.2014.6958994.

[6] R. T. Rajan, G. Leus, and A.-J. van der Veen, "Relative kinematics of an anchorless network," Signal Processing, vol. 157, pp. 266–279, 2019.

[7] Wei, P., Cagle, L., Reza, T., Ball, J., Gafford, J. (2018). LiDAR and camera detection fusion in a real time industrial multi-sensor collision avoidance system. ArXiv, abs/1807.10573.

[8] A. Viquerat, L. Blackhall, A. Reid, S. Sukkarieh, and G. Brooker, "Reactive collision avoidance for unmanned aerial vehicles using doppler radar," in Field and Service Robotics. Springer, 2008, pp. 245–254.

[9] R. Chellappa, G. Qian, and Q.Zheng, "Vehicle detection and tracking using acoustic and video sensors," in 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3. IEEE, 2004, pp. iii–793.

[10] Groenen, P., Mathar, R., Heiser, W. (1995). The majorization approach to multidimensional scaling for Minkowski distances. *Journal of Classification*, 12, 3-19

[11] Abrar, A.S., Luong, A., Spencer, G., Genstein, N., Patwari, N., Minor, M. (2020). Collision prediction from UWB range measurements. arXiv: Signal Processing.