2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

A Distributed Range-Only Collision Avoidance Approach for Low-cost Large-scale Multi-Robot Systems

Ruihua Han[1], Shengduo Chen[1] and Qi Hao[1,2,*]

Abstract— **The challenges of developing low-cost, large-scale multi-robot navigation systems include noisy measurements, a large number of robots, and computing efficiency for collision avoidance. This paper presents a distributed motion planning framework for a large number of robots to navigate with robust collision avoidance using low-cost range only measurements. The novelty of this work is threefold. (1) Developing a distributed collision-free navigation system for a large-scale robot group in which each robot performs motion planning based on the noisy range measurements of neighboring robots; (2) Developing a set of algorithms for each robot to accurately estimate the relative positions and orientations based on the range measurements and relative velocities; (3) Developing a velocity obstacle (VO) based motion planning algorithm for each robot which can take into account of the estimation uncertainties in relative positions and orientations. The proposed approach is tested with various numbers of differential-driven robots in the Gazebo simulator and real-world experiments. Both simulation and experiment results validate the superior performance of the proposed approach compared to other state-of-art technologies.**

I. Introduction

Multi-robot systems have drawn increasing attention because of their advantages in performing complex tasks compared to the single robot, such as surveillance, rescue, formation, and exploration [1]. During group navigation in a 2D environment, each robot's capacity to avoid other robots is highly required, especially for large-scale multi-robot systems. These approaches usually consist of three components: (1) measurement collection, (2) pose estimation, and (3) motion planning.

Despite many efforts in developing collision avoidance schemes for the robot group navigation, three significant challenges need to be addressed:

1) **Low latency distributed computing framework**. Compared to the centralized navigation system, the distributed framework is advantageous in its robustness against local failures and scalability in the group size. However, a successful distributed framework requires low latency in information collection and local processing, i.e., low measurement data throughput and computational complexity.

*Corresponding author

This work is partially supported by the National Natural Science Foundation of China (No: 61773197); the Science and Technology Innovation Committee of Shenzhen City (No: CKFW2016041415372174,GJHZ20170314114424152); and the Nanshan District Science and Technology Innovation Bureau (No: LHTD20170007); and the Intel ICRI-IACV Grant 2019.

[1]The authors are with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, Guangdong, 518055, China. {hanruihuaff@gmail.com haoq@sustc.edu.cn}

[2]Sifakis Research Institute of Trustworthy Autonomous Systems

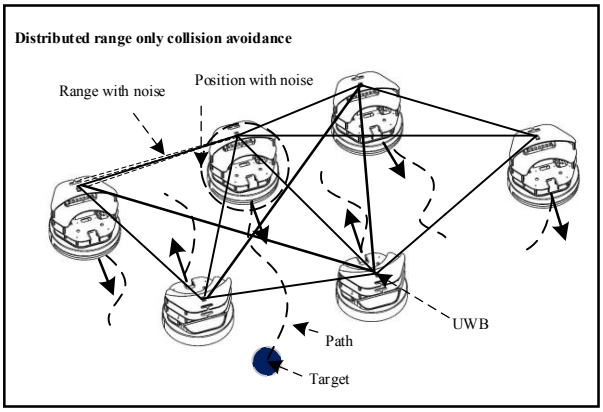

Fig. 1. An illustration of distributed multi-robot collision avoidance with range measurements only.

2) **High-accuracy estimation of relative positions and orientations**. The information on relative poses among robots is the precondition of all collision avoidance approaches. How to estimate the relative poses between robots with high accuracy from low data-throughout measurements is still an open question.
3) **Robust collision avoidance with uncertainties in pose estimation**. The uncertainties in relative pose estimation increase the risk of robot collision. It is vital to develop a robust collision avoidance scheme.

Various approaches have been developed to solve these challenges such as reinforcement learning (RL) based approaches [2]–[7], velocity obstacle (VO) based approaches [8]–[11] and energy-optimal based approaches [12], [13]. Compared with RL and energy-optimal based approaches, VO based methods have lower computational complexities and are convenient for real-time implementation. Most centralized approaches assume that all the robots have a global coordinate system, or each robot has the perfect sensing capability of acquiring the accurate pose information on others [14], [15], an extra tracking system is often required in the real world experiments [16]–[18], incurring high computational costs and limits upon applications in outdoor environments. For distributed approaches, each robot needs to estimate its own pose first using on-board sensors; then, the estimated poses are shared through inter-robot communications [19]–[21]. Usually, expensive high-resolution sensors are unsuitable for applications of large-scale robot groups. Outdoor applications of those sensors highly sensitive to the changing environmental conditions are also limited.

In this paper, a distributed multi-robot collision avoidance approach is proposed with range-only measurements using

TABLE I

A COMPARISON OF THE PROPOSED APPROACH AND STATE-OF-THE-ART APPROACHES TO COLLISION FREE NAVIGATION.

	Approach	Architecture	training stage	Information exchange	Computational complexity	sensoring modality	Position requirement	Advantages	Disadvantages
VO based approaches	ORCA [14]	distributed	no	no	low	tracking system	perfect position	collision-free motion guarantee	need global posion of all robots
	CALU [19]	distributed	no	position and velocity	moderate	LIDAR	estimated position using AMCL	performing ORCA on the real robot with estimated position	only consider the uncertainty in the state estimation process
	PRVO [22]	distributed	no	position and velocity	moderate	tracking system	estimated position with Gaussian distribution	modeling the uncertainties in state estimation and local motion	less robust against the uncertainty in state estimation
RL based approaches	SLCAP [7]	decentralized	required	motion planning policy	high	laser	perfect position	can tackle large robot system and dynamic environment	do not suit for featureless environments
	GA3C-CADRL [6]	decentralized	required	no	high	LIDAR	position with uncertainty	can tackle obstacles with varying number and velocity	high cost measurement
Our approach	–	distributed	no	velocity	low	UWB	estimated position using dead reckoning	low cost, more robust against uncertainty in position , suit for all environments	in need of init state and time to estimate relative position

low-cost UWB (ultra-wideband) modules. The shared velocity information and individual range measurements are used to estimate the relative poses among robots. The uncertainties in pose estimation are taken into account for the VO based collision avoidance algorithm. The main contributions of this work include:

1) Developing a distributed collision avoidance architecture, in which each robot performs its own motion planning and shares the velocity information with each other. Such a distributed framework is robust against local failures and enables large-scale robot groups.
2) Developing a particle filter based algorithm to estimate the relative poses among robots from the range only measurements and shared information on velocities.
3) Developing an extended reciprocal collision avoidance algorithm to tackle the uncertainties in the estimated poses guarantees the robust performance of collision avoidances in the real-world applications.
4) Developing both simulation and experiment platforms for large-scale robot group navigation. The proposed approach is validated using multiple Turtlebots equipped with UWB modules in comparison with other state-of-the-art methods.

The rest of the paper is organized as follows. Section II discusses the related approaches to multi-robot collision avoidance. Section III describes the system setup and problem statement. Section IV presents the proposed approach. Section V provides simulation and experiment results. Section VI concludes the paper and outlines future work.

II. RELATED WORK

Table I summarizes a comparison of various approaches in terms of processing architecture, training requirements, sensing modality, shared information, position requirement, and computational complexity. It can be seen that RL based approaches usually require a training stage, high-accuracy pose estimation, and are more sensitive to the uncertainties in pose estimation results. By comparison, VO based approaches involve less computational complexity and can be more robust against estimation uncertainties with the help of probabilistic models. Besides, the simulation based training not only consumes more offline computational resources but also imposes a limit upon the scale of the robot group for training.

The VO method constructs a predicted collision area with the information on relative pose and velocity between an individual robot and dynamic obstacles to choose a proper velocity [8]. By sharing certain information among multiple agents, cooperative collision avoidance (CCA), and reciprocal VO (RVO) schemes have been developed for robot groups in various sizes [9], [10]. To further reduce the decision dilemmas or deadlocks for multiple robot motion planning, hybrid RVO (HRVO) [11] and optimal reciprocal collision avoidance (ORCA) schemes [14] have been proposed; the former combines the VO and RVO to reduce the symmetry of collision areas and the latter employs liner programming to perform motion planning under constraints on velocities. The ORCA method has also been extended for non-holonomic (NH) robots, called NH-ORCA [15].

However, these VO based approaches usually assume perfect sensing of the poses of robots and require a new tracking system for their real-world applications [16], [18]. Collision avoidance with localization uncertainty (CALU) and convex outline collision avoidance under localization uncertainty (COCALU) have been developed to deal with uncertainties in pose estimation, where each robot performs the adaptive Monte-Carlo localization (AMCL) and derives uncertainty models based on the particle filter while performing the NH-ORCA or ClearPath [19], [20]. However, these approaches only consider the uncertainties in the state estimation process. Thus, the probabilistic RVO (PRVO) has been developed, which combines the RVO and probabilistic constraints, to address the uncertainties in both state estimation and locomotion [22]. However, the PRVO approach needs further simplification for real-world implementation. Besides, all the current VO based methods heavily rely on expensive high-resolution sensors, such as camera and LiDAR, unsuitable for applications of large-scale robot groups and in featureless environments.

In our work, the proposed distributed multi-robot collision avoidance approach only uses the range measurements and exchanges information on velocities among neighboring robots. The NH-ORCA method is upgraded to determine the optimal velocity for each differential-driven robot. The impact of the uncertainties in state estimation upon the individual robot is reduced since only the relative poses and velocities are used to construct the VO space. The NH-ORCA is extended by increasing the radius according to the uncertainty in the estimated position. Besides, the range

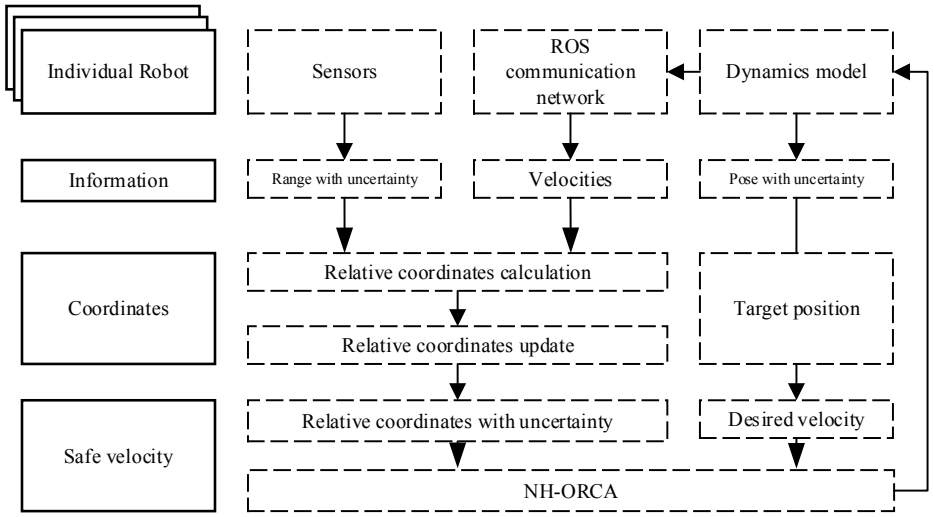

Fig. 2. The system framework of the proposed distributed range only multi-robot collision avoidance system.

measurements between robots can be easily obtained using low-cost UWB modules. Since the proposed approach does not need a training stage, there is no limit on the size of the robot group in applications.

III. System Setup and Problem Statement

A. System Framework

The framework of range only multi-robot collision avoidance is shown in Fig 2. First, for robot i in a group of robots, the range sensor like UWB tag, which can detect the distances of other robots, is equipped. The velocities of detected robots can be obtained via the subscriber-publisher mechanism of the ROS [23] communication network. The state of the differential-drive robot includes the position and orientation in time t can be denoted as $s_i^t = [x_i^t, y_i^t, \theta_i^t]$. Each robot is controlled by a pair of translational and rotational velocities $[v_i^t, \omega_i^t]$. Second, based on the range measurements and the exchanged information on relative velocities, the relative poses of the other robots in the coordinate system of robot i are estimated and updated. Finally, the optimal velocity that guarantees the collision-free motion of the robot is chosen by using NH-ORCA, given the input of relative poses of other robots with uncertainty models.

B. Non-Holonomic Robots Optimal reciprocal collision avoidance

NH-ORCA is an extension of ORCA that guarantees the collision free motion of multiple non-holonomic robots. The locomotions of holonomic and non-holonomic robots in the velocity of $\mathbf{v}_H$ is shown in Fig. 3(a), which has a tracking error ξ_{track}. The NH-ORCA includes three main steps. First, the tracking error is added with the robot radius to construct the safety envelopes and collision areas, as shown in Fig. 3 (b-c). Secondly, the set of allowed holonomic velocities S_{AHV} is computed using the velocity $\mathbf{v}_H$ to limit the tracking error bound. It is represented by a convex polygon P_{AHV}. Finally, within the set of safe velocities computed by

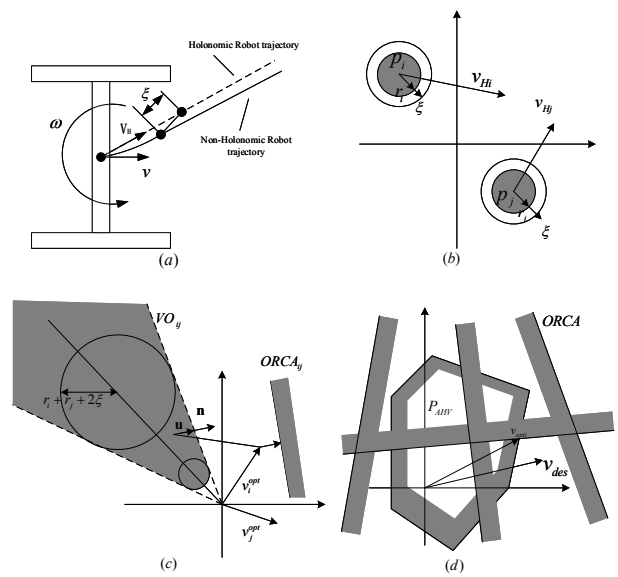

Fig. 3. A comparison of holomonic and non-holomonic robots: (a) locomotions, (b) safety envelopes, (c) collision areas of VO and ORCA, and (d) motion planning of ORCA.

P_{AHV} and ORCA, the optimal velocity closest to the desired velocity is chosen and mapped into the corresponding non-holonomic control inputs, as shown in Fig. 3 (d).

C. Problem Statement

For a group of non-holonomic robots navigating in the 2D environment, the desired velocity is the velocity that guides robots to the target position directly without the consideration of obstacles within each time window. The main problems of the range-only multi-robot collision avoidance include (1) how to estimate the robot poses with range measurements and exchanged information on velocities in a distributed way; (2) how to find a safe velocity for each robot to achieve a collision-free and time-efficient motion within a time horizon τ given the robot poses with uncertainties.

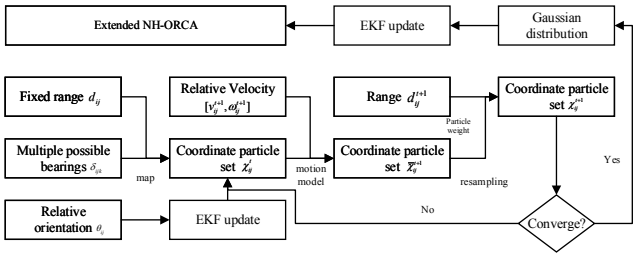

Fig. 4. The flowchart of relative poses estimation.

For a group of l robots, each robot can receive rang measurements r and relative velocity measurements v from neighboring robots at each time step. For robot i, the probability of relative pose p_{ij}^t of neighboring robot j is derived from the range measurements r_{ij}^t and velocity measurements v_{ij}^t over a period of time, $P(p_{ij}^t | r_{ij}^{1:t}, v_{ij}^{1:t})$. Thus, the optimization objective of each robot under N estimated relative poses should be:

$$\begin{aligned} &\arg\min_{v_i} E[T | p_{i,1:N}^{1:t}] \\ &s.t.\ \forall j \in [1, N] \\ &d_{ij} > 2r \end{aligned} \tag{1}$$

where, T is the travel time, d_{ij} is the distance between two robots, r is the robot radius. The goal of the proposed approach is to calculate the optimal velocity to guide each robot to a time efficient and collision free motion under the estimated the relative pose derived from range and velocity measurements.

IV. Range based Collision Avoidance

We propose a distributed framework to utilize the ranges among robots to estimate the relative poses in the robot coordinate system. The estimated poses with uncertainties are used as the input of NH-ORCA to choose the optimal velocity. Besides, the NH-ORCA is extended to take into account of uncertainties in pose estimation. The flowchart of the proposed approach is shown in Fig. 4. In the distributed framework, all the robots perform the same collision avoidance algorithm and choose their own optimal velocities independently. Although those robots have different coordinate systems, their control vectors and orientations can be regarded as in the same coordinate system if the robot orientations at the beginning are pre-known.

For a robot i with the control vectors $[v_i^t, \omega_i^t]$ and orientation θ_i^t at time t, the velocity in x and y direction is $\left[v_{ix}^t, v_{iy}^t\right]$, where, $v_{ix}^t = v_i^t * \cos\theta_i^t$, $v_{iy}^t = v_i^t * \sin\theta_i^t$. The relative velocity and orientation of robot j with respect to robot i at time t should be $v_{ij}^t = [v_{jx}^t - v_{ix}^t, v_{jy}^t - v_{iy}^t]$ and $\theta_{ij}^t = \theta_j^t - \theta_i^t$ respectively. The relative coordinates $[p_{ijx}, p_{ijy}]$ of robot j can be mapped from the range d_{ij} and bearing δ_{ij} as following:

$$\begin{aligned} p_{ijx} &= d_{ij} * \cos\delta_{ij} \\ p_{ijy} &= d_{ij} * \sin\delta_{ij} \end{aligned} \tag{2}$$

Thus, the relative state of robot j at time t can be denoted as $[p_{ijx}^t, p_{ijx}^t, \theta_{ij}^t]$. At each time step, the relative orientation can be fused by extended Kalman filter (EKF). During EKF process, the predicted orientation θ_{ijpre}^t can be calculated from the relative angular velocity ω_{ij}^t by the motion model. The relative orientation θ_{ij}^t from the robots is the observation to correct the estimation [24]. However, the relative bearing δ_{ij} is unknown which leads to the unknown coordinates. Thus, we estimate the relative coordinates utilizing the particle filter as described in algorithm 1. Firstly, for m particles at time t, the coordinates of particle k ($k \in [1, m]$) are initialized with the fixed range d_{ij}^t and randomly distributed bearing δ_{ijk}^t between 0 and 2π. Secondly, at the next time $t+1$, each particle k utilizes the relative velocities v_{ij}^t to calculate a new predicted relative coordinate $[p_{ijk}^{t+1}, p_{ijk}^{t+1}]$ depend on the motion model described as following:

$$[p_{ijk}^{t+1}, p_{ijk}^{t+1}] = [p_{ijk}^t, p_{ijk}^t] + v_{ij}^t * \Delta t \tag{3}$$

This motion model is the approximate estimate of the differential motion model between each time step. The shorter the time step, the smaller the error.

These coordinates can be mapped back to the range and bearing by the following equation:

$$\begin{aligned} d_{ijk}^{t+1} &= \sqrt{\left(p_{ijkx}^{t+1}\right)^2 + \left(p_{ijky}^{t+1}\right)^2} \\ \delta_{ijk}^{t+1} &= \arctan(p_{ijky}^{t+1} / p_{ijkx}^{t+1}) \end{aligned} \tag{4}$$

Actually, the closer the mapped ranges are to the range measurements d_{ij}^{t+1}, the higher the accuracy of the coordinates will be. Thus, the particle weight is generated by the difference $diff_{ijk}^{t+1}$ between d_{ij}^{t+1} and d_{ijk}^{t+1}. We utilize a Gaussian distribution with zero-mean and variance φ_w to generate the probability of each particle as weight.

$$\varpi_{ijk}^{t+1} = f(diff_{ijk}^{t+1}, 0, \varphi_w) \tag{5}$$

Where, $f(x, 0, \varphi_w)$ is the probability density function of Gaussian distribution. Specifically, variance φ_w is decided by the uncertainty of range measurement. Finally, the low variance resampling step is applied to draw a new particle set from the previous one depending on the particle weight [25]. After some time, the coordinates of the particle set will converge decided by the variance of bearing set. This set represents the appropriation of the coordinates of robot j respect to robot i. However, the particles may converge in an incorrect and adjacent pose because of the uncertainties or kidnapped problems. To recover from the failures, the coordinates of parts of the particles are set uniformly depend on current bearing after the initial convergence. For a particle set with current average bearing δ_{ij}^t, the coordinates of n particles ($n < m$) should be calculated from n bearings which obey the uniform distribution between $\delta_{ij}^t - \delta_{range}$ and $\delta_{ij}^t + \delta_{range}$, where δ_{range} is the distribution interval. After several convergences under this situation, the particle set tends to converge at the correct pose.

We use a Gaussian distribution, which extracts the mean and variance from a particle set to represent the relative coordinates, including x and y direction. To save the computational cost, after particle convergence, the relative coordinates with Gaussian distribution are kept and updated

using EKF localization by relative velocities and range measurements. It should be noted that when the moving direction is perpendicular to the range distribution, the range measurements are unchanged and cannot filter the particles. However, in this situation, the collision will never occur.

In multi-robot systems, each robot performs this algorithm to estimate the relative coordinates with others. However, the computational cost of implementing the estimation at the same time is high, which influences the real-time performance. Thus, the estimation is performed sequentially by the value of distance.

Algorithm 1 Relative Coordinates Estimation

Require: For robot i and j, range measurement d_{ij}^t, relative orientation θ_{ij}^t, relative velocity v_{ij}^t

Initialize m a coordinates particle set with range and uniformly distributed bearing between 0 and 2π
for time $t = 1...N$ **do**
  for particle $k = 1$ to m **do**
    Calculate new coordinates by Equation (3)
    Generate particle weight by Equation (5)
  end for
  Low variance resampling
  if particles converge initially **then**
    Generate n uniform distributed bearings depend on current average bearing
    Reset the coordinates of partial particles
    if particles converge continuously **then**
      extract the particle set as a Gaussian distribution
    end if
  end if
end for

The NH-ORCA does not consider the uncertainty in robot position. Thus we extend this approach to tackle the estimated coordinates with Gaussian distribution. The uncertainty of estimated relative states from the previous step can be represented by the covariance φ which is a 3x3 matrix. The diagonal of this matrix $[\varphi_{11}, \varphi_{22}, \varphi_{33}]$ is the variance in the direction of x, y, θ respectively. Depending on the property of Gaussian distribution, the three times of standard deviation are used to represent the maximum error in x and y direction:

$$\begin{aligned} \xi_x &= 3\sqrt{\varphi_{11}} \\ \xi_y &= 3\sqrt{\varphi_{22}} \end{aligned} \tag{6}$$

In the process of range-only filter estimation, there are more uncertainties in the vertical direction of the range. The maximum error $\xi_{\max}$ should be:

$$\xi_{\max} = \sqrt{{\xi_x}^2 + {\xi_y}^2} \tag{7}$$

To guarantee the collision-free motion for robot i, the radius of the robot j should increase by the maximum error $\xi_{\max}$ that should be $r_j + \xi_{\max}$. Thus, during the NH-ORCA, the radius of robot i and j should be $r_i + \xi_{track}$ and $r_j + \xi_{\max} + \xi_{track}$ respectively. Then, the NH-ORCA is performed with the inflated radius to calculate the optimal velocity.

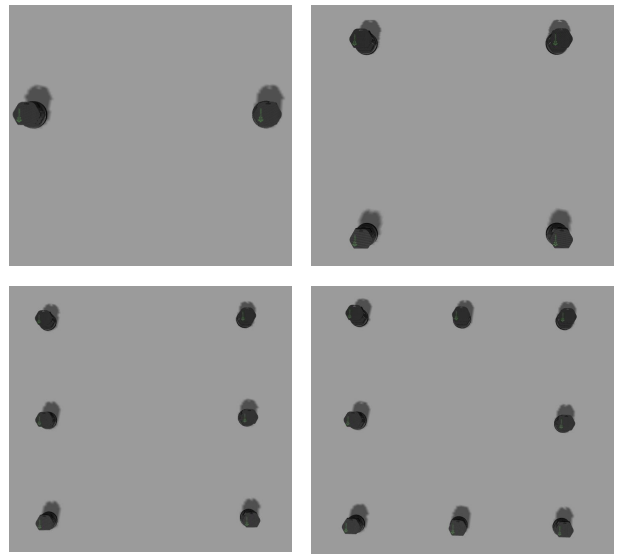

Fig. 5. Four simulation scenarios with multiple Turtlebots.

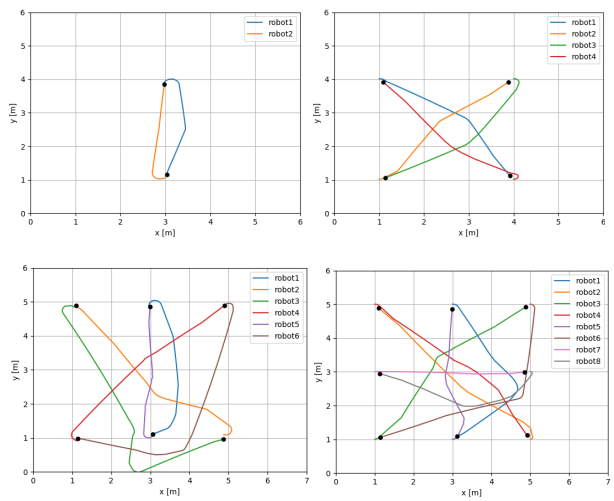

Fig. 6. Robot trajectories of the four simulation scenarios.

During the estimation process, before particles converge, the time complexity for m particles and l neighboring robots should be $O(m * l)$. After convergence, estimated states are updated by EKF step. Thus, the time complexity should be $O(l)$. In the NH-ORCA step, which is used to calculate the optimal velocity, the time complexity should also be $O(l)$.

V. Experiments and Results

A. Simulation Setup

Gazebo can simulate multi-robot systems in different environments [26]. ROS provides numerous libraries and tools to help achieve various functionalities of the robot group. We validate the proposed approach with various numbers of robots in Gazebo. As shown in Fig. 5, each simulation scenario is composed of multiple Turtlebots, which are popular open-source differential-drive robot platforms. Gazebo can simulate the range measurements among robots with

TABLE II
PARAMETER SELECTION

Estimation step		Extended NH-ORCA step	
parameter	value	parameter	value
particle number m	300	robot radius	0.3m
particle number n	40	tracking error	0.05m
variance φ_w	0.02m	time horizon	10s
sampling time	0.05s	max neighbors	8

different levels of noise. In the beginning, multiple robots were placed with pre-known orientations to simplify the pose estimation. Parts of the key parameters of our approach are selected, as shown in Table II. Specifically, particle number m influences the accuracy of estimated relative poses dramatically. More particles will get more precise results but with higher time complexity. Variance φ_w affects the particle convergence speed and accuracy, which is important in real-time navigation. The trajectories of various numbers of robots generated by our approach are shown in Fig. 6. It can be seen that our approach can choose the optimal velocity in a distributed way for each robot to achieve the collision-free motion successfully.

B. Experiment Setup

Our approach is also tested in real-world experiments with multiple Turtlebots. The experiment setup is shown in Fig. 7, where multiple Turtlebots are placed with pre-known orientations. Each robot is equipped with a UWB tag that can measure the distance from other UWB tags. All the robots are in the same ROS network and share the velocity information with each other. Each robot performs the same algorithm and optimizes its own velocity.

Fig. 7. The experiment setup of multiple Turtlebots.

C. Results and Discussions

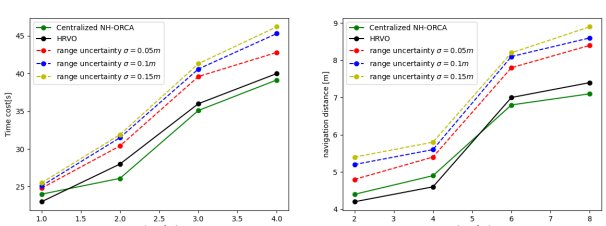

Fig. 8. A comparison of differential approaches in terms of navigation time and distance.

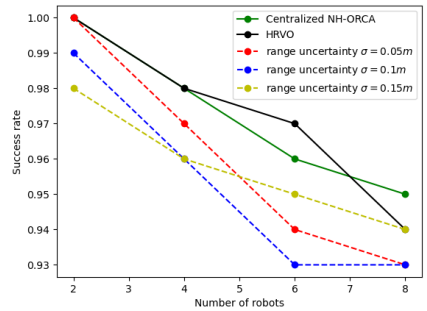

Fig. 9. A comparison of differential approaches in terms of success rate.

We use three indices to measure the performance of our approach: the success rate, navigation time, and distance. This approach is performed in simulation with the number of robots from 2 to 8. We compare three approaches with different range uncertainties represented by the standard deviation $\sigma_{range} = 0.05m$, $\sigma_{range} = 0.1m$, $\sigma_{range} = 0.15m$. The HRVO and NH-ORCA are also tested to compare. Besides, the NH-ORCA is performed with a centralized framework to compare. It uses the global positions and velocities of all robots without noise as input. HRVO and NH-ORCA are the state-of-art VO based approaches to tackle the multi-robot collision avoidance problem, especially for differential-driven robots. Different configuration and scenario are tested with 50 times, and the results are averaged, which are illustrated in Fig. 8 and 9. Specifically, the success rate demonstrates the effectiveness of collision avoidance. The navigation distance and time grow up with the number of robots. Compare to the ideal situation that there is no noise during the computation process, our range based approach can achieve similar collision-free motion but with more time cost and traveled distance. The performance is determined by the accuracy of the range, sampling time, and velocities measurements.

VI. CONCLUSION

In this paper, a range-only distributed collision avoidance approach for large-scale multi-robot systems has been presented. Each robot measures the distance of neighboring robots via UWB modules and shares the velocities with each other via the ROS communication network. The range measurements and relative velocities are utilized to estimate the relative poses of robots by using a particle filter. The probabilistic distributions of relative poses are recursively updated and integrated with an NH-ORCA scheme to achieve robust collision-free motion planning of differential driven robots. The simulations and experiments have been performed with various numbers of Turtlebots in a 2D environment. The results show that distributed collision-free navigation can be achieved using low-cost range-only measurements. Compare to other state-of-art VO based approaches, our approach does not need the global localization and has similar effectiveness of collision avoidance. Our future work includes developing a learning scheme for the proposed approach and testing the proposed approach with more robots.

REFERENCES

[1] R. N. Darmanin and M. K. Bugeja, "A review on multi-robot systems categorised by application domain," in *2017 25th Mediterranean Conference on Control and Automation (MED)*. IEEE, 2017, pp. 701–706.

[2] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in neural information processing systems*, 2017, pp. 6379–6390.

[3] P. Long, W. Liu, and J. Pan, "Deep-learned collision avoidance policy for distributed multiagent navigation," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 656–663, 2017.

[4] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 285–292.

[5] T. Fan, P. Long, W. Liu, and J. Pan, "Fully distributed multi-robot collision avoidance via deep reinforcement learning for safe and efficient navigation in complex scenarios," *arXiv preprint arXiv:1808.03841*, 2018.

[6] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3052–3059.

[7] P. Long, T. Fanl, X. Liao, W. Liu, H. Zhang, and J. Pan, "Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6252–6259.

[8] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[9] Y. Abe and M. Yoshiki, "Collision avoidance method for multiple autonomous mobile agents by implicit cooperation," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 3. IEEE, 2001, pp. 1207–1212.

[10] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1928–1935.

[11] J. Snape, J. Van Den Berg, S. J. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacle," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 696–706, 2011.

[12] A. J. Häusler, A. Saccon, A. P. Aguiar, J. Hauser, and A. M. Pascoal, "Energy-optimal motion planning for multiple robotic vehicles with collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 867–883, 2015.

[13] Y. Zhao, Y. Wang, M. Zhou, and J. Wu, "Energy-optimal collision-free motion planning for multiaxis motion systems: An alternating quadratic programming approach," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 1, pp. 327–338, 2018.

[14] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*. Springer, 2011, pp. 3–19.

[15] J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in *Distributed autonomous robotic systems*. Springer, 2013, pp. 203–216.

[16] R. A. Knepper and D. Rus, "Pedestrian-inspired sampling-based multi-robot collision avoidance," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2012, pp. 94–100.

[17] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot navigation in formation via sequential convex programming," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 4634–4641.

[18] A. Levy, C. Keitel, S. Engel, and J. McLurkin, "The extended velocity obstacle and applying orca in the real world," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 16–22.

[19] D. Hennes, D. Claes, W. Meeussen, and K. Tuyls, "Multi-robot collision avoidance with localization uncertainty." in *AAMAS*, 2012, pp. 147–154.

[20] D. Claes, D. Hennes, K. Tuyls, and W. Meeussen, "Collision avoidance under bounded localization uncertainty," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 1192–1198.

[21] D. Claes and K. Tuyls, "Multi robot collision avoidance in a shared workspace," *Autonomous Robots*, vol. 42, no. 8, pp. 1749–1770, 2018.

[22] B. Gopalakrishnan, A. K. Singh, M. Kaushik, K. M. Krishna, and D. Manocha, "Prvo: Probabilistic reciprocal velocity obstacle for multi robot navigation under uncertainty," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1089–1096.

[23] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[24] H. Fang, N. Tian, Y. Wang, M. Zhou, and M. A. Haile, "Nonlinear bayesian estimation: From kalman filtering to a broader horizon," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 2, pp. 401–417, 2018.

[25] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[26] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.