

RRDV: Robots Rendez-Vous Detection Using Time-Synchronized Ultrasonic Sensors

Trifun Savić

Wattson Elements/Falco

Paris, France

trifun.savic@wefalco.fr

Said Alvarado-Marin

Inria

Paris, France

said-alexander.alvarado-marin@inria.fr

Filip Maksimovic

Inria

Paris, France

filip.maksimovic@inria.fr

Thomas Watteyne

Inria

Paris, France

thomas.watteyne@inria.fr

Abstract—In this paper we propose RRDV, a system for robot-to-robot encounter detection. We use low-cost ultrasound sensor and time-synchronized mobile robots to detect when two robots are facing one another. Ultrasound ranging is triggered by the control application on a computer. The application sends a ranging command to the gateway, which broadcasts it to the mobile robots over the radio. Robots synchronize their ultrasound trigger pin with the start of frame event and send back the notifications with measured distances using Time-Division Multiple Access (TDMA). The system then finds the encounters by searching for timestamps where the difference in distance reported by two robots is less than 1 cm. In the current implementation, the system achieves a 20 Hz distance measurement update rate. RRDV is validated experimentally using 5 mobile robots which are controlled by the users and moved randomly. We implemented a Computer Vision (CV) algorithm for tracking mobile robots as they move and detect when they are facing one another. The CV algorithm is used as the ground truth for the experimental evaluation. The results show 96.7% successfully detected robot encounters, when the duration of the encounter is more than 5 s.

Index Terms—Multi-robot Systems, Swarm, Ultrasound, IoT, Computer Vision

I. INTRODUCTION

The recent research in multi-robot systems on autonomous driving, coordination and formation control enables the development of many applications, including exploration and mapping, search and rescue missions, warehouse automation and industrial production [1]–[3]. Mobile robots of different sizes, shapes and capabilities cooperate to perform complex tasks. Depending on the algorithm used on the mobile robots, multi-robot systems are centralized or distributed. In a centralized system, the information collected by the robots is transferred to the central control unit for processing. The control unit also computes the path planing for the robots taking into account the inputs from a localization system. A distributed system allows for the computation of the robots' positions and the processing of the gathered information to be performed locally [4], [5]. We also distinguish a hybrid and cooperative schemes where the information gathered from one robot is shared with the group of its closest neighbors to improve the system's performance [6].

In any multi-robot system, especially in swarm applications, knowing a position of the mobile robot relative to the infrastructure or to other robots is necessary [7]. Many constrained

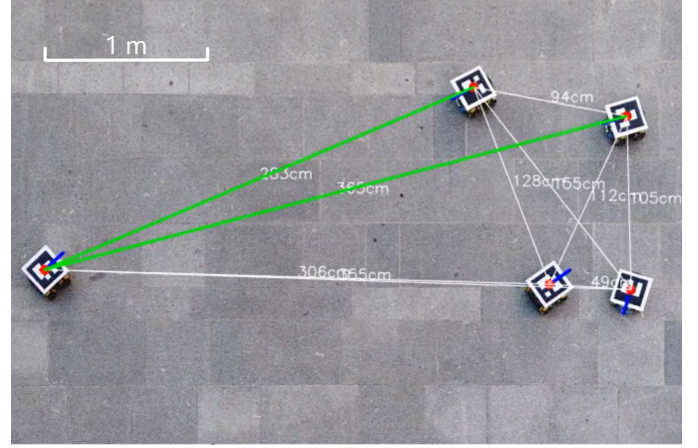


Fig. 1. In our experimental evaluation, five robots drive around randomly. RRDV detects when two robots are facing one another. We call this an “encounter”; it is depicted as a green line.

localization systems have been proposed in the literature, which, depending on the application requirements, enable robot positioning and tracking [8]. The information about the position is crucial for the robot's path planing. In distributed swarm systems, each robot needs to know where other robots are in order to avoid mistaking the other robot for an obstacle. Moreover, robots need to be able to detect when they are close to the docking stations, in order to change/charge their batteries. As an addition to the localization system, it is equally important for robots to be able to detect one another, including to differentiate other robots from permanent obstacles.

This paper proposes RRDV, a system for robot-to-robot detection using an ultra low-cost ultrasound sensor. This type of sensor is ubiquitous in robotic systems, and used to measure the distance to obstacles. Without changing the hardware, we modify the function of this sensor so that, when a robot is in front of another robot, their sensors detect each other's ultrasonic pulse, allowing them to detect one another. The intuition is that, during an encounter, the distance reported by other robots' sensor is very similar. RRDV uses the wireless radio, equipping each robot to tightly synchronize them. Moreover, the robots use their radio to share their measured distances with a central entity, which analyses them and identifies the encounters between robot pairs.

We implement RRDV on a five-robot demonstrator. We use

a computer as the control application we call “gateway”. The gateway broadcasts a frame to all mobile robots signaling the start of a ranging cycle, using the BLE physical layer at 2.4 GHz. The robots arm a timer when they start receiving that frame; when that timer elapses, they activate the trigger pin of the ultrasound sensor. This causes all robots to send an ultrasonic pulse at the same time. When a robot collects the ultrasound measurements, it sends the range measurement in a wireless frame to the gateway using Time-Division Multiple Access (TDMA). The current implementation of the RRDV detects encounters at 20 Hz. To evaluate the performance of our demonstration system, we use Computer Vision (CV) software to track the robots and detect an encounter, using a high-resolution video of the experiment (Fig. 1). We use the results from the CV as our ground truth data and compare it with the RRDV encounter detection. We achieve 96.7% accuracy, when the duration of the encounter is more than 5 s.

Several ultrasound-based localization and ranging solutions are proposed in the literature [9]–[11]. Usually, researchers develop a custom ultrasound transmitter/receiver, time-synchronize them over radio or visible light, and measure Time of Arrival (ToA) to calculate distance. This approach could be used to detect a robot-to-robot encounter. Nevertheless, designing a custom sensor is very costly. Moreover, custom sensors are hard to find and are not easy to integrate in a different multi-robot system. Instead, we use the HC-SR04, perhaps the most ubiquitous low-cost off-the-shelf ultrasound sensor, which typically costs less than 5 USD. Given that the RRDV is completely implemented in firmware/software, using it simply means updating the firmware¹ on an existing robot. Mwaffo et al. [12] present a decentralized algorithm for control and state estimation to autonomously balance a group of robots in a circular formation. The algorithm works in two sequences: “pause”, where robots stop, collect and process the measurements from the sensors, and “go”, where robots accelerate to reach a desired spacing to its closest pursuant robot. The authors also use the same HC-SR04, and mount on a rotating platform on the robot, to measure the distance to other robots. They use the HC-SR04 in its normal way, treating a robot as an obstacle to detect. If they were using RRDV, the robots would explicitly detect an encounter.

The remainder of the paper is organized as follows. Section II describes how we use the HC-SR04 ultrasound sensor. Section III presents RRDV. Section IV details its implementation. Section V discusses the experimental setup. Section VI shows the experimental results. Section VII discusses four applications that could benefit from RRDV. Finally, Section VIII concludes this paper.

II. DETECTING ENCOUNTERS USING THE HC-SR04

RRDV identifies pairs of mobile robots that are facing one another. We call these “encounters”. RRDV does so (re-)using

¹ As an online addition to this paper, all the source code used is published under an open-source BSD licence at <https://github.com/DotBots/RRDV>



Fig. 2. The HC-SR04 is ubiquitous in robotic platforms to measure distance to obstacles. It is very easy to find and very cheap. RRDV re-uses it to detect encounters between robots. Note that the CWL-1601 is an equivalent sensor, operating at 3.3 V.

the off-the-shelf HC-SR04 already present on many robots. As shown in Fig. 2, the HC-SR04 has two ultrasound transducers: one for transmitting an ultrasonic pulse, one for receiving it. In its normal operation, that pulse bounces off an obstacle in front of the sensor; the HC-SR04 uses the speed of sound to turn that round-trip time into distance.

The key is that RRDV tightly synchronizes the robots using wireless. This allows the robots to trigger the transmission of their ultrasonic pulse at exactly the same time. Most of the time, this makes no difference, each robot measures the distance to an obstacle in front of it. But if a pair of robots face one another, the RRDV sensor on one robot receives the pulse from the other robot, rather than its pulse that has bounced off an obstacle. Because the two robots have emitted their pulse at the same time, they both measure the same distance. That is, in a swarm of robots, if two robots measure the same distance, there is a high likelihood they are facing one another. If the robots are measuring the same distance to an obstacle, they will detect a false encounter. We can easily solve this by having robots move together, and verify if the encounter detection is still valid.

Fig. 3 illustrates the operation of the HC-SR04. We call l_1 and l_2 the output of sensor 1 and sensor 2, respectively. If the sensors are facing an obstacle, the distance measurement of the sensors correspond to $l_1 = d_1$ and $l_2 = d_2$ (Fig. 3 a). If the sensors are facing one another, $l_1 = l_2 = l$ and the distance between the two sensors is calculated as $d = 2l$, assuming that the sensors are time synchronized, and we detect an encounter (Fig. 3 b). Due to the inaccuracy of the sensors and the imperfection of the time synchronization, the measured distances l_1 and l_2 will be slightly different. Therefore, in our implementation, if the difference between l_1 and l_2 is below a threshold Δl , we consider they are the same.

III. RRDV

RRDV has three core elements (Fig. 4): control application, gateway, robots. We detail each in the next paragraphs.

RRDV is a centralized system, managed by the **control application**, running on a computer. We implement the control application in Python. The application communicates with the gateway over UART, using High-Level Data Link Control (HDLC) framing. One ultrasound ranging cycle starts when

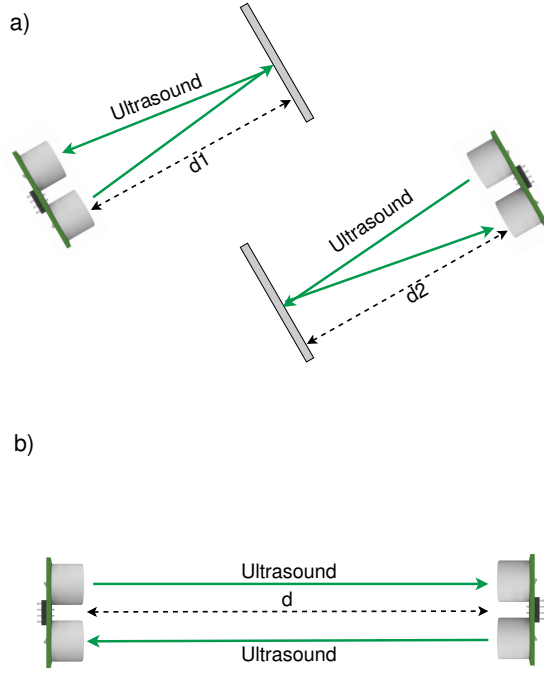


Fig. 3. Ultrasound sensors operation in the RRDV system. *a)* Normal operation, the HC-SR04 measures the distance to some obstacle. *b)* Encounter detection, the HC-SR04 pair measure the distance to one another.

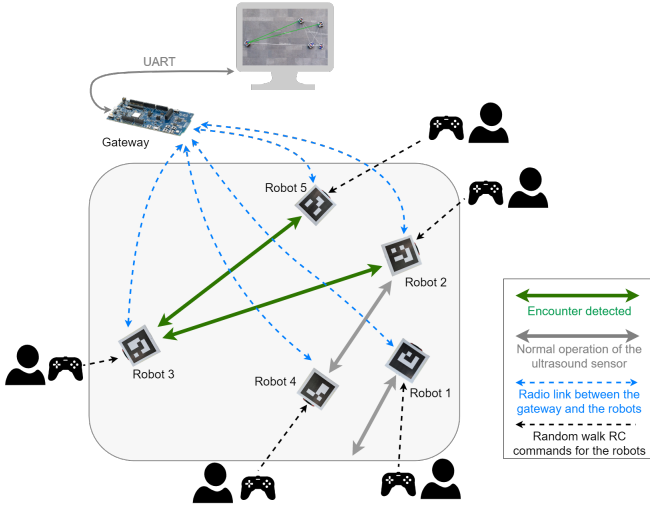


Fig. 4. Diagram of the proposed system demonstrating encounter detection in the experimental evaluation.

the application sends a 32-bit bitmask to the gateway. Each bit in the command corresponds to the unique identifier (ID) of a robot. If a bit in the bitmask is set, the corresponding robot triggers the ultrasound measurement and sends back the notification with measured distance. After issuing a command, the application waits for the incoming serial notification frames from the gateway. These frames contain the recorded distance measurements and robot IDs respectively. When the application receives all the frames, it saves them to a Comma-Separated Values (CSV) file. All robots range at the same time.

One ultrasound ranging cycle takes 50 ms; RRDV has a 20 Hz update rate.

Fig. 5 is the chronogram of the RRDV state machine. Both gateway and robot use a single timer peripheral to implement RRDV. We reuse the Bluetooth Low Energy Long Range (BLE LR) physical layer at 2.4 GHz as a radio link between the mobile robots and the gateway. When the gateway receives an HDLC frame from the control application, it parses the data and prepares a radio frame. The frame contains the gateway ID and 32-bit bitmask. The gateway broadcasts that frame to all robots over its radio, then waits to receive frames from the robots. It waits for a duration that depends on the number of robots and how those are scheduled in the TDMA communication scheme (described in Fig. 6). After parsing the frames it received from the robots, the gateway sends that information over its serial port to the control application.

Robots receive frames from the gateway, and use that as a trigger for their HC-SR04 sensors. A robot starts by listening for frames from the gateway. On reception, it starts a timer and configures it to trigger the HC-SR04 when it expires. It then inspects the received frame, and cancels the timer if its bit is *not* set in the bitmask contained in the frame. This use of the timer allows the HC-SR04 to be triggered at a perfectly deterministic time, without any jitter introduced by the speed of execution of the CPU. After it has triggered the HC-SR04, it captures the output of the “echo” pin, which encodes the distance measured in its pulse width. The robot then waits for its right TDMA timeslot, and sends the measured duration to the gateway over the radio.

IV. IMPLEMENTATION

We implement RRDV state machine shown in Fig. 5 on nRF52840-DK boards. We use one nRF52840-DK connected to a computer as a gateway; the others are installed inside the robots. The nRF52840 System-on-Chip (SoC) has a 64 MHz ARM Cortex-M4 processor and supports different 2.4 GHz radio standards, such as BLE and IEEE 802.15.4. In this implementation, we use the BLE LR PHY physical layer at 125 kbps, but implement our own protocol stack. The nRF52840 is equipped with Programmable Peripheral Interconnect (PPI) that allows precise synchronization between peripherals, for example timers, radio, GPIO, etc. This feature is crucial to our implementation of RRDV as it allows precise timing and eliminates the need for CPU activity to carry on tasks for synchronization. In our case, we chain all events that are shown in Fig. 5 using PPI in order to achieve tight time-synchronization of the system. This is essential for good performance: as we are measuring distance using the ultrasound signals, we need synchronization accuracy below 100 μ s to have cm-level distance accuracy. In order to verify the synchronization error, we trigger a pin on each robot and observe them using an oscilloscope. We collect 1000 samples; the mean absolute synchronization error is 0.05 ns with a standard deviation of 109 ns. This indicates our implementation, and the hardware support of the nRF52840, offer a



Fig. 5. Chronogram of the RRDV state machine representing the activity of a gateway and mobile robot. The figure shows their radio activity (red line), events and timers, value of the global variables and the state of the debug pins.

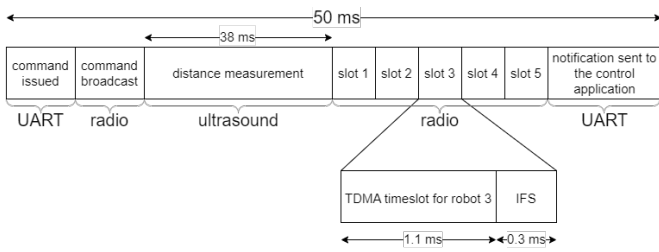


Fig. 6. One ultrasound ranging cycle in the RRDV system.

level of synchronization which is perfectly compatible with our application.

As shown in Fig. 2, the HC-SR04 has two logical pins. The Trig pin is an input pin to the HC-SR04; the nRF52840-DK sends a 10 us pulse to start the ultrasound transmission. The Echo pin is the output pin of the HC-SR04; the sensor encodes the measured distance in the pulse width of the echo pin. According to the sensor's datasheet, 58 us pulse width

corresponds to 1 cm distance. We measure the sensor's field of view by having two HC-SR04 face one-another. We move one along a 1 m cone from the other, until they stop receiving each other's ultra-sonic pulse. This happens at a relative angle of 20°, indicating their field of view is a cone of aperture 40°.

To test our approach we use five off-the-shelf Exost Hyper-Drift radio controlled cars. The cars communicate with their remote controllers using a proprietary 2.4 GHz protocol allowing up to 10 cars to be driven at the same time. Even though the RC cars and the RRDV devices both communicate on the same frequency band, we have not noted any interference between them. As shown in Fig. 7, we install an nRF52840-DK connected to a HC-SR04 inside each car's chassis. In this experiment, RRDV is completely independent from the RC car.

V. EXPERIMENTAL SETUP

We place five cars in a 7×4.2 m² area, each remote controlled by five volunteers. A computer is on the side,

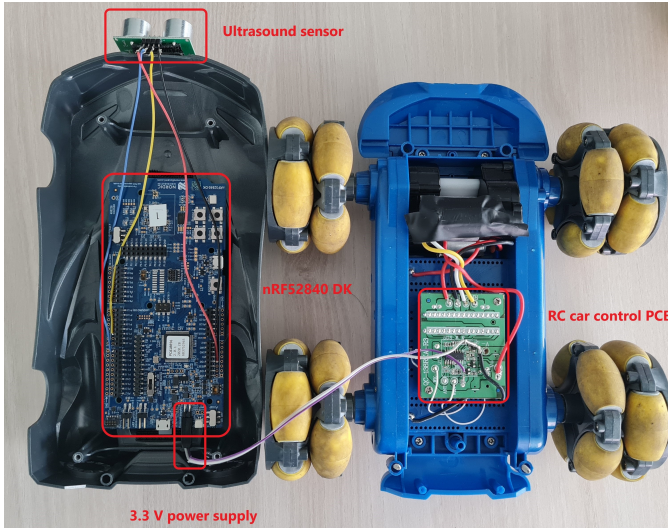


Fig. 7. An RC car equipped with an nRF52840-DK and a HC-SR04. We used five of these to evaluate the performance of RRDV.

running the software, and having the nRF52840-DK and HC-SR04 in each car take ultrasonic distance measurements as the cars are driven around in random patterns. During the experiment, we place a GoPro HERO Black 10, 5 m about the area and record a video at 4K 30 fps, which we will use as ground truth. The entire experiment takes 12.5 min.

We place large ArUco markers [13] on top of each car, so we can determine the position and orientation in each frame of the video using CV. We choose ArUco markers because they are easy to visually track, are well supported by the OpenCV [14] computer vision library, and provide both position and orientation data. To determine in the video when two cars are facing one another, we “draw” 40° cones in front of each ArUco marker, representing the field of view of the HC-SR04. If two cars are contained in each other’s cone at the same time, with no obstacle in-between, we consider that as an encounter. In case a car is involved in two or more encounters, we consider only the one with the shortest distance as valid. We place visible markers in corner of the area to correct the camera perspective with a homography transformation in the CV software. We have the computer display a large timer on its screen, and make sure the camera records that, so we can efficiently synchronize the video to the RRDV data.

VI. RESULTS

This section details the analysis of the data, including fine-tuning of the encounter detection routing running on the computer, and the resulting performance of RRDV. We evaluate the performance in terms of success (both video and RRDV detect an encounter), miss (the video indicates there is an encounter, but RRDV misses it) and false positive (RRDV detects an encounter, but the video shows it didn’t happen).

Based on the speed of the cars, we know that an encounter lasts for at least 1 s. We also know that two successive encounters between the same two cars cannot happen within

TABLE I
A THRESHOLD OF 1.0 CM RESULTS IN THE HIGHEST SUCCESS.

threshold Δl	success	miss	false positive
0.5 cm	53.90%	46.10%	5.00%
1.0 cm	72.30%	27.66%	12.82%
1.5 cm	71.63%	28.37%	17.89%
2.0 cm	69.50%	30.50%	20.33%
2.5 cm	65.96%	34.04%	23.77%

3 s. We hence start our analysis by cleaning up the data, removing spurious measurements based on these two timing. These measurements can be caused by dropped ultrasonic pulses.

We then consider the value of threshold Δl , below which distance measurements are considered the same. This value allows for a trade-off between success and false positive. We compute success and false positives for five value of Δl , see Table I. A threshold of 1.0 cm results in the highest success. We use that value for the remainder of this section.

Fig. 8 is a timeline of the different encounters. It shows 10 sub-plots, one of each robot pair (there are 5 robots, so $\frac{5 \cdot (5-1)}{2} = 10$ undirected pairs of robots). In each, the lines at the top represent the periods over which the camera witnesses an encounter; the dots at the bottom represent encounters as detected by RRDV. When both camera and RRDV detect an encounter at the same time, that’s a success. A camera detection alone is a miss. A RRDV detection alone is a false positive. The performance of RRDV consists in classifying the events as success, miss and false positive, and counting the number of encounters of each category.

Fig. 9 shows the results for different distances between robots and different encounter duration. The top plot show that, the closer the robots, the better RRDV works and that, with this setup, RRDV is best used for robots are less than 2 m apart. Similarly, the bottom plot shows that, the longer robots stay facing one another, the better RRDV works (i.e. the higher the likelihood it will successfully detect an encounter). If the robots face one another for more than 5 s, the success rate is 96.7%.

In this implementation, RRDV offers an encounter detection frequency of 20 Hz. This depends on the number of robots. To allow for fast update rates when there is a large number of robots, the control application could be optimized by having it dynamically modify the bitmap in the command frame. This way, pairs of robots that are likely to be facing one another could be ranging more often, while others, which are known for example to be in different areas of the deployment, would not be.

VII. APPLICATIONS BENEFITING FROM RRDV

Existing multi-robot systems that use both radio communication and an ultrasound sensor could easily take advantage of RRDV without requiring any hardware update. This section lists four applications that could benefit from RRDV.

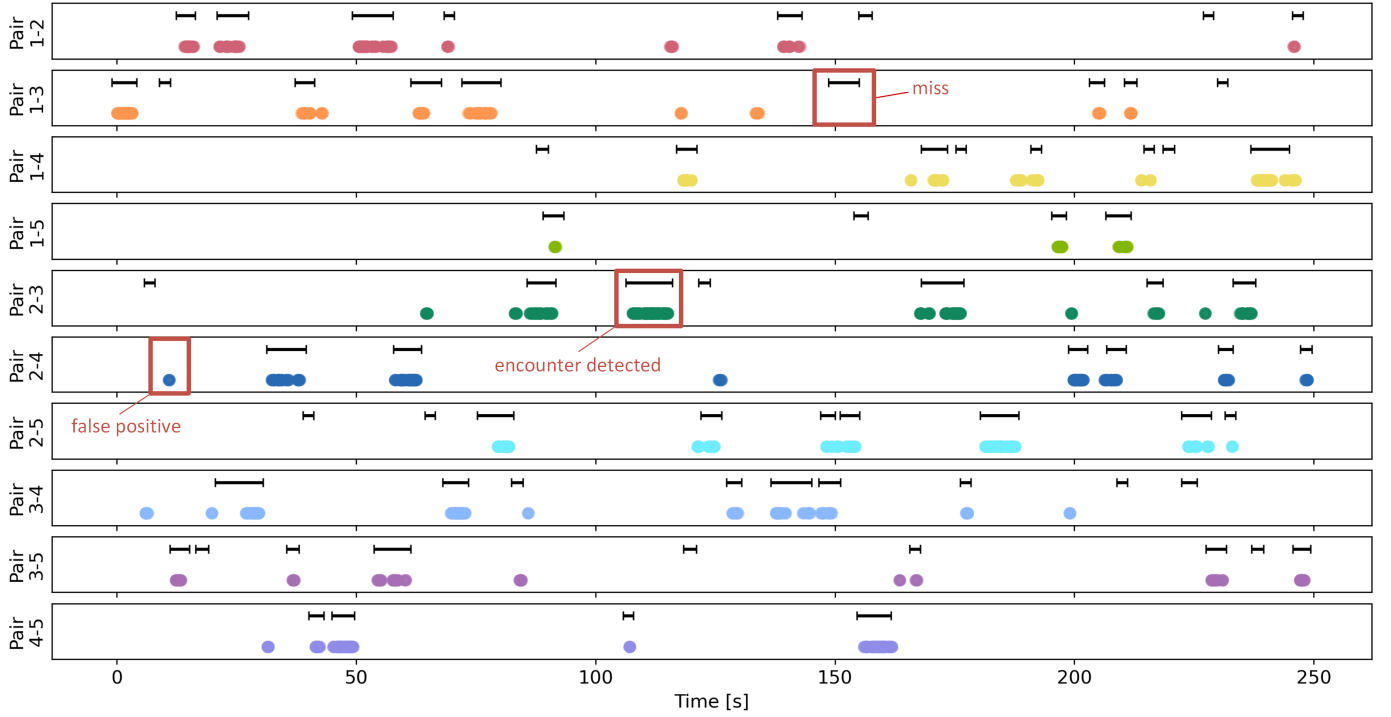


Fig. 8. Timeline of the encounters, as detected by the camera and RRDV. There is a subplot for each robot pair. A line at the top of a subplot represents the periods over which the camera witnesses an encounter. A dots at the bottom of a subplot represents encounters as detected by RRDV.

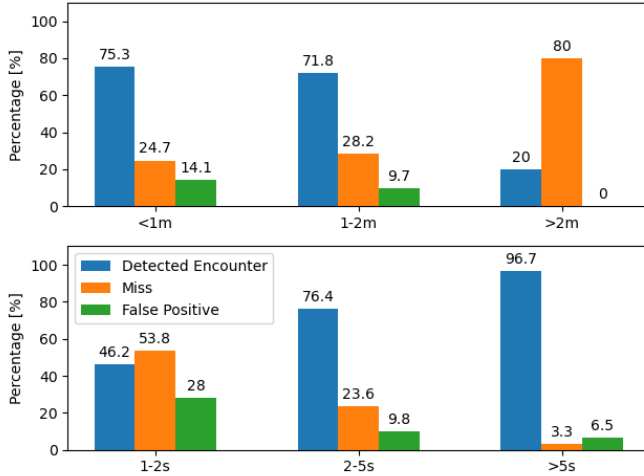


Fig. 9. Encounter detection accuracy at different distances between mobile robots (top) and different duration of the encounters (bottom).

Collision avoidance. When many robots move in an area to perform tasks, there is a high likelihood they will bump into one another, even when complex algorithms are used for controlling their movement [15]. RRDV can be used in the navigation routines on a robot to differentiate between static obstacles and moving robots, resulting in more efficient swarm movements.

Mapping. Swarm of robots are used to collectively carry out exploration and mapping of unknown areas [7]. These robots

can be updated with RRDV to avoid mistaking another robot for an obstacle. Since RRDV does not alter a robot's ability to range, the sensor on a robot would continue to be used in its mapping routines.

Docking stations. When robots need to charge/change their batteries, they typically need to dock [16]. Detecting when a robot is in front of the docking station is necessary for assisting the robot to stop at the right place. RRDV can be used to recognize when a robot is near the docking station. Using RRDV, a robot can position itself at the right angle and stop safely at the docking station. Here, both robots and docking stations are equipped with RRDV.

Security. Using RRDV, a robot can recognize that it is close to another robot. They can use this to trigger the exchange of messages over the radio and establish a secure communication link. RRDV can also be used for physical layer security: two robots can encode their security keys by modulating the distance measurement (driving back-and-forth) removing the opportunity for an eavesdropper to hear the keys.

VIII. CONCLUSION

This paper introduces RRDV, that allows robots to detect they are facing one another in a multi-robot systems. What makes RRDV powerful is that it reuses ultrasonic sensors such as HC-SR04 already present on many robots; RRDV can be seen as a software update. Robots are time-synchronized and use that timing information to trigger the ultrasonic pulses of the sensors. The system detects an encounter when the two robots involved report the same distance. We validate

our system by equipping five RC cars. We implement a CV algorithm to track the positions of those cars, information we use as ground truth in our analysis. The results show that RRDV has an accuracy of 96.7%, when the robots are facing each other for more than 5 s.

REFERENCES

- [1] H. Ma, W. Hönig, L. Cohen, T. Uras, H. Xu, T. S. Kumar, N. Ayanian, and S. Koenig, "Overview: A Hierarchical Framework for Plan Generation and Execution in Multirobot Systems," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 6–12, 2017.
- [2] C. K. Peterson, D. W. Casbeer, S. G. Manyam, and S. Rasmussen, "Persistent intelligence, surveillance, and reconnaissance using multiple autonomous vehicles with asynchronous route updates," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5550–5557, 2020.
- [3] A. Gautam and S. Mohan, "A Review of Research in Multi-Robot Systems," in *International Conference on Industrial and Information Systems (ICIIS)*. IEEE, 2012, pp. 1–5.
- [4] N. Ahmed, J. Cortes, and S. Martinez, "Distributed Control and Estimation of Robotic Vehicle Networks: Overview of the Special Issue," *IEEE Control Systems Magazine*, vol. 36, no. 2, pp. 36–40, 2016.
- [5] Y. Rizk, M. Awad, and E. W. Tunstel, "Cooperative Heterogeneous Multi-Robot Systems: A Survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, pp. 1–31, 2019.
- [6] H. Rezaee and F. Abdollahi, "A Decentralized Cooperative Control Scheme With Obstacle Avoidance for a Team of Mobile Robots," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 347–354, 2013.
- [7] R. Abu-Aisheh, F. Bronzino, M. Rifai, B. Kilberg, K. Pister, and T. Watteyne, "Atlas: Exploration and Mapping with a Sparse Swarm of Networked IoT Robots," in *International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2020, pp. 338–342.
- [8] T. Savić, X. Vilajosana, and T. Watteyne, "Constrained Localization: A Survey," *IEEE Access*, vol. 10, pp. 49 297–49 321, 2022.
- [9] O. De Silva, G. K. Mann, and R. G. Gosine, "An Ultrasonic and Vision-Based Relative Positioning Sensor for Multirobot Localization," *IEEE Sensors Journal*, vol. 15, no. 3, pp. 1716–1726, 2014.
- [10] F. Rivard, J. Bisson, F. Michaud, and D. Létourneau, "Ultrasonic Relative Positioning for Multi-Robot Systems," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2008, pp. 323–328.
- [11] S. Elvira, A. de Castro, and J. Garrido, "ALO: An Ultrasound System for Localization and Orientation Based on Angles," *Microelectronics Journal*, vol. 44, no. 10, pp. 959–967, 2013.
- [12] V. Mwaffo, J. S. Curry, F. L. Iudice, and P. De Lellis, "Pause-and-Go Self-Balancing Formation Control of Autonomous Vehicles Using Vision and Ultrasound Sensors," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 6, pp. 2299–2311, 2020.
- [13] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.
- [14] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
- [15] Y. Zhou, H. Hu, Y. Liu, and Z. Ding, "Collision and Deadlock Avoidance in Multirobot Systems: A Distributed Approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1712–1726, 2017.
- [16] B. W. Minten, R. R. Murphy, J. Hyams, and M. Micire, "Low-Order-Complexity Vision-Based Docking," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 922–930, 2001.