



Chunhyung Cho¹ and Jonghoek Kim^{2,*}

- ¹ Electronic and Electrical Convergence Department, Hongik University, Sejong 30016, Korea
- ² Electronic and Electrical Department, Sungkyunkwan University, Suwon 16419, Korea

* Correspondence: jonghoek@gmail.com

Abstract: This paper considers multi-robot systems, such that each robot has a radar for detecting its neighbor robots. We consider a practical scenario in which a radar sensor contains measurement noise, and the environmental disturbance generates process noise in a robot's maneuvering. We consider a 3D scenario such that the network can be split initially. For instance, complete failures of one or more robots can split the network. Considering 3D environments, the goal of our paper is to let all robots rendezvous in a distributed manner so that the network connectivity can be recovered even after the network is split. Robust distributed rendezvous control is designed so that the network connectivity is maintained (or recovered) during the maneuvering of a robot. To recover the network connectivity, we adaptively control the robot's radar footprint by increasing the transmission power level (adjust the amplifier in the transmitter). To the best of our knowledge, this paper is novel in applying a radar with a variable sensing range in order to make all robots rendezvous in 3D environments. We address MATLAB simulations to demonstrate the outperformance of our rendezvous approach with variable range radars by comparing it with the state-of-the-art in multi-robot rendezvous controls.

Keywords: networked robots; sensor networks; distributed rendezvous control; network connectivity; variable sensing range; variable range radar

1. Introduction

Networked robots can perform many tasks, such as formation controls [1–3] or workspace monitoring [4]. Such robot networks have become feasible due to advances in communication networking [5–8].

Suppose that each robot does not know its position in the global coordinate system, i.e., a robot is not localized using global positioning systems (GPS). We require a *distributed* controller such that every robot moves based only on relative position measurements.

During the robots' maneuvering in formation controls, robots need to maintain network connectivity. In order to make all robots gather close to each other, distributed rendezvous controls can be applied occasionally. In other words, distributed rendezvous controls are the most basic coordination task for a network of mobile robots. Multi-robot rendezvous is an initial step for the creation of more complicated formation tasks and can be particularly useful for autonomous recharging or self-maintenance [9].

This article considers the distributed rendezvous problem of multiple robots, such that each robot can only measure nearby (neighbor) robots using a radar sensor, and each robot moves based on radar sensor measurements. Since GPS is not accessible, a robot cannot move towards a specified rendezvous point directly.

For rendezvous controls, each robot uses a radar sensor, whose footprint is proportional to the signal intensity (power) of the transmitter [10]. Supply modulated transmitters for variable amplitude radar systems were addressed in [11]. Suppose that a robot, say A, emits signals for detecting its neighbor robots. Once the signals generated from A are reflected from a nearby robot, say B, then A can estimate the elevation angle, azimuth



Citation: Cho, C.; Kim, J. Robust Distributed Rendezvous Using Multiple Robots with Variable Range Radars. *Appl. Sci.* 2022, *12*, 8535. https://doi.org/10.3390/ app12178535

Academic Editor: Dario Richiedei

Received: 4 August 2022 Accepted: 25 August 2022 Published: 26 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). angle, and range to *B* using a 3D multiple signal classification (MUSIC) algorithm [12]. In this way, *A* can measure the relative position of its neighbor robot *B*.

Our article considers a practical scenario in which radar has a limited sensing range while containing measurement noise. Further, the environmental disturbance generates process noise in a robot's maneuvering. For instance, currents (for underwater robots) or wind gusts (for aerial robots) can generate process noise in robots' maneuvers.

To the best of our knowledge, distributed rendezvous control in the literature assumed that all robots are initially deployed such that every robot is connected to any other robot utilizing multi-hop links. This assumption is called the *initial connectivity assumption*, which implies that the entire network is not split initially. Our 3D rendezvous control is developed to cope with a case where this assumption is not satisfied initially.

The initial connectivity assumption may not hold in various scenarios. In practice, sensing measurement noise or process noise in a robot's motion can split the initial robot network. Moreover, complete failures of one or more robots can split the initial network. For example, Figure 1 plots a scenario where the failure of one robot splits the network. In this figure, a robot is marked with a dot, and the network connection between two robots is plotted with line segments. In the left subfigure, the robot with a red dot is completely broken. In this case, the networked system changes to the network in the right subfigure. See that the complete failure of one robot (red dot) splits the network.



Figure 1. This figure plots a scenario where the failure of one robot splits the network. In this figure, a robot is marked with a dot, and na etwork connection between two robots is plotted with line segments. In the left subfigure, the robot with a red dot is completely broken. In this case, the network system changes to the network in the right subfigure. See that the complete failure of one robot (red dot) splits the network.

Considering 3D environments, the goal of our paper is to let all robots rendezvous in a distributed manner so that the network connectivity can be recovered even in the case where the network is split. In other words, we develop a 3D distributed rendezvous control so that all robots become connected to each other even after the network is split.

In order to handle the case where the initial connectivity assumption does not hold, our paper assumes that a robot's radar sensor can enlarge its footprint by increasing the transmission power level (adjusting the amplifier in the transmitter). In other words, we consider the case where a robot has a variable range radar, such that its sensor footprint is adjustable. In our paper, we control the sensing range of each robot adaptively in order to achieve distributed rendezvous even in the case where the entire network is split initially. To the best of our knowledge, our paper is novel in applying a radar with a variable sensing range in order to make all robots rendezvous in 3D environments. The proposed distributed rendezvous control is novel since we do not require the initial connectivity assumption.

Using computer simulations, we show that our rendezvous control with an adaptive sensing range can handle the case where the initial connectivity assumption is not satisfied. We show the superiority of the proposed 3D rendezvous algorithm compared to the state-of-the-art in multi-robot rendezvous controls [13–15].

The article is organized as follows: Section 2 presents the literature review of this paper. Section 3 introduces assumptions and definitions. Section 4 introduces our distributed

rendezvous algorithm. Section 5 presents the computer simulation results of our distributed rendezvous algorithm. Section 6 provides the conclusions.

2. Literature Review

Considering 2D environments, many papers [16–21] addressed distributed rendezvous algorithms. Reference [18] addressed distributed rendezvous algorithms, considering limited visibility of identical robots. Reference [16] addressed connectivity-preserving control strategies for the rendezvous problem. Considering 2D environments, [17] addressed distributed rendezvous algorithms considering robots with bounded control inputs. Reference [13] presented rendezvous controllers based on local sensing measurements of every robot. Reference [22] presented distributed rendezvous controllers so that multiple robots rendezvous at multiple rendezvous points simultaneously.

Considering cluttered 3D environments, [23] addressed distributed control laws to make all spherical underwater robots rendezvous. However, [23] requires that a robot can be identified using local sensors and that a tree structure is built based on local sensing measurements. However, the radar sensors considered in our paper cannot be used to identify one robot from another.

The authors of [9,24,25] proposed a distributed controller to achieve rendezvous, even when some robots in the system are broken. The references [9,24,25] assumed that a broken robot does not move or does not follow a prescribed policy. Reference [9] assumed that probabilities associated with robot failures are available, and their rendezvous algorithms rely on these probability distributions to compute stochastic optimal control strategies to maximize performance in an expected sense. However, in practice, robot failure probabilities may not be available. In our paper, we do not use the probabilities associated with robot failures. Recall [9,24,25] assumed that a broken robot does not move or does not follow a prescribed policy. However, our paper considers the case where a robot is completely broken. For instance, enemy attacks can completely destroy a robot, as plotted in Figure 1. This distinguishes our paper from [9,24,25].

For achieving distributed rendezvous in arbitrary dimensions, circumcenter algorithms were proposed in [13–15] to avoid the loss of existing links between robots. Circumcenter algorithms do not require that a robot can be identified using local sensors. Thus, circumcenter algorithms are suitable for distributed rendezvous using radar sensors. Furthermore, in circumcenter algorithms, a tree structure does not have to be built based on local sensing measurements.

The proposed rendezvous controls with variable range radars are based on circumcenter algorithms. Similarly to circumcenter algorithms, our rendezvous approach can handle arbitrary dimensions and does not require that a robot can be identified using local sensors.

As far as we know, other papers on rendezvous controls did not consider sensor specifications required for multi-robot rendezvous. Our paper is distinct from [13–15] since we consider robots with specific sensors: variable range radars.

Circumcenter algorithms in [13–15] require the initial connectivity assumption. To the best of our knowledge, every distributed rendezvous control in the literature requires the initial connectivity assumption. In our paper, we control the sensing range of each robot adaptively in order to achieve distributed rendezvous even in the case where the initial connectivity assumption does not hold.

Considering 3D environments, we design a distributed rendezvous control with a variable sensing range so that the network connectivity is maintained (or recovered) during a robot's maneuvering. Once the rendezvous is achieved, the sensing range of every robot can decrease in order to reduce the power consumption of its radar.

To the best of our knowledge, our article is novel in applying radars with a variable sensing range in order to make all heterogeneous robots rendezvous in 3D environments. Our distributed rendezvous control is unique since we do not require the initial connectivity assumption. Using computer simulations, we show the superiority of the proposed 3D rendezvous algorithms compared to the state-of-the-art circumcenter algorithms (in [13–15]).

3. Assumptions and Definitions

We introduce assumptions and definitions that are utilized in our distributed rendezvous control. G = (V(G), E(G)) is a graph with vertex set V(G) and edge tuple E(G) [26]. Let min(a, b) denote a small value between a and b.

There are *N* robots in total. Let u_i denote the *i*-th robot, where $i \leq N$. $\mathbf{u}_i \subset \mathcal{R}^3$ is the 3D position of robot u_i where $i \leq N$.

We consider discrete-time systems. Let dt denote the sampling interval. $\mathbf{u}_i(k)$ denotes \mathbf{u}_i at time-step k.

Considering the process noise, such as wind, the motion model of u_i is

$$\mathbf{u}_i(k+1) = \mathbf{u}_i(k) + \mathbf{v}_i(k) * dt + \mathbf{n},\tag{1}$$

where $\mathbf{n} = [randn * D, randn * D, randn * D]$ where D > 0 is a constant presenting the process noise strength, and *randn* generates a Gaussian noise with mean 0 and variance 1.

In (1), $\mathbf{v}_i(k)$ is the velocity input of u_i at time-step k. This simple dynamic model (1) is commonly used in multi-robot systems [27–33].

This article further considers a practical scenario where a robot moves with a bounded speed. The maximum speed of u_i is M_s . In other words, $\|\mathbf{v}_i(k)\| \le M_s$ for all k and i. M_s is determined by the hardware specifications of a robot.

Each robot has a radar with a limited sensing range in order to measure the relative position (elevation, azimuth, and range) of a nearby robot. We assume that a robot's radar sensor can increase its footprint by increasing the transmission power level (adjust the amplifier in the transmitter). In other words, we consider the case where a robot has a variable range radar, such that its sensor footprint is adjustable. Since each robot can change its radar sensing range, we consider heterogeneous robots with a distinct sensing range.

Let us define $S_i(k)$ as the radar sensing range of u_i at time-step k. S^u is the upper bound for $S_i(k)$ for all i and k.

Suppose that $\|\mathbf{u}_j(k) - \mathbf{u}_i(k)\| \le S_i(k)$. In this case, u_i can measure u_j using its radar. Note that increasing the footprint of the radar cannot cause a detection in the case of obstructed LOS.

Let $\mathbf{R}_{j,i}(k)$ denote the relative position (radar) measurement of u_i at time-step k. $\mathbf{R}_{j,i}(k)$ is given as the radar of u_i measures u_j . Considering the radar measurement noise, we have

$$\mathbf{R}_{j,i}(k) = \mathbf{u}_j(k) - \mathbf{u}_i(k) + \mathbf{n}_R,\tag{2}$$

where $\mathbf{n}_R = [randn * D_R, randn * D_R, randn * D_R]$ where $D_R > 0$ is a constant presenting the measurement noise strength, and *randn* generates a Gaussian noise with mean 0 and variance 1. For instance, \mathbf{n}_R can be generated due to the shape of a robot, since radar sensors measure the boundary of a nearby robot.

We introduce the adjacency graph, A = (V(A), E(A)), in which every vertex in V(A) represents a robot. Every directed edge, say $\{u_i, u_j\} \in E(A)$, indicates that u_i can detect u_j using its radar. $\{u_i, u_j\} \in E(A)$ implies that $||\mathbf{u}_j(k) - \mathbf{u}_i(k)|| \le S_i(k)$.

See that *A* is a directed graph with directed edges. $\{u, v\}$ is distinct from $\{v, u\}$, since an edge is directed. Even in the case where *u* can measure *v* using its radar, *v* may not measure *u* using its radar. This is due to the fact that the sensing range of *u* may be distinct from that of *v*.

We say that v is a *neighbor* of u in the case where u can detect v using its radar. N_u denotes the set of neighbors of u.

We say that all robots achieved *rendezvous* in the case where the relative distance between any two robots is less than a positive constant, say $\delta > 0$. The goal of our paper is to let all robots rendezvous using distributed control laws, which are based on radar measurements with a variable sensing range.

4. Distributed Rendezvous Controllers

Our paper is based on circumcenter algorithms in [13–15]. The original circumcenter algorithms are as follows.

We need to address one definition beforehand. Let p_s denote a set of points in \mathcal{R}^3 . The center of the smallest-radius circle enclosing p_s is called the *circumcenter* of p_s .

In circumcenter algorithms, each robot performs the following tasks at each time-step k:

- 1. it detects its neighbors using its local sensor (radar in our paper).
- 2. it computes the circumcenter of the point set comprised of its neighbors and of itself.
- 3. it moves toward this circumcenter while maintaining connectivity with its neighbors.

The authors of [15] showed that, when implemented over a 1D space, it is not necessary to enforce the connectivity constraint, as in the third step of the above algorithm. Assuming that a robot can access its orientation in global coordinate systems (inertial measurement units on each robot can be used for orientation measurements), we can extend the 1D algorithm to arbitrary dimensions by means of a circumcenter algorithm implemented in parallel. The parallel circumcenter algorithm in [15] is as follows.

Each robot performs the following tasks at each time-step *k*:

- 1. it detects its neighbors using its local sensor (radar in our paper).
- 2. it projects the detected neighbor positions to each axis of its frame.
- 3. it computes the circumcenters of each of the projected sets of positions on each axis.
- 4. it moves to the point whose coordinates are given by each of those circumcenters while maintaining connectivity with its neighbors.

The computation of the circumcenter of a bounded set is a strictly convex problem and, in particular, a quadratically constrained linear program [14]. In the parallel circumcenter algorithm, a circumcenter is derived over a 1D space, not over a 3D space. Thus, the parallel circumcenter algorithm is desirable, considering the computational load. Therefore, our rendezvous approach is based on the parallel circumcenter algorithm.

In the last step of the above algorithm, a robot, say u_i , moves to the point whose coordinates are given by each of those circumcenters while maintaining connectivity with its neighbors. We next explain this maneuver in detail.

Let $\mathbf{T}(k)$ denote the point whose coordinates are given by each of those circumcenters. Recall that M_s denotes the maximum speed of a robot. We define

$$\mathbf{Q}(k) = \mathbf{u}_i(k) + M_s * dt * \frac{\mathbf{T}(k) - \mathbf{u}_i(k)}{\|\mathbf{T}(k) - \mathbf{u}_i(k)\|}.$$
(3)

 $\mathbf{Q}(k)$ is set such that $\|\mathbf{Q}(k) - \mathbf{u}_i(k)\| = M_s * dt$. In addition, $\mathbf{Q}(k)$ is set such that it exists on the line segment connecting $\mathbf{u}_i(k)$ and $\mathbf{T}(k)$.

Let $\lambda \leq 1$ denote a positive constant. Let

$$\mathbf{q}(\lambda) = (1 - \lambda) * \mathbf{u}_i(k) + \lambda * \mathbf{Q}(k)$$
(4)

denote a point on the line segment connecting $\mathbf{u}_i(k)$ and $\mathbf{Q}(k)$. As λ changes from 0 to 1, $\mathbf{q}(\lambda)$ changes from $\mathbf{u}_i(k)$ to $\mathbf{Q}(k)$. Since $\|\mathbf{Q}(k) - \mathbf{u}_i(k)\| = M_s * dt$,

$$\|\mathbf{q}(\lambda) - \mathbf{u}_i(k)\| \le M_s * dt \tag{5}$$

is satisfied as λ changes from 0 to 1.

Let $I = \begin{bmatrix} \frac{1}{M}, \frac{2}{M}, ..., 1 \end{bmatrix}$ denote the equally spaced interval between 0 and 1. There are *M* elements in interval *I*. While changing λ from the first element in *I* to the last element, we find **q**(λ), such that

$$\|\mathbf{q}(\lambda) - \mathbf{w}\| < S_i(k) \tag{6}$$

for all $w \in N_{u_i}$. This implies that $\mathbf{q}(\lambda)$ maintains connectivity with all robots in N_{u_i} .

Once $\mathbf{q}(\lambda)$ is found, which satisfies (6) for all neighbors of u_i , we let λ^* denote the found λ . Then, $\mathbf{q}(\lambda^*)$ is set as the waypoint of u_i at time-step k.

Using (5), we have $\|\mathbf{u}_i(k) - \mathbf{q}(\lambda^*)\| \le M_s * dt$. Thus, u_i moves toward $\mathbf{q}(\lambda^*)$ by setting its velocity command as

$$\mathbf{v}_i(k) = \frac{\mathbf{q}(\lambda^*) - \mathbf{u}_i(k)}{dt}$$
(7)

in (1).

In the case where no element in *I* satisfies (6), then u_i stops moving by setting

v

$$_{i}(k) = 0 \tag{8}$$

in (1).

Updating the Sensing Range Adaptively

The circumcenter algorithm was originally introduced in [13]. Consider the ideal case where there is no process noise or measurement noise. In addition, assume that the initial connectivity assumption is satisfied. In this ideal case, ref. [14] proved the convergence of the circumcenter algorithm for any switching sequence of adjacency graphs that contains a strongly connected graph every *l* instants of time, for some fixed l > 0.

In practice, the initial connectivity assumption may not hold. Moreover, process noise, limited communication range, or measurement noise can disconnect the network frequently. In order to handle these practical problems, we need to increase network connectivity as much as possible.

The proposed strategy is to increase the network connectivity by increasing the sensing range of each robot adaptively. By increasing the sensing range of each robot adaptively, we can make the switching sequence of adjacency graphs contain a strongly connected graph more frequently. Then, according to [14], rendezvous can be achieved more robustly. Once the rendezvous is achieved, the sensing range of every robot can shrink to reduce the power consumption of its radar.

Recall that $S_i(k)$ is the sensing range of u_i at time-step k. In addition, recall that S^u is the upper bound for $S_i(k)$ for all i and k.

Let $N^{min} > 0$ denote a lower threshold for the number of neighbors. Recall that N_u denotes the set of neighbors of u. In the case where $N_{u_i} < N^{min}$, we need to increase the number of neighbors for u_i . In the case where $N_{u_i} < N^{min}$, $S_i(k)$ increases using

$$S_i(k+1) = \min(S_i(k) + \beta, S^u).$$
⁽⁹⁾

Here, $\beta > 0$ is a constant. Using (9), the sensing range for u_i increases, until it reaches S^u . Note that in the case where $N_{u_i} \ge N^{min}$, (9) is not applied.

In (9), $\beta > 0$ is a tuning parameter. As β increases, the sensing range of u_i can increase more rapidly. A large sensing range can increase the network connectivity, which improves the rendezvous performance. However, a large sensing range increases the power consumption of a robot.

Moreover, N^{min} is a tuning parameter in our controls. As N^{min} increases, the network connectivity increases, which improves the rendezvous performance. However, increasing N^{min} increases the power consumption of each robot since the sensing range of a robot needs to increase using (9). The effect of changing N^{min} is discussed in the Simulation section.

The original circumcenter algorithms in [13–15] require the initial connectivity assumption. For instance, consider the case where a robot is not connected to any other robot initially. This isolated robot cannot move based on local sensing measurements since it has no local interaction initially.

In our paper, we can remove an isolated robot by increasing its radar sensing range. Thus, our paper does not require the initial connectivity assumption. Once the rendezvous is achieved, the sensing range of every robot can decrease in order to reduce the power consumption of its radar.

5. Simulation Results

This section addresses the MATLAB simulation results to demonstrate the effectiveness of the proposed distributed rendezvous scheme with variable range radars. Using computer simulations, we show the superiority of the proposed 3D rendezvous algorithm compared to the original circumcenter algorithms in [13–15].

The simulation settings are as follows. In MATLAB simulations, the radar sensing noise (2) uses $D_R = 0.01$ km. In addition, we consider the case where u_1 is static. Thus all robots need to rendezvous at u_1 . Since u_1 is static, the motion model of u_1 is

$$\mathbf{u}_1(k+1) = \mathbf{u}_1(k) + \mathbf{n} \tag{10}$$

instead of (1). In our MATLAB simulations, (1) and (10) use $\mathbf{n} = [randn * D, randn * D, randn * D]$. Here, D > 0 is a constant presenting the process noise strength, and *randn* generates a Gaussian noise with a mean of 0 and a variance of 1. We use D = 0.05 km.

At time-step 0, we set the maximum sensing range $S_i(0)$ as $S_i(0) = 5$ km for all $i \in \{1, 2, ..., N\}$. The upper bound for $S_i(k)$ is set as $S^u = 50$ km. In addition, (9) uses $\beta = 5$ km, and $N^{min} = 5$.

The sampling interval is set as dt = 1 second. Furthermore, the maximum speed of each robot is 200 m/s. The rendezvous algorithm is finished when every robot is within $\delta = 3$ (km) from each other. In the case where the rendezvous is not achieved, the simulation continues for up to T = 300 seconds.

Initially, robots are randomly deployed in the workspace with size $20 \times 20 \times 20$ in km. Since $S_i(0) = 5$ km for all $i \in \{1, 2, ..., N\}$, the initial connectivity assumption may not hold depending on the initial position of a robot.

Original circumcenter algorithms in [13–15] require the initial connectivity assumption for convergence. However, the proposed rendezvous algorithm with an adaptive sensing range does not require this assumption.

Considering the case where N = 10, Figure 2 plots the movement of each robot under our distributed rendezvous algorithm. Each robot's position at every *dt* seconds is depicted with a red circle. It takes 95 seconds for all robots to rendezvous at a point.



Figure 2. Considering the case where N = 10, a blue asterisk plots the final position of every robot using the proposed rendezvous control. Each robot's position at every *dt* seconds is depicted with a red circle.

Considering the case where N = 10, Figure 3 plots the final position of every robot using the proposed rendezvous control. See that all robots are positioned close to each other since multi-robot rendezvous is achieved.



Figure 3. Considering the case where N = 10, a blue asterisk plots the final position of every robot using the proposed rendezvous control. See that all robots are positioned close to each other since multi-robot rendezvous is achieved.

Considering the case where N = 10, Figure 4 plots S_i ($i \in \{1, 2, ..., N\}$), the sensing range of each robot, when the rendezvous is done. See that the sensing range of a robot can be distinct from that of another robot.



Figure 4. Considering the case where N = 10, a blue circle plots S_i ($i \in \{1, 2, ..., N\}$), the sensing range of each robot, when the rendezvous is done. The sensing range of a robot can be distinct from that of another robot.

5.1. Monte-Carlo Simulations

Table 1 shows the comparison results. In this table, [P] presents the proposed rendezvous algorithm. [R] presents the original circumcenter rendezvous algorithms in [13–15]. In MATLAB simulations, the radar sensing noise (2) uses the noise parameter as $D_R = 0.01$ km. The motion model (10) uses the process noise parameter as D = 0.05 km.

At each MC simulation, we check whether the rendezvous succeeds or not. Considering the *j*-th MC simulation ($j \in \{1, 2, ..., J\}$), let S(j) denote whether the rendezvous succeeds or not. S(j) = 1 when the rendezvous succeeds at the *j*-th MC simulation. In addition, S(j) = 0 when the rendezvous does not succeed at the *j*-th MC simulation. Let $m(S) = 1/J * \sum_{j=1}^{J} S(j)$.

At each MC simulation, we check how much time it takes to let all of the robots rendezvous. Considering the *j*-th MC simulation ($j \in \{1, 2, ..., J\}$), let $T_r(j)$ denote the time it takes to let all robots rendezvous. In the case where the rendezvous is not achieved in the *j*-th MC simulation, $T_r(j)$ is set as *T*. Let $m(T_r) = 1/J * \sum_{j=1}^{J} T_r(j)$.

Furthermore, we check how much time (in seconds) it takes to run each MC simulation under MATLAB. Considering the *j*-th MC simulation ($j \in \{1, 2, ..., J\}$), let $T_c(j)$ denote the computational time of the *j*-th MC simulation. Let $m(T_c) = 1/J * \sum_{i=1}^{J} T_c(j)$.

Table 1 shows that increasing N (the number of robots) increases both the rendezvous time $m(T_r)$ and the computational load $m(T_c)$. Table 1 shows that only the proposed rendezvous algorithms can succeed in the rendezvous process. Under the proposed control [P], m(S) = 1 regardless of N. This is due to the fact that $S_i = 5$ (km) is rather short for network connectivity, considering random deployment in the workspace. Original circumcenter rendezvous algorithms [R] cannot handle the case where the initial connectivity assumption does not hold.

Alg.	Ν	m(S)	$m(T_r)$	$m(T_c)$
[P]	5	1	70	0.1
[P]	10	1	85	1
[P]	15	1	91	3
[P]	20	1	94	3
[R]	5	0	300	0.1
[R]	10	0	300	1
[R]	15	0	300	1
[R]	20	0	300	1

5.2. The Effect of Changing the Radar Sensing Measurement Noise D_R

Table 2 shows the effect of changing the radar sensing measurement noise D_R in (2). We use $N^{min} = 5$ in this table. Table 2 shows that increasing D_R up to 1 km does not degrade the rendezvous performance significantly. This shows that the proposed rendezvous approach is robust to sensing measurement noise.

Table 2. The effect of changing the sensing measurement noise D_R .

D_R (km)	Ν	m(S)	$m(T_r)$	$m(T_c)$
0.01	10	1	85	1
0.1	10	1	88	1
1	10	0.9	113	1

6. Conclusions

This article introduced a robust distributed rendezvous algorithm in 3D environments. In our paper, we control the sensing range of each robot adaptively, in order to achieve distributed rendezvous even in the case where the entire network is split initially. To the best of our knowledge, this paper is novel in applying radars with a variable sensing range in order to make all robots rendezvous in 3D environments. Computer simulations are utilized to demonstrate the superiority of the proposed robust rendezvous approach. In the future, we will verify the effectiveness of the proposed distributed rendezvous algorithm using real robots with variable range radars.

The proposed approach can be applied to underwater robots as well as aerial robots. Aerial robots can use radar sensors to achieve the rendezvous proposed in our paper. On the other hand, underwater robots can use active sonar sensors to achieve the rendezvous addressed in this paper.

In this paper, we handle the case where a robot can measure the relative position (elevation, azimuth, and range) of its neighbor robot using its radar. In practice, a robot may have range-only sensors for measuring the relative distance to its neighbor robot [34]. The authors of [34] addressed multi-robot rendezvous with range-only measurements. In

the future, we will apply range-only radars with variable sensing ranges in order to make all robots rendezvous at a point.

Author Contributions: This paper was written by J.K., C.C. reviewed the paper and provided ideas for solving the problems in this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Lin, J.; Yang, X.; Zheng, P.; Cheng, H. End-to-end Decentralized Multi-robot Navigation in Unknown Complex Environments via Deep Reinforcement Learning. In Proceedings of the 2019 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, 4–7 August 2019; pp. 2493–2500.
- Hu, J.; Sun, J.; Zou, Z.; Ji, D.; Xiong, Z. Distributed multi-robot formation control under dynamic obstacle interference. In Proceedings of the 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Boston, MA, USA, 6–10 July 2020; pp. 1435–1440.
- Alonso-Mora, J.; Baker, S.; Rus, D. Multi-robot formation control and object transport in dynamic environments via constrained optimization. *Int. J. Robot. Res.* 2017, 36, 1000–1021. [CrossRef]
- 4. Kim, J. Workspace exploration and protection with multiple robots assisted by sensor networks. *Int. J. Adv. Robot. Syst.* 2018, 15, 1729881418792170. [CrossRef]
- 5. Culler, D.; Estrin, D.; Srivastava, M. Overview of Sensor Networks. IEEE Comput. Mag. 2004, 37, 41–49. [CrossRef]
- Souryal, M.R.; Geissbuehler, J.; Miller, L.E.; Moayeri, N. Real-time deployment of multihop relays for range extension. In Proceedings of the 5th International Conference on Mobile Systems, Applications and Services (MobiSys 07), San Juan, Puerto Rico, 11–14 June 2007; ACM: New York, NY, USA, 2007; pp. 85–98.
- Liu, H.; Li, J.; Xie, Z.; Lin, S.; Whitehouse, K.; Stankovic, J.A.; Siu, D. Automatic and robust breadcrumb system deployment for indoor firefighter applications. In Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys 10), San Francisco, CA, USA, 15–18 June 2010; ACM: San Francisco, CA, USA, 2010; pp. 21–34.
- Hoog, J.D.; Cameron, S.; Visser, A. Selection of Rendezvous Points for Multi-Robot Exploration in Dynamic Environments. In Proceedings of the Workshop on Agents in Realtime and Dynamic Environments, International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Toronto, ON, Canada, 10–14 May 2010.
- 9. Park, H.; Hutchinson, S. Robust rendezvous for multi-robot system with random node failures: an optimization approach. *IEEE Trans. Cybern.* **2018**, *42*, 1807–1818. [CrossRef]
- Jaleel, H.; Rahmani, A.; Egerstedt, M. Duty cycle scheduling in dynamic sensor networks for controlling event detection probabilities. In Proceedings of the 2011 American Control Conference, San Francisco, CA, USA, 29 June–1 July 2011; pp. 3233– 3238.
- 11. Zai, A.H. Efficient Supply-Modulated Transmitters for Variable Amplitude Radar. Ph.D. Thesis, University of Colorado, Denver, CO, USA, 2015.
- 12. Li, Y.C.; Choi, B.; Chong, J.W.; Oh, D. 3D Target Localization of Modified 3D MUSIC for a Triple-Channel K-Band Radar. *Sensors* 2018, 18, 1634. [CrossRef] [PubMed]
- 13. Ando, H.; Oasa, Y.; Suzuki, I.; Yamashita, M. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Trans. Robot. Autom.* **1999**, *15*, 818–828. [CrossRef]
- 14. Cortés, J.; Martínez, S.; Bullo, F. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Trans. Autom. Control* 2006, *51*, 1289–1298. [CrossRef]
- Martinez, S.; Bullo, F.; Cortes, J.; Frazzoli, E. On synchronous robotic networks Part II: Time complexity of rendezvous and deployment algorithms. In Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, 12–15 December 2005; pp. 8313–8318.
- Ajorlou, A.; Aghdam, A.G. Connectivity Preservation in Nonholonomic Multi-Agent Systems: A Bounded Distributed Control Strategy. *IEEE Trans. Autom. Control* 2013, 58, 2366–2371. [CrossRef]
- Dimarogonas, D.V.; Johansson, K.H. Decentralized connectivity maintenance in mobile networks with bounded inputs. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 1507–1512.
- Flocchini, P.; Prencipe, G.; Santoro, N.; Widmayer, P. Gathering of asynchronous robots with limited visibility. *Theor. Comput. Sci.* 2005, 337, 147–168. [CrossRef]

- Park, H.; Hutchinson, S. A distributed optimal strategy for rendezvous of multi-robots with random node failures. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 1155 – 1160.
- Lin, J.; Morse, A.S.; Anderson, B.D.O. The multi-agent rendezvous problem. In Proceedings of the IEEE International Conference on Decision and Control, Maui, HI, USA, 9–12 December 2003; pp. 1508–1513.
- He, L.; Zhang, J.; Hou, Y.; Liang, X.; Bai, P. Time-Varying Formation Control for Second-Order Discrete-Time Multi-Agent Systems with Directed Topology and Communication Delay. *IEEE Access* 2019, 7, 33517–33527. [CrossRef]
- Parasuraman, R.; Kim, J.; Luo, S.; Min, B.C. Multipoint Rendezvous in Multirobot Systems. *IEEE Trans. Cybern.* 2018, 50, 310–323. [CrossRef] [PubMed]
- Kim, J. Three dimensional distributed rendezvous in spherical underwater robots considering power consumption. *Ocean Eng.* 2020, 199, 107050. [CrossRef]
- 24. Park, H.; Hutchinson, S.A. Fault-Tolerant Rendezvous of Multirobot Systems. IEEE Trans. Robot. 2017, 33, 565–582. [CrossRef]
- Park, H.; Hutchinson, S. An efficient algorithm for fault-tolerant rendezvous of multi-robot systems with controllable sensing range. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 358–365.
- 26. Douglas, B.W. Introduction to Graph Theory, 2nd ed.; Prentice Hall: Hoboken, IL, USA, 2001.
- Garcia de Marina, H.; Cao, M.; Jayawardhana, B. Controlling Rigid Formations of Mobile Agents under Inconsistent Measurements. *IEEE Trans. Robot.* 2015, 31, 31–39. [CrossRef]
- Krick, L.; Broucke, M.E.; Francis, B.A. Stabilization of infinitesimally rigid formations of multi-robot networks. In Proceedings of the 2008 47th IEEE Conference on Decision and Control, Cancun, Mexico, 9–11 December 2008; pp. 477–482.
- 29. Paley, D.A.; Zhang, F.; Leonard, N.E. Cooperative Control for Ocean Sampling: The Glider Coordinated Control System. *IEEE Trans. Control Syst. Technol.* **2008**, *16*, 735–744. [CrossRef]
- Ji, M.; Egerstedt, M. Distributed Coordination Control of Multiagent Systems While Preserving Connectedness. *IEEE Trans. Robot.* 2007, 23, 693–703. [CrossRef]
- Kim, J. Constructing 3D Underwater Sensor Networks without Sensing Holes Utilizing Heterogeneous Underwater Robots. *Appl. Sci.* 2021, 11, 4293. [CrossRef]
- Kim, J.; Kim, S. Motion control of multiple autonomous ships to approach a target without being detected. *Int. J. Adv. Robot. Syst.* 2018, 15, 1729881418763184. [CrossRef]
- Luo, S.; Kim, J.; Parasuraman, R.; Bae, J.H.; Matson, E.T.; Min, B.C. Multi-robot rendezvous based on bearing-aided hierarchical tracking of network topology. *Ad Hoc Netw.* 2019, *86*, 131–143. [CrossRef]
- 34. Zheng, R.; Sun, D. Multirobot rendezvous with bearing-only or range-only measurements. *Robot. Biomimetics* 2014, 1, 4. [CrossRef]