

# Improving Collision Avoidance Behavior of a Target-Searching Algorithm for Kilobots

Alexander Horst<sup>1</sup>, Egemen Kaba<sup>2</sup>, Rolf Dornberger<sup>3</sup>, Thomas Hanne<sup>4\*</sup>

<sup>1, 2, 3, 4</sup> School of Business, Institute for Information Systems

University of Applied Sciences and Arts Northwestern Switzerland  
Olten/Basel, Switzerland

Email: <sup>1</sup> alexander.horst@students.fhnw.ch, <sup>2</sup> egemen.kaba@students.fhnw.ch, <sup>3</sup> rolf.dornberger@fhnw.ch,  
<sup>4</sup> thomas.hanne@fhnw.ch

\*Corresponding Author

**Abstract**— Collision avoidance in the area of swarm robotics is very important. The lacking ability of such collision avoidance is mentioned as one important reason for the sparse distribution of the small test robots named Kilobots. In this research paper, two new algorithms providing a collision avoidance strategy are presented and compared with previous research results. The first algorithm uses randomness to decide which one of several approaching Kilobots are stopped for a defined time before starting to move again. The second algorithm tries to determine the assumed position of approaching Kilobots based on its radio signal strength and then to move away in the opposite direction by rotation. The results, especially of the second algorithm, are promising as the number of collisions can be significantly reduced.

**Keywords**— collision avoidance, Kilobots, swarm robotics, optimization, V-REP, target-surrounding

## I. INTRODUCTION

Former research by Zhong et al. [1] demonstrates a search and target surrounding swarm robotic algorithm using ten Kilobots. While one Kilobot acted as a target, the nine others acted as searchers, trying to approach the target while avoiding a collision among them. The results of using real swarm robots were then compared with the results of a computer simulation. The aim of their project was twofold. First, they aimed to develop an algorithm, that incorporates collision avoidance, because this was not considered explicitly, but proposed by some earlier research. Second, it is not easily feasible to accurately imitate a robot's movements, its sensations, and communication within a computer-based simulation. Therefore, the researchers implemented the algorithms (search and target surrounding as well as collision avoidance) not only within a simulation environment, but also with genuine Kilobots to compare both.

Collision avoidance of swarm robots is important, but still not sufficiently researched [2]. Therefore, this paper proposes and introduces two novel approaches based on radio signal strength (RSS) through an infrared channel, randomness and controlling the movements of Kilobots. The paper is structured as follows: In Section II the research methodology is presented. Related work is discussed in Section III. The considered collision avoidance approach and its improvement are outlined in Section IV. Section V presents experimental setup whereas results of the

experiments are shown in Section VI. Sections VII and VIII present the conclusions and suggestions for future research.

## II. RESEARCH QUESTION AND RESEARCH METHODOLOGY

In the area of the Kilobots, the lack of collision avoidance is an annoying side effect, which, even in the worst case, has severe effects such as overturning of individual bots and thus losing those [2]. This practical problem has already been mentioned in other papers as the reason, why Kilobots are currently not more common [2]. In addition, this effect drastically prevents the scalability of Kilobots. If, for instance, 10000 Kilobots are being used, space becomes a critical resource and single bots can topple. This may trigger a domino effect that will disable further bots. Thus, appropriate collision avoidance becomes crucial to avoid defect robots and the following research question can be formulated: How can a collision avoidance procedure be implemented for sensor-poor swarm intelligence using the example of Kilobots?

In order to answer the question, the target searching algorithm proposed by Zhong et al. [1] has been analyzed. Based on this, we have considered improvements regarding more effective collision avoidance capabilities and have implemented in two different algorithms. The algorithm variants are run on the V-REP simulation environment and metrics are gathered for each run. In order to make the results comparable, the same number of test runs are performed with the algorithm introduced in [1] and with our introduced two algorithms.

The direct comparison is then used to conclude whether the new algorithms reduce the number of collisions or not. Results, positive or negative, will, on the one hand, raise attention towards the importance of collision avoidance and on the other hand show future research potential.

## III. COLLISION AVOIDANCE IN SWARM ROBOTICS

Collisions and collision avoidance are important topics in robot research as collisions are significant risks to the good shape and condition of the robot and its reliable operations. In addition, collisions provide risks to the environment of the robot, especially when it is operating in areas together with humans or, in general, in unknown or dynamically changing environments. The general problem has been investigated in many publications. For instance,



surveys on approaches to collision-free navigation are given in [3] – [12]. Additional problems regarding collisions usually arise when several robots operate in a shared environment. In particular, swarm robot scenarios are prone to mutual collisions due to the larger number of robots, the frequently considered operations in unknown or changing environments, or because of the often limited sensorial information due to simple and inexpensive robot concepts [13] – [18]. On the other hand, communication between robots allow for better self-organized coordination and decision making which may help avoiding collisions [19] – [23].

For our considered Kilobot approach in swarm robotics, the general reasons for poor collision avoidance can be distinguished into two areas: First, the effects of noise in the communication ability of Kilobots shall be described and how this can cause collisions. Secondly, the area of hardware limitations of Kilobots is addressed.

#### A. Noise and its Effect on Collision Avoidance

The paper by Stender et al. [24] simulated the collaboration of a swarm of micro-robots to determine specific points in 2D space for control purposes. The researchers laid emphasis on a broad array of practical problems that were also encountered in [1]. The first problem is noise, which arises from the location measurement of the robots, which in turn yields noisy fitness function evaluations. This noise may lead to severe problems for each Kilobot to communicate within its surroundings leading to collisions.

The limitation of using a single infrared frequency in also one of the origins of noise: Since all Kilobots communicate on the same frequency, it can happen that two signals overlap [25]. Even with the supported collision avoidance (Carrier Sense Multiple Access/Collision Avoidance, CSMA/CA) methods, the bandwidth of the robots is greatly reduced and can, therefore, lead to collisions [25]. CSMA/CA describes a principle for collision avoidance, when several entities access the same transmission channel [28], i.e. the same infrared frequency is used simultaneously by various Kilobots.

#### B. Communication and Physical Limitations

The problem of limited communication ranges of micro-robots is mentioned by Stender et al. [24], leading to an incomplete information sharing as well as misinformed Kilobots. Furthermore, limited movement capabilities of the physical robots are discussed. This is also mentioned by Rubenstein et al.: the low-cost locomotion with its limited odometry (use of motion sensor data to estimate changes in position over time) makes moving precisely over a long distance difficult [25]. A target-oriented control of the Kilobots is therefore almost impossible. As mentioned by Zhong et al. [1], these limitations should be taken into account at an early stage, before the real-life testing of a novel approach is already implemented [1].

Dimidov et al. [26] describe the lack of local sensing abilities and low processing powers as major physical downsides of a Kilobot. This lack of sensing abilities is also mentioned by Valentini et al. [27] as one major downside,

why Kilobots are not yet more widely used. These physical limitations of a single Kilobot are usually disturbing an entire swarm of Kilobots. The researchers point out that, due to these and further limitations, complex algorithms cannot be implemented in Kilobots. E.g. neither the memory and the computing power, nor the battery capacity allow implementing complex moving and sensing strategies directly onto the Kilobots [26].

An additional type of physical limitations is discussed by Valentini et al. [27]. They identified as a cause for Kilobots not yet being suitable for large-scale experiments: Simple procedures such as turning of the robots, erecting them after a turnover, or recharging their batteries may become hardly feasible, particularly when the size of the swarm increases and each robot is surrounded by many others moving around. Having an appropriate collision avoidance in place does not necessarily tackle this kind of problems, but can eventually lead to an easier handling of the Kilobots, because a reduced number of fallen robots potentially reduces the set-up time of an experiment and the number of necessary replacements due to hardware damage.

### IV. OPTIMIZING CURRENT COLLISION AVOIDANCE APPROACHES

First, an overview of the target-searching algorithm used in Zhong et al. [1] is given. Subsequently, two modifications (phase 1 and phase 2) are proposed to improve the algorithm. These algorithms are primarily intended to be run on the simulation software V-REP [31].

#### A. The Current Algorithm

The target-searching algorithm introduced by Zhong et al. [1] is described in the listing in Fig. 1 and is representing a simple algorithm for collision avoidance.

```

if previous_direction was right then
    move left
else
    if previous_direction was left
    then
        move right
    else
        randomly decide which direction to take
end

```

Fig. 1. Behavior of original collision avoidance algorithm in pseudo code

This algorithm is executed, when the Kilobot is still roaming around and the estimated distance of the last received message is falling under a certain threshold of a predefined sensed collision distance.

#### B. Applied Optimization Techniques

The inspiration for modifying this algorithm lies in the work of Khorbotly et al. [29] and Oh et al. [30]. They modified the behavior of the Kilobots using RSS to estimate distances and heuristics to avoid the collisions.

The aim is to increase the capability of avoiding a collision, thus reducing the risk of a Kilobot's turnover and increasing its efficiency of fulfilling the task. In order to be able to examine the feasibility of our proposed extensions of the collision avoidance algorithm in a differentiated way, we

have proceeded in two phases: In the first phase, randomness is used to decide, which one of several approaching Kilobots has to be stopped for a defined time before starting to move again. In the second phase, a randomly chosen Kilobot additionally tries to determine the assumed position of another Kilobot based on RSS in order to move away in the opposite direction by rotation. The different behaviors are illustrated in Figure 2.

For all simulations done in V-REP, we define a Kilobot as collided, if the estimated distance between two Kilobots is equal or less than a diameter of such a robot, because the sensors are mounted in the middle of each device.

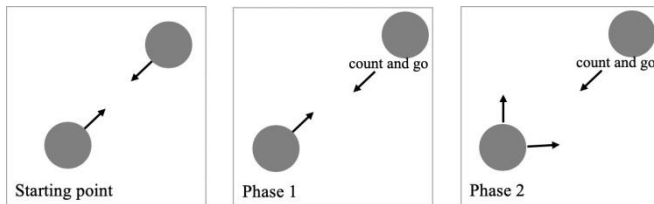


Fig. 2. Different Kilobot behaviors on the approximation of two Kilobots.

These two phases result in two algorithms, which are described in more detail in the following sections.

1) Phase one: If two roaming Kilobots approach each other, the Kilobots identify each other and decide randomly, which one of them will enter the waiting behavior or will continue its path. If a Kilobot has decided to wait, it sets a counter, which is decreased for every iteration of the main program loop. If this counter is depleted, the Kilobot resets some control variables and resumes its path following its default behavior. In our experiments, it has turned out that, in the ideal case, the Kilobot waits for a fixed pre-defined number of iterations, currently 50. The relevant part of the algorithm is shown in Figure 3.

```

Initialize a counter which decreases e
very iteration of the main execution loop

Set motion to stop in every iterat
ion

If counter = 0
reset behaviour

```

Fig. 3. Behavior of original collision avoidance algorithm in pseudo code.

2) Phase two: In the second phase, in addition to the waiting behavior described in phase one for Kilobot A, the movements of Kilobot B are also influenced.

Earlier, the behavior that Kilobots communicate only on a shared infrared frequency was mentioned as a disadvantage. As an overcome, blocking the frequency by sending a state flag in the message was not used as it proved to be unpractical. Instead, following the principle that Kilobots should remain as autonomous as possible, both Kilobots decide randomly, which behavior they want to engage, when there is a danger of colliding with each other.

For example, Kilobot A can randomly decide to move away from the stagnant Kilobot B. To do that, one of the

two motors is switched off, initiating a rotational movement. After a certain number of main iterations, Kilobot A checks, whether or not the distance to Kilobot B has been increased. If this is the case, it continues. Otherwise, it chooses the other direction for the remaining of the maneuver. Again, after a certain number of iterations, Kilobot A walks forward in an attempt to increase the distance to Kilobot B even further.

The whole maneuver lasts for a defined number of iterations. Currently 50 iterations proved in our tests to be a feasible setting: The direction is checked again after 10 iterations. Ideally, the Kilobots walk then straight for 40 iterations. After Kilobot B's counter has expired, it resumes its original path. The relevant part of the algorithm is shown in the following pseudo-code in Figure 4.

```

Initialize a counter which decreases e
very iteration of the main execution loop

Initialize the next motion direction to l
eft

Save the distance measurement of
the last message

If Counter hit threshold to check the direc
tion then
    If the distance measurement of the
    last message is smaller than
    the previously saved distance
    measurement
        then change the motion
        to the opposite direc
        tion otherwise leave it as is

If Counter hit thresholds to change behav
iour then set motion to go straight

If the counter reached 0, reset behaviour

```

Fig. 4. Behavior of original collision avoidance algorithm in pseudo code

This algorithm implements a behavior, in which the Kilobots must wait for each other to pass, attempting to create a safer environment for every robot. The main drawback is clearly a decreased search performance of the swarm, because certain swarm agents possibly have to wait for a few additional iterations. A Kilobot can determine that another robot is within its communication radius and roughly predict the direction via RSS. However, it is not possible to determine the exact direction of the other robot. The rotation away from the direction, where the signal originates, presented in phase two, can, therefore, be misinterpreted. This misinterpretation may lead to the result that the Kilobot rotates directly in the direction another Kilobot is approaching from. The proposed modification is heavily limited by the capabilities of the Kilobot. As they do not have any directional proximity sensors and no ability to determine its position on a grid, the algorithm must rely purely on RSS and the implemented communication protocol based on infrared. Additionally, the Kilobots are fragile to a certain degree, in which inappropriate handling could cause a turnover or even worse a hardware damage.

## V. EXPERIMENT SETUP

The researchers in [1] have used V-REP [31] and arranged the Kilobots according to the setup shown in Figure 5 on the left side. After one simulation test run, the Kilobots approached the target in the middle without paying attention to collisions or providing information about them as shown in Figure 5 on the right side. As mentioned, all algorithms were extended by a logging function, which adds a message to the status bar of the simulation software indicating, which Kilobots collided. After each simulation run, the whole status bar was copied, stored in a separate file and searched for the string indicators. The results were gathered in a table for further analysis. Since the number of collisions can vary greatly depending on the simulation run, ten simulations have been performed for every setup (see Tables 1 and 2) in order to average the results.



Fig. 5. Experiment setup (left) and the result of one simulation (right) according to [1].

As described in the previous section, the algorithm was modified with the two phases. The first phase with integrated counter and the second phase with integrated counter and motion control of a Kilobot. For these two phases individually, too, ten simulation runs have been performed (see Tables 3 to 6).

## VI. RESULTS AND COMPARISON

In the first part, the data of the simulation runs are presented in tabular form without being evaluated. In the second part, the results are critically evaluated and compared with each other.

### A. Results

Table I shows the results of the baseline that can be achieved with the setup of [1], where an average of 4.4 Kilobots collided (at least once) over 10 runs.

During the simulation of different algorithms, a correlation between collision avoidance and search success was determined. Adjustments that lead to enhanced collision avoidance had a potentially negative effect on the quality of the search results. Therefore, in addition to the effects on the number of collisions, the percentage of Kilobots that found the target is also taken into account.

Here, Table II shows the number of Kilobots per simulation run that were able to find the target Kilobot resulting in an average of 3.7 successful searchers with the setup of [1]. Table III shows the results of the implemented phase 1 algorithm. With this setup in place, an average of 4.3 Kilobots collided (at least once) over 10 runs.

Table IV shows per simulation run the number of Kilobots that were able to find the target Kilobot resulting in

an average of 2.7 successful searchers with the setup of phase 1. Table V shows the results of the implemented phase 2 algorithm. With this setup in place, an average of 3.7 Kilobots collided over 10 runs. Table VI shows per simulation run the number of Kilobots that were able to find the target Kilobot resulting in an average of 2.4 successful searchers with the setup of phase 2.

TABLE I. OVERVIEW OF KILOBOTS COLLIDED WITH THE SETUP OF [1]

	run 1	run 2	run 3	run 4	run 5	run 6	run 7	run 8	run 9	run 10
Kilobot#0 collided	X	X	X	X	X	X	X	X		X
Kilobot#1 collided	X	X	X		X	X		X	X	X
Kilobot#2 collided										X
Kilobot#3 collided			X					X		
Kilobot#4 collided					X	X				
Kilobot#5 collided	X									X
Kilobot#6 collided			X		X	X		X	X	
Kilobot#7 collided	X	X			X	X	X			X
Kilobot#8 collided		X	X	X	X	X	X	X	X	X
Kilobot#9 collided										
<b>Total</b>	<b>4</b>	<b>4</b>	<b>5</b>	<b>2</b>	<b>6</b>	<b>6</b>	<b>3</b>	<b>5</b>	<b>3</b>	<b>6</b>

TABLE II. OVERVIEW OF KILOBOTS THAT FOUND THE TARGET WITH THE SETUP OF [1]

Bots found target	run 1	run 2	run 3	run 4	run 5	run 6	run 7	run 8	run 9	run 10
<b>Total</b>	<b>5</b>	<b>5</b>	<b>2</b>	<b>3</b>	<b>2</b>	<b>5</b>	<b>3</b>	<b>3</b>	<b>5</b>	<b>4</b>

TABLE III. OVERVIEW OF KILOBOTS COLLIDED WITH THE SETUP OF PHASE 1

	run 1	run 2	run 3	run 4	run 5	run 6	run 7	run 8	run 9	run 10
Kilobot#0 collided	X	X		X	X		X	X	X	X
Kilobot#1 collided	X	X					X	X	X	X
Kilobot#2 collided		X	X					X	X	
Kilobot#3 collided						X				
Kilobot#4 collided		X	X							
Kilobot#5 collided									X	
Kilobot#6 collided	X	X	X	X						
Kilobot#7 collided	X	X	X		X	X	X	X	X	X
Kilobot#8 collided	X		X	X	X	X		X	X	X
Kilobot#9 collided										
<b>Total</b>	<b>5</b>	<b>6</b>	<b>5</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>5</b>	<b>6</b>	<b>4</b>

TABLE IV. OVERVIEW OF KILOBOTS THAT FOUND THE TARGET WITH THE SETUP OF PHASE 1

Bots found target	run 1	run 2	run 3	run 4	run 5	run 6	run 7	run 8	run 9	run 10
<b>Total</b>	<b>5</b>	<b>3</b>	<b>4</b>	<b>4</b>	<b>2</b>	<b>1</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>

TABLE V. OVERVIEW OF KILOBOTS COLLIDED WITH THE SETUP OF PHASE 2

	run 1	run 2	run 3	run 4	run 5	run 6	run 7	run 8	run 9	run 10
Kilobot#0 collided	X	X		X	X	X	X	X		X
Kilobot#1 collided	X	X	X		X	X	X	X	X	X
Kilobot#2 collided			X					X		
Kilobot#3 collided			X							
Kilobot#4 collided			X	X				X		
Kilobot#5 collided	X						X		X	X
Kilobot#6 collided					X		X	X		X
Kilobot#7 collided								X	X	
Kilobot#8 collided			X	X				X	X	
Kilobot#9 collided										
<b>Total</b>	<b>3</b>	<b>2</b>	<b>5</b>	<b>2</b>	<b>4</b>	<b>2</b>	<b>4</b>	<b>7</b>	<b>4</b>	<b>4</b>

TABLE VI. OVERVIEW OF KILOBOTS THAT FOUND THE TARGET WITH THE SETUP OF PHASE 2

Bots found target	run 1	run 2	run 3	run 4	run 5	run 6	run 7	run 8	run 9	run 10
<b>Total</b>	<b>3</b>	<b>4</b>	<b>3</b>	<b>1</b>	<b>2</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>1</b>

### B. Discussion and Comparison of Results

As already explained, results on collision avoidance must be considered separately from the quality of the target

search. The initial setup of [1] has a relatively high number of collisions with an average of 4.4 collisions on 10 Kilobots. At the same time, 3.7 Kilobots are able to locate the target. As Zhong et al. [1] did not focus primarily on collision avoidance; this relatively high number of collisions was acceptable.

Applying phase 1, the number of collisions could slightly be reduced to an average of 4.3 collisions (decrease of 2.3 %). At the same time, the search success rate decreased strongly to 2.7 Kilobots successfully fulfilling the task (decrease of 27%). The nearly missing improvement in collision avoidance is now accompanied by a significant deterioration in search capabilities. Therefore, phase 1 does not seem to be a valid option.

In phase 2, the number of collisions could be reduced to an average of 3.7 collisions (decrease of almost 16%). At the same time, the search success rate decreased to 2.4 Kilobots successfully fulfilling the task (decrease of 35%). Compared to the baseline from [1], collision avoidance has decreased significantly. However, this also significantly applies to the search capabilities of Kilobots. If the searchability of the Kilobots is counted as less important, phase 2 would be a promising option. If fulfilling the task of finding the target robot has high priority, neither phase 1 nor phase 2 for collision avoidance are valid extensions to the standard approach of [1].

## VII. CONCLUSIONS

Collision avoidance is discussed as one of the main reasons for the still sparse distribution of swarm robots, particularly Kilobots, in the real world [27]. Therefore, this paper examines novel approaches to improve collision avoidance using Kilobots based on the target-searching algorithm introduced by Zhong et al [1]. Two different extensions of their algorithm (called phase 1 and phase 2) have been implemented and the results are compared to the simulation data of Zhong et al. [1].

In phase 1, one of two potentially colliding Kilobots is brought to a temporary standstill. In phase 2, one of the Kilobots is brought to a standstill and the other Kilobot rotates around its own axis in order to move away from the opponent.

Both phases achieved an improvement in collision avoidance by decreasing collisions by 2.4% respectively 35%. However, the search capability of the Kilobots was severely impaired. Since a reduced number of collisions should not occur at the expense of the actual goal, the finding of a target, both algorithms are not ideal and cannot be used in their present form to sufficiently improve collision avoidance without reducing the ability to find the target. Future research must now address if and how the successes of these two phases can be achieved without compromising the search ability.

## VIII. FURTHER RESEARCH

On the basis of the present paper, three concrete next research topics have been identified: Firstly, as already stated in the conclusion, the two algorithms (phase 1 and 2) presented are capable of reducing collisions, but only at the

expense of the actual goal, namely finding a target. Future research could take over the second phase algorithm presented here and investigate how it could be optimized so that the search function is not so severely restricted. Maybe, the value of 50 waiting iterations is better for collision avoidance, but worse for target finding.

Secondly, if the second phase algorithm is optimized to such an extent that the search quality no longer deviates too much from the baseline in [1], the question of generalizability also arises. If the algorithm can also be used in larger Kilobots swarms or even beyond Kilobot applications with swarm robots, e.g. swarms of drones.

Lastly, this paper has not addressed the hardware limitations of the Kilobots. Further research could, therefore, be carried out in the field of whether and how the Kilobot hardware could be optimized without breaking the basic principles of Kilobot swarm robots, cheap manufacturing cost and low calculation power.

## REFERENCES

- [1] V. J. Zhong, R. R. Umamaheshwarappa, R. Dornberger, and T. Hanne, "Comparison of a real Kilobot robot implementation with its computer simulation focussing on target-searching algorithms," In 2018 International Conference on Intelligent Autonomous Systems (ICoIAS) (pp. 160-164). IEEE, March 2018
- [2] G. Valentini, A. Antoun, M. Trabattoni, B. Wiandt, Y. Tamura, Y., E. Hocquard, and M. Dorigo. "Kilogrid: a novel experimental environment for the Kilobot robot", *Swarm Intelligence*, 12(3), 245-266, 2018.
- [3] C. H. Everett, Survey of collision avoidance and ranging sensors for mobile robots. *Robotics and Autonomous Systems*, 5(1), 5-67, 1989.
- [4] J. P. Laumond (ed.), *Robot motion planning and control* (Vol. 229). New York: Springer, 1998.
- [5] B. M. Albaker, N. A. Rahim, "A survey of collision avoidance approaches for unmanned aerial vehicles", In 2009 international conference for technical postgraduates (TECHPOS) (pp. 1-7). IEEE, December 2009.
- [6] B. M. Albaker, N. A. Rahim, "Unmanned aircraft collision detection and resolution: Concept and survey", In 2010 5th IEEE Conference on Industrial Electronics and Applications (pp. 248-253). IEEE, June 2010.
- [7] T. Kruse, A. K. Pandey, R. Alami, A. Kirsch, Human-aware robot navigation: A survey. *Robotics and Autonomous Systems*, 61(12), 1726-1743, 2013.
- [8] M. Hoy, A. S. Matveev, A. V. Savkin, Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey. *Robotica*, 33(3), 463-497, 2015.
- [9] H. Pham, S. A. Smolka, S. D. Stoller, D. Phan, J. Yang, A survey on unmanned aerial vehicle collision avoidance systems. *arXiv preprint arXiv:1508.07723*, 2015.
- [10] P. A. Lasota, T. Fong, J. A. Shah, A survey of methods for safe human-robot interaction. Now Publishers, 2017.
- [11] S. Haddadin, A. De Luca, A. Albu-Schäffer, Robot collisions: A survey on detection, isolation, and identification. *IEEE Transactions on Robotics*, 33(6), 1292-1312, 2017.
- [12] M. G. Mohanan, A. Salgoankar, A survey of robotic motion planning in dynamic environments. *Robotics and Autonomous Systems*, 100, 171-185, 2018.
- [13] H. He, S. Kamburugamuve, G. C. Fox, W. Zhao, Cloud based real-time multi-robot collision avoidance for swarm robotics. *International Journal of Grid and Distributed Computing*, 9(6), 339-358, 2016.
- [14] E. G. Hernández-Martínez, E. Aranda-Bricaire, Convergence and collision avoidance in formation control: A survey of the artificial potential functions approach (pp. 105-126). Rijeka, Croatia: INTECH Open Access Publisher, 2011.

- [15] Y. Mulgaonkar, A. Makineni, L. Guerrero-Bonilla, V. Kumar, Robust aerial robot swarms without collision avoidance. *IEEE Robotics and Automation Letters*, 3(1), 596-603, 2017.
- [16] M. Hamer, L. Widmer, R. D'andrea, Fast generation of collision-free trajectories for robot swarms using GPU acceleration. *IEEE Access*, 7, 6679-6690, 2018.
- [17] S. H. Arul, A. J. Sathyamoorthy, S. Patel, M. Otte, H. Xu, M. C. Lin, D. Manocha, LSwarm: Efficient collision avoidance for large swarms with coverage constraints in complex urban scenes. *IEEE Robotics and Automation Letters*, 4(4), 3940-3947, 2019.
- [18] J. N. Yasin, M. H. Haghbayan, J. Heikkonen, H. Tenhunen, J. Plosila, "Formation maintenance and collision avoidance in a swarm of drones", In *Proceedings of the 2019 3rd International Symposium on Computer Science and Intelligent Control* (pp. 1-6), September 2019.
- [19] T. Schmickl, K. Crailsheim, Trophallaxis within a robotic swarm: bio-inspired communication among robots in a swarm. *Autonomous Robots*, 25(1), 171-188, 2008.
- [20] T. Schmickl, R. Thenius, C. Moeslinger, G. Radspieler, S. Kernbach, M. Szymanski, K. Crailsheim, Get in touch: cooperative decision making based on robot-to-robot collisions. *Autonomous Agents and Multi-Agent Systems*, 18(1), 133-155, 2009.
- [21] N. R. Hoff, A. Sagoff, R. J. Wood, R. Nagpal, "Two foraging algorithms for robot swarms using only local communication", In *2010 IEEE International Conference on Robotics and Biomimetics* (pp. 123-130). IEEE, December 2010.
- [22] B. R. Trilaksono, "Distributed consensus control of robot swarm with obstacle and collision avoidance", In *2015 2nd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)* (pp. 2-2). IEEE, October 2015.
- [23] A. R. Shirazi, Y. Jin, A strategy for self-organized coordinated motion of a swarm of minimalist robots. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 1(5), 326-338, 2017.
- [24] M. Stender, Y. Yan, H. B. Karayaka, P. Tay, and R. Adams, "Simulating micro-robots to find a point of interest under noise and with limited communication using particle swarm optimization", In *SoutheastCon 2017* (pp. 1-8). IEEE, March 2017.
- [25] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors", In *2012 IEEE International Conference on Robotics and Automation* (pp. 293-298). IEEE, May 2012.
- [26] C. Dimidov, G. Oriolo, G., and V. Trianni, "Random walks in swarm robotics: an experiment with Kilobots", In *International Conference on Swarm Intelligence* (pp. 185-196). Springer, Cham, September 2016.
- [27] Z. Gao, N. Zhang, and H. Han, "Analysis and Design of Collision Avoidance Optimization Algorithm Based on CSMA/CA", In *2018 International Symposium on Communication Engineering & Computer Science (CECS 2018)*. Atlantis Press, July 2018.
- [28] S. Khorbotly, G. El-Howayek, A. Morrison, and A. Roggow, An RSS based triangulation method for robot tracking in robotic swarms, in *2017 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS)*, 2017, pp. 15.
- [29] H. Oh, A. R. Shiraz, and Y. Jin, Morphogen diffusion algorithms for tracking and herding using a swarm of kilobots, *Soft Comput*, vol. 22, no. 6, pp. 1833-1844, March 2018.
- [30] Arul, S. H., Sathyamoorthy, A. J., Patel, S., Otte, M., Xu, H., Lin, M. C., & Manocha, D. (2019). LSwarm: Efficient Collision Avoidance for Large Swarms with Coverage Constraints in Complex Urban Scenes. *arXiv preprint arXiv:1902.08379*.
- [31] V-REP User Manual. [Online]. Available: <http://www.coppeliarobotics.com/helpFiles/index.html>. [Accessed: 23-May-2019]