

Swarm Robotics: Simulators, Platforms and Applications Review

Cindy Calderón-Arce ^{1,†} , Juan Carlos Brenes-Torres ^{2,†}  and Rebeca Solis-Ortega ^{1,*,†} 

¹ School of Mathematics, Costa Rica Institute of Technology, Cartago 30101, Costa Rica; ccalderon@tec.ac.cr

² Mechatronics Academic Area, Costa Rica Institute of Technology, Cartago 30101, Costa Rica; juanbrenes@tec.ac.cr

* Correspondence: rsolis@tec.ac.cr

† These authors contributed equally to this work.

Abstract: This paper presents an updated and broad review of swarm robotics research papers regarding software, hardware, simulators and applications. The evolution from its concept to its real-life implementation is presented. Swarm robotics analysis is focused on four aspects: conceptualization, simulators, real-life robotics for swarm use, and applications. For simulators and robots, a detailed comparison between existing resources is made. A summary of the most used swarm robotics applications and behaviors is included.

Keywords: swarm robotics; simulators; robots; review



Citation: Calderón-Arce, C.; Brenes-Torres, J.C.; Solis-Ortega, R. Swarm Robotics: Simulators, Platforms and Applications Review. *Computation* **2022**, *10*, 80. <https://doi.org/10.3390/computation10060080>

Academic Editor: Yudong Zhang

Received: 9 April 2022

Accepted: 12 May 2022

Published: 24 May 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/>).

1. Introduction

Swarm robotics, as a concept, appeared around three decades ago; since then, a large amount of research on it has taken place. Beni and Wang, in their 1989 paper [1], were the first to introduce the concept of swarm intelligence into the field of robotics. Although the concept of swarm robotics dates back to the early 1990s, it was not until 2000 that great interest in research in this topic began to rise. Dorigo et al. in [2] showed that trend in swarm research, by analyzing how many times the phrase “swarm robotics” appeared in papers, both SCOPUS and Google Scholar.

Swarm robotics have been researched for more than 20 years. During this time, software for simulating swarm behavior and real robotic swarms has been created, adapted and tested in different situations. For this reason, most of the work related to swarm robotics is considered to cover four main areas: (i) evolution of swarm robotics concept and characteristics; (ii) software that can be used to simulate algorithms and/or behavior applicable to swarm robotics; (iii) real-life robots that can be used or are supposed to be used as a swarm; (iv) applications where swarm robotics have been used, either simulated or real.

This paper is organized as follows: Section 2 presents a definition of swarm robotics, its main objective, and its characteristics. In Section 3, software for simulating swarm algorithms is presented. Section 4 summarizes some robot platforms that have been created or adapted to be used in real swarm applications. In Section 5, possible applications for swarm robotics are displayed. Finally, the conclusions are presented in Section 6.

2. Swarm Robotics: Definition and Characteristics

Swarm robotics is a dynamic research field that integrates two important concepts: Swarm Intelligence (SI) and Multi-Robotics System (MRS).

SI is inspired by simple behaviors and self-organization of organisms such as ants, birds, bees, fishes, bacteria, wolves, and so on [3]. It can be defined as the collective intelligence that emerges from simple interactions between a large group of autonomous agents [4]. SI systems usually consist of a simple agents group, where all individuals interact locally with each other and their environment, with the purpose of achieving a

global behavior [5]. Generally, agents do not possess global data about the state of its world, nor the task they are working on.

On the other hand, MRS consists of multiple robots sharing a workspace in order to perform assigned tasks. These tasks can be difficult or impossible to accomplish for a single robot [6]. There are several advantages of using MRS, such as an increasing ability to resolve task complexity and improving performance, reliability and simplicity in design [7].

Swarm robotics (SR) is an area of knowledge within multi-robotics systems, and has its origins in the application of Swarm Intelligence to the field of robotics [8]. Particularly, SR studies how to design a large number of relatively simple physically embodied agents. SR seeks a collective behavior based on local interactions among agents and their environment [5,9].

In order to be considered a swarm and not only a MRS, a group of robots should have most of the following characteristics:

- *Autonomy*: it is defined as the ability to perform planned tasks based on current state and sensing, without human intervention [10]. Thus, all robots of the swarm must have a real body and be able to physically interact with the world without outside intervention [5,9].
- *Homogeneity*: each robot must have the same design, functionalities and share the same control algorithm [11]. Homogeneity is usually wanted because heterogeneity lowers the degree of redundancy, which may reduce robustness if it is not considered carefully [4]. However, it has become more common to find works that employ a restrained heterogeneous swarm. In these, robots are designed as a small set of different agent types, that can be put together in order to accomplish a task collaboratively [12].
- *Large number*: the formal definition of a swarm implies having “a large number” of agents, however, a minimum group size is difficult to decide on and justify. For instance, a swarm of 10 to 20 robots may seem enough for laboratory tests, but if they are deployed in an area of several square kilometers, this number may seem insignificant [9].
- *Limited capabilities*: a key idea in SR systems is to use relatively simple robots. Thereby, robots might not be able to efficiently carry out tasks on their own, but they would be highly efficient by cooperating with others [5]. As mentioned in [9], even though robots must be simple, this does not impose any restrictions on the hardware or software complexity of the robots. The simplicity of individual robots should not be taken in absolute terms, but relative to the task.
- *Communication and control schemes*: two main approaches can be used to manage communication and control of a robotic system: centralize and decentralize. Centralized schemes have a main entity which collects and synthesizes data from all the agents and, in some occasions, tells them how they should operate on a global level [4]. They have the advantage of offering direct control over each agent and making it easy to predict the overall system behavior. On the other hand, decentralized systems use distributed communication and control mechanisms [4]. Among their advantages are the following: (i) It reduces delays and bottle necks associated with centralized processing; (ii) It reduces failures associated with agent loss; (iii) It naturally exploits parallelism.

The swarm algorithm and collective behavior should be effective enough to allow scalability, robustness and flexibility of the system.

Scalability is accomplished when the system can operate under a wide range of group sizes (from a small number to several thousand individuals or more) without decreasing performance [4,9]. In [8] it is mentioned, as a good practice, to use local sensing and communication to accomplish scalability [8].

Robustness is defined as fault tolerance and fail-safety [12]. The system should be able to continue functioning even with the loss or malfunction of some robots. Robustness can

be attributed to (i) redundancy in the system, in which any individual can be compensated by another one, (ii) decentralized coordination and (iii) simplicity of the individuals [9].

Finally, flexibility refers to the ability of adapting to new, different and/or changing requirements of the environment [8].

Even though SR use derives plenty of advantages, it is often difficult to design a robot's behavior in order to achieve the desired global performance [11]. For this reason, simulation, modeling and learning approaches are used.

Simulation, where a virtual scenario and swarm are created, is used to analyze robot capabilities (sensors and actuators) and programming algorithms. By replicating simulation experiments, the swarm performance can be improved without run-time-consuming experiments with real robots [11]. Since it is not always possible to build large groups of physical robots, simulations are a good option to test swarm algorithms on a large scale [5].

The next section presents some of the most used simulation software for swarm robotics research.

3. Simulators for Swarm Algorithms

Simulators are an intermediate step between abstraction, such as mathematical models, and real validation using physical robots. Simulations and modeling are some of the most used tools to analyze and validate swarm robotics systems [13]. In that sense, Brambilla et al. summarized in [14] more than 60 studies related to collective behaviors in swarm robotics. More than 50% of them present results obtained only by simulations or modeling.

Simulation use provides an easier, faster and even safer way to perform experiments than using real robots. However, it is not easy to select the best software to simulate or model a robotic swarm. Erez et al. in [15] established some measures of simulation performance, based on typical numerical challenges for robotics. They concluded that each platform performs its best in the type of system it was designed and optimized for. Thus, it is hard to compare different multi-robot simulators, because each one has been developed with different objectives.

3.1. Software Examples

There are many kinds of platforms, software, and algorithms to simulate robots based on different objectives and types of robot. For example, there are 2D and 3D simulators with specific algorithms to design, analyze and compare swarms. In this subsection, some of the most popular software and platforms that could be used to simulate swarm robotics will be covered.

Stage is a free C++ library that simulates multiple mobile robots with a size capacity of up to 100,000 agents. It presents advantages such as code reuse, transparency, experiment replication and modification. *Stage* is able to simulate mobile robots with several sensors and actuator models, such as infrared rangefinders, scanning laser rangefinder, color-blob tracking, fiducial tracking, bumpers, grippers and mobile robot bases with global localization (odometric) [16]. It provides good performance on tasks such as exploration and foraging, but not for modeling trail-following or cooperative behaviors.

Massink et al. [17] presented a method to analyze swarm robotics systems using *Bio-PEPA*. *Bio-PEPA* is a high-level modeling language proposed in [18] to analyze biochemical systems, and it allows to perform stochastic simulation, fluid flow analysis and model checking. Additionally, its structure also allows modeling scalable systems and space-time characteristics adaptable to swarm robotics, modeling specific behavior of individual entities, such as robots, and its interactions between the various species defining how robots interact.

TeamBots is a 2D Simulator for multi-agent mobile robotics research, which is easy, free and able to run the simulation code on real robots. *TeamBots* is still available, but its development appears to have stopped in 2000 [19]. It supports prototyping simulation with the same control systems that can be run on real mobile robots.

Swarm-bots is a robotic simulator in which collective interaction is exploited by the swarm intelligence mechanism. Its control layer can be extended to the physical level and its use has been reported in experiments with up to 40 robots [16]. It is a powerful tool, but it is not publicly available. It allows to simulate properties such as robustness, flexibility, and it has the ability to solve complex problems by exploiting parallelism and self-organization [20].

Open Dynamics Engine (ODE) is a dynamics engine capable of multi-agent simulation and which could, consequently, be used for robotics simulation. ODE simulates chains of rigid bodies connected and constrained by different types of joints. It has a built-in collision detection system and implements hard contacts using non-penetration constraint [21].

Gazebo is a 3D simulator for outdoor environments. It generates realistic sensor feedback and physically consistent interactions between objects, allowing the user to select between multiple dynamics engines, but it has the disadvantage of running slowly with large populations [22]. Gazebo has been used to compare algorithms for navigation and grasping in a controlled environment.

Webots is a development environment used to model, program, and simulate mobile robots. It simulates sensors and actuators and controller programs could be transferred to commercially available real robots [23]. Webots is able to model some of the most popular robots in swarm robotics such as EPFL Alice, K-Team Khepera, Mondada's E-Puck, but it has the disadvantage of not scaling data [16].

Microsoft robotics studio (MSRS) is based on dynamics engines, such as Gazebo and Webots, and its aim is to provide software standards for robot control. It has been used not only in industry, but also for academic objectives [24]. Robotics studio is free to use, but not to modify, nor distribute, the source code is not publicly available, it does not allow to simulate more than one robot at a time [16] and it is a discontinued Windows environment.

USARSim is a free 3D simulator similar to Gazebo, Webots and Microsoft Robotics Studio. It is open-source and can be used for research and education [25]. USARSim was originally developed and used for the RoboCup Urban Search And Rescue competition, but it can be extended to model different arbitrary application scenarios [16].

ARGoS is a modular and multi-engine simulator for heterogeneous swarm robotics, i.e., it allows to use multiple physics engines of different types and assigns them to different parts of the environment. Controllers, sensors, actuators, physics engines and visualizations can be included in the design robot [26].

CoppeliaSim (previously V-REP: Virtual Robot Experimentation Platform) is quite similar to Webots and permits fast development of algorithms, with code that can be transferred to real robotic hardware [27].

In addition to this software, there are algorithms and methods to design swarm robots and their behaviors. For instance, Francesca and Birattari worked on automatic design methods for robot swarm control [28]. They formulated an optimization problem to choose the best design option from a search space with different settings. The researchers classified the method as off-line when the design occurs and terminates before the robot swarm is deployed in its operational environment, and as on-line when the design takes place when the robot swarm has been already deployed in its operational environment. They indicated that automatic design is a viable and promising approach to programming the control of robot swarms.

Table 1 summarizes some important features of simulation software for robots, it is a useful reference to choose one for swarm robotics simulations.

Table 1. Features of most used simulation software.

Software/Feature	Last Update	Free	Open-Source	Linux	MacOS	Windows	Swarm	Source (Accessed: 11 May 2022)
Stage	2020	✓	✓	✓	✓		✓	Repository: https://github.com/rtv/Stage
Bio-PEPA	2010	✓		✓	✓	✓	✓	Website: https://homepages.inf.ed.ac.uk/jeh/Bio-PEPA/biopepa.html
TeamBots	2000	✓	✓	✓	✓	✓	✓	Website: https://www.cs.cmu.edu/~trb/TeamBots/
Swarm-bots	2014			✓	✓		✓	Website: https://www.swarm-bots.org/
ODE	2022	✓	✓	✓	✓			Website: https://www.ode.org/
Gazebo	2019	✓	✓	✓	✓		✓	Website: http://gazebo.org/
Webots	2021	✓	✓	✓	✓	✓		Website: https://cyberbotics.com/
MSRS	2015	✓				✓		Not available
USARSim	2013	✓	✓	✓	✓	✓	✓	Repository: https://sourceforge.net/projects/usarsim/
ARGoS	2022	✓	✓	✓	✓	✓	✓	Website: https://www.argos-sim.info/
CoppeliaSim	2022	✓	✓	✓	✓	✓		Website: https://www.coppeliarobotics.com/

3.2. From Simulation to Reality

Most of the time, it is not possible to simulate all aspects of reality. For instance, some simulated sensors and actuators are free of noise and interference. Moreover, simulators usually do not take into account all environment interactions such as friction, luminosity, temperature, wind or dust. Robot internal interactions are also challenging to consider on simulators. This may include battery life and energy level, component degradation and failure, sensors and actuators limitations due to static friction, backlash, hysteresis or saturation.

Simulated data sets and environments are far from interactions, details, and situations experienced in the real world. In that sense, virtual reality comes to generate realistic and plausible data for a wide variety of problems. For instance, UnrealROX is an environment built to reduce that reality gap, in which robot agents explore environments and interact with real objects in a simulated world [29].

Barca and Sekercioglu [4] analyzed how to get closer to a practical swarm robotic system for real-world applications. They presented aspects, such as robotic devices, that can be used in swarm robotic systems and challenges to take into account for real-world applications. They focused their study on ways to interact efficiently with the environment by means of selecting appropriate communication and control schemes, defining behaviors, traits, functions and supporting control, connectivity, and energy consumption. The potential advantages associated with proper control of robot swarms are some of the most relevant motivations for this research.

In that sense, the use of real robots to validate collective behavior is a fundamental tool. Real robot experiments allow the transfer of collective behavior to different robotic platforms and hardware [14]. It helps to identify which collective behaviors are realizable in practice and to distinguish differences between prototypes and real robot systems.

However, real-world experiments are time-consuming and error-prone tasks. Gupta et al. [30] mention that currently most real swarm robotics experiments face security issues. This is mainly because there have not been many studies about security protocols adapted to swarm robotics, therefore, communications can be intercepted or disturbed by an attacker.

Another issue is communication within the swarm and between the swarm and the environment. For the swarm to be fully autonomous, it should provide its own means of communication, however, the most used technology (ad hoc WLAN networks) provides communication with a limited range and it is susceptible to individual robot failures. An infrastructure-based network can provide more stable communication, but it requires the installation of hardware, such as base stations. This issue causes many industrial projects to still rely on centralized communication [31].

In addition, Dorigo et al. [2] comment that options related to materials, biohybrid solutions and new ways of storing and transmitting energy could help address some of the current issues related to the hardware of real robot swarms.

Therefore, synthetic data generation has become increasingly popular since it is faster to generate by automatic computation [29]. Consequently, real robot experiments should not be considered to validate collective behaviors in real-world applications, but it is useful to test them against realistic noise patterns in sensors and actuators [14].

On the other hand, mixed reality can be a valuable tool for robotics research and development. It establishes interactions between physical and virtual objects in physical or virtual environments. Moreover, it could reduce the gap between simulation and real implementation by an object combination, including robots, sensors and humans [32].

4. Real-Life Swarm Robotics Platforms

Research on swarm robotics uses simulations and real-life implementations to test and evaluate swarm behavior designs. Most swarm robotics use several identical mobile robots as swarm agents. A buoyant development of these robots has been noted in the last few years, all of them looking for a balance between cost and capabilities. This section covers the most prominent real-life robots for swarm use, found in the literature of the last decade, which are summarized in Tables 2 and 3.

Table 2. Main robots for use in swarm robotics.

Robot/Feature	Introduction Date	Developer	Commercially Available	Cost
E-puck	2004	École Polytechnique Fédérale de Lausanne (EPFL)	✓	USD 1000
Khepera IV	2015	K-Team (EPFL spin-off)	✓	USD 3200
Kilobot	2010	Harvard University	✓	USD 130
Swarmanoid: footbot	2011	Future and Emerging Technologies (FET-OPEN) project		-
Colias	2014	University of Lincoln		USD 32 (GBP 25)
Mona	2017	University of Manchester		USD 129 (GBP 100)
Psi Swarm	2016	University of York		-
GRITSBot	2015	Georgia Tech		USD 50 (parts)
Thymio	2011	Mobsya (EPFL spin-off)	✓	USD 173

Mobile robots such as *E-puck*, *Khepera* and *Kilobot* have been widely popular among swarm robotics researchers, appearing in publications across the globe and still present nowadays. It should be noted that these robots share the feature of being commercially available, which increases their accessibility to research groups.

E-pucks were developed by École Polytechnique Fédérale de Lausanne (EPFL) and are differential drive robots with two stepper motors. Notably, they can perform odometry by tracking step counts [33]. *E-pucks* features 8 infrared (IR) sensors, a color camera, which can be used for obstacle detection, and some other sensors such as accelerometers and microphones [34]. Inter-robot communication is achieved by using the IR sensors and Bluetooth. Additionally, they are controlled by a dsPIC30 microcontroller [35]. An *E-puck* robot is sold at a retail price of USD 1000 [36].

Khepera robots are notorious in academia, they are developed and sold by K-Team (an EPFL spin-off). *Kheperas* have gone through several iterations, with *Khepera IV* being the latest. It is a differential drive robot with two DC motors and several options for odometry: magnetic encoders, accelerometer and gyroscope. Additionally, it presents a large array of sensors to be used for obstacle detection, such as 12 IR sensors, 5 ultrasonic sensors,

and a color camera [37]. Communication with other robots or with a base is achieved by Bluetooth or WiFi [38]. Moreover, they present a control architecture with an ARM Cortex-A8 processor and a dsPIC33 microcontroller [39]. *Khepera IV* robots have a price of approximately USD 3200, with documentation and support available in [36].

The *Kilobot* robot represented a paradigm change in swarm robotics when introduced, by opting for a fairly limited hardware structure at a very low cost. It was developed by Harvard University and was promoted as being able to reach a swarm of a thousand robots [40]. Each *Kilobot* uses one wide-angle IR sensor to detect obstacles and even to detect or communicate to other robots [41,42]. Unlike most of the reviewed robots, *Kilobots* are not differential drive robots with wheels, they have legs with vibrating motors for locomotion (slip-stick principle), which also limits their locomotion to flat even surfaces [43,44]. Additionally, each robot is controlled by an Atmega328 microcontroller and can be bought for USD 130 [36].

Project *Swarmanoid* presented a new swarm structure by developing several kinds of mobile robots that collaborate towards a goal [45]. In this paper, the *Footbot* robot is reviewed due to its similarity to the other robots under evaluation. It is a differential drive robot that uses two motors to move two wheels and two tracks. Additionally, it has an accelerometer and gyroscope for odometry. The *Swarmanoid Footbot* features a ring of 24 IR proximity sensors, 4 IR distance sensors in a rotating platform, and 2 cameras, all of which can be used for obstacle detection. Furthermore, it features eight IR sensors at the base and an RFID reader and communicates to other robots by using color LEDs or WiFi. *Footbot* has an ARM 11 microprocessor and two microcontrollers. Finally, *Swarmanoid* was a joint Future and Emerging Technologies (FET-OPEN) project led by Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle—Université Libre de Bruxelles. The cost has not been specified in the literature and the robots are not commercially available.

The *Colias* robot is an effort from the University of Lincoln. It is a differential drive robot with two motors, nine IR sensors for obstacle detection and inter robot communication, and a light sensor [46,47]. Additionally, it is controlled by an Atmega168 and an Atmega-644 microcontroller. Notably, the reported cost is approximately USD 32, which is the lowest among the reviewed robots [48].

University of Manchester's *Mona* robot has a relatively low cost of USD 129, with five IR sensors and several communication options (WiFi, XBee, and RF 2.4 GHz transceiver) [49]. It is a differential drive robot with magnetic encoders for odometry and an Atmega328 microcontroller [50].

Psi Swarm is a robot created by the University of York, with no commercial availability, nor price specified. It has a differential drive configuration, with optical encoders, accelerometer, and gyroscope that can be used for odometry. Furthermore, its sensors suite features eight IR sensors for obstacle detection, five IR base sensors, and a color sensor. It has an ARM Cortex M3 microcontroller and Bluetooth or RF communication (with an RF 433 MHz transceiver) [51,52].

Georgia Tech have presented their *Robotarium* project, which uses their *GRITsbot* robots. These robots are not meant to be commercialized, but rather to be used remotely by researchers [53]. *GRITsbot* is a differential drive robot with stepper motors. Odometry is performed by tracking the step count and using an accelerometer and gyroscope. Obstacle avoidance is achieved by the use of 6 IR sensors. Control is performed by an ESP8266 board, as well as an Atmega168 and an Atmega328 microcontroller. Communication is documented to be conducted by WiFi and a RF 2.4 GHz transceiver. Finally, each robot is documented to have a cost of USD 50 in parts [54].

The last reviewed robot, *Thymio*, was introduced in 2013 by Mobsya (an EPFL spin-off). It is a differential drive robot intended for research in robotics and STEM education [55,56]. It has seven IR sensors for obstacle detection and short-range inter-robot communication. It also has two IR base sensors, an accelerometer, a microphone, and a temperature sensor [57,58]. However, in its basic configuration, *Thymio* lacks

some features needed for higher education in robot control and navigation [59]. Its control architecture was not disclosed and it has a retail price of USD 173 [60].

Table 3. Hardware features of main robots for swarm use.

Robot/Feature	Sensors						Communication				Controller			
	Differential Drive	Encoders	Infrared (IR)	Ultrasonic	Accelerom./Gyro	Camera	Microphone	IR Short-range	Bluetooth	WiFi	Radio Frequency	Other	Microcontroller(s)	Microprocessor
E-puck	✓		✓		✓	✓	✓	✓	✓				✓	
Khepera IV	✓	✓	✓	✓	✓	✓	✓		✓	✓			✓	✓
Kilobot			✓					✓					✓	
Swarmanoid	✓		✓			✓				✓		✓	✓	✓
Colias	✓		✓					✓					✓	
Mona	✓	✓	✓							✓	✓	✓	✓	
Psi Swarm	✓	✓	✓		✓				✓		✓		✓	
GRITSBot	✓		✓		✓					✓	✓		✓	✓
Thymio	✓		✓		✓		✓	✓						

5. Swarm Robotics Applications

Swarm robotics can be applied in many fields, for example, formations, mapping, localization, path planning, object transport and manipulation, coverage, aggregation, foraging, object assembly, self-assembly, tracking, and others [4,5,8,9,11,12].

In this section, we present some of the latest works made with SR. We divide our search results into two categories: (i) those that employed a real life swarm and (ii) those that were made mainly in simulated environments. For each of these categories, if found, we will present the specific platform or software that was used.

5.1. Navigation

Navigation in swarm robotics refers to the scenarios where a robot, with limited sensing and localization capabilities, is able to reach a target in an unknown location with the help of other robots [11].

Cardona and Calderon in [61] develop research on swarm robotics navigation aimed at victim detection. The navigation strategy is based on the application of particle swarm theory, where the attraction and repulsion forces of swarm particle systems are used to avoid obstacles, keep the swarm compact, and navigate to a target location. Once an agent finds a victim, it separates from the main swarm by creating a sub-swarm. The sub-swarm agents use a modified rendezvous consensus algorithm to perform formation control around the victim. This research was simulated using Matlab and the Virtual Robot Experimentation Platform (V-Rep) software; they conducted experiments using 23 agents (drones) and 6 victims distributed in five places.

Nedjah and Luneque in [62] propose a collective strategy for swarm navigation in scenarios with obstacles and in the presence of two and four clusters of robots. The strategy is based on wave algorithms and is achieved following a sequence composed of recruitment, alignment, and movement subtasks. The evaluation of its approach was performed by simulations using V-Rep, and the tests demonstrate the effectiveness of the method.

5.2. Foraging

In SR, foraging refers to finding items scattered in environments and bringing them back to a specific area called the “nest” [11].

Talamali et al. in [63] propose a collective foraging system based on virtual pheromones. It was tested both as a computer simulation and with 200 real robots. The physics-based

simulations were conducted with ARGoS. For the real swarm, an augmented reality for kilobot software (ARK) was used in order to simulate the pheromone. The results showed that simple behaviors generally reduce the impact of the reality gap and preserve consistent dynamics in reality.

Castello et al. in [64] present a division of labor algorithm for a simple foraging task that acts to maintain a target amount of food at the nest despite consumption rates that vary over time. Its algorithm, the adaptive response threshold model (ARTM), showed to be efficient in achieving adaptive workload distribution for a small-sized robot swarm. ARTM also reduces a common problem of real robots: the duration of collisions among them. The experiments were carried out both in simulation and with a real swarm. For the simulation, a multi-robot simulation library STAGE was used. For the physic swarm, they used five e-pucks.

Foraging behavior was used to validate the swarm performance of Kilobot robots. In this scenario, one robot was assigned the “food” role and another—the “nest” role. Next, the rest of the robots try to reach the food and take it to the nest. Some robots may assume the “beacon” role to help others reach food and nest easily. In [41], a swarm of 100 Kilobots showed to be capable of such collective behavior.

5.3. Exploration

Exploration in SR systems refers to the collective behavior in which robots cooperate to explore an environment in order to accomplish some tasks such as monitoring, surveillance, space coverage, and others [14].

Duarte et al. in [65] propose the use of swarm robotics systems to carry out marine environmental monitoring missions, focusing on the control of temperature. They conduct a simulation-based evaluation of the robot’s performance over large areas and with large swarm sizes. Moreover, they implemented their proposal in a real aquatic swarm composed of eight units. The results showed that the use of swarm systems is useful in environmental monitoring tasks that involve covering areas.

Solis and Calderón in [66] present a scheme to solve the problem to explore, cover, and find paths in dangerous environments. The proposal consisted of three phases: exploration, mapping, and path planning. For the first phase, a cellular automata algorithm along with a stigmergy approach for the communication between the agents was used. In the second phase, the explored cell was stored in order to create a grid map. Finally, in the path planning phase, a multiobjective problem was used to minimize distance and danger, through a graph obtained by an adapted RRG, and a genetic algorithm known as modified NSGA-II. The simulation was carried out with the Processing software.

5.4. Aggregation

Aggregation is one of the most fundamental swarm behaviors frequently observed in nature. It refers to the task in which each individual positions themselves close enough to each other in one specific place [12].

Amjadi et al. in [67] propose a method that combines the BEECLUST algorithm with pheromone-following behavior to solve the problem of finding the source of a chemical leakage and clean the contaminated area. They conducted experiments with a simulated model of a Mona robot in the Webots software. The effects of population size and robot speed on the ability of the swarm in a decontamination task were analyzed. The results showed the feasibility of deploying robotic swarms in an exploration and cleaning task in an extreme environment.

In this same context, but with real-life robots, Colias robots were validated by implementing BEECLUST behavior with swarm sizes from 5 to 20 robots. Robots showed a coherent response with aggregation characteristics. Nonetheless, aggregation decreased in time as the number of robots increased [46,47].

Mona robots were also validated by implementing BEECLUST behavior. It used a swarm of five robots in an educational activity. Agent aggregation was effectively observed, but no further swarm sizes were tested [50].

Notably, in Colias and Mona implementations, comparing different swarms was a challenging task. Behavior characteristics, such as aggregation time or size of aggregate, depend heavily on the scenario configuration. For example, sources of variation on a scenario are arena size, robot size, robot speed, aggregation cue, and sensor noise.

On the other hand, Ramroop et al. in [68] present an implementation of a bio-inspired aggregation algorithm, in which the goal is for the swarm to find an optimal thermostatic zone in which to converge. The experiments were carried out both in simulation and with real robots using the Mona robot. For the simulation, they used the software Player/Stage in order to analyze the swarm behavior of a large number of robots. Additionally, for the real implementation, a swarm of five Mona robots was programmed. The results showed reliable aggregations and a population-dependent swarm performance.

5.5. Other Applications

Collective Decision or Consensus behaviors are the general names for the collective generation of a choice in a swarm. E-puck robots were validated with this type of behavior. Each robot sent its current action (right or left wall following) to the rest of the swarm. Therefore, based on actions from fellow robots, a single agent can change its direction with a probabilistic decision. In addition, the power of the communication transmission was varied during the experiment. An inverse correlation between power intensity and time to achieve consensus has been observed [69].

Thymio robots have been validated both as an educational tool and as a robotic swarm platform. The latter was achieved by implementing a Collective Decision algorithm inspired by house-hunting honeybees [70]. In this case, Thymio robots were assigned a role as bee or nest. Bees looked out for a nest and committed to it based on a value-sensitive decision. Consequently, consensus was achieved in a few minutes in all cases [56].

A branch of Consensus behavior is *Rendezvous* algorithms. They, in general, consist of a group of robots that seek to meet in a physical space and create a cluster. For example, GRITsbot robots were validated by performing random walks on two agents until they enter a mutual sensing range and reach each other. Successful rendezvous was achieved with two robots in [54] and with six robots in [53].

A biologically inspired behavior, commonly used in robotic swarms, is *Formation Control*. Kilobots were also validated with this behavior in [41]. A swarm of six Kilobots successfully achieved a collective performance by following a leader through a path. Additionally, GRITsbots were validated with Formation Control in [53]. A group of six robots started at random positions and then achieved a hexagon formation.

The Swarmanoid project implemented a *Search and Retrieval* behavior to demonstrate its approach to swarm robotics. First, Footbots and Eye-bots (aerial robots) performed an exploration of a real human environment until they found a target object. Second, the third type of robot (Hand-bot) grasped the object. Finally, Hand-bot and Footbots retrieved the target object to a goal area [45].

Several models of Khepera robots have been used in swarm research and demonstration. A swarm implementation of the latest model, Khepera IV, is found in [38] with a group of 10 robots. Two swarm demonstrations were presented: first, a patrol exercise to test backup capabilities within the group, and then, a search exercise to find a specific object.

Finally, Psi Swarm robots were used and tested in [52] with a mixed swarm group of 10 Psi Swarm robots and 15 E-Pucks robots. No specific behavior, or performance, was detailed.

Table 4 summarizes the most used applications of swarm robotics, comparing real-life implementations and simulations.

Table 4. Swarm robotics applications.

Application	Real-Life Swarm		Swarm Simulation	
	Implemented	Platforms	Implemented	Software
Navigation			✓	Matlab V-Rep
Foraging	✓	E-pucks Kilobots	✓	ArGoS ARK Stage
Exploration	✓	Unnamed marine robot	✓	Processing
Aggregation	✓	Colias Mona	✓	Webots Player/Stage

It is also important to mention that most of the time, the swarm size depends on the number of robots that companies or research agencies have in stock for arbitrary reasons [31], and robots may not be intended for all swarm applications.

6. Discussion

As it has been shown, swarm robotics have great potential to solve (or improve) a large number of problems and situations. However, applying SR in real-life environments is currently limited by existing technology: environment interaction, noise, sensors, actuators, infrastructure-based network, security protocols, energy, components degradation, and failures, as detailed in Section 3.2.

Nonetheless, real swarm robotics implementations are important and necessary to test and suggest real solution to real problems. Simulations and virtual reality represent a previous step on the way to reality.

Thus, many experiments that have been carried out so far can be classified into two types: (i) simulations of algorithms and (ii) creation and/or use of existing mobile robots in controlled environments.

For simulations, as presented in Section 3.1, there is specialized software for SR, general robotics simulators, and even software libraries that can be adapted for swarm research. Choosing “the best” among them will depend on the corresponding research goal.

If the main purpose is to test or design a robot for a swarm, taking into account its locomotion, sensors, and physics, then ODE, MSRS, and CoppeliaSim present the most suitable features. These programs allow to simulate robot models as similar to real life as possible. However, they were not created to simulate collective swarm behaviors, but some modifications can be made to use them for this purpose.

Similarly, Webots offers the possibility of using different existing robot models. It presents an appropriate environment for testing collective robot behaviors, although it is not easily scalable.

On the other hand, if the aim is to test swarm robotics algorithms, software such as Stage, Gazebo, and ARGoS may seem more suitable. These three programs are among the most used when behavior simulation is the main focus. They offer a balance between the simulation of algorithms and the physics of robots. While many of them can be used to simulate most swarm applications, Stage offers the best performance for swarms with a large number of agents (up to 100,000).

Despite not being created for swarm robotics research, Bio-PEPA software can be adapted to simulate the biological behaviors of general swarms.

There is a wide variety of real-life swarm robotic platforms. The spectrum of possibilities range from inexpensive and limited robot agents to commercial robots full of resources. However, there is no rule for defining the right swarm size, most of them depend on the creators’ or developers’ possibilities and sponsors.

For example, robots such as Kilobot, Thymio, Mona, and Colias have been offered as an economic option. Nevertheless, they have the downside of presenting some limited

features. In practice, this may cause some collective behaviors to not be implemented or to need to rely on external hardware (such as cameras, screens, and projectors).

It can be noted, based on the list of features presented in Section 4, that Khepera IV robots have the larger sensor and processing suite. Thus, this robot appears in a broad range of education and research publications, not limited to swarm robotics. In contrast, they are the most expensive robot platform among the ones reviewed, which limits the ability to scale up the swarm size.

The selection of the most suitable robot for real testing will depend on (i) economic aspects, (ii) the amount and type of sensors, and (iii) the communication type needed for a specific swarm application.

7. Conclusions

Swarm robotics combines Swarm Intelligence with Multi-Robot Systems. In order to be considered a swarm, a group of robots should have autonomy, homogeneity, large number, limited capabilities, and a specific communication and control scheme. In most cases, swarms of robots are designed and defined for a specific behavior or application.

Simulation and modeling are useful tools to approach or deepen the designs and behaviors of robots. In general, robot simulators are developed with a specific objective, for that reason, most of the time, it is not possible to compare them. Each researcher must take into account the specific characteristics of each software, and choose the one whose features best suit their needs.

Virtual reality is a possible middle step between simulation and reality in robotics research. Even though it is not possible to simulate all aspects of reality, these kinds of simulations are often used to allow having a glimpse of how a swarm will behave in real life. These types of experiments have been increasing, mainly because they are faster to generate and less expensive and time-consuming.

However, working and experimenting with real-life robots is a necessary stage towards evolving the research around the subject. Nowadays, robots are developed with different intentions, for instance, they might focus on education or industrial production. When robots are intended for swarm robotics research, usually, they are validated by creating a robotic swarm with a known behavior, and then the collective response is tracked.

Most real robot experiments presented in the literature were performed in controlled environments, far from the scenarios which they are assumed to recreate. This shows that there are still many open problems around the subject. Only further research will help in taking swarm robotics to next level.

This review presents references to compare and choose software, simulators, applications, behaviors, and robots to explore and research swarm robotics. Moreover, it provides options for transitioning from simulations to real swarm robotic implementations.

Author Contributions: Author Contributions: Conceptualization, C.C.-A., J.C.B.-T. and R.S.-O.; methodology, C.C.-A., J.C.B.-T. and R.S.-O.; formal analysis, C.C.-A., J.C.B.-T. and R.S.-O.; data curation, C.C.-A., J.C.B.-T. and R.S.-O.; writing—original draft preparation, C.C.-A., J.C.B.-T. and R.S.-O.; writing—review and editing, C.C.-A., J.C.B.-T. and R.S.-O. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Costa Rica Institute of Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Special thanks to the Research and Outreach of the Costa Rica Institute of Technology (VIE, ITCR), for their support of the PROE project (code VIE 1440036). These results are part of that project.

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

References

1. Beni, G.; Wang, J. Swarm intelligence in cellular robotic systems. In *Robots and Biological Systems: Towards a New Bionics?* Springer: Berlin/Heidelberg, Germany, 1993; pp. 703–712.
2. Dorigo, M.; Theraulaz, G.; Trianni, V. Swarm Robotics: Past, Present, and Future [Point of View]. *Proc. IEEE* **2021**, *109*, 1152–1165. [\[CrossRef\]](#)
3. Mavrovouniotis, M.; Li, C.; Yang, S. A survey of swarm intelligence for dynamic optimization: Algorithms and applications. *Swarm Evol. Comput.* **2017**, *33*, 1–17. [\[CrossRef\]](#)
4. Barca, J.C.; Sekercioglu, Y.A. Swarm robotics reviewed. *Robotica* **2013**, *31*, 345–359. [\[CrossRef\]](#)
5. Khaldi, B.; Cherif, F. An overview of swarm robotics: Swarm intelligence applied to multi-robotics. *Int. J. Comput. Appl.* **2015**, *126*, 31–37. [\[CrossRef\]](#)
6. Jose, K.; Pratihari, D.K. Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods. *Robot. Auton. Syst.* **2016**, *80*, 34–42. [\[CrossRef\]](#)
7. Khamis, A.; Hussein, A.; Elmogy, A. Multi-robot task allocation: A review of the state-of-the-art. In *Cooperative Robots and Sensor Networks 2015*; Springer: Cham, Switzerland, 2015; pp. 31–51.
8. Senanayake, M.; Senthoooran, I.; Barca, J.C.; Chung, H.; Kamruzzaman, J.; Murshed, M. Search and tracking algorithms for swarms of robots: A survey. *Robot. Auton. Syst.* **2016**, *75*, 422–434. [\[CrossRef\]](#)
9. Şahin, E. Swarm robotics: From sources of inspiration to domains of application. In Proceedings of the SAB 2004 International Workshop, Santa Monica, CA, USA, 17 July 2004; pp. 10–20.
10. Haidegger, T.; Galambos, P.; Rudas, I. Robotics 4.0—Are we there yet? In Proceedings of the 2019 IEEE 23rd International Conference on Intelligent Engineering Systems (INES), Gödöllő, Hungary, 25–27 April 2019.
11. Bayındır, L. A review of swarm robotics tasks. *Neurocomputing* **2016**, *172*, 292–321. [\[CrossRef\]](#)
12. Hamann, H. *Swarm Robotics: A Formal Approach*; Springer: Cham, Switzerland, 2018.
13. Nedjah, N.; Junior, L.S. Review of methodologies and tasks in swarm robotics towards standardization. *Swarm Evol. Comput.* **2019**, *50*, 100565. [\[CrossRef\]](#)
14. Brambilla, M.; Ferrante, E.; Birattari, M.; Dorigo, M. Swarm robotics: A review from the swarm engineering perspective. *Swarm Intell.* **2013**, *7*, 1–41. [\[CrossRef\]](#)
15. Erez, T.; Tassa, Y.; Todorov, E. Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 4397–4404.
16. Vaughan, R. Massively multi-robot simulation in stage. *Swarm Intell.* **2008**, *2*, 189–208. [\[CrossRef\]](#)
17. Massink, M.; Brambilla, M.; Latella, D.; Dorigo, M.; Birattari, M. Analysing robot swarm decision-making with Bio-PEPA. In Proceedings of the 8th International Conference, ANTS 2012, Brussels, Belgium, 12–14 September 2012; pp. 25–36.
18. Ciocchetta, F.; Hillston, J. Bio-PEPA: A framework for the modelling and analysis of biological systems. *Theor. Comput. Sci.* **2009**, *410*, 3065–3084. [\[CrossRef\]](#)
19. Balch, T. *Behavioral Diversity in Learning Robot Teams*; Technical Report; Georgia Institute of Technology: Atlanta, GA, USA, 1998.
20. Mondada, F.; Pettinaro, G.C.; Guignard, A.; Kwee, I.W.; Floreano, D.; Deneubourg, J.L.; Nolfi, S.; Gambardella, L.M.; Dorigo, M. SWARM-BOT: A new distributed robotic concept. *Auton. Robots* **2004**, *17*, 193–221. [\[CrossRef\]](#)
21. Smith, R. Open Dynamics Engine, V0.5 User Guide. 2005. Available online: <http://ode.org/ode-latest-userguide.pdf> (accessed on 11 May 2022).
22. Hoffman, E.M.; Traversaro, S.; Rocchi, A.; Ferrati, M.; Settini, A.; Romano, F.; Natale, L.; Bicchi, A.; Nori, F.; Tsagarakis, N.G. Yarp based plugins for gazebo simulator. In Proceedings of the International Workshop on Modelling and Simulation for Autonomous Systems, Rome, Italy, 5–6 May 2014; pp. 333–346.
23. Michel, O. Cyberbotics Ltd. Webots™: Professional mobile robot simulation. *Int. J. Adv. Robot. Syst.* **2004**, *1*, 5. [\[CrossRef\]](#)
24. Jackson, J. Microsoft robotics studio: A technical introduction. *IEEE Robot. Autom. Mag.* **2007**, *14*, 82–87. [\[CrossRef\]](#)
25. Carpin, S.; Lewis, M.; Wang, J.; Balakirsky, S.; Scrapper, C. USARSim: A robot simulator for research and education. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Rome, Italy, 10–14 April 2007; pp. 1400–1405.
26. Pinciroli, C.; Trianni, V.; O’Grady, R.; Pini, G.; Brutschy, A.; Brambilla, M.; Mathews, N.; Ferrante, E.; Di Caro, G.; Ducatelle, F.; et al. ARGoS: A modular, multi-engine simulator for heterogeneous swarm robotics. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 5027–5034.
27. Rohmer, E.; Singh, S.P.; Freese, M. V-REP: A versatile and scalable robot simulation framework. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 1321–1326.
28. Francesca, G.; Birattari, M. Automatic design of robot swarms: Achievements and challenges. *Front. Robot. AI* **2016**, *3*, 29. [\[CrossRef\]](#)
29. Martinez-Gonzalez, P.; Oprea, S.; Garcia-Garcia, A.; Jover-Alvarez, A.; Orts-Escolano, S.; Garcia-Rodriguez, J. Unrealrox: An extremely photorealistic virtual reality environment for robotics simulations and synthetic data generation. *Virtual Real.* **2019**, *24*, 271–288. [\[CrossRef\]](#)
30. Gupta, M.; Saxena, D.; Kumari, S.; Kaur, D. Issues and applications of swarm robotics. *Int. J. Res. Eng. Technol. Sci.* **2016**, *6*, 1–5.

31. Schranz, M.; Umlauf, M.; Sende, M.; Elmenreich, W. Swarm robotic behaviors and current applications. *Front. Robot. AI* **2020**, *7*, 36. [CrossRef]
32. Hoenig, W.; Milanes, C.; Scaria, L.; Phan, T.; Bolas, M.; Ayanian, N. Mixed reality for robotics. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 5382–5387.
33. Mondada, F.; Bonani, M.; Raemy, X.; Pugh, J.; Cianci, C.; Klapotocz, A.; Magnenat, S.; Zufferey, J.C.; Floreano, D.; Martinoli, A. The e-puck, a robot designed for education in engineering. In Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions, Castelo Branco, Portugal, 7 May 2009; Volume 1, pp. 59–65.
34. Chen, J.; Gauci, M.; Price, M.J.; Groß, R. Segregation in swarms of e-puck robots based on the brazil nut effect. In Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, Valencia, Spain, 4–8 June 2012; Volume 1, pp. 163–170.
35. Pitonakova, L.; Winfield, A.; Crowder, R. Recruitment near worksites facilitates robustness of foraging E-puck swarms to global positioning noise. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4276–4281.
36. Products—RoadNarrows Robotics. Available online: <https://roadnarrows-robotics.github.io/> (accessed on 11 May 2022).
37. Peralta, E.; Fabregas, E.; Farias, G.; Vargas, H.; Dormido, S. Development of a Khepera IV Library for the V-REP Simulator. *IFAC-PapersOnLine* **2016**, *49*, 81–86. [CrossRef]
38. Varadharajan, V.S.; St-Onge, D.; Adams, B.; Beltrame, G. SOUL: Data sharing for robot swarms. *Auton. Robots* **2019**, *44*, 377–394. [CrossRef]
39. Soares, J.M.; Navarro, I.; Martinoli, A. The Khepera IV mobile robot: Performance evaluation, sensory data and software toolbox. In Proceedings of the Robot 2015: Second Iberian Robotics Conference, Lisbon, Portugal, 19–21 November 2015; pp. 767–781.
40. Pinciroli, C.; Talamali, M.S.; Reina, A.; Marshall, J.A.; Trianni, V. Simulating Kilobots within ARGoS: Models and experimental validation. In Proceedings of the International Conference on Swarm Intelligence, Shanghai, China, 17–22 June 2018; pp. 176–187.
41. Rubenstein, M.; Ahler, C.; Hoff, N.; Cabrera, A.; Nagpal, R. Kilobot: A low cost robot with scalable operations designed for collective behaviors. *Robot. Auton. Syst.* **2014**, *62*, 966–975. [CrossRef]
42. Valentini, G.; Ferrante, E.; Hamann, H.; Dorigo, M. Collective decision with 100 Kilobots: Speed versus accuracy in binary discrimination problems. *Auton. Agents Multi-Agent Syst.* **2016**, *30*, 553–580. [CrossRef]
43. Reina, A.; Cope, A.J.; Nikolaidis, E.; Marshall, J.A.; Sabo, C. Ark: Augmented reality for kilobots. *IEEE Robot. Autom. Lett.* **2017**, *2*, 1755–1761. [CrossRef]
44. Dimidov, C.; Oriolo, G.; Trianni, V. Random walks in swarm robotics: An experiment with kilobots. In Proceedings of the International Conference on Swarm Intelligence, Bali, Indonesia, 25–30 June 2016; pp. 185–196.
45. Dorigo, M.; Floreano, D.; Gambardella, L.M.; Mondada, F.; Nolfi, S.; Baaboura, T.; Birattari, M.; Bonani, M.; Brambilla, M.; Brutschy, A.; et al. Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robot. Autom. Mag.* **2013**, *20*, 60–71. [CrossRef]
46. Arvin, F.; Murray, J.; Zhang, C.; Yue, S. Colias: An autonomous micro robot for swarm robotic applications. *Int. J. Adv. Robot. Syst.* **2014**, *11*, 113. [CrossRef]
47. Arvin, F.; Murray, J.C.; Shi, L.; Zhang, C.; Yue, S. Development of an autonomous micro robot for swarm robotics. In Proceedings of the 2014 IEEE International Conference on Mechatronics and Automation, Tianjin, China, 3–6 August 2014; pp. 635–640.
48. Arvin, F.; Xiong, C.; Yue, S. Colias-φ: An autonomous micro robot for artificial pheromone communication. *Int. J. Mech. Eng. Robot. Res.* **2015**, *4*, 349–353. [CrossRef]
49. Arvin, F.; Espinosa, J.; Bird, B.; West, A.; Watson, S.; Lennox, B. Mona: An affordable open-source mobile robot for education and research. *J. Intell. Robot. Syst.* **2019**, *94*, 761–775. [CrossRef]
50. Arvin, F.; Mendoza, J.L.E.; Bird, B.; West, A.; Watson, S.; Lennox, B. Mona: An affordable mobile robot for swarm robotic applications. In Proceedings of the UK-RAS Conference on Robotics and Autonomous Systems, Coimbra, Portugal, 26–28 April 2017; pp. 49–52.
51. Hilder, J.; Horsfield, A.; Millard, A.G.; Timmis, J. The Psi swarm: A low-cost robotics platform and its use in an education setting. In Proceedings of the Annual Conference Towards Autonomous Robotic Systems, Sheffield, UK, 26 June–1 July 2016; pp. 158–164.
52. Millard, A.G.; Redpath, R.; Jewers, A.M.; Arndt, C.; Joyce, R.; Hilder, J.A.; McDaid, L.J.; Halliday, D.M. ARDebug: An augmented reality tool for analysing and debugging swarm robotic systems. *Front. Robot. AI* **2018**, *5*, 87. [CrossRef]
53. Pickem, D.; Glotfelter, P.; Wang, L.; Mote, M.; Ames, A.; Feron, E.; Egerstedt, M. The robotarium: A remotely accessible swarm robotics research testbed. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 1699–1706.
54. Pickem, D.; Lee, M.; Egerstedt, M. The GRITSBot in its natural habitat—a multi-robot testbed. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 4062–4067.
55. Riedo, F.; Chevalier, M.; Magnenat, S.; Mondada, F. Thymio II, a robot that grows wiser with children. In Proceedings of the 2013 IEEE Workshop on Advanced Robotics and its Social Impacts, Tokyo, Japan, 7–9 November 2013; pp. 187–193.
56. Vitanza, A.; Rossetti, P.; Mondada, F.; Trianni, V. Robot swarms as an educational tool: The Thymio’s way. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881418825186. [CrossRef]

57. Mondada, F.; Bonani, M.; Riedo, F.; Briod, M.; Pereyre, L.; Rétornaz, P.; Magnenat, S. Bringing robotics to formal education: The thymio open-source hardware robot. *IEEE Robot. Autom. Mag.* **2017**, *24*, 77–85. [[CrossRef](#)]
58. Wahby, M.; Petzold, J.; Eschke, C.; Schmickl, T.; Hamann, H. Collective change detection: Adaptivity to dynamic swarm densities and light conditions in robot swarms. In Proceedings of the 2018 Conference On Artificial Life: A Hybrid of the European Conference on Artificial Life (ECAL) and the International Conference on the Synthesis and Simulation of Living Systems (ALIFE), Newcastle, UK, 29 July–2 August 2019; pp. 642–649.
59. Guzzi, J.; Giusti, A.; Di Caro, G.A.; Gambardella, L.M. Mighty thymio for university-level educational robotics. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
60. Thymio Educational Robot—Roboshop. Available online: <https://www.robotshop.com/en/thymio-educational-robot.html> (accessed on 11 May 2022).
61. Cardona, G.A.; Calderon, J.M. Robot swarm navigation and victim detection using rendezvous consensus in search and rescue operations. *Appl. Sci.* **2019**, *9*, 1702. [[CrossRef](#)]
62. Junior, L.S.; Nedjah, N. Efficient strategy for collective navigation control in swarm robotics. *Procedia Comput. Sci.* **2016**, *80*, 814–823. [[CrossRef](#)]
63. Talamali, M.S.; Bose, T.; Haire, M.; Xu, X.; Marshall, J.A.; Reina, A. Sophisticated collective foraging with minimalist agents: A swarm robotics test. *Swarm Intell.* **2020**, *14*, 25–56. [[CrossRef](#)]
64. Castello, E.; Yamamoto, T.; Dalla Libera, F.; Liu, W.; Winfield, A.F.; Nakamura, Y.; Ishiguro, H. Adaptive foraging for simulated and real robotic swarms: The dynamical response threshold approach. *Swarm Intell.* **2016**, *10*, 1–31. [[CrossRef](#)]
65. Duarte, M.; Gomes, J.; Costa, V.; Rodrigues, T.; Silva, F.; Lobo, V.; Marques, M.M.; Oliveira, S.M.; Christensen, A.L. Application of swarm robotics systems to marine environmental monitoring. In Proceedings of the OCEANS 2016—Shanghai, Shanghai, China, 10–13 April 2016; pp. 1–8.
66. Solis-Ortega, R.; Calderon-Arce, C. Multiobjective problem to find paths through swarm robotics. In Proceedings of the 2019 3rd International Conference on Automation, Control and Robots, Prague, Czech Republic, 11–13 October 2019; pp. 12–21.
67. Amjadi, A.S.; Raoufi, M.; Turgut, A.E.; Broughton, G.; Krajník, T.; Arvin, F. Cooperative pollution source localization and cleanup with a bio-inspired swarm robot aggregation. *arXiv* **2019**, arXiv:1907.09585.
68. Ramroop, S.; Arvin, F.; Watson, S.; Carrasco-Gomez, J.; Lennox, B. A bio-inspired aggregation with robot swarm using real and simulated mobile robots. In Proceedings of the Annual Conference Towards Autonomous Robotic Systems, Bristol, UK, 25–27 July 2018; pp. 317–329.
69. Cianci, C.M.; Raemy, X.; Pugh, J.; Martinoli, A. Communication in a swarm of miniature robots: The e-puck as an educational tool for swarm robotics. In Proceedings of the International Workshop on Swarm Robotics, Rome, Italy, 30 September–1 October 2006; pp. 103–115.
70. Reina, A.; Bose, T.; Trianni, V.; Marshall, J.A. Effects of spatiality on value-sensitive decisions made by robot swarms. In *Distributed Autonomous Robotic Systems*; Springer: Cham, Switzerland, 2018; pp. 461–473.