

Article

Not peer-reviewed version

---

# From Network Sensors to Intelligent Systems: A Decade-Long Review of Swarm Robotics Technologies

---

[Fouad Chaouki Refis](#)\*, [Nassim Ahmed Mahammedi](#), [Chaker Abdelaziz Kerrache](#), [Sahraoui Dhelim](#)

Posted Date: 22 August 2025

doi: 10.20944/preprints202508.1417.v1

Keywords: swarm robotics; systematic review; swarm intelligence; metaheuristics; Robot Operating System (ROS); hardware; software; standardization



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

## Article

# From Network Sensors to Intelligent Systems: A Decade-Long Review of Swarm Robotics Technologies

Fouad Chaouki Refis <sup>1,\*</sup>, Nassim Ahmed Mahammed <sup>2</sup>, Chaker Abdelaziz Kerrache <sup>3</sup> and Sahraoui Dhelim <sup>4</sup>

<sup>1</sup> Department of Computer Science, University of Batna 2, Algeria

<sup>2</sup> The Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg, Luxembourg City, L-1855, Luxembourg

<sup>3</sup> Laboratoire d'Informatique et de Mathématiques, University Amar Telidji of Laghouat, Laghouat, Algeria

<sup>4</sup> The School of Computing, Dublin City University, Ireland

\* Correspondence: fouad.refis@etu.univ-batna2.dz

## Abstract

Swarm Robotics (SR) is a relatively new field, inspired by the collective intelligence of social insects. It involves using local rules to control and coordinate large groups (swarms) of relatively simple physical robots. Important tasks that robot swarms can handle include demining, search, rescue, and cleaning up toxic spills. Over the past decade, the research effort in the field of swarm robotics has intensified significantly in terms of hardware, software, and systems integrated developments, yet challenges still exist; particularly concerning standardization, scalability, and affordable deployment. To contextualize the state of swarm robotics technologies, this paper provides a systematic literature review (SLR) of swarm robotic technologies published from 2014 to 2024, with an emphasis on how hardware and software subsystems have co-evolved. We contribute an overview of 40 studies in peer-reviewed journals along with a well-defined and replicable systematic review protocol. The protocol describes criteria for including and excluding studies; and outlines a data extraction approach. We explored trends in sensor hardware, actuation methods, communication devices, energy systems, as well as an examination of software platforms to produce swarm behavior, covering meta-heuristic algorithms and generic middleware platforms such as ROS. Our results demonstrate how dependent hardware and software are to achieve swarm intelligence, the lack of uniform standards for their design, and the pragmatic limits which hinder scalability and deployment. We conclude by noting ongoing challenges and proposing future directions for developing interoperable, energy-efficient swarm robotics (SR) systems incorporating machine learning (ML).

**Keywords:** swarm robotics; systematic review; swarm intelligence; metaheuristics; Robot Operating System (ROS); hardware; software; standardization

## 1. Introduction

The world outside our minds is the primary source of knowledge for us as humans. Since the earliest stages of human history, nature has profoundly inspired and shaped their thoughts. It continues to be a source of innovation, influencing everything from the tools we use to the stories we tell. This occurs through cognitive processes that transform environmental observations into actionable knowledge. Artificial intelligence (AI) was inspired by the biological intelligence of humans, and swarm robotics (SR) was inspired by how nature's swarms function. SR draws inspiration from social insects such as ants, bees, and termites, which coordinate and perform complex tasks through self-organization without centralized control [1–5]. Every technology relies on a foundation of hardware and software for effective implementation.

Similarly, SR technology is built upon hardware and software inspired by natural swarm intelligence (SI) algorithms. These so-called metaheuristic algorithms address various optimization problems. A metaheuristic is an "iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search spaces using learning strategies to structure information to find efficiently near-optimal solutions" [6]. The term is derived from two Greek words, meta and heuristic, and the concept was coined by Glover in 1986 [7]. As already mentioned, metaheuristic algorithms are used to solve optimization problems, which are problems that "ask for minimal or maximal values of an objective function on a given domain" [8]. However, hardware poses several challenges. No standard architecture for SR exists due to the diversity of SR systems. Hardware refers to the physical elements or devices used to implement swarm robotic systems. These elements include sensors, actuators, processors, communication modules, energy sources, and mechanical structures. The goal of both hardware and software in SR is to build robots that can mimic the complex swarm behavior of social animals, so the robots need to exhibit swarm characteristics, which include robustness, flexibility, scalability, decentralization, and autonomy [2,9,11].

Despite the considerable advances in hardware and software design associated with SR systems, challenges persist. The lack of a unified framework for hardware components, software architectures, or communication protocols in SR systems makes it difficult to compare research findings, develop cost-effective solutions, and transition SR systems from research labs to real-world applications. Furthermore, this lack of standardization limits cross-platform applications. There have been literature reviews providing insight into the different aspects of SR. However, few have offered a holistic review that considers both hardware and software in SR systems, while also accounting for the economic and practical constraints of developing SR systems.

In this paper, we conduct a systematic review to introduce the state-of-the-art for both the hardware and software elements of SR systems, focusing on the past decade of research. We chose a systematic review because it is a transparent and structured method to compile and evaluate research findings on a specific topic or question. The aim is to minimize the bias associated with single studies and non-traditional reviews [12]. a systematic review "renders itself amenable for replication" [13]. In contrast to previous reviews that treat hardware and software separately, our review illustrates the interplay between the two. We compare different hardware designs and evaluate the software systems that enable swarm intelligence, like the Robot Operating System (ROS). Additionally, we identify key trends, challenges, and future directions for standardizing and optimizing SR systems for future applications.

## 2. The SLR Protocol

A comprehensive and rigorous systematic literature review starts with a detailed and predefined protocol that outlines all the steps and methods to be followed during the review. This protocol includes: Research questions, an exclusion and inclusion criteria, and protocol execution procedures.

### 2.1. Research Questions

- What hardware components should SR be equipped with?
- What software should SR be equipped with?
- What future research directions are suggested for SR, and what gaps have been identified in the reviewed literature?

### 2.2. Exclusion & Inclusion Criteria

- The papers must be published between 2014-2024.
- The papers are written in English.

- The papers are the results of searching in Google Scholar, Google search engine, and Semantic Scholar.
- Exclude duplicate papers that contain redundant information.
- Papers must be four pages or longer to exclude works lacking scientific methodology and sufficient detail.
- Papers that are inaccessible are excluded.
- Exclude papers that do not contribute to the research after reviewing them.

### 2.3. Quality Assessment Criteria

#### 1. Research Design & Methodology:

\* Clearly defined research objectives: Does the paper outline its contributions and research goals?

\* Description of methodology: Is the development or experimental methodology sufficiently described?

\*Software and hardware validation: Have the suggested software and hardware components undergone adequate testing and validation?

#### 2. Technical Contribution:

\* Innovation and novelty: Does the paper offer new software or hardware solutions or enhancements?

\* Technical depth: Is the technical content rigorous and detailed enough?

#### 3. Experimental Evaluation:

\* Experiments: Are they appropriate and well-designed?

#### 4. Relevance and Impact:

\*Swarm robotics relevance: To what extent does the work relate to the field of swarm robotics?

\*Can the proposed solution be practically implemented?

### 2.4. Execution

Keyword selection was based on the study subject and applied to paper titles. The keywords used were: In Google Scholar and Semantic Scholar, those keywords were used: (swarm and (robot or robots or robotics) and (design or hardware or software)) + how to create an SR. In Google search engine: hardware design implementation of swarm robots.

The results are:

99 papers → Google Scholar.

72 papers → Semantic scholar.

25 papers → Google search engine.

196 paper → total.

After applying the criteria outlined in subsections **B** and **C**, 40 out of 196 papers met the requirements, representing 20.41% of the total.

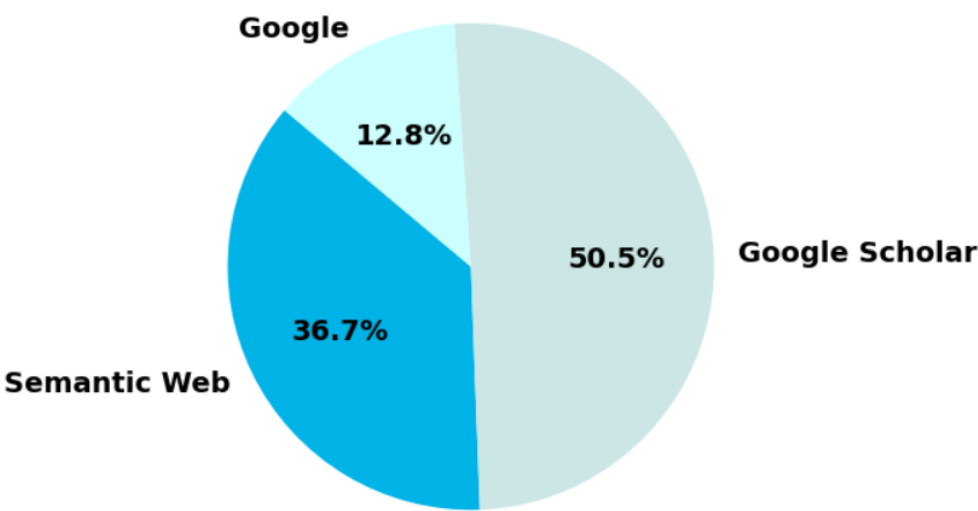


Figure 1. Distribution of the 196 scientific papers across different search engines

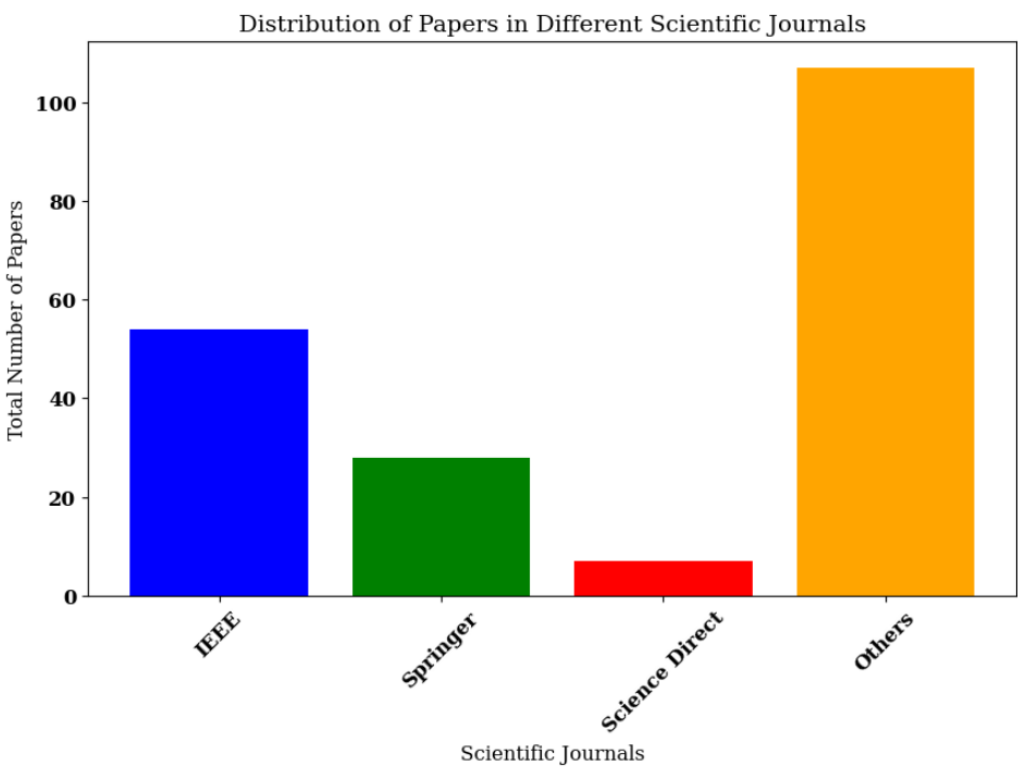
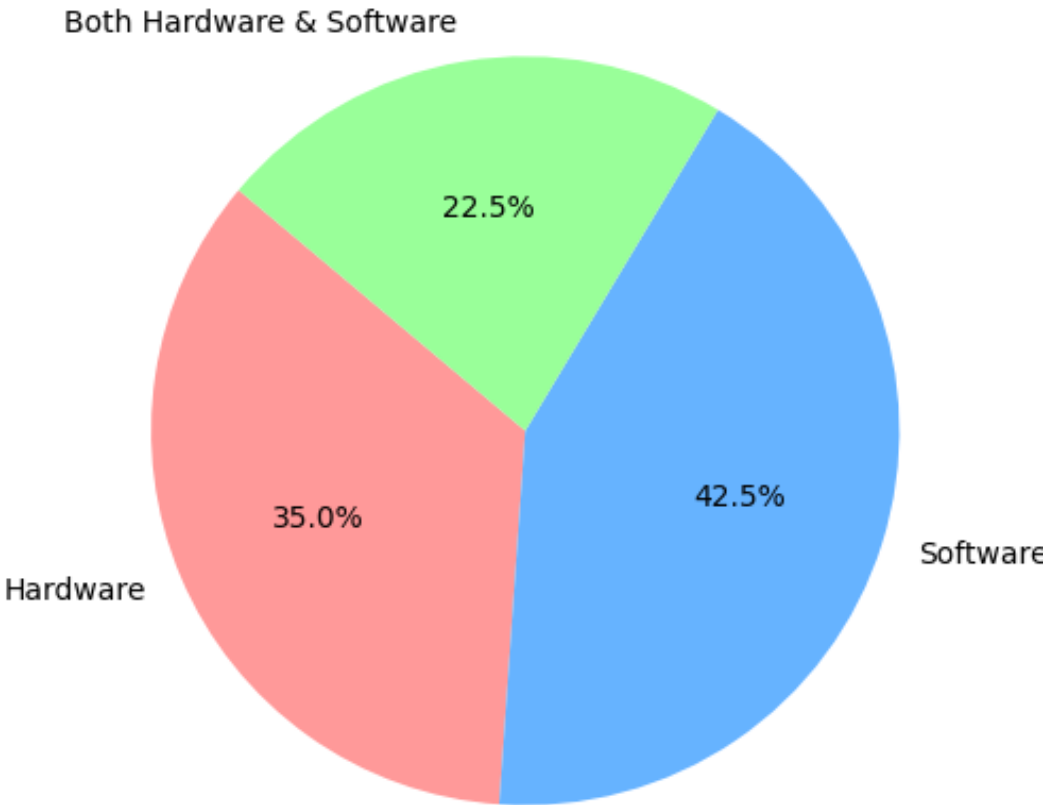


Figure 2. Distribution of the scientific papers regarding the scientific journals



**Figure 3.** Distribution of the 40 selected papers regarding their focus area (14 Hardware, 17 Software, 9 Both Hardware & Software)

Before proceeding, it is worth noting that the SR field needs an “urgent standardization in several aspects, including the robots hardware and software” as Nedjah and Junior stated [14]. According to them, this lack of standardization is one of the main reasons SR applications are not yet in our daily lives. The goal is to design a swarm of robots of low-cost and easy to control their cooperation as to yield intelligent collective behaviors, there are attempts to solve the problems that are facing the standardization but they are ad-hoc. Each SR model has its own sensors, programming language, and actuators. This diversity complicates project migration across platforms due to hardware or software incompatibilities. The goal of unification has yet to be accomplished. However, there are some promising attempts like Villemure et al. [15] hardware/software open platform for SR development. The process of building SR is divided into two parts: the hardware design and the software design.



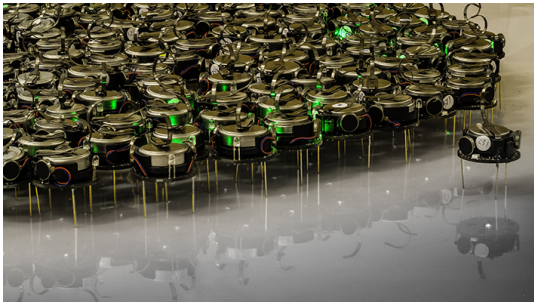


Figure 4. A swarm of robots [59]

3. Hardware Design

While the software is the intelligent core, providing a virtual environment that governs the robot’s operations, it is the hardware that executes the instructions dictated by the software, translating them into physical actions. The challenge with hardware is the significant variation required for different environments: in the air (aerial SR), on land (terrestrial SR), or underwater (aquatic SR). The hardware variations in swarm robots are also driven by the specific requirements of the task, as demonstrated by the experimental results of Salman et al [16]. For example, cameras and vision sensors are more compatible with navigation tasks, while infrared or thermal sensors are better suited for rescue operations, and chemical sensors are useful for environmental monitoring. Furthermore, building SR systems under economic constraints differs significantly from building them without such constraints, as hardware prices vary, and the hardware differs between homogeneous and heterogeneous SR systems [17]. As noted by Salman et al [16], the hardware specifications of each individual robot, the design of the control software orchestrating their behavior, and the optimal swarm size are heavily contingent upon the distinctive nature of the collective mission at hand, as well as the economic constraints imposed on the project. However, while "a universal swarm design methodology does not exist" [salman2019], a general hardware architecture is common across most swarm robots (see Figure 5).

3.1. Sensors

The sensory unit used by the SR is essential for perceiving "the surrounding environment to the controller—a process known as mapping" [17]. This process is vital for accomplishing tasks such as detecting and avoiding obstacles, detecting neighboring robots, and navigation. It is also used to collect sensory data that allows the SR to make decisions. Each SR system can be equipped with various sensors(see Figure 6), such as cameras, proximity sensors, range sensors, GPS, humidity sensors, temperature sensors, chemical sensors. (Sensor categories, their names, the robots equipped with them, power consumption, and sensor prices are listed in Table 1).

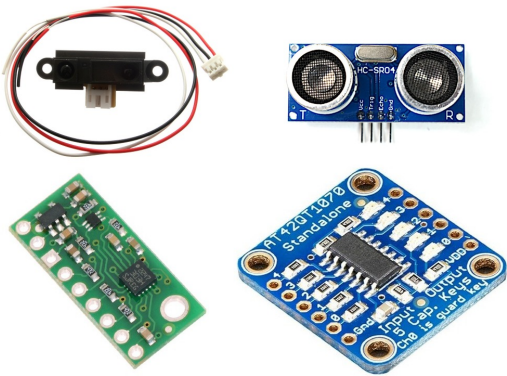


Figure 5. Sensors used in SR [60–63]

Table 1. Summary of Hardware Components Used in Swarm Robotics Platforms

Category	Component (Model/Type)	Function / Usage	Specs (Power, Range, Cost)	Platform(s) / Ref.
Sensors	GP2Y0A21YK IR Sensor	Obstacle avoidance, range detection	30–50 mA, 10–80 cm, \$10	S-Bot [29]
	HC-SR04 Ultrasonic Sensor	Distance measurement, anti-collision	15 mA, 2–400 cm, \$2–5	UBswarm, Mustafa SR [18,29]
	Realsense D400 Camera	Depth sensing, 3D mapping	1.6 W, USB-powered, \$150	SwarmUS [15]
	DS18B20 Temp Sensor	Water temperature sensing	1 mA, digital, \$5	Aquatic SR (Costa) [21]
	AT42QT1070 Touch Sensor	Capacitive human interaction	2 µA, \$1	Zooids [20]
Actuators	DC Gear Motors (e.g., Solarbotics)	Wheeled movement	35–150 mA, 3–12V DC	Colias, UBswarm, S-Bot [18,24,29]
	SG90 Servo Motors	Wheel drive (modified)	250 mA peak, 5V, \$3–5	HeRo [36]
	Piezoelectric Actuators	Micro-scale movement	mW range, low voltage	Kilobot (implied) [25]
	Omni-wheel Drive	Holonomic motion	Variable, platform-specific	Cellulo [19]
	Propeller Motors (NTM, EMAX)	Aquatic propulsion	140 W, 11.1V LiPo	Jeff, Costa SR [21, 25]
Communication Modules	ESP8266 Wi-Fi	Long-range comms, ROS link	70–170 mA, 20–50 m, \$2–10	HeRo, GRITSbot [28,36]
	PmodWiFi (SPI)	High-speed Wi-Fi data	250 mA, 400 m, \$20–30	UBswarm [? ]
	nRF24L01+ RF Chip	RF link with PC/server	15 mA, 800–1000 m, \$1–5	Zooids [20]
	RN-42 Bluetooth	App/tablet communication	15–50 mA, 10–15 m, \$10–30	Cellulo [19]
	IR Sensors (Long/Short Range)	Obstacle avoidance, basic comms	30 mA, 1–5 m, \$5–20	Colias, S-Bot [24], [29]
Power Sources	3.7V Li-Po Battery	Main control and drive power	600–1200 mAh, USB, \$5–15	Colias, Zooids [20, 24]
	11.1V Li-Po Pack	High power, aquatic missions	880 mAh ×8, 120 min	Jeff [25]
	Wireless Charging Dock	Automatic recharge	400 mAh, 40 min runtime	Robotarium (GRITSbot) [? ]
	Dual Battery (Motor + Logic)	Noise isolation, redundancy	4.2V Li-Ion ×2	Abuelhaija SR [38]



Figure 6. Locomotion mechanisms used in SR [64–67]

Variations in robotic sensors may result in robots perceiving information differently, which influences the control unit to generate different actuation commands for the actuators based on the sensory input. The infrared (IR) proximity sensor is the most commonly used sensor, as it is small, easy to mount, and can detect objects at short ranges of 5–15 cm, depending on the object’s color [19]. Nedjah and Junior note that IR sensors are present in most robots [14]. These sensors work by emitting infrared light, which is reflected off objects—closer objects cause stronger reflection intensity.(The mathematics behind the work of IR sensors have been illustrated in equation (1) represents the sensor response function which is a response an IR sensor. While the equation (2) represents the cumulative sensor response and the equation (3) represents the sensor model in Colias SR).

Reviewing SR hardware papers from 2014 to 2024, we observe different sensors implemented in SR projects. In the cellular project, the robots were equipped with a downward-facing camera [19].



The UBswarm, developed at the RISC lab at the University of Bridgeport, uses ultrasonic as well as photoelectric (infrared) proximity sensors [18]. In the Zooids project, a flexible electrode is wrapped and embedded within the 3D-printed housing, facilitating capacitive touch sensing. An integrated capacitive touch sensing circuit (Atmel AT42QT1070) is included to detect user touch [20].

Aquatic SR systems, as described by Costa et al. [21], incorporate sensors into every robot, including temperature sensor, digital compass, and a GPS receiver. The GPS receiver, a GlobalTop FGPMOPA6H module, provides position updates with 5 Hz frequency and interfaces with the single-board computer via UART protocol. It is equipped with a 26 dB gain GPS antenna for more powerful signal quality and position accuracy of  $\pm 3$  m. The digital compass unit, an STMicroelectronics LSM303D magnetometer, provides information by compensating based on the robot's pose. Due to interference, the magnetometer is placed in a secondary enclosure in the vessel's prow. Temperature data are obtained from the onboard SBC temperature sensor and a Maxim DS18B20 sensor, positioned in the vessel's bottom, to measure water temperature. The DS18B20 sensor provides digital temperature readings with 12-bit resolution and interfaces with the SBC via the One-Wire standard protocol.

In Bartmess et al's [22] project to build fast, low-cost swarm robots, a webcam is positioned above a table to observe the tabletop surface and the robots on it. The webcam connects to the computer through a USB interface and is an off-the-shelf component that adheres to the cost constraints set for the vision system. The webcam derives its power supply from the USB connection. In Shang's PhD thesis [23], the robot is equipped with two infrared photoelectric sensors. The responses of each individual reflective point within the sensor's viewing range are added up to determine the IR sensor's output magnitude:

In the Colias SR [24], only IR proximity sensors are used. The following formula serves as a mathematical model for the reflected infrared value that a sensor measures :

$$s(x, \theta) = \frac{\alpha_c \cos \theta}{x^2} + \beta_c \quad (1)$$

$s(x, \theta)$  sensor output signal.

$x$  is the distance of the obstacle.

$\theta$  is the angle of incidence with the surface.

$\alpha_c$  includes several parameters, such as the reflectivity coefficient, output power of emitted IR, and sensor sensitivity.

$\beta_c$  is the amplifier's offset value plus the effect of ambient light.

Another aquatic SR project, "Jeff," by Mintchev et al [25], involves an autonomous underwater vehicle (AUV) equipped with several sensors and communication systems. Underwater navigation in this system relies on a pressure sensor and a gyroscope to measure depth. An accelerometer and a magnetometer are used to control orientation and locomotion, and additional sensor payloads (e.g., temperature sensors, chemical concentration sensors, and cameras) can be integrated into Jeff's modular shell. The SwarmUS project is equipped with an A2M8 RPLidar and a D400 series Realsense camera [15].

In their project, Mustafa et al [26] utilize the HC-SR04 ultrasound sensor from ElecFreaks. Operating at 5V DC, the sensor uses sonar to measure distances to objects. Each module consists of an ultrasonic transmitter, receiver, and control circuit. The sensor emits ultrasound pulses at a constant frequency of 40 kHz with 8 cycles per burst and captures echoes lasting milliseconds. With an accuracy of about 3 mm and a pulse travel range of 2–500 cm, the sensor is used for collision avoidance with both static and dynamic objects, including other robots in the swarm and the physical boundaries of the 3D environment.

The 25g pico quadrotor with a protective cage is equipped with a wireless camera installed on the bottom, capable of providing real-time video feedback [27]. Another SR project has been equipped with additional introspective sensors (specifically motor current and temperature sensors) [28]. The S-Bot

robot uses a variety of sensors for different purposes, including the Sharp distance sensor 2Y0A21, the ultrasonic distance sensor HCSR04, and a general-purpose proximity sensor that uses an infrared emitter and receiver pair. The Sharp GP2Y0A21YK0F measures distances between 10 and 80 cm, with its output voltage matching the detection distance. In contrast, the HCSR04 has a range of 2 cm to 400 cm with an accuracy of 3 mm. Through the use of an onboard potentiometer, the general-purpose proximity sensor's operating range can be adjusted from 2 to 15 cm. Noted for its compact design and low power consumption [29,30], the Bulubot prototypes also utilize two types of sensors. Firstly, the Sharp GP2Y0A41SK sensor, To detect obstacles, a short-range infrared proximity sensor is employed. Second, for light following, a 10 mm light-dependent resistor (LDR) is used [31]. Bump sensors, mounted upward allow the direction of impact to be detected on the bottom circuit board, with an average error of  $8.1^\circ$ . These bump sensors are used for obstacle avoidance and to estimate the angle of impact [32]. The mROBerTO platform features both a proximity-sensing module and a swarm-sensing module [33]. Their research explored SR systems for material handling tasks and enhancing solar energy conservation efforts. In Kumar et al.'s work [34], the SR is equipped with IR proximity sensors, temperature sensors, humidity sensors, and LDRs.

The RiBot project places an IR sensor (TSOP75436WTT) on the back of the robot [35]. The HeRo robot [36] avoids obstacles and collisions with other robots by using only infrared proximity sensors. The three TCRT5000 long-range sensors that make up the IR sensory system are positioned in front of the robot and have a 20 cm range. In the simulation conducted by Shang et al. [37], two downward-pointing IR sensors are located at the front of the robot. For another project [38], Infrared sensors and switches were combined. An Analog to Digital Converter (ADC) module on the ATtiny85 microcontroller detects and measures voltage variations across specific I/O pins. This capability is used to determine the robot's distance and read the voltage from an infrared receiver.

### 3.2. Actuators and Locomotion Mechanisms

Just as muscles facilitate movement and coordination in humans, actuators enable the movement and operation of the robot's joints and linkages (see Figure 7), responding to control signals received from the controller. Actuators help the robot withstand gravity, inertia, and other external forces during operation [39]. In Mustafa et al's work [26], Wheels and servomotors are the two primary parts of their SR platform. Two side wheels that are driven by servomotors that rotate continuously provide the robot with movement. Servomotors, in contrast to conventional motors, can be individually controlled and only need the rotational angle to move. An omnidirectional ball caster wheel supports the robot's rotation, enabling it to swivel in any direction.



**Figure 7.** Communication and networking hardware used in SR [68–71].

The SwarmUS platform [15] focuses on the coordination, communication, and localization elements required for swarm behavior rather than directly addressing actuators. Wheels are the Pioneer 2DX with SwarmUS's primary feature. Conversely, "Jeff," the aquatic SR [25], is adaptable for underwater swarm operations because it uses a combination of DC motors and specially made mechanisms to control movement. One DC motor regulates the cam and piston mechanism in Jeff's buoyancy system, allowing for depth control and up/down movement. In order to facilitate attraction and repulsion during docking procedures, the docking system also uses a DC motor to control the orientation of a magnet inside the docking station.

Other aquatic SR actuators are represented in the work of Costa et al [21], where two DC motors each drive a propeller through a shaft. Two particular models are mentioned in the paper: the Emax 2215/25 950 kv 2-3S and the NTM Prop Drive Series 28-30A 750 kv/140 W. Furthermore, the HobbyKing 50 A Boat ESC 4 A UBEC and other electronic speed controllers (ESCs) regulate the direction and speed of the DC motors, enabling accurate robot movement. The Colias robot [24] is driven by two tiny DC motors that use direct gears and two 2.2 cm-diameter wheels, allowing it to reach a top speed of 35 cm/s. Using distinct H-bridge controllers and pulse-width modulation (PWM) techniques, each motor's rotational velocity is individually controlled.

DC motor drivers power each motor, with an average power consumption of  $35 \pm 5$  mA under no-load conditions and up to  $150 \pm 20$  mA when stalled. In their work on building fast low-cost SR, Bartmess et al. [22] used two DC motors per robot for propelling the robot around a table. These motors are connected to the wheels, allowing differential drive by varying the speed of each wheel. The design includes motor controller ICs that connect to the DC motors. These ICs receive PWM signals from the SoC and change them into signals that can drive the motors, which lets you control speed and direction.

In the Zooids project [20], the actuators are micro DC motors with wheels, which allow the robots to move and navigate on flat surfaces, supporting dynamic and interactive swarm user interfaces. The UB Robot Swarm [18] uses DC motors for driving the robot's wheels, with various types of motors such as Solarbotics gear motors, Micro-metal gear motors, and Tamiya gearbox motors. For manipulating things with a robot arm, geared DC motors are also used. Hitec HS-422 servo motors move the arm and gripper, giving the robots precise control over movement and enabling them to grasp objects.

The Cellulo robots [19] are equipped with omnidirectional ball drive actuators for locomotion, enabling holonomic movement, meaning they can move in any direction and change direction instantaneously. Patil et al. [17] reviewed actuators used in various swarm robot platforms and concluded that DC motors are widely used for driving Wheels and tracks on platforms like E-puck, Alice, Sumobot, Swarm-Bot, AutoBot, and CYBOTS. Some platforms, like E-puck and Nanokhod, use stepper motors for precise control. Servo motors, on the other hand, allow for positional and motion control for locomotion and other tasks. Microrobots use piezoelectric actuators because they are small and accurate. Robots like TerminatorBot use gear motors to move things more powerfully. In his SR design, Shang [23] used DC motors and two wheels. The pico quadrotor [27] uses four DC brushed motors as its main actuators in aerial SR. These motors create the thrust and torque needed to control movement and orientation. The Robotarium's GRITSbots [28] use stepper motors as their main actuators. This lets them move with great accuracy and run a variety of swarm robotics algorithms correctly. The S-Bot robot [29] has a caster wheel on the front for support and two DC geared motors to control its movement. The drive motors let the robot turn left or right and move forward and backward. Distance sensors help the motor system work better. In his master's thesis, Demir [31] used four DC motors, each operating at 60 rpm, alongside two TB6612FNG motor drivers to drive the robot's legs.

The r-one robot [32] employs DC gearmotors to drive the robot's wheels, providing locomotion and maneuverability. A 100:1 gearbox coupled with the motors allows for precise speed control and high torque output. The robot's gripper attachment uses an S-75 sub-micro servo motor to control the movement of the gripper paddles, enabling it to grasp and release objects with varying force.

In mROBerTO [33], two 4mm nano coreless DC motors are used in a differential drive configuration, eliminating the need for additional gearing, simplifying the design and reducing size. Vibration motors, though not used in mROBerTO, are mentioned as an alternative locomotion method used in Kilobot. However, Although vibration motors are not used in mROBerTO, they are employed in Kilobot as an alternative locomotion method. However, they offer lower precision and are less effective for long-distance movement compared to DC motor-based systems.

Kumar et al [34] worked on SR for solar energy conservation, where they employed DC motors to drive the robots' wheels, enabling movement. They also used L293D motor driver ICs to control the speed and direction of the DC motors. RiBot [35] is equipped with a micro step gear motor ("MF03G" by Seiko Precision Inc.) for actuating the robot's tail (caudal peduncle), allowing it to mimic the tail movements of a real zebrafish. HeRo [36] uses two SG90 micro servo motors modified for continuous rotation to drive its wheels and control its movement.

The final SR project introduced is by Abuelhaija et al. [38], who used two DC motors per robot. These motors facilitate movement and they are controlled by a DRV8833 dual H-bridge motor driver chip. This setup allows for bidirectional control of the motors, enabling the robots to move forward, backward, and turn.

### 3.3. Communication And Networking

"In multi-robot systems, communication media are used for inter-robot communication to share information and make collective decisions." [29] Most often, wireless communication is employed when a scenario involving mobile robots is completed. [24] The choice of communication module depends on several factors [18], including coverage, price, and power consumption.

- Communication Distance Capability: The X-Bee modules have more extensive communication range in comparison to the Bluetooth Bee modules.
- Data Transmission Speed: The PmodWiFi module, when mixed with the SPI interface, facilitates a higher rate of data transfer comparing to the X-Bee and Bluetooth Bee modules.
- Energy Consumption: Each module exhibits distinct power requirements.

Moustafa et al [26] in their hardware robotic platform For very short distances (10 m or less), they use an integrated communication interface for Bluetooth 4.1, and for larger areas (100 m), they use an 802.11n wireless LAN, and an FM receiver operating in the 65-108 MHz FM bands. These wireless capabilities are enabled by the Cypress CYW43438 wireless chip. The various wireless communication options are utilized for facilitates short and long-range communication for the swarm robot system. SwarmUS has been equipped with communication system utilizes a Wi-Fi network for data exchange, with broadcast messages for updating shared information and unicast messages for sending commands. It employs a messaging system based on Protobuf and RPC mechanisms to route communications between agents, hosts, and other system components [15]. moving to the aquatic SR Jeff [25] is equipped with Blue LED units used for communication and distance sensing, each unit has two pairs of LEDs, One pair with 1m range, 60° beam for communication, Another pair with 0.5m range, 120° beam for obstacle detection. 14 electrodes inspired by electrosensory capabilities of fish Enables communication and localization through electric field detection/interpretation with the Range of 250-500mm, more then that it is also have Two loudspeakers, one on each side with the Range of 0.5-1m, and finally a Microphone that Receives acoustic signals from a floating station Used for a virtual fence system to confine the swarm within a designated area. Colias [24] uses infrared (IR) technology for both communication and sensing, it utilizes short-range IR bump sensors for basic obstacle avoidance, while the long-range IR proximity sensors provide both environmental sensing capabilities (obstacle detection and range estimation) as well as a means for direct communication between robots in the swarm. Back to another aquatic SR developed by costa et al [21] is equipped with A TP-Link TL-WN722N Wi-Fi adapter with a high-gain antenna allows robots to communicate wirelessly with each other over a range of 40 meters on the water. The Fast, Low cost 16 SR developed by bartmess et al [22] is equipped with a Wi-Fi router capable of transmitting a 2.4GHz signal to the robots. The router will be communicating to the robots

Utilizing a UDP connection, the system enables rapid and frequent communication. It establishes a connection with the computer through an Ethernet cable, facilitating simultaneous connections with up to 16 devices. Remarkably, it can transmit 16 packets, each with a size of 16 kilobytes, within a mere 100 milliseconds. These capabilities are attributed to the Zooids component. zooids SR by M. Le Goc et al [20] are equipped with two main elements of communication and networking hardware: 2.4GHz Radio Chip (Nordic nRF24L01+) Enables wireless centralized communication between each Zooid and a master computer, also it Allows sending back information status to the master. The UB Robot Swarm[18] uses several communication modules depending on the specific robot and its role within the swarm: X-Bee modules: They offer wireless communication for both indoor and outdoor environments. They operate using serial communication (Tx/Rx) which are compatible with other communication modules.

Bluetooth Bee Modules: Like X-Bee, they use serial communication for short-range communication within the swarm.

PmodWiFi modules: provides wireless communication through WiFi connectivity. It uses the SPI mode to transmit data and receive it, offering faster data rates compared to serial communication.

Cellulo SR project [19] use wireless RN-42 Bluetooth for communication. Patil et al in their review on SR hardware [17] they mentions various communication and networking options used in swarm robotics, each with its own advantages and drawbacks "Short-Range Communication" like Infrared (IR) transceivers/sensors it is Low cost, simple implementation, excellent for short-range robot-to-robot communication and obstacle detection, but its range is limited, susceptible to interference from ambient light, requires line-of-sight. Also Ultrasonic sensors which are Longer range than IR, can measure distance and angle, but it is Sensitive to object material and surface properties, less accurate than IR. For the Long-Range Communication another hardwares mentioned in the review like the Radio Frequency (RF) modules which is Longer range than IR or ultrasonic, good for complex environments, but the Potential interference from other RF devices, can be more expensive. Another one is bluetooth it is Relatively low-cost, readily available modules, good for moderate data transfer but is short in its range then the (RF), and finally Wireless LAN (Wi-Fi) which is High bandwidth, good for large data transfers but it is Higher power consumption, potential interference issues, more expensive. in his thesis about Hardware Variation in Robotic Swarm [23], shang categorizes the communication methods into three main types and provides examples of each technology category:

#### 1. Interaction via Communication:

- Short-range wireless transmission: This often involves the use of radio frequency (RF) modules like Zigbee or Bluetooth.
- Infrared broadcasting communication: IR transceivers enable robots to exchange information using infrared light.
- LEDs for information indication: Robots can use LEDs on board to let other robots know what state they're in or other information.

#### 2. Interaction via Sensing:

- Sonar: Ultrasonic sensors help robots to detect the distance and direction of nearby robots without direct communication.

#### 3. Interaction via the Environment:

- Virtual pheromones: This method uses the environment to "remember" things, just like ants do with pheromones. The actual implementation can be different; for example, robots can use wireless communication to keep track of virtual pheromone information.



The 25g swarm pico quadrotors [27] primarily use ZigBee communication modules for wireless communication. The Robotarium [28] primarily relies on WiFi (IEEE 802.11 B/G/N) for communication between the robots and the central server. Each GRITSBot is equipped with an ESP8266 chip that provides WiFi capabilities with a bandwidth of up to 54 MBit/s. The S-bot [28] As demonstrated, the CC2500 Serial Communication Module is a low-cost 2.4 GHz transceiver intended for extremely low-power wireless applications. The module is intended to operate in the ISM (Industrial, Scientific, and Medical) and SRD (Short Range Device) frequency bands, which are 2400–2483.5 MHz. The CC2500 features a standard configuration baud rate of 9600, a variable baud rate RS232 UART interface, and a programmable device address (255 per channel). In Demir's master thesis [31] the focus of his work is on the mechanical design, leg optimization, and individual robot control for achieving flocking behavior, so he doesn't mention any communication hardware. In the r-one SR [32], 8 infrared transmitters, 8 infrared receivers, a 2.4 GHz radio with a 2 Mbps data rate, and a USB port are used for communication. In order to communicate with the user, The robot has three pushbuttons, three distinct arrays, and a VLSI1053 audio chip with MIDI playback for user interaction., each consisting of five LEDs emitting light. Another SR is mROBerTO [33], it is equipped with two types of communication hardware wireless RF "ANT<sup>TM</sup>" for low-power mesh networking and "BLE" for higher bandwidth and an infrared multi-channel system for local robot-to-robot communication and relative positioning. This suite of hardware enables effective communication, networking and localization capabilities for the SR system. In their work kumar et al [34] primarily focuses on Bluetooth modules for communication between the swarm robots and potentially with a computer Each robot is equipped with a Bluetooth module, Utilizes serial communication mode (Tx and Rx) for data exchange, Enables decentralized communication among the robots. While HeRo [36] is equipped with a NodeMCU v3 board which has a built-in ESP8266 microprocessor that provides WiFi communication capabilities. This allows the robot to connect to a network and communicate with other devices, including a central computer running the Robot Operating System (ROS). In Abuelhaija et al's [38] work the communication of SR in this project rely on an indirect form of communication through infrared (IR) sensors, By measuring the strength of the received IR signal, a robot can estimate the distance to other robots within its line of sight. This information is then used by the control algorithm to guide the robot's movement and achieve swarming behavior.

### 3.4. Power Source

Power sources to a robot are like the engine to the car [40], so it is important to equip the SR with the appropriate power source depending on factors like the size of the robot its mission. moustafa et al [26] A rechargeable, power-efficient 3.7 V lithium-ion polymer battery was used in their SR platform. A boost converter is part of the system to raise the battery voltage to the necessary level, and a specialized lithium battery charger module that provides charging/discharging management, temperature control, and various protection features to ensure safe and reliable operation of the battery power source. SwarmUS [15] equipped with 11.1 V LiPo battery A DC/DC converter is used to step down the robot's battery voltage to power the SwarmUS boards, and the Hiveboard distributes power to the connected Beeboards. about Jeff the aquatic SR is equipped with an 8 Li-Po cells, each with an 880 mAh capacity. A lithium battery pack stores energy for up to 120 minutes of autonomy. located in Jeff's stern. it have also a Battery Status Monitoring Circuits Dedicated circuits monitor the battery's state of charge and remaining capacity. This information is crucial for Jeff's cognitive capabilities, allowing it to make informed decisions about energy usage and potentially seek recharging when needed. Another SR is Colias [24], utilizes A 3.7V, 600 mAh lithium-polymer battery serves as the primary power source. This battery is expandable up to 1200 mAh for increased autonomy, The lower board of the robot houses a dedicated power management system. This system monitors and controls the power consumption of various robot functions. The aquatic SR developed by costa et al [21] utilizes two LiPo batteries, one dedicated to powering the motors and propulsion, and the other for control, processing, and sensing components. The control battery is regulated by an SBEC to provide a stable 5V DC supply. For the Fast, Low-cost 16 robots developed by Bartmess [22] they use Lithium-Ion Battery (500mAh), a circuit



manages the charging process of the Li-Ion battery, preventing overcharging and undercharging to ensure safety and battery longevity, a Voltage Regulator (LDL1117S33R): This component regulates the voltage from the battery to a steady 3.3V, which is required for powering the ESP8285 SoC and other components. Each zooids robot SR [20] is powered by a 100 mAh LiPo battery. The robots' motors, radio module, microcontroller, and LED use the majority of their power. the zooids are able to move for an hour and continue to function for even longer when used normally. UB Robot Swarm [20] are powered by NiMH or LiPo batteries, chosen for their size, weight, and power characteristics. Power distribution and management involve considering the current consumption of individual components, as well as environmental factors and operational patterns. the paper of the cellulo project [19] does not mentioned the details of the power supply of their SR but it is a rechargeable battery because cellulo have a USB port for recharging it. In the Review paper about SR hardware[17] the authors mention several power supply like the Rechargeable lithium batteries which are the most prevalent choice due to high energy density, compact size, and light weight specially Lithium-polymer (Li-Po) batteries which are favored for safety and thin profile, SR typically operate on a voltage range of 5V to 25V DC power. also they mention some factors that influencing battery choice:

- Robot size and weight: Smaller robots require smaller, lighter batteries.
- Power consumption: Robots with more sensors, actuators, and processing power need higher-capacity batteries.
- Mission duration: Longer missions necessitate batteries with longer run times.

The energy supply hardware used in the pico quadrotor described in the paper[27] is a 3.7V, 340mAh Lithium Polymer (LiPo) battery. the robotarium (GRITSbots) [28] uses 400 mAh LiPo battery: This is the onboard energy storage for each robot, allowing them to operate for up to 40 minutes, and a Wireless charging system. S-Bot [29] has 12 Volts rechargeable battery to power the whole system. charging using external chargeur. The r-one robot [32] is equipped with the following energy supply hardware: 3.7V 2000mAh Lithium Polymer (LiPo) Battery, The robot can be charged via the USB port, An additional charging method is available through a docking connector. The mROBerTO [33], utilizes the following energy supply hardware: Three 3.7V Li-Po batteries connected in parallel Voltage divider and ADC port for battery monitoring. They conducted a survey on swarm robotics material handling and how it can be used to conserve solar energy, Kumar et al [34] pointed to Li-Po batteries as the primary power source, supplemented by miniature solar panels that harvest solar energy to recharge the batteries and extend their operational time. Additional hardware components like solar charge controllers and power meters are employed to manage and optimize energy usage and storage. The RiBot [35] uses a small 40mAh rechargeable LiPo battery as its main power source. The battery can be conveniently recharged through contacts in the robot's "eyes" without needing to disassemble the device. With continuous tail movement, the battery provides around 23 minutes of runtime, while intermittent tail usage extends the operating time beyond 1 hour. The HeRo SR, is equipped with 3.7V 1000mAh Li-Po battery and 5V step-up boost converter this converter efficiently steps up the voltage from the 3.7V battery to the required 5V level. Finally the paper of abuelhaija et al [38], The paper describes their SR as being powered by Two rechargeable 4.2V lithium-ion batteries, One battery is dedicated to powering the motor circuit, while the other supplies energy to the control circuit, This separation is likely done to isolate potential electrical noise from the motors and ensure stable power delivery to the sensitive control electronics.

#### 4. Software Design

A "software is the product that software professionals build and then support over the long term. It encompasses programs that execute within a computer of any size and architecture, content that

is presented as the computer programs execute, and descriptive information in both hard copy and virtual forms that encompass virtually any electronic media" [41]. The aim of any software developed for SR is to achieve swarm behavior and enable this emergent property to emerge from simple local rules. As for the hardware, Nadjah and Junior [14] introduce the SR software as a field rich with potential but fragmented, needing greater standardization and direction to transition successfully to real-world applications. Establishing common foundations and embracing techniques like automatic design are highlighted as important steps forward.

At the heart of swarm robotics lies the software that enables and governs the collective intelligence and coordination among individual robots (General software structure of swarm robotics). The software components of swarm robotic systems are responsible for tasks, and the implementation of swarm algorithms that dictate the emergent behaviors of the swarm.

To synthesize current knowledge and practices in swarm robotics software development, Table 2 represents a comparative summary between the SR platforms reviewed by this paper. The first paper to be reviewed [15] by É. Villemure et al, offers a unique approach to swarm robotics development by providing a software and hardware framework that can be integrated with existing robots. This review specifically focuses on the software aspects of SwarmUS, it is composed of different components citing: HiveMind (core firmware running on Hiveboard), HiveConnect (handles Wi-Fi networking on ESP32), HiveMindBridge (C++ library for host robot communication), Buzz Programming Language (for developing swarm behaviors), HiveAR (Android app for human-swarm interaction), also swarmUS Buzz integration for efficient swarm behavior development. Abstraction and modularity for flexible robot integration by the separation between the swarm intelligence (HiveMind) and robot functionalities (HiveMindBridge), it has a Simulation and real-world deployment capabilities. It's The ability to cross-compile Buzz scripts into ROS nodes and utilize the Gazebo simulator facilitates testing and development before the deployment on physical robots. moving to Costa et al [21] in their project to Design and Development of an Inexpensive Aquatic Swarm Robotics System, The paper details a well-structured and comprehensive software architecture for controlling and managing a swarm of aquatic robots, the software has a (Raspberry Controller) Runs on each robot using a Raspberry Pi and Raspbian OS, it uses effectively open-source libraries (Pi4J, WiringPi) for hardware interaction, promoting code reusability and community involvement, also it comes with User-friendly interface. It Enables real-time control and deployment of waypoints, geo-fences, and obstacle information, and Logs commands and messages for offline analysis and debugging. For the communication The system utilizes a Wi-Fi network for inter-robot and robot-base station communication. Bartmess et al [22] develop a Fast, Low SR, it comes with a Modular design with separate vision processing and robot control modules promotes clarity and maintainability, it uses an ORB algorithm and vision targets for localization is a well-established approach, and a UDP communication protocol. Finally the paper demonstrates a solid foundation for a low-cost SR system, Addressing the potential concerns and exploring additional features can further enhance the system's robustness, flexibility, and scalability. Regarding Zooids [20] The paper highlights the software architecture enabling complex swarm behaviors and user interaction with the robots. It introduces a layered architecture : an Application Layer (Defines desired swarm behaviors/goals), Simulation Layer, Server Layer (Dispatches commands), Hardware Layer. For the control the zooids use strategies like Hungarian Algorithm for efficient swarm reconfiguration, HRVO for real-time collision avoidance, PID control for precise individual robot positioning. It is also Addressing limitations and further development could unlock the full potential of such swarm systems. The Cellulo paper [19] presents an innovative and distinctive methodology for educational robotics, centering around the utilization of a swarm comprising numerous compact robotic units, affordable, and haptic-enabled robots. The software design plays a crucial role in achieving the platform's goals, the software design effectively supports the platform's goals of versatility, practicality and ubiquity in educational robotics. While some limitations exist, the flexible and scalable architecture shows promise for further development and exploration of swarm robotics in education. The architecture is decentralized, it comes with several software components like Vision-Based, haptics controller, Bluetooth 2.1

and a tablet with user interface. the Robotarium [28] software demonstrates a well-designed approach to enabling safe(using the Safety Measures through simulation-based verification and safety barrier certificates), with a focus on simulation-based verification, safety barriers, and user-friendly interfaces. The software's emphasis on safety and accessibility makes it a valuable platform for advancing the field. mROBerTO [33] is another SR platform, regarding the software mROBerTO it comes with Onboard ARM processor allows for complex swarm algorithms, communication via ANT and Bluetooth Low Energy (BLE), Programmed in C++ using open-source SDK for more control. Next is HeRo [36] an SR platform that comes with a ROS integration that Offers modularity and reusability by leveraging existing ROS packages, it Provides standardized communication between robot and external computers, also it uses Simple Arduino firmware focused on essentials like sensors, motors, ROS communication. In Abuelhaja et al's paper [36] The robots are programmed in AVR assembly using Atmel Studio, The code is organized into subroutines. Each subroutine is likely responsible for a specific task like motor control, sensor reading, or implementing a part of the AI algorithm. The core AI algorithm revolves around interpreting sensor readings and translating them into robot movement. The paper presents a second-order polynomial equation used to determine the relationship between IR sensor readings (representing distance to other robots) and the duration of forward movement. Salman et al [16] offers insightful information about the automatic creation of robot swarm control software, focusing on economic limitations, they Introduce the "Waffle" platform within the AutoMoDe framework, emphasizing the assembly and fine-tuning of pre-defined modules for mission-independent control software. Utilizes Probabilistic Finite State Machines (PFSMs) to generate control software, dictating robot actions based on sensor input and internal parameters Offers flexibility and reduced human intervention, it is allowing for mission-specific selection and combination of modules, Optimizes parameters within each module automatically to enhance performance for the chosen mission and all of that can be done under the economical constraint based on mission requirements, those economical constraints significantly shape software design and resulting behaviors. moving to Rubenstein et al paper [42] that introduces some algorithms to realise the self assembly behavior in a thousand SR called kilobots, all Kilobots run an identical program containing the self-assembly algorithm and target shape image, highlighting the decentralized nature of the system with no central controller. Edge-following, gradient formation, and localization through neighbor trilateration are the three basic collective behaviors they employ; These primitive behaviors are combined into a finite-state automaton that dictates each robot's sequence of actions based on its current state and sensor data, its an innovative algorithmic design that enables complex collective self-assembly to result from interacting with numerous basic robots with restricted capabilities, however there is a lack of details about the used software. Next is Franceska et al paper [43] in which they introduce an automatic method for designing control software for swarm robots which is the AutoMoDe-Chocolate, the paper compares Chocolate with other automatic and manual design methods: beginning with the Reference Model that defines the e-puck robot's capabilities, formalizes sensor inputs and actuator outputs, and provides a consistent comparison platform. Then the Automatic Design Methods like Vanilla that Uses pre-existing modules to create a probabilistic finite state machine with F-Race optimization, EvoStick that evolves a feedforward neural network using evolutionary algorithms and Chocolate that enhances Vanilla with the Iterated F-Race algorithm for better optimization. The Manual Design Methods have also been mentioned like U-Human (Unrestricted design by human experts) and C-Human (Constrained design using the same modules as Vanilla and Chocolate). In Hasselmann's phd thesis [44] about "the automatic modular design of control software for robot swarms" Hasselmann makes a good source of knowledge about the SR software, he focuses on higher-level design methodologies, architectures, and bridging simulation-reality gaps, introducing the direct neuroevolution and modular neural network behaviors as a Control Software Architecture and mentioning also Behavior trees as an alternative. back to the mentioned standardization problem SwarmTalk [45] represents an important step towards standardized cross-platform communication and benchmarking for swarm robotics. It focuses on efficiency, portability and ease of use are valuable for resource-constrained swarms, the

paper Addresses critical need for standardized swarm robot communication, it comes with a Portable design with simple driver interface for easy cross-platform use, it has Minimal resource requirements for memory constrained platforms. next is Buzz [46] is a programming language oriented to SR design, Buzz introduces important innovations like the swarm construct for more nuanced, dynamic swarm behaviors. The extensible design, situated communication, and stigmergy mechanisms provide powerful tools for heterogeneous swarm programming. Francesca and Birattari [47] in their paper highlights the diversity of software approaches used, with a shift towards modularity emphasizing the need for more structured research, comparisons, and benchmarking in the domain of automatically designing control software for swarm robotics, especially leveraging evolutionary techniques. Establishing standards would enable objective assessment of different software designs and optimization methods, they mentioned the Monolithic Neural Networks, Probabilistic Finite State Machines, Parametric Control Architectures, Modular Architectures. More than that they talked about the Off-line methods using simulations and evolutionary algorithms, and the On-line methods for robots to adapt software while operating, like embodied evolution. Toolkits are so useful and make the work much easier for the developers, that is what de andrade et al [48] offers by Pyswarming which is a valuable contribution enabling swarm robotics research by prioritizing behavior implementation in a cross-platform, Python-based package. Its strengths include accessibility, existing algorithm library, simple simulation, and customizability. The next paper [49] presents a layered software architecture for autonomous UAV swarms designed for firefighting with an architecture built on ROS. This software architecture boasts a modular design, with a layered approach (four layers) that enables independent development and testing, enhancing scalability and future improvements. It has good path planning 14 that uses a PRM-based global path planner to make the best paths for each UAV, taking into account the types of formations. Line-of-sight smoothing makes paths even better, which cuts down on mission time. Centralized and decentralized methods both work to avoid collisions, which keeps navigation safe by managing inter-UAV. Deep Reinforcement Learning (DRL) gives UAVs the ability to act on their own, letting them make smart choices about how to avoid obstacles and optimize their paths in new places. Toolkits are very useful to developers in all CS branches for facilitating, in the case of SR Akkaya et al [50] present PILOT, a software toolkit designed for creating data-intensive distributed applications in robotic swarm scenarios, PILOT presents an innovative actor-oriented paradigm well-suited for modular swarm programming. It is explicit state-space modeling, streaming data handling, and machine learning integration are strengths. Nevertheless, there is a lack of presented information about algorithm library, implementation specifics, scalability evaluation, and scope of learning integration to fully assess its effectiveness as a comprehensive swarm software toolkit. Swarmie project [51] is a swarm robotics platform developed at NASA Kennedy Space Center, it comes with Strong choice of ROS framework providing modularity, inter-process communication, standard message types, UI tools, and simulation capabilities; it promotes code reuse and collaboration. This approach aligns with the sound design principle of separating functionality into distinct modular nodes. In terms of designing SR software Brambilla et al [52] presents a novel approach to designing swarm robotics software called "Property-Driven Design". Instead of the traditional "code-and-fix" method, it advocates a top-down approach focusing on desired properties and formal verification through model checking, The property-driven design approach presents a promising formal methods-based, in terms of Formal Specification & Verification The use of PCTL and Markov chains allows for formally specifying desired swarm behaviors and verifying them through model checking. This helps ensure the software achieves intended goals, reducing the risk of unexpected outcomes. Property-driven design prioritizes design over implementation details. By promoting the idea of prescriptive model that acts as a blueprint. Gansari and Buiu [53] present a novel approach to integrating heterogeneous swarms of robots through a ROS-based software framework. The proposed system effectively addresses the challenge of integrating robots with different hardware and software characteristics. This is crucial for real-world applications where diverse robot capabilities are needed. The system integrate ROS as the backbone provides several benefits like Modularity(by using ROS nodes), Scalability (ROS's network



capabilities making the system scalable to larger swarms), Flexibility(The five working modes) and finally it is open source. Chamanbaz et al [54] describe a hardware/software suite to enable swarming behavior in a variety of robots. For The software, a Python module called "marabunta," exhibits several noteworthy features:

- Modular design: separating robot control "body", communication "network", and swarming behavior "behavior" into distinct classes.
- Platform agnostic : different robots can be integrated by creating platform-specific body classes (eBot and e-puck examples given).
- Communication flexibility and Enables heterogeneity.
- Provides "MockBody" and "MockNetwork" classes for simulation and rapid prototyping of swarm algorithms without real hardware. And it is Python-based for ease of development.

Table 2. Analytical Comparison of Swarm Robotics Software Platforms and Architectures

Platform / Architecture	Key Strengths	Limitations	Best Use Case	Reference
SwarmUS	ROS + Buzz integration, modular, real-to-simulation continuity	Requires experience with multiple tools	Advanced real-world hybrid deployments	[15]
Aquatic System	Real-time Java control, low-cost Raspberry Pi, user-friendly interface	Limited modularity and scalability	Low-cost aquatic swarms with basic missions	[21]
Zooids	Human-swarm interaction, high-frequency coordination, layered architecture	Limited to tabletop, not scalable for field robots	UI/HCI and real-time swarm interaction studies	[20]
Cellulo	Haptic feedback, education-focused, decentralized design	Limited in complexity and application domains	Educational swarm robotics for K–12 or university labs	[19]
Robotarium	Safety mechanisms, remote testing, real-to-sim pipeline	Limited hardware control; depends on centralized server	Cloud-based experiments, large-scale coordination testing	[28]
HeRo	Simple Arduino+ROS setup, cost-effective, modular firmware	Minimal swarm-specific middleware	Low-cost teaching and entry-level experimentation	[36]
Waffle (AutoMoDe)	Modular, automatically optimized control under constraints	Not flexible beyond mission-specific designs	Automated, constraint-aware controller generation	[16]
Kilobots	Extremely scalable, low-cost, decentralized	Lacks computation and sensor diversity	Self-organization studies and large-scale swarm behavior	[42]
AutoMoDe-Chocolate	PFSM + optimization with structured modules	Limited expressiveness for novel tasks	Comparative benchmarking of automatic vs manual design	[10]
AutoMoDe-Vanilla	Simple PFSM composition; human-constrained design space	Less optimal than Chocolate or Maple	Baseline for automated swarm control	[56]
AutoMoDe-Maple	Modular behavior trees, automatic tuning	Higher complexity, new method under exploration	Rich control strategies for evolving swarms	[57]
Buzz	Swarm-specific DSL, stigmergy, and situated communication	Steep learning curve; lacks IDE tools	Flexible programming of heterogeneous swarms	[46]
SwarmTalk	Lightweight communication API; cross-platform abstraction	Middleware only; lacks behavioral logic	Efficient communication layer for custom control stacks	[45]
Pyswarming	Pythonic interface, built-in algorithms, fast prototyping	Limited to simulation environments	Teaching, prototyping, and algorithm development	[48]
Marabunta	Modular Python framework, supports heterogeneity and simulation	Requires manual platform adaptation	Platform-agnostic simulation + physical deployment bridge	[54]
PILOT	Actor-oriented toolkit for ML and distributed control	Limited real-world validation; documentation lacking	Research in ML-integrated distributed swarm programming	[50]
Swarmie (NASA)	Modular ROS architecture, formation control, inter-process messaging	Application-specific (lunar mining); limited public toolkit	Engineering-oriented missions, e.g., resource foraging	[51]
Property-Driven Design	Formal methods; uses PCTL and model checking	High complexity; early-stage usability	Safety-critical swarm verification with provable guarantees	[52]
ROS-Heterogeneous Framework	Five-mode abstraction, modular, ROS-based	Custom-built; limited general documentation	Unified control of heterogeneous multirobot teams	[53]
EvoStick (Neural Net)	Evolves FFNNs using evolutionary algorithms	Prone to overfitting; lacks interpretability	Optimizing reactive swarm behavior from scratch	[56]

Parker et al [55] provides a broad overview of multi-robot systems with a particular emphasis on hardware. While it mentions some specific software examples. While the paper focuses primarily on hardware aspects, it touches on important swarm software principles like behavior-based control that uses simple reactive behaviors that combine into emergent swarm behavior, Distributed algorithms for swarm behaviors like dispersion, leader following, clustering. Back to Franceska et al [56] they develop Vanilla first in 2014 then chocolate in 2015, chocolate has been mentioned earlier in this review, now talking about vanilla, paper focuses on comparing different approaches to designing control software for swarm robots, specifically for the e-puck robot platform. The software architectures compared are:

1. AutoMoDe-Vanilla (Vanilla): is a modular approach-based automatic design process. It assembles pre-existing parametric modules representing low-level behaviors and conditions (like black-floor, neighbor-count, etc.), to synthesize control software in the form of a probabilistic finite state machine.
2. EvoStick: is an implementation of evolutionary robotics for automatic design. It makes use of an unhidden feed-forward neural network.
3. U-Human: is a manual design approach where human experts have complete freedom to design the control software using an API that accesses the robot's sensors and actuators.
4. C-Human: is an additional manual design method in which human experts are limited to using the same parametric modules and control architecture as Vanilla.

The project emphasizes a modular design approach, particularly in the Vanilla and C-Human methods, which employs pre-existing modules to confine the plan space. This methodology helps bridge the reality gap by balancing complexity and interpretability. Both Vanilla and C-Human employ a probabilistic finite state machine (PFSM) architecture, making them well suited for modeling the behaviors of robot swarms. In contrast, EvoStick employs neural networks, offering greater representational power but at the risk of overfitting, which can increase the reality gap. In their paper [57] Kuckling et al explore the using of behavior trees as a control architecture for automatically designing swarm robot software. It presents a new method called Maple, which assembles and fine-tunes pre-existing modules into a behavior tree. It introduces a Novel application of behavior trees to automatic swarm control design, Allows complex behaviors by combining sub-trees greater expressiveness with two-way control transfers and Validated with real-world e-puck robot experiments showed comparable or superior performance to other methods. Gianduja [58], is a technique for automatically designing robot swarms that exhibit communication-based behaviors. The paper focuses on the software design process and evaluates the resulting control software through simulations and physical experiments. It utilizes a modular design approach, allowing for flexible and reactive behavior based on probabilistic rules. Robots communicate via a single locally broadcasted message with emergent semantics, meaning its interpretation depends on the evolved behavior; it belongs to the AutoMode family.

## 5. Answering the Research Questions

Based on the reviewed papers, the answers to the research questions will be as follows:

### 5.1. What Are the Hardware Components That SR Should Be Equipped with?

Based on our review, a two-part process for designing SR systems can be identified:

1. **Hardware Design:** Focuses on choosing the appropriate actuators, sensors, communication modules, and power sources based on the specific environment and task.
2. **Software Design:** Emphasis is placed on developing software that enables emergent swarm behavior inspired by natural swarm intelligence algorithms. The software needs to manage the robot actions, communication, and coordination.



About the components:

The hardware: composed of various parts:

- Actuators: DC motors, servo motors, wheels, legs, propellers, vibration motors, and specialized mechanisms for buoyancy (in aquatic robots).
- Sensors: Infrared (IR) sensors, ultrasonic sensors, cameras, GPS, chemical sensors, temperature sensors, humidity sensors, and touch sensors.
- Communication: Bluetooth, Wi-Fi, ZigBee, RFID, IR, and even electric field detection in aquatic robots.
- Power Sources: Rechargeable batteries (often LiPo), solar panels, and charging systems.

This hardware can vary based on:

- Environment: Different environments require specialized actuators and sensors.
- Task: The specific task influences the hardware selection.
- Economic Constraints: Budgetary limitations may dictate the choice of more affordable components.

## 5.2. What Software Should SR Be Equipped with?

Software is critical for enabling swarming behavior to emerge from local rules and interactions between individual robots; therefore, the essential software components with which robot swarms should be equipped, based on this review, are:

- Application Layer: Software that shapes the swarm behaviors and targets using task specific algorithms like Behavior trees and finite state machines.
- Inter- Communication Layer: Software that Facilitate information exchange and coordination using protocols like WiFi, Bluetooth and ZigBee.
- Control Layer: Software that process the sensor data and control actuators.
- Simulation and development Layer: For testing the robots before the real world deployment like the mentioned Gazebo simulator, also frameworks like Robot operating system (ROS), Buzz and AutoMoDe to programme the SR and the modular software design to ease the development process.
- Hardware Abstraction Layer: Provide interfaces for human users to control and monitor the swarm like HiveAR for real time interaction.

There are real platforms that are used to achieve the swarm behavior and to design and build swarm robots like Like AutoMoDe-Chocolate, Vanilla, and Maple. This mentioned software can be merged with machine learning techniques for better swarm coordination.

### 5.3. What Future Explorations Are Suggested for SR, and What Gaps Have Been Identified in the Reviewed Papers?

First, standardization across SR hardware and software platforms is urgently needed. Setting up common protocols and architectures will make it easier for many people to use, work together, and connect different SR systems. Another important area of research is the automatic design of software for SR. Continued development of modular architectures is needed to make SR systems more flexible and scalable. Modular designs will make it easy to quickly change the way things work, adapt to changing environments, and add new robotic agents or features to existing swarms without any problems. Additionally, combining swarm robotics with cutting-edge artificial intelligence and machine learning methods presents significant opportunities. We can make very smart and adaptable swarm systems by giving each robot the ability to learn and letting the swarm work together to process and adapt to data. In conclusion, looking into new materials, miniaturization methods, and ways to collect energy will make swarm robotics even more powerful. Making robotic agents smaller, more efficient, and able to take care of themselves will allow swarms to work in harder and harder places for longer periods of time.

Regarding the identified gaps in the reviewed papers:

- **Standardization:** The most significant gap is the lack of a universal design methodology or standardized hardware platform. This makes it difficult to compare results, share code, or easily migrate projects between platforms.
- **The Economic Constraints:** The need for low-cost SR is emphasized, but few papers provide detailed comparisons of different cost-effective hardware options. More research on balancing performance with cost is needed. However, the paper by Salman et al., previously reviewed, presents waffle platform, a significant advancement in automatic design methods for SR by addressing the simultaneous creation of hardware and control software under economic constraints.
- **Self-Repair and Robustness:** There isn't much talk about how to make SR that are strong and can fix themselves in tough situations.
- **Automatic Design Methods:** While automatic design methods show promise, they are still under development. More research is needed on their effectiveness.
- **Environmental Adaptation:** The papers mostly look at certain environments, such as ground, aquatic, and aerial environments. We need to do more research on how to make SR that can work in a variety of environments that are always changing.
- **Communication Protocols:** There's a need for standardized, efficient, and robust communication protocols for SR. Existing protocols often lack scalability and flexibility.
- **Swarm Behavior Modeling:** Developing robust models and tools for predicting and analyzing swarm behavior is important for designing effective SR.
- **Real-World Deployment:** Many papers focus on simulation or small-scale experiments.
- **Human-Swarm Interaction:** While some papers mention interfaces, more research is needed on designing intuitive and user-friendly interfaces that enable effective human control and management of large-scale swarms.

- Ethical Considerations: The ethical implications of using SR are rarely discussed.

## 6. Conclusion

In summary, this systematic literature review examined publications from the last decade regarding advancements in the swarm robotics field focusing on hardware-software interplay. Following a rigorous process, this review presents a wealth of information on both hardware and software aspects. The review highlights trending technologies in hardware (sensors, networking, actuators, power supply) and software (HAL, ROS, individual robot control, SI algorithms, simulation tools, communication protocols). Moreover, it discusses the trends in integrating machine learning and modular architectures into the field. The review also identifies critical gaps in the field including the urgent need for standardization of the hardware and the software of SR. It also examines economic constraints and their impact, proposes solutions, and addresses environmental diversity challenges. The paper covered the opportunities for the future research in the field. Finally, the review offers a concise overview of trending technologies from the last decade that can help researchers by providing insights to laying groundwork for advancing the swarm robotics field, emphasizing the importance of interdisciplinary collaboration and innovation to unlock the full potential of these systems in both academic and practical applications.

## References

1. G. Beni, "From swarm intelligence to swarm robotics," in *International Workshop on Swarm Robotics*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 1–9.
2. E. Şahin, "Swarm robotics: From sources of inspiration to domains of application," in *International workshop on swarm robotics*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 10–20.
3. P. G. F. Dias, M. C. Silva, G. P. Rocha Filho, P. A. Vargas, L. P. Cota, and G. Pessin, "Swarm robotics: A perspective on the latest reviewed concepts and applications," *Sensors*, vol. 21, no. 6, p. 2062, 2021.
4. Y. Tan, *Handbook of research on design, control, and modeling of swarm robotics*. IGI global, 2015.
5. S. J. Russell and P. Norvig, *Artificial intelligence a modern approach*. London, 2010.
6. J. P. Kelly and I. H. Osman, *Meta-heuristics: theory and applications*. Kluwer Norwell, 1996.
7. C.-W. Tsai and M.-C. Chiang, *Handbook of metaheuristic algorithms: from fundamental theories to advanced applications*. Elsevier, 2023.
8. H. T. Jongen, K. Meer, and E. Triesch, *Optimization theory*. Boston, MA: Springer US, 2004.
9. E. Şahin, S. Girgin, L. Bayindir, and A. Turgut, "Swarm robotics," in *Swarm intelligence: introduction and applications*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 87–100.
10. A. Francesca, et al., "AutoMoDe-Chocolate: Automatic design of control software for robot swarms," *Swarm Intelligence*, vol. 9, no. 2-3, pp. 125-152, 2015.
11. H. Tello-Rodríguez and L. Torres-Treviño, "Characterization of Environment Using the Collective Perception of a Smart Swarm Robotics Based on Data from Local Sensors," in *Smart Technology: First International Conference, MTYMEX 2017, Monterrey, Mexico, May 24–26, 2017, Proceedings*, Cham: Springer International Publishing, 2018, pp. 133–140.
12. J. Jesson, F. M. Lacey, and L. Matheson, *Doing your literature review: Traditional and systematic techniques*, 2011.
13. D. Pati and L. N. Lorusso, "How to write a systematic review of the literature," *HERD: Health Environments Research & Design Journal*, vol. 11, no. 1, pp. 15–30, 2018.
14. N. Nedjah and L. S. Junior, "Review of methodologies and tasks in swarm robotics towards standardization," *Swarm and Evolutionary Computation*, vol. 50, p. 100565, 2019.
15. É. Villemure, P. Arsenault, G. Lessard, T. Constantin, H. Dubé, L.-D. Gaulin, X. Groleau, S. Laperrière, C. Quesnel, and F. Ferland, "Swarmus: An open hardware and software on-board platform for swarm robotics development," *arXiv preprint arXiv:2203.02643*, 2022.
16. M. Salman, A. Ligot, and M. Birattari, "Concurrent design of control software and configuration of hardware for robot swarms under economic constraints," *PeerJ Computer Science*, vol. 5, p. e221, 2019.
17. M. Patil, T. Abukhalil, and T. Sobh, "Hardware architecture review of swarm robotics system: Self-reconfigurability, self-reassembly, and self-replication," *International Scholarly Research Notices*, vol. 2013, 2013.

18. M. Patil, T. Abukhalil, S. Patel, and T. Sobh, "Ub robot swarm—design, implementation, and power management," in *2016 12th IEEE International Conference on Control and Automation (ICCA)*, IEEE, 2016, pp. 577–582.
19. A. Özgür, S. Lemaignan, W. Johal, M. Beltran, M. Briod, L. Pereyre, F. Mondada, and P. Dillenbourg, "Cellulo: Versatile handheld robots for education," in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, 2017, pp. 119–127.
20. M. Le Goc, L. H. Kim, A. Parsaei, J.-D. Fekete, P. Dragicevic, and S. Follmer, "Zoooids: Building blocks for swarm user interfaces," in *Proceedings of the 29th annual symposium on user interface software and technology*, 2016, pp. 97–109.
21. V. Costa, M. Duarte, T. Rodrigues, S. M. Oliveira, and A. L. Christensen, "Design and development of an inexpensive aquatic swarm robotics system," in *Oceans 2016-Shanghai*, IEEE, 2016, pp. 1–7.
22. P. C. M. Bartmess and P. Ernst, "Fast, low-cost swarm robots," Springer, 2018.
23. B. Shang, "Hardware variation in robotic swarm and behavioural sorting with swarm chromatography," Ph.D. dissertation, University of Southampton, 2017.
24. F. Arvin, J. C. Murray, L. Shi, C. Zhang, and S. Yue, "Development of an autonomous micro robot for swarm robotics," in *2014 IEEE International Conference on Mechatronics and Automation*, IEEE, 2014, pp. 635–640.
25. S. Mintchev, E. Donati, S. Marrazza, and C. Stefanini, "Mechatronic design of a miniature underwater robot for swarm operations," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 2938–2943.
26. N. Moustafa, A. Gálvez, and A. Iglesias, "A general-purpose hardware robotic platform for swarm robotics," in *Intelligent Distributed Computing XII*, Springer, 2018, pp. 259–271.
27. Y. Mulgaonkar, G. Cross, and V. Kumar, "Design of small, safe and robust quadrotor swarms," in *2015 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2015, pp. 2208–2215.
28. D. Pickem, P. Grotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, "The robotarium: A remotely accessible swarm robotics research testbed," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 1699–1706.
29. D. A. Patil, M. Y. Upadhye, F. Kazi, and N. Singh, "Multi robot communication and target tracking system with controller design and implementation of swarm robot using arduino," in *2015 International Conference on Industrial Instrumentation and Control (ICIC)*, IEEE, 2015, pp. 412–416.
30. M. T. Lakshmi and A. Lalitha, "Design and implementation of Arduino based multi robot for target tracking system," 2020.
31. Ç. A. Demir, "Design of a low-cost swarm robotic system for flocking," M.S. thesis, Middle East Technical University, 2019.
32. J. McLurkin, A. McMullen, N. Robbins, G. Habibi, A. Becker, A. Chou, H. Li, M. John, N. Okeke, J. Rykowski, et al., "A robot system design for low-cost multi robot manipulation," in *2014 IEEE/RSJ international conference on intelligent robots and systems*, IEEE, 2014, pp. 912–918.
33. J. Y. Kim, Z. Kashino, T. Colaco, G. Nejat, and B. Benhabib, "Design and implementation of a millirobot for swarm studies—mroberto," *Robotica*, vol. 36, no. 11, pp. 1591–1612, 2018.
34. S. P. Kumar, A. Ravindiran, S. Meganathan, N. O. Roberts, and N. Anbarasi, "Swarm robot materials handling paradigm for solar energy conservation," *Materials Today: Proceedings*, vol. 46, pp. 3924–3928, 2021.
35. F. Bonnet, Y. Kato, J. Halloy, and F. Mondada, "Infiltrating the zebrafish swarm: design, implementation and experimental tests of a miniature robotic fish lure for fish—robot interaction studies," *Artificial Life and Robotics*, vol. 21, pp. 239–246, 2016.
36. P. A. Rezeck, H. Azpurua, and L. Chaimowicz, "Hero: An open platform for robotics research and education," in *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*, IEEE, 2017, pp. 1–6.
37. B. Shang, R. Crowder, and K.-P. Zauner, "Simulation of hardware variations in swarm robots," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, 2013, pp. 4066–4071.
38. A. Abuelhaija, A. Jebrein, and T. Baldawi, "Swarm robotics: Design and implementation," *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 10, no. 2, 2020.
39. M. N. J. E. Jacob, *Simplified Robotics: An Illustrative Guide to Learn Fundamentals of Robotics, Including Kinematics, Motion Control, and Trajectory Planning*. BPB publications, India, 2022.
40. R. Arrick and N. Stevenson, *Robot building for dummies*. John Wiley & Sons, 2003.
41. R. S. Pressman, *Software engineering: a practitioner's approach*. Palgrave macmillan, 2005.
42. M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795–799, 2014.

43. G. Francesca, M. Brambilla, A. Brutschy, L. Garattoni, R. Miletitch, G. Podevijn, A. Reina, T. Soleymani, M. Salvaro, C. Pincirolì, *et al.*, "Automode-chocolate: automatic design of control software for robot swarms," *Swarm Intelligence*, vol. 9, pp. 125–152, 2015.
44. K. Hasselmann, "Advances in the automatic modular design of control software for robot swarms: Using neuroevolution to generate modules," Ph.D. dissertation, Université libre de Bruxelles, Brussels, Belgium, 2023.
45. Y. Zhang, L. Zhang, H. Wang, F. E. Bustamante, and M. Rubenstein, "Swarmtalk-towards benchmark software suites for swarm robotics platforms," in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, 2020, pp. 1638–1646.
46. C. Pincirolì and G. Beltrame, "Swarm-oriented programming of distributed robot networks," *Computer*, vol. 49, no. 12, pp. 32–41, 2016.
47. G. Francesca and M. Birattari, "Automatic design of robot swarms: achievements and challenges," *Frontiers in Robotics and AI*, vol. 3, p. 29, 2016.
48. E. M. de Andrade, A. C. Fernandes, and J. S. Sales, "Pyswarming: a research toolkit for swarm robotics," *Journal of Open Source Software*, vol. 8, no. 89, p. 5647, 2023.
49. Á. Madridano, A. Al-Kaff, P. Flores, D. Martín, and A. de la Escalera, "Software architecture for autonomous and coordinated navigation of uav swarms in forest and urban firefighting," *Applied Sciences*, vol. 11, no. 3, p. 1258, 2021.
50. I. Akkaya, S. Emoto, and E. A. Lee, "Pilot: An actor-oriented learning and optimization toolkit for robotic swarm applications," in *International Workshop on Robotic Sensor Networks*, 2015.
51. K. A. Stolleis, "Swarming robot design, construction and software implementation," Tech. Rep., 2014.
52. M. Brambilla, A. Brutschy, M. Dorigo, and M. Birattari, "Property-driven design for robot swarms: A design method based on prescriptive modeling and model checking," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 9, no. 4, pp. 1–28, 2014.
53. M. Gansari and C. Buiu, "Software system integration of heterogeneous swarms of robots," *Journal of Control Engineering and Applied Informatics*, vol. 19, no. 3, pp. 49–57, 2017.
54. M. Chamanbaz, D. Mateo, B. M. Zoss, G. Tokić, E. Wilhelm, R. Bouffanais, and D. K. Yue, "Swarm-enabling technology for multi-robot systems," *Frontiers in Robotics and AI*, vol. 4, p. 12, 2017.
55. L. E. Parker, D. Rus, and G. S. Sukhatme, "Multiple mobile robot systems," *Springer handbook of robotics*, pp. 1335–1384, 2016.
56. G. Francesca, M. Brambilla, A. Brutschy, L. Garattoni, R. Miletitch, G. Podevijn, A. Reina, T. Soleymani, M. Salvaro, C. Pincirolì, *et al.*, "An experiment in automatic design of robot swarms: Automode-vanilla, evostick, and human experts," in *Swarm Intelligence: 9th International Conference, ANTS 2014, Brussels, Belgium, September 10–12, 2014. Proceedings 9*, Springer, 2014, pp. 25–37.
57. J. Kuckling, A. Ligot, D. Bozhinoski, and M. Birattari, "Behavior trees as a control architecture in the automatic modular design of robot swarms," in *International conference on swarm intelligence*, Springer, 2018, pp. 30–43.
58. K. Hasselmann, F. Robert, and M. Birattari, "Automatic design of communication-based behaviors for robot swarms," in *Swarm Intelligence: 11th International Conference, ANTS 2018, Rome, Italy, October 29–31, 2018, Proceedings 11*, Springer, 2018, pp. 16–29.
59. L. Williams, "FEATURE: How swarms of robots will revolutionise the construction industry," Aug. 14, 2019. [Online]. Available: [https://imecheweresources.blob.core.windows.net/imeche-web-content/images/default-source/default-album/dscf4314\\_lr1-edit\\_lr1.jpg?sfvrsn=c438b12\\_0](https://imecheweresources.blob.core.windows.net/imeche-web-content/images/default-source/default-album/dscf4314_lr1-edit_lr1.jpg?sfvrsn=c438b12_0)
60. "HC-SR04 Ultrasonic Distance Sensor." [Online]. Available: [https://imecheweresources.blob.core.windows.net/imeche-web-content/images/default-source/default-album/dscf4314\\_lr1-edit\\_lr1.jpg?sfvrsn=c438b12\\_0](https://imecheweresources.blob.core.windows.net/imeche-web-content/images/default-source/default-album/dscf4314_lr1-edit_lr1.jpg?sfvrsn=c438b12_0)
61. "Sharp GP2Y0A21YK IR proximity sensor.jpg." [Online]. Available: [https://upload.wikimedia.org/wikipedia/commons/8/89/Sharp\\_GP2Y0A21YK\\_IR\\_proximity\\_sensor.jpg](https://upload.wikimedia.org/wikipedia/commons/8/89/Sharp_GP2Y0A21YK_IR_proximity_sensor.jpg)
62. "Adafruit Standalone 5-Pad Capacitive Touch Sensor Breakout - AT42QT1070 [ADA1362]." [Online]. Available: [https://m.media-amazon.com/images/I/61Go+qnompL.\\_AC\\_SX522\\_.jpg](https://m.media-amazon.com/images/I/61Go+qnompL._AC_SX522_.jpg)
63. "LSM303D 3D Compass and Accelerometer Carrier with Voltage Regulator." [Online]. Available: <https://a.pololu-files.com/picture/0J4935.1200.jpg?7d52096304294be12fb09e46e661c656>
64. "DC Gear Motor, DC 6V Gear Motor High Torque 1:1000 for Toy Car Model Reduction Gearbox 10/15/20RPM(6V 10RPM)." [Online]. Available: [https://m.media-amazon.com/images/I/61nJ1X1u5L.\\_SL1001\\_.jpg](https://m.media-amazon.com/images/I/61nJ1X1u5L._SL1001_.jpg)



65. "TowerPro SG 90 Micro Servo Motor." [Online]. Available:[https://m.media-amazon.com/images/I/61yflwAxe0L.\\_SX522\\_.jpg](https://m.media-amazon.com/images/I/61yflwAxe0L._SX522_.jpg)
66. "EMAX Multicopter motor MT2213 (With Prop1045 Combo) 935KV." [Online]. Available:[https://rees52.com/cdn/shop/files/Drone\\_Multirotor\\_Motor\\_combo.png?v=1740738578&width=713](https://rees52.com/cdn/shop/files/Drone_Multirotor_Motor_combo.png?v=1740738578&width=713)
67. "Stepper motor." [Online]. Available:<https://stock.adobe.com/dz/images/stepper-motor-of-cnc-linear-axis-drive-of-3d-machine/84747521>
68. "ESP8266 WiFi Module." [Online]. Available:<https://grobotronics.com/images/thumbnails/570/570/detailed/135/ESP8266-01-112.png>
69. "RN-42 Bluetooth Module." [Online]. Available:<https://media.parallax.com/wp-content/uploads/2020/07/13151324/30086.png.webp>
70. "2.4GHz Wireless Transceiver nRF 24L01." [Online]. Available:[https://store.fut-electronics.com/cdn/shop/products/nrf24l01\\_1024x1024.jpeg?v=1566906321](https://store.fut-electronics.com/cdn/shop/products/nrf24l01_1024x1024.jpeg?v=1566906321)
71. "Pmod WiFi: WiFi Interface 802.11g." [Online]. Available:<https://cdn02.plentymarkets.com/qxzolkbuxzfb/item/images/100705/middle/24083-gross.jpg.avif>

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.