Scalability, Robustness, and Resilience

Jonas Kuckling, Alexandre Pacheco, Andreagiovanni Reina, Heiko Hamann

Synonyms

expandability; size-robustness; fault-tolerance; flexibility; antifragility

Definitions

Scalability

Scalability is the property of a multi-robot system to perform its tasks across varying system sizes efficiently. In cases where resources (space, communication channels, etc.) are shared, the shared resource can either be fixed or change with the system size. Both should be considered, up-scaling and down-scaling, that is, proportionally maintained performance for increased system sizes but also for decreased system sizes.

Robustness and Resilience

Robustness and resilience describe the response of a multi-robot system to internal or external disturbances, such as communication noise, attacks, or component failures. A typical distinction between the two terms is based on the effects of the disturbance (Munoz et al 2022). A system is robust when it maintains its performance in the presence of disturbances or uncertainties in the system parameters. A system is resilient when it recovers from a drop in performance caused by errors and disturbances. It is also possible to distinguish the two on the basis of how the system reacts to the disturbances (Prorok et al 2021). Robust systems remain in the same state and no structural transformations take place, while resilient systems transform themselves (e.g., through structural re-organization and growth) to adapt to the newly perceived conditions.

Overview

Multi-robot systems offer ways to implement systems with desirable properties, such as scalability, robustness, and resilience. However, implementing multi-robot systems with these desirable properties requires dedicated design, accounting for robot interactions and coordination. For example, *scalability* can be achieved by supporting the addition or removal of robots without compromising overall system functionality; *robustness* can be achieved by including redundancies and enabling multiple robots to perform the same task; and *resilience* can be achieved by designing self-organized group behaviors that autonomously adapt to environmental disturbances.

Swarm robotics, in particular, follows design paradigms for multi-robot systems which promote the realization of multi-robot systems exhibiting these desirable properties (Hamann 2018). For example, robots in a swarm are limited to local communications to prevent critical bottlenecks or points of failure in the communication medium or processing of messages.

Key Research Findings

Scalability

As defined above, scalability is the ability of a multi-robot system to adapt to different system sizes, that is, to keep functioning efficiently for different numbers of robots N in the system. Scaling up and down can be challenging. When scaling up ($N' > N$), shared resources may get scarce. When scaling down ($N' < N$), robots may not meet often enough for a required collaborative feature of the considered task. Scalability in system size is surely a unique challenge within robotics that is inherent to multi-robot systems. However, outside of robotics, scalability is a widely known challenge, too. For example, the analysis of network traffic (e.g., radio or car traffic) has a long tradition of testing for scalability. The most prominent field to mention is computation, in particular, parallel and distributed computing. In the following, the focus is first on what can be learned from this computational perspective of scalability and then we focus on the actual robotics challenge.

Computational perspective

There is vertical and horizontal scalability in the context of enhancing the capacity and performance of computing systems. Vertical scalability, or scaling up, increases the capacity of a single server by adding more resources such as CPUs, memory, or storage; whereas horizontal scalability, or scaling out, enhances system performance by adding more servers or nodes to distribute the workload across multiple machines. The more relevant concept here is horizontal scalability as we add or remove robots rather than expanding capabilities of individual robots. The key metric for scalability in computing is speedup. We define speedup S as

$$S = \frac{T_1}{T_p}\,, \tag{1}$$

for time T_1 it takes to solve the problem on a single processor and time T_p to solve the problem on p processors.

Research of parallel and distributed computing has produced scalability models or laws. Most of them are phenomenological, that is, they are neither derived from first principles nor from the microscopic mechanisms governing the behavior of the system's components. Instead, they are based on high-level insights and the creativity of researchers. The two most prominent and most cited scaling laws are probably Amdahl's Law and Gustafson's Law. Amdahl's Law is considered to be a rather pessimistic look at parallel computing and was probably intended to be so. In that sense, Gustafson's Law is the more optimistic response. In addition, Gunther's Universal Scalability Law (USL) is worth mentioning. It addresses especially the challenge of finding the best system size N_{opt} as system performance (or speedup) may also decrease once available shared resources are depleted.

Amdahl's Law

Amdahl's Law (Amdahl 1967) is used to find the maximum improvement possible by enhancing a particular part of a system. Amdahl defines

$$S(N) = \frac{N}{1+\sigma(N-1)}\,, \qquad (2)$$

where $S(N)$ is the theoretical speedup of the execution of the whole task, $0 \leq \sigma \leq 1$ is the fraction of time spent on serial parts of the task (i.e., parts that cannot be parallelized), and N is the number of processors. Amdahl's Law is considered to be a pessimistic take on parallel programming as it predicts a bounded scalability of performance. The performance saturates at a constant value of σ.

Gustafson's Law

Gustafson's Law (Gustafson 1988), also known as Gustafson–Barsis's Law, addresses the shortcomings of Amdahl's Law by focusing on the scaling of the workload with the system. Often applied in high-performance computing scenarios where the workload can be scaled with the system. Gustafson's Law is defined as

$$S(N) = N + (1-N)\sigma\,, \qquad (3)$$

where we assume that the serial part $0 \leq \sigma \leq 1$ of the task decreases with system size N. This is connected to the assumption of problem scaling. We assume that the problem size (or work package size) scales with system size N. This assumption turns out to be reasonable as the serial part is often the administrative work to divide and allocate tasks.

Universal Scalability Law

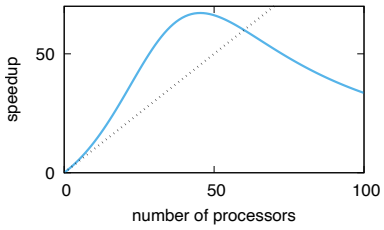

Fig. 1 Retrograde scalability according to the Universal Scalability Law.

The Universal Scalability Law (Gunther 1993, 2007, USL) is a model used to predict the scalability of a system, particularly in the context of parallel processing and computing. It extends Amdahl's Law and incorporates elements to account for the overhead caused by increased parallelism. The USL is expressed mathematically as

$$S(N) = \frac{N}{1+\alpha(N-1)+\beta N(N-1)}\,, \tag{4}$$

where α is the coefficient representing the overhead from serialization or contention (similar to the non-parallelizable portion σ in Amdahl's Law), β is the coefficient representing the overhead from coherency delay or communication between processing units. Following this model we can distinguish several qualitatively different scenarios: (1) when $\alpha = \beta = 0$, the model implies perfect linear scalability; (2) when $\beta = 0$ but $\alpha > 0$, it reduces to Amdahl's Law; (3) $\beta > 0$ accounts for the additional overhead that comes into play as more processors are added, which is neither considered in Amdahl's nor in Gustafson's Laws. This scenario, called 'retrograde scalability' (analogous to the concept of diminishing returns from economics), had been often overlooked in the literature. However, to create good models of scalability, it is useful to consider (Hamann and Reina 2022). Hence, the USL is particularly suited for understanding real-world systems where interactions between processors or tasks can significantly impact performance as the system scales.

Scaling and managing resources online

In computing of today, it is crucial to manage resources dynamically, for example, in microservice architectures, cloud computing (e.g., virtual machines), high-performance computing environments, and real-time data processing. Elasticity is defined as the skill of automatically scaling resources up or down based on current demands. This belongs to the more general category of load balancing and resource management (Leinberger et al 2000).

Robotics perspective

In multi-robot systems, scalability has long been recognized as a critical challenge in distributed control (Tegling and Sandberg 2019; Li et al 2021), and is often cited as one of the primary advantages of swarm robotics (Şahin 2004; Hamann 2018). As a result, extensive research has been conducted on this topic; however, most of these studies are empirical and tailored to specific scenarios. Similar to scalability challenges in other fields, multi-robot systems face difficulties in efficiently managing shared resources. The most prominent ones in this context are the shared space of the group of mobile robots and their communication channels. Additionally, the presence of too many concurrently operating active sensors (e.g., sonar, infrared) can also become a bottleneck due to an interference challenge.

In swarm robotics, an important property to study scalability is the swarm density, that is number of robots N per area A: $\rho = \frac{N}{A}$. The generally claimed high scalability of robot swarms may hold if the density ρ is kept constant.

We would need to provide additional space for each additional robot (and similarly reduce space for each removed robot). In turn, the system's scalability is contested once swarm density is increased. In a sense, that is the stress test for scalability.

Despite the importance of scalability in multi-robot systems, it was rarely studied as the main subject but rather as a symptom in given use cases. Rare exceptions include the following.

Rosenfeld et al (2008) investigate scalability through extensive simulations in foraging and search tasks. They propose composite coordination strategies to reduce robot-robot interference. They focus on specific methods, such as mild repulsion and random movement, dropping carried objects when near teammates, and backing off for a fixed duration. Their main finding is that coordination strategies that more effectively reduce interference support better scalability, while non-adaptive methods lead to sharp declines in group performance.

Hayes (2002) reports a quantitative analysis of how robot group size affects performance in collective search tasks. They derive analytical expressions for optimal group size and system performance of the form

$$S(N) = \frac{N}{a + bN^2} \,, \tag{5}$$

for number of robots N, and constants a and b. Simulations confirm the analytical predictions, demonstrating how cost tradeoffs influence the optimal swarm size and offering concrete design guidelines for efficient, scalable multi-robot systems.

When reviewing empirical studies on scalability of multi-robot systems (see for example this small representative collection: Rosenfeld et al (2008); Pini et al (2011); Talamali et al (2020)), particularly on large-scale systems, a common trend emerges: the average performance, as a function of robot density, consistently follows a similar pattern. This pattern aligns with the third scenario of the USL: retrograde scalability (see Fig. 1). This observation suggests a fundamental relationship between system size and performance that holds across a wide variety of otherwise diverse systems (e.g., traffic systems: Iannini and Dickman (2016), vascular flow in plants: Jensen et al (2013)). Models like the USL or models suggested specifically for swarm robotics (Hamann 2013) try to model this relationship. While these models can help to predict scalability properties they are phenomenological, that is, they describe the effect without explaining it based on first principles. It is important to get a deeper understanding of the underlying mechanisms to learn also how to design for scalability.

Modeling scalability in multi-robot systems from first principles

Hamann and Reina (2022) tried to resolve that and developed a general model of scalability, unifying models of scalability across different domains and deriving a population model starting from a microscopic model. The model uses a three-state system to describe interactions between units and derives system-wide performance from these interactions. It is versatile enough to apply to various fields, including robotics and computing, and demonstrates different scalability behaviors, including linear,

sublinear, and superlinear speedups. The three robot states are: *solo* (robot without contact to other robots), *grupo* (robot in contact with one or a few robots), and *fermo* (robot in congestion, deadlock). The assumption is that any of these three states can be ascribed to any robot at any time. The model is a system of ordinary differential equations (ODE):

$$\begin{cases} \frac{ds}{dt} = & -2k_1s^2 - k_2sg - k_3sf + k_4g \\ \frac{dg}{dt} = & 2k_1s^2 + k_2sg + k_3sf - k_4g \\ & \quad -2k_5g^2 - k_6gf + k_7f \\ \frac{df}{dt} = & \quad 2k_5g^2 + k_6gf - k_7f\,, \end{cases} \tag{6}$$

where the system variables s, g, and f represent the number of robots in states *solo* S, *grupo* G, and *fermo* F; and parameters k_i describe the dynamics of the system. This system's fixed points for different initializations provide predictions of how the swarm robot systems will scale.

This model was later analyzed by Kuckling et al (2024) who showed that this model even predicts a specific 2-phase system behavior. It was first observed empirically that such two phases can coexist. For a given swarm size (or swarm density) one observes system runs that perform well and other runs that perform poorly. This bimodal distribution of system performance over independent runs reveals that studying scalability in multi-robot systems based on mean system performance (as seen in Fig. 1) can be fundamentally misleading. The arguably graceful degradation once swarm density goes beyond peak system performance may not exist in real-world systems. They may either continue in the high-performance phase or break down in a system deadlock.

The 2-phase system behavior indicates a critical system behavior as in a bifurcation and that was shown by Kuckling et al (2024) for the above ODE system (eq. 6) through non-linear systems analysis. The consequence, especially for performance optimized systems, is that the designer might accidentally put them on the edge. A small change in the system then can cause a disastrous breakdown of system performance. To avoid this scenario, the system should be optimized for individual performance (best individual performance is reached for smaller swarm sizes before the system performance peak in Fig. 1). Other options would be general means to maximize scalability and to avoid deadlocks or congestions.

For applications of multi-robot systems, we would like to have methods that can guarantee, or at least predict, system scalability for a given swarm density. Also an online distributed early-warning system would be desirable. However, applicable results are pending.

Improving scalability through offline methods

In multi-robot literature there are many approaches to enable system size scalability. These aim to manage access to shared resources, effectively increasing the system size at which the system can still safely run.

A common approach is the partition of an intractably large task into smaller ones (Vieira et al 2009). The problem partitions can then be efficiently solved without fear of congestions or deadlocks

and coordination between the partitions ensures that the complete task is solved. Unfortunately, there exists no general partition scheme and solutions need to be found for the particular task at hand.

Another scalability challenge occurs in human-robot interaction, where scalability is important to allow multiple users to interact with multiple robotic systems simultaneously. A typical scalable architecture might function in two-phases: a first phase establishing many-to-many connections, connecting users and robots; and a second phase of one-to-one interactions in which the user interacts directly with selected robots (Tews et al 2003).

Online scalability

In literature, usually scalability in multi-robot systems is understood as an offline property, that is, the system is supposed to scale to a certain system size or (swarm) density while assuming robot conservation during a mission. We start with a given number N of active robots and N remains time-invariant. Only few exceptions exist in the literature that study an online distributed self-organizing mechanism to allow the robot team to regulate system size or swarm density adaptively in a distributed approach at runtime (akin to the concept of elasticity in distributed computing).

Mayya et al (2019) reduce interference in robot swarms applied to a distributed object collection task. Each robot autonomously decides to remain or retreat based on local proximity encounters, minimizing interference without explicit communication between robots. Experimental results demonstrate that this algorithm effectively regulates robot density, enhancing the overall efficiency and productivity of the swarm.

Wahby et al (2019) take a different approach by adapting the robot system to external effects that change the robot density. Each individual robot within the swarm monitors local conditions and shares information with neighbors, enabling the swarm to collectively detect swarm density and to adjust. Experiments with ten robots demonstrate that this density awareness significantly improves the swarm's performance. This implements a practical adaptive approach for fluctuating robot densities.

Fuady et al (2025) proposed an approach, which addresses online scalability at the level of task participation. They dynamically adjusts the size of a self-organized subset of robots involved in collective decision-making, allowing the swarm to scale its task-specific size at runtime. While the physical robot population remains constant, this mechanism implements a form of online functional scalability by selectively engaging only as many robots as required for reliable collective decisions.

Robustness and Resilience

In addition to scalability, multi-robot systems must also display robustness and resilience before being deployed in real-world applications. While computer hacking and malfunction can lead to important information and service losses, malfunctioning robots also threaten the environment in which they operate, including all agents with whom they share space, for example, humans, animals,

vehicles, and other robots. As the system size increases, it can become impossible to monitor the concurrent behavior of a large number of robots to detect and preempt performance degradation or system failures as external observers. Hence, it is important to design collective behaviors or mechanisms to make the multi-robot system inherently robust and resilient (Hunt and Hauert 2020).

In addition to interacting with the environment through sensing and actuation, distributed robotic systems are characterized by information exchange and robot coordination (Gielis et al 2022). Without this aspect, they would merely be a collection of independently operating robots with no interaction among them (e.g., 3D visual installation of drones that only communicate with a base control station). Hence, guaranteeing the robustness of the individual robot control (e.g., by employing controller safety margins to withstand model uncertainties) addresses only part of the problem of building a robust multi-robot system. A common form of robustness analysis in multi-robot systems involves adding constant unbiased disturbances, such as white noise, on information acquisition and/or transfer (Chin et al 2023; Garg and Panagou 2021), while more sophisticated analyses may also account for systematic biases (De Masi et al 2021) and spatiotemporal correlation (Bartashevich and Mostaghim 2021; Rausch et al 2019).

Applications that, to operate properly, require constant connectivity among robots or with an external network need to take into consideration inherent robots' mobility, which introduces the additional challenge of maintaining the network of robots connected, critical to certain applications (e.g., formation control; Ji and Egerstedt 2007; Kan et al 2010; Coppola et al 2019). Robots that separate from the group or move into network-denied areas can act using outdated information or fail to report critical information to others.

Types of disturbances

Robustness and resilience analysis should, however, not only be limited to random noise and network connectivity but include individual robot failures (e.g., dirty sensors, software errors, damaged components, robot breakdown), harmful cyberattacks (e.g., content manipulation, jamming), and unforeseen environmental changes that can affect the entire system (e.g., obstructed motion or communication paths, infrastructure breakdown, low visibility due to smoke).

Faults and breakdowns

Robots can spread false and incorrect information or stop communicating (potentially causing information loss or disappearance of communication links in the network) due to malfunctioning hardware, software errors, or damaged components. In multi-robot systems, most research efforts have focused on proposing functioning collective behaviors (often only assessing robustness to noise). State-of-the-art swarm robotics systems can suffer catastrophic system breakdowns when even a single malfunctioning robot spreading incorrect information is introduced into the system (Strobel et al 2023; Van Calck et al 2023). There is a growing interest by the community in analyzing the

system's robustness and resilience to robot faults and designing mechanisms to withstand them (Hunt and Hauert 2020). Multi-robot systems operating in structured environments (for example, in warehouses) can benefit from constant supervision and prompt identification of faulty robots, triggering intervention from repair operators. Differently, robot teams operating 'in the wild,' without human supervision or automated surveillance (e.g., space exploration, underwater operations, wildlife monitoring), may need dedicated mechanisms to prevent system-wide failures when a minority of robots stops functioning correctly.

Adversarial attacks

Multi-robot networks, particularly decentralized ones such as robot swarms, can also be vulnerable to attacks perpetrated by malicious actors. These attacks can be performed both externally (e.g., jamming the communication channel) and internally (e.g., when a hacker gains control over one or more robots). Several types of attacks can be perpetrated against a multi-robot system. Through a Sybil attack (Douceur 2002), a small minority of robots forges many fake identities to gain control over the system or interfere with its operations. Jamming of communication channels or physical tampering can cause a 'denial of service' on some robots (similar to traditional cyberattacks on the internet), that become unable to carry out their operations (Multerer et al 2017; Xu et al 2021). Information tampering has been recently studied in the form of robots that deliberately spread false information (e.g., zealot or contrarian robots in opinion dynamics (De Masi et al 2021; Reina et al 2023), copycat robots in collective learning (Pacheco et al 2024)).

Since the disturbances caused by adversarial actors are often concealed by their perpetrators, they cannot be easily identified. Therefore, it may be required to deploy cryptographic security (e.g., message signing) to safeguard the operation of the multi-robot system, particularly during deployments in unstructured outdoor environments where the possibility of adversarial attacks is higher (e.g., in defense applications or chaotic metropolitan areas).

Environmental disturbances

While a common approach has been to create a disturbance-free environment for the robots (e.g., purposely-built warehouses (Everett et al 1995; Carlson and Murphy 2003)), to deploy multi-robot systems in most outdoor applications the robots must be able to robustly operate in unstructured environments (e.g., urban) without incurring faults or overly relying on environment infrastructure.

Infrastructure breakdown can compromise the robots' ability to communicate or coordinate with one another. For safe real-world deployment, multi-robot systems need alternative sensing/communication modalities to maintain liveness (which is the system's ability to stay operational) (Lamport 1977). While redundant mechanisms may appear inefficient, relying on more than one sensing or communication technologies, for example, multiple localization systems (such as GPS, radio waves, or sound), can prove vital as one

technology compensates when another fails.

Robust and resilient collective behaviors

Solutions should be chosen considering the application scenario at hand, as each system can be exposed to different kinds of threats to different extents. In structured environments, robustness is primarily centered on guaranteeing performance in the face of sensor noise and network failures. However, additional potential disturbances can arise when considering interactions with humans and long-term autonomy (Egerstedt et al 2018). Systems interacting with external actors (e.g., urban robots in public places) or comprising a large number of robots (impossible real-time monitoring) are exposed to further threats (e.g., hacking, jamming, and tampering). These dangers are exacerbated in systems composed of robots belonging to different stakeholders, where aligned interests (hence cooperation) among robots cannot be guaranteed.

Real-world deployments should consider the integration of multiple solutions, anticipating the wide variety of disturbances that may occur. The following types of robust and resilient approaches to the design of multi-robot systems can be identified.

Robust bio-inspired behaviors

System robustness can be achieved even without identifying and actively neutralizing the source of disturbance (Park and Hutchinson 2017). Instead, the multi-robot system can be inherently robust to them, for example, the group could collectively filter the sensor noise or maintain unperturbed performance when part of the group stops working. Natural systems are often robust because organisms evolved to survive in uncertain and hostile environments. Swarm robotics solutions are predominantly inspired by collective animal behavior, with successful solutions in several tasks, for example, bee-inspired decision-making (Reina et al 2023), ant-inspired pheromone navigation (Ducatelle et al 2010), cockroach-inspired aggregation (Garnier et al 2005). These bio-inspired algorithms can be simple and have minimal memory and computation requirements, making them suitable for large-scale systems of minimal machines. Nevertheless, they are able to sustain high disturbance levels, robot breakdowns, and information biases, for example, through leveraging group dynamics and robot redundancy (Hamann 2018). However, such simple systems may be susceptible to issues like digital double counting (where old messages are treated as new) or malicious attacks, potentially propagating errors such as incorrect pheromone levels (Aswale et al 2022) or voting decisions (Strobel et al 2023; Zhao et al 2023). Hence, dedicated mechanisms to enhance security are needed.

Maintaining connectivity

Several applications require the robots to remain within the communication range of one another to avoid system-wide failures (Olfati-Saber et al 2007). One approach consists of constructing and maintaining k-connected

networks that ensure connectivity despite the failure of up to $k-1$ robots through redundancy (Dashti et al 2022; Guerrero-Bonilla et al 2017). To maintain the robot network connected despite environmental disturbances and imprecise robot motion, a popular approach is to employ control barrier functions to constrain robot movements and avoid network fragmentation (Capelli et al 2021). Such methods typically rely on tools from graph theory, or Laplacian spectrum analysis and Lyapunov-like conditions (Ames et al 2019). They have the advantage of providing theoretical guarantees on the system's *robustness*; however, these guarantees only apply when the underlying model of the multi-robot system is adequate, and disturbances stay within specified bounds. However, establishing good models and bounds is challenging. The nonlinear interactions between (potentially many) robots make the modeling and analysis difficult, often limited to small groups or limiting assumptions (e.g., complete system observability, semi-synchronous communication). Moreover, unpredictable and dynamic environments can expose the multi-robot system to unforeseen situations, out of the established bounds. Hence, system's robustness can be further improved with algorithms that tolerate temporary disconnections, such as partition-tolerant methods (Winfield 2000) or that resiliently reconfigure their network to restore connectivity during attacks (Saulnier et al 2017).

Cooperative behaviors to detect faults

An asset of multi-robot systems is the redundancy of robots which can exploit "their many eyes" to detect faults. Robots normally operate in the same space and can frequently exchange information and inspect each other's actions. For example, robots can compare each other's observations to report any significant difference in their and others' sensing or actuation (Tarapore et al 2015; Lee et al 2022). These cooperative strategies contribute to the *resilience* of multi-robot systems, by allowing them to detect and isolate faulty members to maintain safe operations, rather than merely withstanding their presence. While effective, they can only be employed when robots reliably share trustworthy information (e.g., internal state, sensor readings) as deceitful agents can potentially evade fault detection.

Resilience to Byzantine robots

In adversarial scenarios, the compliance of the robots in detecting and neutralizing faults cannot be guaranteed, as malicious robots may attempt to conceal their faults and mislead the detectors (Mitra et al 2019). Under a Byzantine fault model (Pease et al 1980), faults are erratic and cannot be associated with any particular robot.

Byzantine fault tolerance is extensively studied in multi-robot rendezvous, where robots converge at a common location in a shared environment Park and Hutchinson (2017); Bouzid et al (2009). While solutions vary depending on the system's assumptions, one necessary assumption common to most rendezvous algorithms is that robots can externally observe the state of other robots. Reaching an agreement on internal hidden states that cannot be reliably measured poses further challenges.

Several approaches achieve resilience to Byzantine robots in consensus achievement by combining local outlier-removal rules (e.g., W-MSR; LeBlanc et al 2012) with mechanisms that maintain sufficient network connectivity. The underlying connectivity requirement is commonly formalized through r-robustness (Zhang et al 2015), which ensures that, even after discarding potentially corrupted information, each robot maintains enough honest neighbors to support resilient consensus (LeBlanc et al 2013). System resilience can be further enhanced by enabling robots to update a reputation system used to weigh information they receive (Usevitch and Panagou 2019; Yemini et al 2025; Tang et al 2025).

Guaranteeing network connectivity among mobile robots can be challenging in adversarial settings (Cavorsi et al 2024; Lee and Panagou 2025); therefore, decentralized robot networks can exploit global knowledge and circumvent connectivity requirements by using globally synchronized ledgers, such as blockchains (Dorigo et al 2024; Ferrer 2023). In such systems, reaching a correct agreement on internal states is only possible if $N > 3F$, where N is the swarm size and F is the number of faulty agents (Pease et al 1980). Exploiting blockchains as a distributed and tamper-proof digital memory for robot swarms can enable a high swarm resiliency (Strobel et al 2023), which can potentially be achieved using reputation systems based on economic principles (Van Calck et al 2023). Information uploaded on the blockchain by the robots cannot be altered or refuted, creating a system whose members are accountable for their actions, ultimately increasing trust in the system and facilitating the integration of multi-robot systems into our societies (Pacheco et al 2025).

Future Directions for Research

Scalability, robustness, and resilience are fundamental challenges to design multi-robot systems for practical, real-world use cases. Studying scalability will allow developing large-scale multi-robot systems that can be deployed without the risk of congestion or deadlocks. By increasing robustness and resilience, multi-robot systems can operate even under adverse conditions.

Scalability

In terms of scalability, two research directions may become increasingly important. First, it is necessary to better understand the mechanisms behind the scalability of multi-robot systems. The theoretical analysis of the scalability model by Hamann and Reina (2022) indicates that multi-robot systems might exhibit hysteresis effects (Kuckling et al 2024). This implies that a system, once congested, might not be easily recoverable anymore. However, this theoretical insight has not yet been shown in practice. By gaining a deeper understanding of scalability dynamics, it might be possible to better characterize multi-robot systems in an effort to determine system sizes for optimal performance.

A second important research direction will be the online regulation of scalability (Mayya et al 2019). By

developing novel methods for multi-robot systems to detect and regulate local robot densities, the region of high-performing system sizes can be extended. Even if online density regulating multi-robot systems do not offer increased performance, they reduce the risk of catastrophic effects and critical system breakdowns caused by congestion.

Robustness and Resilience

Future research on robustness and resilience should focus on developing new collective behaviors to bring the robustness and resiliency of multi-robot systems to the needed standards for real-world deployment, safeguarding both humans and the environment where they operate. Progress in robustness and resilience will enable the deployment of multi-robot settings in scenarios where repairing the robots or removing sources of disturbance is not possible (e.g., replacing faulty sensors during remote applications).

In addition to developing specific solutions to given problems, another important direction of research should focus on integrating different mechanisms into a single system. Existing studies have too often only focused on one possible protocol and type of disturbance at a time.

Given the inevitability of noise in robotics, an interesting concept is *antifragility* (Taleb 2012; Ilan 2023), where the system improves its performance when subject to disturbances, rather than suffering from them. A possible approach to building antifragile systems is to resort to inspiration from natural systems, which evolved in highly noisy and unreliable contexts, and likely converged to robust strategies that exploit noise and disturbances (e.g., increase exploration, avoid stagnation, or break symmetry) (Tsimring 2014; Jhawar et al 2020; Zakir et al 2024).

References

Amdahl GM (1967) Validity of the single processor approach to achieving large scale computing capabilities. In: AFIPS Conference Proceedings, ACM, pp 483–485

Ames AD, Coogan S, Egerstedt M, Notomista G, Sreenath K, Tabuada P (2019) Control barrier functions: Theory and applications. In: 2019 18th European control conference (ECC), IEEE, pp 3420–3431

Aswale A, López A, Ammartayakun A, Pinciroli C (2022) Hacking the colony: On the disruptive effect of misleading pheromone and how to defend against it. In: Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, AAMAS '22, p 27–34

Bartashevich P, Mostaghim S (2021) Multi-featured collective perception with evidence theory: tackling spatial correlations. Swarm Intelligence 15(1-2):83–110, DOI 10.1007/s11721-021-00192-8

Bouzid Z, Potop-Butucaru M, Tixeuil S (2009) Byzantine Convergence in Robots Networks: The Price of Asynchrony. Technical report, ???

Capelli B, Fouad H, Beltrame G, Sabattini L (2021) Decentralized connectivity maintenance with time delays using control barrier functions. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 1586–1592

Carlson J, Murphy R (2003) Reliability analysis of mobile robots. In: 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422), vol 1, pp 274–281 vol.1, DOI 10.1109/ROBOT.2003.1241608

Cavorsi M, Sabattini L, Gil S (2024) Multirobot adversarial resilience using control barrier functions. IEEE Transactions on Robotics 40:797–815, DOI 10.1109/TRO.2023.3341570

Chin KY, Khaluf Y, Pinciroli C (2023) Minimalistic collective perception with imperfect sensors. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 8862–8868

Coppola M, Guo J, Gill E, de Croon GC (2019) Provable self-organizing pattern formation by a swarm of robots with limited knowledge. Swarm Intelligence 13(1):59–94

Dashti ZAZS, Deplano D, Seatzu C, Franceschelli M (2022) Resilient self-organizing networks in multi-agent systems via approximate random k-regular graphs. In: 2022 IEEE 61st Conference on Decision and Control (CDC), pp 6448–6453, DOI 10.1109/CDC51059.2022.9992489

De Masi G, Prasetyo J, Zakir R, Mankovskii N, Ferrante E, Tuci E (2021) Robot swarm democracy: The importance of informed individuals against zealots. Swarm Intelligence 15(4):315–338, DOI 10.1007/s11721-021-00197-3

Dorigo M, Pacheco A, Reina A, Strobel V (2024) Blockchain technology for mobile multi-robot systems. Nature Reviews Electrical Engineering 1:264–274, DOI 10.1038/s44287-024-00034-9

Douceur JR (2002) The Sybil attack. In: International workshop on peer-to-peer systems, Springer, pp 251–260

Ducatelle F, Di Caro GA, Förster A, Gambardella L (2010) Mobile stigmergic markers for navigation in a heterogeneous robotic swarm. In: Swarm Intelligence, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 456–463

Egerstedt M, Pauli JN, Notomista G, Hutchinson S (2018) Robot ecology: Constraint-based control design for long duration autonomy. Annual Reviews in Control 46:1–7

Everett HR, Gage DW, Gilbreath GA, Laird RT, Smurlo RP (1995) Real-world issues in warehouse navigation. In: Wolfe WJ, Chun WH (eds) Mobile Robots IX, International Society for Optics and Photonics, SPIE, vol 2352, pp 249 – 259, DOI 10.1117/12.198975

Ferrer EC (2023) If blockchain is the solution, robot security is the problem. Frontiers in Blockchain 6, DOI 10.3389/fbloc.2023.1181820

Fuady S, Tarapore D, Soorati MD (2025) SubCDM: Collective decision-making with a swarm subset. arXiv preprint arXiv:250800467

Garg K, Panagou D (2021) Robust control barrier and control Lyapunov functions with fixed-time convergence guarantees. In: 2021 American Control Conference (ACC), IEEE, pp 2292–2297

Garnier S, Jost C, Jeanson R, Gautrais J, Asadpour M, Caprari G, Theraulaz G (2005) Aggregation behaviour as a source of collective decision in a group of cockroach-like-robots. In: Capcarrère MS, Freitas AA, Bentley PJ, Johnson CG, Timmis J (eds) Advances in Artificial Life, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 169–178

Gielis J, Shankar A, Prorok A (2022) A critical review of communications in multi-robot systems. Current robotics reports 3(4):213–225

Guerrero-Bonilla L, Prorok A, Kumar V (2017) Formations for resilient robot teams. IEEE Robotics and Automation Letters 2(2):841–848, DOI 10.1109/LRA.2017.2654550

Gunther NJ (1993) A simple capacity model of massively parallel transaction systems. In: CMG National Conf., pp 1035–1044

Gunther NJ (2007) Guerrilla Capacity Planning. Springer

Gustafson JL (1988) Reevaluating Amdahl's law. Commun ACM 31(5):532–533, DOI 10.1145/42411.42415

Hamann H (2013) Towards swarm calculus: Urn models of collective decisions and universal properties of swarm performance. Swarm Intelligence 7:145–172

Hamann H (2018) Swarm robotics: A formal approach, vol 221. Springer

Hamann H, Reina A (2022) Scalability in computing and robotics. IEEE Transactions on Computers 71(6):1453–1465, DOI 10.1109/TC.2021.3089044

Hayes AT (2002) How many robots? Group size and efficiency in collective search tasks. In: Asama H, Arai T, Fukuda T, Hasegawa T (eds) Distributed Autonomous Robotic Systems 5, Springer, Tokyo, pp 289–298, DOI 10.1007/978-4-431-65941-9_29

Hunt ER, Hauert S (2020) A checklist for safe robot swarms. Nature Machine

Intelligence 2(8):420–422, DOI 10.1038/s42256-020-0213-2

Iannini MLL, Dickman R (2016) Kinetic theory of vehicular traffic. American Journal of Physics 84(2):135–145

Ilan Y (2023) Making use of noise in biological systems. Progress in Biophysics and Molecular Biology 178:83–90, DOI 10.1016/j.pbiomolbio.2023.01.001

Jensen KH, Kim W, Holbrook NM, Bush JW (2013) Optimal concentrations in transport systems. Journal of the Royal Society Interface 10(83):20130,138

Jhawar J, Morris RG, Amith-Kumar UR, Danny Raj M, Rogers T, Rajendran H, Guttal V (2020) Noise-induced schooling of fish. Nature Physics 16(4):488–493, DOI 10.1038/s41567-020-0787-y

Ji M, Egerstedt M (2007) Distributed coordination control of multiagent systems while preserving connectedness. IEEE Transactions on Robotics 23(4):693–703, DOI 10.1109/TRO.2007.900638

Kan Z, Dani AP, Shea JM, Dixon WE (2010) Ensuring network connectivity during formation control using a decentralized navigation function. In: 2010 - MILCOM 2010 MILITARY COMMUNICATIONS CONFERENCE, pp 531–536, DOI 10.1109/MILCOM.2010.5680411

Kuckling J, Luckey R, Avrutin V, Vardy A, Reina A, Hamann H (2024) Do we run large-scale multi-robot systems on the edge? More evidence for two-phase performance in system size scaling. In: 2024 IEEE International Conference on Robotics and Automation (ICRA), IEEE

Lamport L (1977) Proving the correctness of multiprocess programs. IEEE Transactions on Software Engineering SE-3(2):125–143, DOI 10.1109/TSE.1977.229904

LeBlanc HJ, Zhang H, Sundaram S, Koutsoukos XD (2012) Consensus of multi-agent networks in the presence of adversaries using only local information. In: International Conference on High Confidence Networked Systems

LeBlanc HJ, Zhang H, Koutsoukos X, Sundaram S (2013) Resilient asymptotic consensus in robust networks. IEEE Journal on Selected Areas in Communications 31(4):766–781, DOI 10.1109/JSAC.2013.130413

Lee H, Panagou D (2025) Maintaining strong *r*-robustness in reconfigurable multi-robot networks using control barrier functions. In: 2025 IEEE International Conference on Robotics and Automation (ICRA), pp 1–7, DOI 10.1109/ICRA55743.2025.11127420

Lee S, Milner E, Hauert S (2022) A data-driven method for metric extraction to detect faults in robot swarms. IEEE Robotics and Automation Letters 7(4):10,746–10,753

Leinberger W, Karypis G, Kumar V, Biswas R (2000) Load balancing across near-homogeneous multi-resource servers. Technical report, Department of Computer Science and Engineering, University of Minnesota

Li JS, Alonso CA, Doyle JC (2021) Frontiers in scalable distributed control: Sls, mpc, and beyond. In: 2021 American Control Conference (ACC), IEEE, pp 2720–2725

Mayya S, Pierpaoli P, Egerstedt M (2019) Voluntary retreat for decentralized interference reduction in robot swarms. In: 2019 international conference on robotics and automation (ICRA), IEEE, pp 9667–9673

Mitra A, Richards JA, Bagchi S, Sundaram S (2019) Resilient distributed state estimation with mobile agents: overcoming byzantine adversaries, communication losses, and intermittent measurements. Autonomous Robots 43(3):743–768, DOI 10.1007/s10514-018-9813-7

Multerer T, Ganis A, Prechtel U, Miralles E, Meusling A, Mietzner J, Vossiek M, Loghi M, Ziegler V (2017) Low-cost jamming system against small drones using a 3d mimo radar based tracking. In: 2017 European Radar Conference (EURAD), pp 299–302, DOI 10.23919/EURAD.2017.8249206

Munoz A, Billsberry J, Ambrosini V (2022) Resilience, robustness, and antifragility: Towards an appreciation of distinct organizational responses to adversity. International Journal of Management Reviews 24(2):181–187, DOI 10.1111/ijmr.12289

Olfati-Saber R, Fax JA, Murray RM (2007) Consensus and cooperation in networked multi-agent systems. Proceedings of the IEEE 95(1):215–233, DOI 10.1109/JPROC.2006.887293

Pacheco A, Vos SD, Reina A, Dorigo M, Strobel V (2024) Securing federated learning in robot swarms using blockchain technology. 2409.01900

Pacheco A, Zhao H, Strobel V, Roukny T, Dudek G, Reina A, Dorigo M (2025) Swarm oracle: Trustless blockchain agreements through robot swarms. arXiv preprint arXiv:250915956

Park H, Hutchinson SA (2017) Fault-tolerant rendezvous of multirobot systems. IEEE Transactions on Robotics 33(3):565–582, DOI 10.1109/TRO.2017.2658604

Pease M, Shostak R, Lamport L (1980) Reaching agreement in the presence of faults. J ACM 27(2):228–234, DOI 10.1145/322186.322188

Pini G, Brutschy A, Birattari M, Dorigo M (2011) Task partitioning in swarms of robots: Reducing performance losses due to interference at shared resources. In: Cetto JA, Filipe J, Ferrier JL (eds) Informatics in Control Automation and Robotics, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 217–228

Prorok A, Malencia M, Carlone L, Sukhatme GS, Sadler BM, Kumar V (2021) Beyond robustness: A taxonomy of approaches towards resilient multi-robot systems. arXiv preprint arXiv:210912343

Rausch I, Reina A, Simoens P, Khaluf Y (2019) Coherent collective behaviour emerging from decentralised balancing of social feedback and noise. Swarm Intelligence 13(3–4):321–345, DOI 10.1007/s11721-019-00173-y

Reina A, Zakir R, De Masi G, Ferrante E (2023) Cross-inhibition leads to group consensus despite the presence of strongly opinionated minorities and asocial behaviour. Communications Physics 6(1):1–11, DOI 10.1038/s42005-023-01345-3

Rosenfeld A, Kaminka GA, Kraus S, Shehory O (2008) A study of mechanisms for improving robotic group performance. Artificial Intelligence 172(6-7):633–655

Şahin E (2004) Swarm robotics: From sources of inspiration to domains of application. In: International workshop on swarm robotics, Springer, pp 10–20

Saulnier K, Saldaña D, Prorok A, Pappas GJ, Kumar V (2017) Resilient flocking for mobile robot teams. IEEE Robotics and Automation Letters 2(2):1039–1046, DOI 10.1109/LRA.2017.2655142

Strobel V, Pacheco A, Dorigo M (2023) Robot swarms neutralize harmful Byzantine robots using a blockchain-based token economy. Science Robotics 8(79):eabm4636, DOI https://doi.org/10.1126/scirobotics.abm4636

Talamali MS, Bose T, Haire M, Xu X, Marshall JA, Reina A (2020) Sophisticated collective foraging with minimalist agents: A swarm robotics test. Swarm Intelligence 14(1):25–56

Taleb NN (2012) Antifragile: Things that Gain from Disorder. Random House

Tang Y, Lv Y, Zhou J, Ogorzałek M (2025) Resilient consensus in open multi-agent systems. IEEE Control Systems Letters 9:1261–1266, DOI 10.1109/LCSYS.2025.3580518

Tarapore D, Lima PU, Carneiro J, Christensen AL (2015) To err is robotic, to tolerate immunological: fault detection in multi-robot systems. Bioinspiration & Biomimetics 10:016,014, DOI 10.1088/1748-3190/10/1/016014

Tegling E, Sandberg H (2019) Noise-induced limitations to the scalability of distributed integral control. Systems & Control Letters 130:23–31, DOI 10.1016/j.sysconle.2019.06.005

Tews AD, Mataric MJ, Sukhatme GS (2003) A scalable approach to human-robot interaction. In: 2003 IEEE international conference on robotics and automation (Cat. No. 03CH37422), IEEE, vol 2, pp 1665–1670

Tsimring LS (2014) Noise in biology. Reports on Progress in Physics 77(2):026,601, DOI 10.1088/0034-4885/77/2/026601

Usevitch J, Panagou D (2019) Resilient leader-follower consensus to arbitrary reference values in time-varying graphs. IEEE Transactions on Automatic Control 65(4):1755–1762

Van Calck L, Pacheco A, Strobel V, Dorigo M, Reina A (2023) A blockchain-based information market to incentivise cooperation in swarms of self-interested robots. Scientific Reports 13:20,417, DOI 10.1038/s41598-023-46238-1

Vieira MA, Govindan R, Sukhatme GS (2009) Scalable and practical pursuit-evasion with networked robots. Intelligent Service Robotics 2(4):247–263

Wahby M, Petzold J, Eschke C, Schmickl T, Hamann H (2019) Collective change detection: Adaptivity to dynamic swarm densities and light conditions in robot swarms. In: Artificial life conference proceedings,

MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , pp 642–649

Winfield AFT (2000) Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots. In: Distributed Autonomous Robotic Systems, vol 4, Springer, Tokyo, Japan, pp 273–282, DOI 10.1007/978-4-431-67919-6_26

Xu Y, Deng G, Zhang T, Qiu H, Bao Y (2021) Novel denial-of-service attacks against cloud-based multi-robot systems. Information Sciences 576:329–344, DOI 10.1016/j.ins.2021.06.063

Yemini M, Nedić A, Goldsmith AJ, Gil S (2025) Resilient distributed optimization for multi-agent cyberphysical systems. IEEE Transactions on Automatic Control

Zakir R, Dorigo M, Reina A (2024) Miscommunication between robots can improve group accuracy in best-of-n decision-making. In: Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2024), IEEE Press, Piscataway, NJ, pp 9014–9021, DOI 10.1109/IROS58592.2024.10802464

Zhang H, Fata E, Sundaram S (2015) A notion of robustness in complex networks. IEEE Transactions on Control of Network Systems 2(3):310–320, DOI 10.1109/TCNS.2015.2413551

Zhao H, Pacheco A, Strobel V, Reina A, Liu X, Dudek G, Dorigo M (2023) A generic framework for byzantine-tolerant consensus achievement in robot swarms. In: Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2023), IEEE, pp 8839–8846, DOI 10.1109/IROS55552.2023.10341423

Cross-References

Task and Role Allocation; Reconfigurable Systems