How foundation models will revolutionize robot swarms

Volker Strobel[1*], Marco Dorigo[1], Mario Fritz[2]

[1]IRIDIA, Université Libre de Bruxelles, Brussels, Belgium

[2]CISPA Helmholtz Center for Information Security, St. Ingbert, Germany

* Corresponding author. Email: volker.strobel@ulb.be

ONE-SENTENCE SUMMARY

We explore how foundation models will revolutionize robot swarms' controller synthesis and decision-making capabilities.

ABSTRACT

Robot swarms are composed of many, typically simple, robots that accomplish complex tasks through local communication and decentralized coordination. Traditionally, robot controllers are designed prior to a mission using programming code. This process requires substantial development effort and limits the flexibility of the swarm. We discuss how onboard foundation models (FMs) could revolutionize this process through two complementary approaches. The first approach uses FMs as swarm designers to synthesize robot controllers and perform high-level planning. The second approach uses FMs as swarm operators to facilitate robot-robot collaboration and human-swarm interaction.

INTRODUCTION

Robot swarms are multi-robot systems that consist of many robots operating without a central unit of control (*1*). Their decentralized control makes them more flexible and scalable than other multi-robot systems (*2*), and, thanks to their large population size, they can potentially continue functioning even if some robots fail (*3*). Therefore, robot swarms could be ideal for applications that require rapid responses to unforeseen events, such as large-scale disasters (*4*).

Communication and interaction protocols of robot swarms are usually manually specified by experts during the design phase (*5*). This process can be intricate, time-consuming, and prone to errors. Even though there is research on automating this process (*6–9*), the developed approaches usually still require expert knowledge (such as formal specification languages), and the hard-coded controllers cannot flexibly react to unforeseen events (such as the presence of an injured person (*10*)) when the robot swarm is deployed.

Recent advances in generative artificial intelligence have led to the development of foundation models (FMs) (*11*): neural networks trained on very large datasets that are capable of processing and generating text, images, and videos. We believe that the above-mentioned issues in robot swarms could be addressed with FMs. The most widely used FMs are large language models (LLMs), which are trained on vast amounts of text and can process and generate language. These models have evolved into multimodal variants such as vision-language models (VLMs) that can process and reason about both images and text. Vision-language-action models (VLAs) extend these capabilities to interact with physical environments by generating control commands for robots.

FMs have been integrated into various technologies, including virtual assistants and search engines (*12*). Such integrated systems are called agentic AIs, as they can autonomously make decisions and perform tasks, such as sending an email or organizing travel (*13*). Recently, it has been proposed to let multiple LLM agents interact and negotiate (*14*). With the advent of LLMs, there have also been proposals to use FMs for synthesizing the controllers of single robots (*15*) as well as integrating them into multi-robot systems (*16,17,18*) in order to enhance their reasoning, planning, and collaboration capabilities. Integrating FMs into robotic systems has, in particular,

led to several new applications (*19*), including task planning (20), motion planning (*16,17*), and human-robot interaction (*15*).

This viewpoint explores how FMs could be integrated into a specific type of multi-robot system: robot swarms. There are several characteristics that distinguish robot swarms from other multi-robot systems (*2*). First, robot swarms are not centrally controlled, and their collective behavior arises exclusively from local interactions among the constituent robots. This makes the micro–macro link problem (*21*) (how to design the behavior of the individual robots to obtain the desired swarm behavior) central to the design of robot swarms. Second, robot swarms are envisioned to operate at much larger scales (hundreds or even thousands of robots) and with resource-constrained hardware. Finally, purely local communication capabilities could lead to long-lasting partitions (and therefore, inconsistent information) in the swarm. These characteristics and constraints call for specific solutions when integrating robot swarms with FMs.

We envision two main approaches for integrating FMs into robot swarms: using FMs as swarm designers to synthesize robot controllers and perform high-level planning (during and/or before deployment) based on natural-language instructions, drawings, diagrams, or demonstrations of intended behavior; and using FMs as swarm operators to incorporate a robot's sensor readings into an FM prompt, interpret this prompt in context, and use the resulting FM response to control the robot's actuators. These two approaches can be combined (together with traditional control) in comprehensive control architectures.

Existing automatic design approaches for robot swarms, such as evolutionary swarm robotics (*8*), modular design (*6*), multi-robot reinforcement learning (*22*), or formal methods (*23*), typically rely on expert knowledge, environmental or human feedback, and explicit objective functions and often require iterative refinement during offline design stages. In contrast, FM-enabled robot swarms could exhibit fundamentally new capabilities, particularly in scenarios that demand flexible responses to completely unforeseen events or intuitive forms of human-swarm interaction. In these settings, controller synthesis is not driven by optimization or learning from feedback, but instead leverages the ability of FMs to generate new code on the fly in a zero-shot manner, based on environmental observations or human inputs (including language, gestures, or drawings). In this viewpoint, we outline both the opportunities and the associated challenges of

such FM-enabled robot swarms. Interested readers can explore those with our LLM2Swarm software (*10*).

Foundation model-enabled robot swarms

In this section, we first present FMs as swarm designers, followed by a description of FMs as swarm operators. For both paradigms, we discuss the associated opportunities we foresee. We conclude with a discussion on how FM designers, FM operators, and traditional control could be combined in comprehensive control architectures.

Foundation models as swarm designers

One way to use FMs to design robot controllers is to employ them as designers for controller synthesis, that is, for the automated generation of code executed by a robot's controller, as well as for task planning. Such on-board FM designers could enable the synthesis of new controllers on-the-fly, for example, when the swarm realizes that it lacks certain functionalities—potentially leading to increased autonomy and flexibility in robot swarm behavior. Moreover, FMs could equip robots with the ability to generate collective plans that allocate robots to different tasks, and to adapt them in real time when unanticipated situations occur (*24*). In addition to on-board FMs, off-board FMs, such as powerful FMs using a cloud service, could be executed during the design stage of a robot swarm. These FMs could translate descriptions of desired collective swarm behaviors specified in natural language or in sketches/diagrams into code for individual robot controllers.

To implement FM designers, we propose a multi-step validation process. The first step is to verify that the synthesized code does not contain syntax errors. This is followed by the intricate task of validating that the generated code produces the expected collective swarm behavior. This step is especially challenging due to the micro–macro link problem (*21*): in swarm robotics, the lack of a central control unit restricts programming to individual controllers (the microscopic level), even though the goal is to obtain macroscopic (swarm-level) collective behavior. If the code is synthesized before deployment, the micro–macro link problem can be addressed by evaluating the resulting swarm behavior in simulation or with real robots. The analysis could be performed

through an evaluation metric (such as the maximum distance between the robots in an aggregation task), human observation, or by the video analysis capabilities of a VLM. The results of this evaluation are then fed into a subsequent iteration of FM prompting, for example, through text-based or schematic prompts. During a live robot swarm deployment, however, validating the control logic is more challenging. One approach might be to run simulations with the generated controllers onboard the robots during deployment and evaluate the resulting behavior through evaluation metrics or VLMs, and reiterate the controller design if necessary. Still, due to the simulation-to-reality gap, this approach cannot account for the wide range of possible real-world scenarios (*6*). Other approaches to evaluate a controller during deployment could include reinforcement learning to explicitly incorporate environmental feedback (such as task completion rates) to adapt and systematically compare controllers (*25*), as well as probabilistic model checking methods grounded in formal mathematical analysis (*26*).

In addition to logic verification, secure deployment of robot swarms also requires verifying whether a synthesized controller contains vulnerabilities—a common issue in FMs (*27*). For example, the controller could contain code that makes it easy for an attacker to take control of the swarm, or logic that enables sabotaging other robots. An initial approach towards mitigating security issues could rely on the Common Weakness Enumeration (CWE), a catalog of vulnerabilities across programming languages and hardware components (*28*). Addressing vulnerabilities specific to robotics could involve the development of LLMs fine-tuned for this task or the use of model checking.

Opportunities enabled by foundation models as swarm designers

Synthesizing controllers via FMs could substantially reduce the time required to design, test, and deploy robot controllers. In addition, it could allow non-experts to develop controllers, reducing costs and potentially expanding the community of users and developers, thereby promoting a more accessible and accelerated adoption of robot swarms.

Automated on-board controller synthesis could enable robot swarms to generate controllers on the fly during deployment. This capability is particularly valuable because experiments with robot swarms are typically conducted in controlled environments that cannot adequately capture

the full range of real-world variability. In contrast, deployment environments may be unstructured and unpredictable, making it almost impossible to anticipate all scenarios or preprogram all necessary capabilities into the swarm before deployment. FM designers may preserve the responsiveness of a robot swarm by carrying out controller synthesis and high-level planning during low-activity phases or prior to deployment.

When using FMs to create controllers, several design choices must be addressed. For instance, one must decide whether FMs should synthesize a complete controller encompassing all functionalities, or only specific components, such as movement logic, communication protocols, or decision-making modules. Another key design choice concerns the timing of controller synthesis. Controller components may be generated by an FM prior to deployment, allowing for human inspection and verification—at the potential cost of reduced autonomy during operation. In contrast, controllers that are synthesized or modified during deployment offer higher adaptability, but preclude direct human verification. A possible compromise is to employ pre-synthesized controller templates that can be adjusted during deployment by varying a limited set of parameters.

Foundation models as swarm operators

For implementing FM operators, we propose to feed the FMs' prompts with sensor information (including messages sent by humans or other robots). To generate these prompts, there is a need for suitable prompt templates for the required task—these can be written by humans or generated by FM designers. The FM responses must then be mapped to robot actions. These mappings can rely on high-level control programs (such as targeted navigation) or low-level control programs (such as setting wheel speed) that are activated when commands from the FM are received. These control programs need to be either pre-programmed or generated by FM designers. To activate the control programs, the on-board FMs need to be instructed to format their responses so that they adhere to the format required by the robot controllers. To this end, the FM prompts should include a robot's application programming interface (API) as part of the system prompt and thus select the desired function(s), such as `navigate(x,y)`. Previously

sent and received messages of robots (or summaries thereof) could be appended to new FM prompts to maintain context.

To implement FM operators, different model classes can be used depending on the application, including LLMs and VLMs. LLMs are well-suited for tasks that primarily involve text-based inputs and outputs, such as communication among robots or between robots and human operators. VLMs, in contrast, are more appropriate for tasks that require visual perception, such as interpreting camera images or video streams from a robot's environment. In heterogeneous robot swarms, the use of robot-specific VLMs would allow the same high-level command—for example, "search for victims"—to be executed differently depending on each robot's (potentially fine-tuned) FM, hardware capabilities, position, sensor inputs, and current task.

Opportunities enabled by foundation models as swarm operators

FM operators may allow for intelligent decision making, adaptive responses to changing environments, and enhanced communication capabilities. We envision two primary areas for FM operators: robot-robot collaboration and human-swarm interaction.

Robot-robot collaboration FMs have the potential to enhance the intelligence of collective behaviors. First, the integration of FMs could expand the reasoning capabilities of robots: pre-trained FMs already incorporate substantial world knowledge and support context-aware reasoning. FM-enabled robots could use these capabilities to respond to unforeseen events, potentially considering how humans would react in similar situations, and respect ethical principles and moral values. For example, if a robot swarm detects a collapsed building, the FMs could decide to let the robots search for potential victims and allocate tasks among the robots even if such a scenario was not explicitly programmed into the robots' controllers (*10*).

Moreover, FMs enable robots to interact through natural language, potentially enabling more sophisticated interactions than those supported by classical controllers. The use of natural language also allows humans to understand such interactions more naturally. This would make robot-robot interactions more transparent, thereby enhancing trustworthiness and accountability, as humans could audit the actions and decisions of the robot swarm. In conclusion,

FMs have the potential to substantially enhance robot-robot collaboration, leading to more effective, autonomous, resilient, and secure robot swarms, particularly in unknown or unstructured environments.

Human-swarm interaction The question of how to enable effective human-swarm interaction and collaboration while minimizing cognitive load remains an open research problem (*29*). FMs may substantially facilitate and advance human-robot and human-swarm interaction, especially in high-stress environments such as search-and-rescue operations (30). For example, chat interfaces or voice commands may allow human operators to communicate with the FM-enabled robots within a swarm, reducing cognitive load through the use of natural language. Such interactions could support both the retrieval of information about the swarm's collective state and the issuance of high-level commands (10,31).

Foundation models may process human commands in context—for example, by taking into account previous interactions, the swarm's current task, and objects detected in the environment—and ask clarifying questions when needed. Moreover, in missions involving nontechnical individuals, such as search-and-rescue operations, an FM could communicate with them in natural language (for example, by asking injured individuals about their condition) while also explaining the swarm's intentions and available actions (such as providing first aid or transporting individuals to a hospital). In addition, the use of VLMs in human-swarm interaction could facilitate learning from demonstration (*32*) and enhance, or even replace, existing automatic design approaches for robot swarms by relying on intuitive demonstrations rather than technical specifications.

Comprehensive control architectures

We envision robot swarms that combine FM designers, FM operators, and traditional control in comprehensive control architectures. Such swarms could balance the trade-offs between flexibility, responsiveness, and autonomy, depending on the robots' mission. High-level orchestration and code synthesis could be implemented using FM designers based on the current

needs of the robot swarm (such as adapting to a novel task or compensating for failed functionality), whereas low-level control and real-time decision-making could be managed by FM operators or traditional controllers to ensure flexible and fast reactions.

For example, during a search-and-rescue mission, each robot could use FM operators to interpret high-level human instructions, communicate with other robots, and make situation-dependent decisions. However, if a robot encounters a situation that requires a specialized navigation routine (such as climbing stairs) that it is not equipped for, it could trigger controller synthesis via FM designers to generate a suitable control routine tailored to the new situation. These generated robot controllers could be complemented by controllers coded by human designers.

For such comprehensive control architectures, it is important to translate an FM's response—which may include high-level commands (such as "search for victims")—into low-level instructions that robots can execute to control their actuators. It is worth noting that such translation is not always necessary, as FMs can operate purely at the communication level; however, certain scenarios may require direct control of a robot's motion hardware (including wheels or rotors). Future research should explore how pre-programmed behavior can be controlled by FM responses, when and how frequently to query the FMs, and how to executegenerated multi-step plans. An example workflow integrating FM designers and FM operators is illustrated in Figure 1.

CHALLENGES

Integrating FMs into swarm robotics enables new approaches to deployment and control, but it also introduces several challenges. These include addressing the micro–macro link problem; managing the transition from controller-based design to specification-driven design; developing training methods that incorporate the robots' perspective; overcoming hardware limitations, sustaining scalability, and addressing partitioning; and preserving controllability, security, and safety. In the following sections, we discuss each of these challenges.

Micro-macro link problem

A fundamental challenge in swarm robotics research is the micro–macro link problem: a robot swarm designer aims to achieve a desired collective (swarm-level) behavior, yet can only program the controllers of individual robots (*21*). In FM-enabled robot swarms, this micro–macro link problem arises both for FM designers and FM operators.

With FM designers, using FMs out of the box—that is, without explicitly accounting for the micro–macro link problem—may lead to collective behaviors that deviate from expectations, as FMs are typically not trained to reason about the relationship between individual behaviors and emergent collective outcomes. Introducing a dedicated logic validation step into the FM design process could therefore help address this issue. For example, when a VLM is used to analyze videos of collective behavior, it could detect deviations from the intended behavior and indicate that further design iterations are required.

For FM operators, addressing the micro–macro link problem is likely to be more challenging, as the FM inputs and outputs are less predictable. One possible approach is to develop FMs tailored to swarm robotics (for example, by fine-tuning existing FMs using feedback from simulations) to better address the micro–macro link problem.

Transition from controller-based to specification-driven design

A potential issue with FM-enabled robot swarms has already been observed in prior work on the automatic generation of robot swarm controllers: such approaches are often not fully automatic. Rather than eliminating manual design effort, they tend to shift it from the explicit programming of controllers to the selection of appropriate conditions that yield desirable outcomes (*5*). For example, in evolutionary swarm robotics, the choice of fitness function and hyperparameters is often critical to success.

A similar shift is likely to occur in FM-enabled robot swarms. Instead of manually writing controllers, human designers might need to craft effective prompts and specify suitable interaction modalities that enable FMs to produce the desired behaviors. To support this process

and alleviate the burden on human designers, we advocate developing design guidelines and conducting systematic comparative studies of different FMs and prompting strategies.

Human perspective vs. robot perspective

The majority of state-of-the-art FMs are not trained on data that reflects how robots perceive the world; instead, they are typically trained on data generated by humans (*33*). As a result, it is necessary to investigate the use of FMs in robotic contexts to assess their capabilities and determine whether substantial differences exist between the perceptual and generative requirements of humans and robots. If such differences are identified, FMs could be trained on robot-generated data or fine-tuned to better capture robot-centric perspectives.

At the same time, differences between human and robot perception may also prove advantageous. Adopting a more human-like perspective could help align robot behavior with ethical standards and social norms, thereby improving public acceptance of and trust in robot swarm, an important factor for successful human-robot interaction (*34,35*). Moreover, non-expert users who may encounter robot swarms incidentally (for example, during environmental monitoring) may be more inclined to trust robot swarms that exhibit human-like behavior and are able to communicate using natural language.

Hardware limitations, scalability, and partitioning

To ensure the autonomy of FM-enabled robot swarms, FMs should be embedded within each robot's hardware, as relying on external infrastructure may be infeasible in many deployment scenarios—such as underwater missions, underground mining, or space exploration—and can introduce communication bottlenecks. Local execution also ensures that data remain onboard, thereby preserving privacy and facilitating compliance with regulations such as the General Data Protection Regulation (GDPR). However, executing FMs locally remains challenging, as robots in swarms typically have limited computational resources. However, recent advances in hardware design are beginning to relax this constraint (36), and current robotics-specific hardware platforms already deliver acceptable performance when running state-of-the-art LLMs (*37*).

Latency can be mitigated by using lightweight FMs for rapid responses (potentially involving only a small subset of actuators) and more powerful FMs for long-term planning—analogous to human fast-and-slow thinking (*38*). Executing large FMs on board edge devices is a very active research field, and several techniques have been proposed to support this, including fine-tuning (*39*), quantization (40–*42*), pruning (*43,44*), distillation (*45*), and flash attention (*46*). These techniques often reduce storage and memory requirements at the cost of decreased accuracy and flexibility. Continued advances in both hardware miniaturization (for example, of microprocessors or graphical processing units) and software optimization will likely enable the execution of increasingly powerful FMs on robots operating within swarms.

Maintaining or improving swarm performance as the number of robots increases poses an additional challenge for FM-enabled robot swarms. As the swarm scales, the number of potential inter-robot interactions increases, and these interactions may need to be incorporated into prompts. Consequently, prompts can become increasingly long, potentially exceeding token limits or introducing delays. Moreover, FMs often favor verbosity over conciseness (*47*), which can further reduce scalability due to high communication overhead.

One way to mitigate the exponential growth of message length is to restrict each robot's communication radius, thereby limiting the number of exchanged messages. An alternative strategy is to develop efficient methods for storing and summarizing the history of inter-robot interactions, for example, via retrieval-augmented generation (RAG) (*48*).

During deployment, motion and communication limitations can partition a robot swarm into disconnected subswarms, thereby disrupting information propagation. As a result, messages exchanged within each subswarm may diverge substantially, and upon reconnection, these divergent information states may lead to incompatibilities between previously isolated subswarms. Partitioning can also induce heterogeneous behavior, as the lack of communication prevents synchronization between robots running independent FM instances. Depending on the task, such heterogeneity may be beneficial by fostering exploratory solution finding, or detrimental, as it may result in inconsistent behavior and degraded performance. When high behavioral consistency is desired, several strategies can be employed, including using identical prompts across all robots, decreasing the FM temperature to reduce stochasticity, introducing

consensus mechanisms within the robot swarm (such as voting on the next action), or constraining the set of functions that FMs are allowed to call. Beyond managing its effects, partitioning may also be mitigated by reducing its likelihood. For instance, FMs could be used to directly control robot actuators (such as drone rotors) in ways that help maintain connectivity. In addition, FMs could monitor communication patterns and trigger control logic to locate and rejoin other subswarms when prolonged disconnections are detected. Nevertheless, additional mechanisms (such as RAG-based summaries of robot interactions) are required to support both the autonomous operation of subswarms and the reconciliation of their locally evolved states after reconnection.

Controllability, security, and safety

A central objective of swarm robotics research is to increase swarm autonomy, and thus decrease reliance on external control or infrastructure (*49*). We argue that FMs have the potential to contribute substantially to this goal. However, as autonomy increases, ensuring controllability, security, and safety of the swarm becomes essential, so that robots perform their intended tasks while remaining compliant with applicable rules and regulations (50). On the one hand, FM integration may enable robot swarms to autonomously execute ethically responsible actions that consider how humans might react in a given situation (*51*). On the other hand, the probabilistic nature of FM outputs raises fundamental questions about the tradeoff between autonomy and controllability in autonomous systems (*52*). In particular, there is no guarantee that the resulting swarm behavior will align with expectations. To mitigate this issue, we have proposed a logic validation procedure for FM designers to improve the robustness, reliability, and predictability of synthesized controllers.

With FM operators, additional security risks arise, analogous to other application domains involving FMs (*12*). These include the potential for malicious users to extract sensitive data from robots through carefully crafted prompts, as well as safety and reliability issues caused by hallucinated outputs (*53*). For instance, an FM may hallucinate the existence of functions in a robot's API, leading it to attempt to control a nonexistent actuator. although such code hallucinations can be mitigated to some extent by explicitly specifying the robot’s API in the

system prompt, they are not entirely avoidable. They are, however, relatively easy to detect, as they typically result in execution errors, and thus require adequate handling and fallback strategies. Another important concern is whether a user—or a compromised robot—could reprogram other robots within the swarm via prompt injection attacks.

FM-enabled robot swarms could ultimately give rise to self-evolving systems, often referred to as open-ended systems, that continuously interact with their environment and autonomously generate new behaviors. Recent work has highlighted that ensuring the safety of such systems is particularly challenging (*54*). In the worst case, FM-enabled robot swarms could lead to a loss of human control and to misalignment between swarm objectives and human goals. This prospect underscores the need for sustained and systematic research in this area. We therefore advocate research approaches that balance the exploration of the potential of FM-enabled robot swarms with a rigorous investigation of their associated risks. Potential risk-mitigation strategies include sandboxing (limiting FMs' access to specific hardware or software components; for example, preventing them from disabling the communication system), allowlisting (restricting FMs to a predefined set of commands), monitoring mechanisms (such as human-in-the-loop supervision or anomaly detection), and redundancy (for example, deploying multiple FMs per robot). Ideally, these measures should be embedded within a comprehensive ethics-by-design framework.

CONCLUSIONS

In this viewpoint, we have discussed the potential of FM-enabled robot swarms, which may help realize the long-standing promise of adaptive and flexible robot swarm behavior in unforeseen situations. Crucially, such situations may involve interactions with laypersons, who need intuitive interfaces to communicate with robot swarms—for example, to ask about its planned actions, provide feedback, or issue control commands. Foundation models are one of the most promising approaches for providing such interfaces, potentially facilitating broader public acceptance of early real-world deployments.

At the same time, realizing this potential raises important concerns. Delegating control to AI models may reduce the controllability of a robot swarm, particularly in safety-critical or open-ended environments. We therefore advocate research efforts that balance the exploration of the

potential of FM-enabled robot swarms with careful investigation of their associated risks. Proactively addressing security concerns will help mitigate these risks and lay the foundation for adaptive, secure, and reliable real-world deployment.

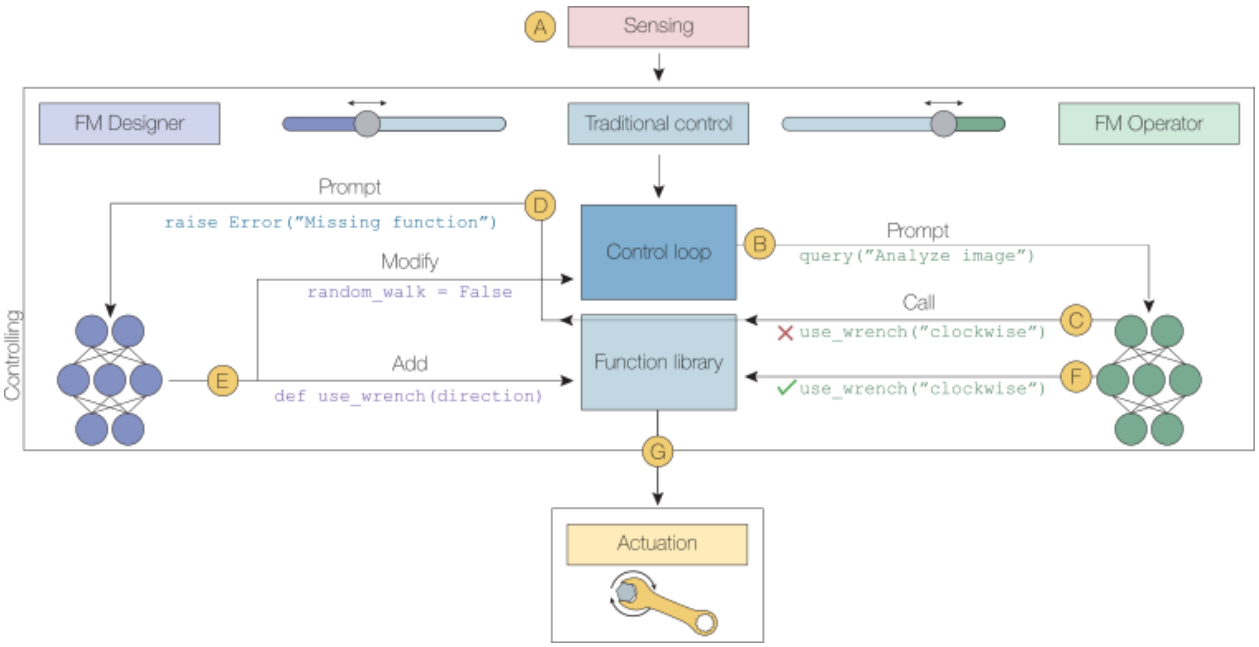

Figure 1: Illustration of a comprehensive controller architecture for FM-enabled robot swarms. As indicated by the sliders, the three main components (FM designer, traditional control, and FM operator) can contribute in different proportions to the control of an individual robot. The illustrated workflow (steps **A**–**G**) exemplifies the interactions among these components: **(A)** a robot uses its camera to capture an image of a loose bolt; **(B)** the control loop queries the FM operator to analyze the image; **(C)** the FM operator identifies the issue and attempts to invoke a manipulation function, which **(D)** fails because the function is not available in the robot's function library; **(E)** the FM designer synthesizes the required control function, disseminates it within the swarm, and modifies the robot's control state by disabling its random-walk behavior; **(F)** the FM operator retries the invocation of the manipulation function; and **(G)** the newly added function is successfully executed, resulting in coordinated actuation.

References and Notes

1. H. Hamann, *Swarm Robotics: A Formal Approach* (Springer) (2018), doi:10.1007/978-3-319 - 74528-2.

2. E. Şahin, Swarm robotics: From sources of inspiration to domains of application, in *Proceedings of the 2004 International Conference on Swarm Robotics (SAB'04)* (Springer, Berlin/Heidelberg) (2004), pp. 10–20, doi:10.1007/978-3-540-30552-1 2.

3. A. F. T. Winfield, J. Nembrini, Safety in numbers: Fault tolerance in robot swarms. *International Journal on Modelling Identification and Control* **1** (1), 30–37 (2006), doi:10.1504/IJMIC.20 06.008645.

4. G.-Z. Yang, *et al.*, The grand challenges of *Science Robotics*. *Science Robotics* **3** (14), eaar7650 (2018), doi:10.1126/scirobotics.aar7650.

5. J. Kuckling, Recent trends in robot learning and evolution for swarm robotics. *Frontiers in Robotics and AI* **10** (2023), doi:10.3389/frobt.2023.1134841.

6. G. Francesca, M. Brambilla, A. Brutschy, V. Trianni, M. Birattari, AutoMoDe: A novel approach to the automatic design of control software for robot swarms. *Swarm Intelligence* **8** (2), 89–112 (2014), doi:10.1007/s11721-014-0092-4.

7. D. Garzon Ramos, M. Birattari, Automatically designing robot swarms in environments populated by other robots: An experiment in robot shepherding, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2024)* (IEEE Press) (2024), pp. 12240–12247, doi:10.1109/ICRA57147.2024.10611309.

8. V. Trianni, E. Tuci, C. Ampatzis, M. Dorigo, Evolutionary swarm robotics: A theoretical and methodological itinerary from individual neuro-controllers to collective behaviours, in *The horizons of evolutionary robotics* (MIT Press), vol. 153 (2014).

9. E. Tolstaya, *et al.*, Learning Decentralized Controllers for Robot Swarms with Graph Neural Networks, in *Proceedings of the Conference on Robot Learning* (PMLR), vol. 100 of *Proceedings of Machine Learning Research* (2020), pp. 671–682.

10. V. Strobel, M. Dorigo, M. Fritz, LLM2Swarm: Robot Swarms that Responsively Reason, Plan, and Collaborate through LLMs, in *NeurIPS 2024 Workshop on Open-World Agents (OWA2024)* (2024), https://arxiv.org/abs/2410.11387.

11. R. Bommasani, *et al.*, On the Opportunities and Risks of Foundation Models (2022), https://arxiv.org/abs/2108.07258.

12. K. Greshake, *et al.*, Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection, in *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, AISec '23 (ACM) (2023), pp. 79—90, doi:10.1145/3605764.3623985.

13. Y. Shavit, *et al.*, *Practices for governing agentic AI systems*, Tech. rep., OpenAI (2023), https://cdn.openai.com/papers/practices-for-governing-agentic-ai-systems.pdf, Accessed April 23, 2025.

14. S. Abdelnabi, A. Gomaa, S. Sivaprasad, L. Schonherr, M. Fritz, Cooperation, Competition, and Maliciousness: LLM-Stakeholders Interactive Negotiation, in *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS 2024)* (Curran Associates Inc., Red Hook, NY, USA) (2024), pp. 83548–83599.

15. S. Vemprala, R. Bonatti, A. Bucker, A. Kapoor, *ChatGPT for Robotics: Design Principles and Model Abilities*, Tech. Rep. MSR-TR-2023-8, Microsoft (2023), https://www.microsoft.com/en-us/research/publication/chatgpt-for-robotics-design-principles-and-model-abilities/.

16. Y. Chen, J. Arkin, Y. Zhang, N. Roy, C. Fan, Scalable Multi-Robot Collaboration with Large Language Models: Centralized or Decentralized Systems?, in *Proceedings of the IEEE*

International Conference on Robotics and Automation (ICRA 2024) (IEEE Press) (2024), pp. 4311–4317, doi:10.1109/ICRA57147.2024.10610676.

17. Z. Mandi, S. Jain, S. Song, RoCo: Dialectic Multi-Robot Collaboration with Large Language Models, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2024)* (IEEE Press) (2024), pp. 286–299, doi:10.1109/ICRA57147.2024.10610855.

18. W. Ji et al., GenSwarm: Scalable Multi-Robot Code-Policy Generation and Deployment via Language Models. npj Robotics 4, 5 (2026), doi:10.1038/s44182-025-00065-w.

19. Y. Kim, *et al.*, A survey on integration of large language models with intelligent robots. *Intelligent Service Robotics* **17** (5), 1091–1107 (2024), doi:10.1007/s11370-024-00550-5.

20. Y. Ding, *et al.*, Integrating action knowledge and LLMs for task planning and situation handling in open worlds. *Autonomous Robots* **47** (8), 981–997 (2023).

21. H. Hamann, H. Worn, A framework of space–time continuous models for algorithm design in¨ swarm robotics. *Swarm Intelligence* **2** (2-4), 209–239 (2008).

22. L. C. Garaffa, M. Basso, A. A. Konzen, E. P. de Freitas, Reinforcement learning for mobile robotics exploration: A survey. *IEEE Transactions on Neural Networks and Learning Systems* **34** (8), 3796–3810 (2021).

23. M. A. Hsieh, V. Kumar, L. Chaimowicz, Decentralized controllers for shape generation with robotic swarms. *Robotica* **26** (5), 691–701 (2008).

24. C. H. Song, *et al.*, LLM-Planner: Few-shot grounded planning for embodied agents with large language models, in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (IEEE Press) (2023), pp. 2998–3009.

25. M. Blais, M. Akhloufi, Reinforcement learning for swarm robotics: An overview of applications, algorithms and simulators. *Cognitive Robotics* **3**, 226–256 (2023), doi:10.1016/j.cogr.2 023.07.004.

26. S. Konur, C. Dixon, M. Fisher, Analysing robot swarm behaviour via probabilistic model checking. *Robotics and Autonomous Systems* **60** (2), 199–213 (2012), doi:10.1016/j.robot.2011.10.005.

27. H. Hajipour, K. Hassler, T. Holz, L. Schonherr, M. Fritz, CodeLMSec Benchmark: Systematically Evaluating and Finding Security Vulnerabilities in Black-Box Code Language Models, in *Proceedings of the 2nd IEEE Conference on Secure and Trustworthy Machine Learning (SATML 2024)* (2024).

28. MITRE Corporation, Common Weakness Enumeration (CWE), https://cwe.mitre.org/ (2025), accessed: April 28, 2025.

29. A. Paas, E. B. J. Coffey, G. Beltrame, D. St-Onge, Towards evaluating the impact of swarm robotic control strategy on operators' cognitive load, in *Proceedings of the 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN 2022)* (IEEE Press) (2022), pp. 217–223, doi:10.1109/RO-MAN53752.2022.9900763.

30. K. Ji, X. Hu, X. Zhang, J. Chen, An LLM-based Framework for Human-Swarm Teaming Cognition in Disaster Search and Rescue, in *NeurIPS 2025 Workshop on LLM Persona Modeling (PersonaLLM)* (2025).

31. D. S. Brown, M. A. Goodrich, S.-Y. Jung, S. Kerman, Two invariants of human-swarm interaction. *Journal of Human-Robot Interaction* **5** (1), 1–31 (2016), doi:10.5898/JHRI.5.1.Brown.

32. I. Gharbi, J. Kuckling, D. G. Ramos, M. Birattari, Show me what you want: Inverse reinforcement learning to automatically design robot swarms by demonstration, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2023)* (2023), pp. 5063–5070.

33. P. Villalobos, *et al.*, Will We Run Out of Data? Limits of LLM Scaling Based on Human-Generated Data, in *Proceedings of the 41st International Conference on Machine Learning*

(ICML) (PMLR), vol. 235 of *Proceedings of Machine Learning Research* (2024), pp. 49523–49544.

34. J. Wilson, S. Hauert, Search space illumination of robot swarm parameters for trustworthy interaction, in *International Symposium on Distributed Autonomous Robotic Systems (DARS 2022)* (Springer) (2022), pp. 173–186.

35. J. B. Lyons, *et al.*, Examining the human-centred challenges of human–swarm interaction. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **383** (2289), 20240140 (2025), doi:10.1098/rsta.2024.0140.

36. S. Jones, M. Studley, S. Hauert, A. F. T. Winfield, A Two Teraflop Swarm. *Frontiers in Robotics and AI* **5** (2018), doi:10.3389/frobt.2018.00011.

37. S. Maheshwari, C. Su, *Technical Blog (Robotics) – NVIDIA JetPack 6.2 Brings Super Mode to NVIDIA Jetson Orin Nano and Jetson Orin NX Modules*, Tech. rep., NVIDIA (2025), https://developer.nvidia.com/blog/nvidia-jetpack-6-2-brings-super-mode-to-nvidia-jetson-orin-nano-and-jetson-orin-nx-modules/, accessed: April 24, 2025.

38. D. Kahneman, *Thinking, Fast and Slow* (Farrar, Straus and Giroux, New York, NY) (2011).

39. J. Howard, S. Ruder, Universal Language Model Fine-tuning for Text Classification, in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)* (ACL) (2018), pp. 328–339, doi:10.18653/v1/P18-1031.

40. T. Dettmers, M. Lewis, Y. Belkada, L. Zettlemoyer, GPT3.int8(): 8-bit Matrix Multiplication for Transformers at Scale, in *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)* (Curran Associates, Inc.) (2022), pp. 30318–30332.

41. J. Lin, *et al.*, AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration, in *Proceedings of the 7th Annual Conference on Machine Learning and Systems (MLSys 2024)* (2024), pp. 87–100.

42. Z. Liu, *et al.*, KIVI : A Tuning-Free Asymmetric 2bit Quantization for KVCache, in *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)* (PMLR), vol. 235 of *Proceedings of Machine Learning Research* (2024), pp. 32332–32344.

43. E. Frantar, D. Alistarh, SparseGPT: Massive language models can be accurately pruned in one-shot, in *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)* (JMLR) (2023), pp. 10323–10337.

44. X. Ma, G. Fang, X. Wang, LLM-Pruner: On the Structural Pruning of Large Language Models, in *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)* (Curran Associates, Inc.) (2023), pp. 21702–21720.

45. G. Hinton, O. Vinyals, J. Dean, Distilling the Knowledge in a Neural Network, in *NIPS 2015 Deep Learning and Representation Learning Workshop* (2015).

46. T. Dao, D. Fu, S. Ermon, A. Rudra, C. Ré, FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness, in *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)* (Curran Associates, Inc.) (2022), pp. 16344–16359.

47. K. Saito, A. Wachi, K. Wataoka, Y. Akimoto, Verbosity Bias in Preference Labeling by Large Language Models, in *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following* (2023).

48. P. Lewis, *et al.*, Retrieval-augmented generation for knowledge-intensive NLP tasks, in *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS 2020)* (Curran Associates Inc., Red Hook, NY, USA) (2020).

49. M. Brambilla, E. Ferrante, M. Birattari, M. Dorigo, Swarm Robotics: A Review from the Swarm Engineering Perspective. *Swarm Intelligence* **7** (1), 1–41 (2013), doi:10.1007/s11721 -012-0075-2.

50. M. Dorigo, A. Pacheco, A. Reina, V. Strobel, Blockchain technology for mobile multi-robot systems. *Nature Reviews Electrical Engineering* **1** (4), 264–274 (2024), doi:10.1038/s44287 - 024-00034-9.

51. M. Graves, Moral Attention Is All You Need. *Theology and Science* **23** (2), 241–248 (2025), doi:10.1080/14746700.2025.2472118.

52. A. F. Winfield, M. Swana, J. Ives, S. Hauert, On the ethical governance of swarm robotic systems in the real world. *Philosophical Transactions of the Royal Society A* **383** (2289), 20240142 (2025).

53. Z. Ji, *et al.*, Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys* **55** (12) (2023), doi:10.1145/3571730.

54. I. Sheth, J. Wehner, S. Abdelnabi, R. Binkyte, M. Fritz, Safety is Essential for Responsible Open-Ended Systems, in *ICLR Workshop on Scaling Self-Improving Foundation Models without Human Supervision* (2025).

Acknowledgements: The authors thank Alessandro Nazzari for providing helpful comments on the manuscript.

Funding: V. S. and M. D. acknowledge support from the Belgian F.R.S.-FNRS, of which they are a Postdoctoral Researcher and a Research Director, respectively. In addition, V. S. acknowledges the Helmholtz Information & Data Science Academy (HIDA) for providing financial support enabling a short-term research stay at CISPA Helmholtz Center for Information Security. This work was also partially funded by ELSA - European Lighthouse on Secure and Safe AI funded by the European Union under grant agreement number 101070617, as well as the German Federal Ministry of Education and Research (BMBF) under the grant AIgenCY (16KIS2012).

Author contributions: V. S. drafted the initial version of the manuscript and structured the core arguments. All authors contributed to the development of the ideas, provided critical feedback, and revised the manuscript.

Competing interests: There are no competing interests to declare.