The automatic off-line design of robot swarms: recent advances and perspectives

David Garzón Ramos, Darko Bozhinoski, Gianpiero Francesca, Lorenzo Garattoni, Ken Hasselmann, Miquel Kegeleirs, Jonas Kuckling, Antoine Ligot, Fernando J. Mendiburu, Federico Pagnozzi, Muhammad Salman, Thomas Stützle, and Mauro Birattari

IRIDIA, Université Libre de Bruxelles

We sketch some recent advances in the automatic off-line design [1] of robot swarms and we discuss our perspective. Our vision is that automatic off-line design will play a major role in the development of swarm robotics and in its applications. Recent discussions have foreseen the milestones that would drive the advance of swarm robotics in the next decade [2, 3]: (i) the appearance of novel robot platforms that can operate in unstructured and dynamic environments [4]; (ii) the development of new methodologies for the design of collective behaviors [5]; (iii) new opportunities to exploit emergence [6]; and (iv) the shift of focus towards applications suited for large groups of coordinated robots [7]—e.g., precision agriculture, ecological monitoring, and city cleaning. Although the future is promising, at present most achievements in swarm robotics research still occur under controlled laboratory conditions [8].

There is a need for robust design methodologies that will enable the transition from laboratory experiments to real-world applications [7, 9]. Today, many researchers promote the adoption of engineering principles in the realization of robot swarms [10]. Yet, no general methodology exists to design the behavior of an individual robot so that a desired collective behavior is obtained. Traditionally, the design process has an iterative nature and is based on trial and error: a human designer manually refines the control software of the individual robots until the desired collective behavior emerges [11]. This procedure is costly, time-consuming, and does not guarantee that the results are reproducible.

Optimization-based design is an alternative approach to the design of collective behaviors for robot swarms [12]. In this approach, an optimization algorithm explores possible instances of control software for the robots and selects the one that maximizes performance on the specific mission at hand—according to a given performance metric. Optimization-based methods can be categorized with respect to different criteria. Common classifications divide them into (i) on-line and off-line methods, and into (ii) semi-automatic and (fully) automatic ones. Although, these classifications are not to be considered as strict—indeed, hybrids exist—they are convenient to appreciate the relative merits of different methods and to properly define expectations on their performance [12]. On-line methods produce control software directly on the robots, while the latter execute the mission; conversely, off-line methods produce control software before the robots are deployed, typically using simulation. In semi-automatic methods, a human designer operates an optimization algorithm that serves as their primary design tool; contrarily, automatic methods do not require any human intervention during the design process.

On-line (both automatic and semi-automatic) and off-line semi-automatic methods will definitely contribute to the advancement and application of swarm robotics. Still, it is our contention that automatic off-line methods will play the most central role. Indeed, they are of general applicability and have the potential to realize robot swarms quickly, with reduced effort, while ensuring sufficiently good performance. On-line methods appear ideal to refine existing solutions—limitations exists that restrict their general applicability. For example, they can explore a relatively small search space, could produce sub-optimal control software that could damage robots and environment (notably in the early phases of the design process), and are applicable only when the robots can assess their own collective performance. Similarly, semi-automatic methods are a useful and promising tool but are labor-intensive. They require the attention of a skilled operator that analyzes the outcome of an optimization process, adjusts parameters and amends the so-called fitness function by adding/removing terms to penalize/promote the emergence of behavioral features, before iterating the process. The need for a human operator is a limitation when one is called to design/refine control software for robot swarms under tight time and cost constraints. Although we believe that automatic off-line design addresses the more general design problem, in the long term, we expect that on-line, off-line, semi-automatic and automatic methods could coexists [12]—hybrid methods could be particularly appealing and appropriate in many applications.

Whereas we deem it the most promising approach, automatic off-line design is not itself free from presenting challenges and open issues. The main problem faced in automatic off-line design (and also in semi-automatic off-line design) is the so-called reality gap [13], that is, the differences between reality and simulation models on which the off-line optimization is based. Due to the reality gap, control software designed off-line typically experiences an important performance drop when ported to the real robots. Even worse, the drop is method-dependent with some design method being more intrinsically robust than others. This has implications on how instances of control software should be assessed and eventually selected before being deployed in reality [14].

Automatic off-line design is currently an early-stage technology that has been mostly demonstrated with laboratory experiments [15]. Important scientific and engineering questions need to be addressed before reaching mature methods that are ready for real-world application. Can we design effective and reliable robot swarms via automatic off-line design? What are the components that influence the effectiveness of a method? How can we conceive a method that is effective? Given a class of missions, which is the most appropriate design method? Which features of a mission make it more or less hard to be tackled? To what extent a design method is robust to the reality gap? What can we do to improve the robustness of a method? How can we characterize and specify a mission or a class of missions that a swarm must perform? To what extent an automatic design method can be ported to other design problems, and vice versa?

Recent advances in automatic off-line design belong mostly in two main approaches: (i) neuro-evolution [16, 17]; and (ii) automatic modular design

(AutoMoDe) [18]. Neuro-evolution is the traditional approach to the automatic design of collective behaviors for robot swarms: each robot is controlled by an artificial neural network whose parameters (and possibly the architecture) are obtained via artificial evolution. As an alternative to neuro-evolution, a few methods have been recently proposed within the AutoMoDe approach [18, 19]. In these methods, the control software of the robots is produced via an optimization-based process that fine-tunes pre-existing software modules and combines them into a modular architecture such as a probabilistic finite-state machine or a behavior tree. The software modules can be produced manually or with the assistance of optimization processes—for example, via evolutionary computation [20]. A number of studies have shown that AutoMoDe is less prone than neuro-evolution to the effects of the reality gap and tend to produce control software that eventually performs better once ported from simulation to reality [18, 19]. For a review of automatic off-line design methods, see [1].

Author Contributions

The manuscript was drafted by DGR and revised by MB. All authors contributed to the elaboration of the ideas presented in the paper, read the text, and provided comments. The research described in the paper was conceived and directed by MB.

Funding

The project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (*DEMIURGE* Project, grant agreement No 681872) and from Belgium's Wallonia-Brussels Federation through the ARC Advanced Project *GbO–Guaranteed by Optimization*. DGR acknowledges support from the *Colombian Ministry of Science, Technology and Innovation* – Minciencias. DB, MK, JK, TS, and MB acknowledge support from the Belgian *Fonds de la Recherche Scientifique* – FNRS.

References

[1] Gianpiero Francesca and Mauro Birattari. Automatic design of robot swarms: achievements and challenges. *Frontiers in Robotics and AI*, 3(29):1–9, 2016.

[2] Guang-Zhong Yang, Jim Bellingham, Pierre E. Dupont, Peer Fischer, Luciano Floridi, Robert Full, Neil Jacobstein, Vijay Kumar, Marcia McNutt, Robert Merrifield, Bradley J. Nelson, Brian Scassellati, Mariarosaria Taddeo, Russell Taylor, Manuela Veloso, Zhong Lin Wang, and Robert Wood. The grand challenges of Science Robotics. *Science Robotics*, 3(14):eaar7650, 2018.

[3] Marco Dorigo, Guy Theraulaz, and Vito Trianni. Reflections on the future of swarm robotics. *Science Robotics*, 5:eabe4385, 2020.

[4] Florian Berlinger, Melvin Gauci, and Radhika Nagpal. Implicit coordination for 3d underwater collective behaviors in a fish-inspired robot swarm. *Science Robotics*, 6(50):eabd8668, 2021.

[5] Nithin Mathews, Anders Lyhne Christensen, Rehan O'Grady, Francesco Mondada, and Marco Dorigo. Mergeable nervous systems for robots. *Nature Communications*, 8(1):439, 2017.

[6] Lorenzo Garattoni and Mauro Birattari. Autonomous task sequencing in a robot swarm. *Science Robotics*, 3(20):eaat0430, 2018.

[7] Edmund R. Hunt and Sabine Hauert. A checklist for safe robot swarms. *Nature Machine Intelligence*, 2:420–422, 2020.

[8] Heiko Hamann, Melanie Schranz, Wilfried Elmenreich, Vito Trianni, Carlo Pinciroli, Nicolas Bredeche, and Eliseo Ferrante. Editorial: designing self-organization in the physical realm. *Frontiers in Robotics and AI*, 7:164, 2020.

[9] Simon Jones, Emma Milner, Mahesh Sooriyabandara, and Sabine Hauert. Distributed situational awareness in robot swarms. *Advanced Intelligent Systems*, 2(11):2000110, 2020.

[10] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.

[11] David St-Onge, Vivek Shankar Varadharajan, Švogor Ivan, and Giovanni Beltrame. From design to deployment: decentralized coordination of heterogeneous robotic teams. *Frontiers in Robotics and AI*, 7:51, 2020.

[12] Mauro Birattari, Antoine Ligot, and Ken Hasselmann. Disentangling automatic and semi-automatic approaches to the optimization-based design of control software for robot swarms. *Nature Machine Intelligence*, 2(9):494–499, 2020.

[13] Nick Jakobi, Phil Husbands, and Inman Harvey. Noise and the reality gap: the use of simulation in evolutionary robotics. In F. Morán, A. Moreno, Juan J. Merelo, and P. Chacón, editors, *Advances in Artificial Life: Third european conference on artificial life*, volume 929 of *Lecture Notes in Artificial Intelligence*, pages 704–720, Berlin, Germany, 1995. Springer.

[14] Antoine Ligot and Mauro Birattari. Simulation-only experiments to mimic the effects of the reality gap in the automatic design of robot swarms. *Swarm Intelligence*, pages 1–24, 2019.

[15] Mauro Birattari, Antoine Ligot, Darko Bozhinoski, Manuele Brambilla, Gianpiero Francesca, Lorenzo Garattoni, David Garzón Ramos, Ken Hasselmann, Miquel Kegeleirs, Jonas Kuckling, Federico Pagnozzi, Andrea Roli, Muhammad Salman, and Thomas Stützle. Automatic off-line design of robot swarms: a manifesto. *Frontiers in Robotics and AI*, 6:59, 2019.

[16] Stefano Nolfi and Dario Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, Cambridge, MA, USA, first edition, 2000. A Bradford Book.

[17] Vito Trianni. *Evolutionary Swarm Robotics.* Springer, Berlin, Germany, 2008.

[18] Gianpiero Francesca, Manuele Brambilla, Arne Brutschy, Vito Trianni, and Mauro Birattari. AutoMoDe: a novel approach to the automatic design of control software for robot swarms. *Swarm Intelligence*, 8(2):89–112, 2014.

[19] Gianpiero Francesca, Manuele Brambilla, Arne Brutschy, Lorenzo Garattoni, Roman Miletitch, Gaëtan Podevijn, Andreagiovanni Reina, Touraj Soleymani, Mattia Salvaro, Carlo Pinciroli, Franco Mascia, Vito Trianni, and Mauro Birattari. AutoMoDe-Chocolate: automatic design of control software for robot swarms. *Swarm Intelligence*, 9(2–3):125–152, 2015.

[20] Jorge Gomes and Anders Lyhne Christensen. Task-agnostic evolution of diverse repertoires of swarm behviours. In Marco Dorigo, Mauro Birattari, Christian Blum, Anders Lyhne Christensen, Andreagiovanni Reina, and Vito Trianni, editors, *Swarm Intelligence – ANTS*, volume 11172 of *LNCS*, pages 225–238, Cham, Switzerland, 2018. Springer.