The Journal of Open Source Software

Crazyswarm2: A ROS 2-based Stack for Bitcraze Crazyflie Multirotor Robots

Wolfgang Hönig [1,2*] **and Kimberly N. McGuire** [3*]

1 TU Berlin, Germany **2** Robotics Institute Germany (RIG) **3** Independent Roboticist, Sweden ***** These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Sébastien Boisgérault

Reviewers:

- @llanesc
- @mrpollo

Submitted: 25 November 2025
Published: unpublished

Summary

Validation of multi-robot and swarm robotics research in the physical world requires a *testbed*, i.e., accessible robots and a software stack that is well tested and simplifies the operation of common use cases. We present Crazyswarm2, a software stack that uses the Robot Operating System 2 (ROS 2) (Macenski et al., 2022) at its core and enables simulation, visualization, and control of commercially off-the-shelf flying robots from Bitcraze AB (Figure 1). These robots are popular amongst researchers because they are fully open (including schematics and low-level firmware), extendible using standardized connectors, and can be easily obtained world-wide. Our software made significant changes to Crazyswarm (Preiss et al., 2017), a popular ROS 1-based stack that has been widely used in the research community for planning, state estimation, controls, and even art. While the high-level API is identical, we used the required breaking changes when moving to ROS 2 to re-visit some core design decisions and enable more sophisticated use-cases compared to the original Crazyswarm.

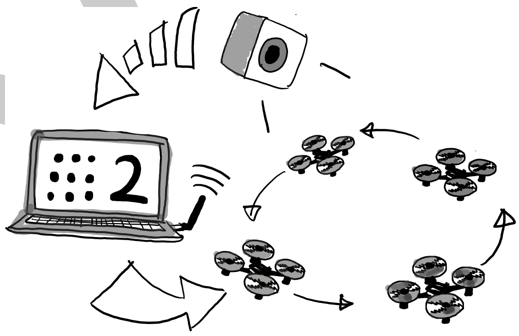

Figure 1: Simple representation of Crazyswarm2's core functionalities

Statement of Need

Testbeds are crucial for research in robotics as they simplify and accelerate data collection and validation experiments. The de-facto standard for physical robots is the Robot Operating System 2 (ROS 2) (Macenski et al., 2022), because robot vendors typically provide drivers and examples using this middleware. Most research labs for flying robots either use large custom-built multirotors with a powerful companion computer (e.g., Baca et al. (2021)) or the Crazyflie robots by Bitcraze AB. However, the official vendor software currently lacks ROS 2 support, has limitations when scaling to larger teams of robots, and does not provide simulation support. We address all of these limitations simultaneously, unlike existing and/or parallel efforts that bring ROS (2) support for the Crazyflie.

State of the Field

There are parallel efforts to mitigate the missing ROS 2 support: CrazyChoir (Pichierri et al., 2023) and AeroStack2 (Fernandez-Cortizas et al., 2023). Both rely on the official Python API, which can present challenges for larger teams, whereas our default backend is written in C++ and offers improved performance for multi-robot scenarios. The focus of CrazyChoir is on distributed optimization and for AeroStack2 on high-level missions. Differences between these frameworks and Crazyswarm2 have been evaluated at the "Aerial Swarm Tools and Applications" workshop at the Robotics Science and Systems conference (Kimberly McGuire, 2024).

A newer development is Dynamic Swarms Crazyflies (Vinzenz Malke, 2025), which has an interesting distributed architecture, where each Crazyflie is controlled by a single ROS 2 node. Each node uses topics to communicate with a radio node that handles all the communication. However, this approach requires an adjusted version of the vendor-maintained software and firmware to be installed and flashed, which may present long-term maintainability challenges.

There are also some dedicated existing simulation tools for the Crazyflie robot, e.g., CrazySim (Llanes et al., 2024), see our recent survey paper on simulation tools for a more detailed list (Dimmig et al., 2024). Most simulators are developed as separate tools that have different communication interfaces and APIs compared to those of the real robots. In Crazyswarm2, the simulation is integrated as a backend, allowing to seamlessly test ROS 2 user-code simply by changing a launch file flag.

Software Design

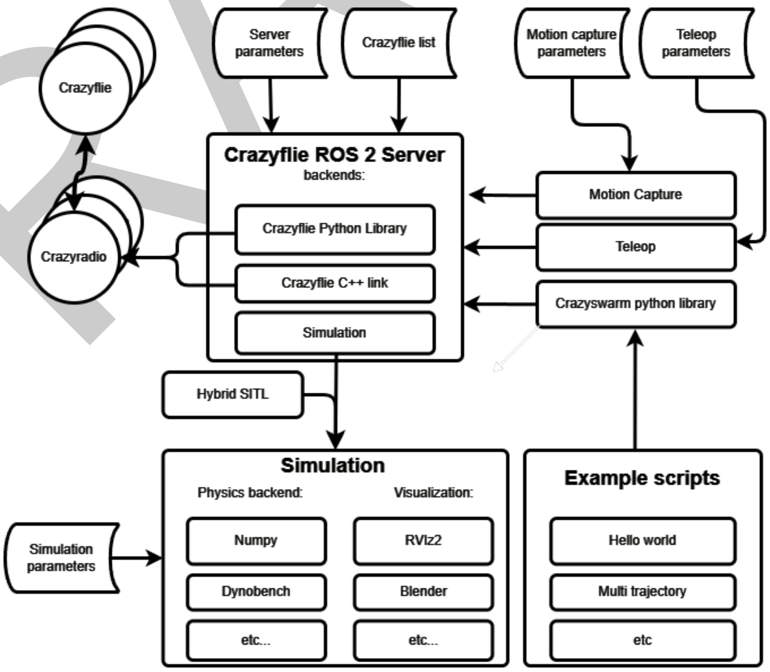

Figure 2: Architecture of Crazyswarm2

The architecture of Crazyswarm2 can be found in Figure 2. The Crazyflie server is the node that connects the Crazyflies to the ROS 2 framework. Its main components include the Crazyflie Server, the simulation framework, and a separate Python library that simplifies the command handling through ROS 2.

The Crazyflie server receives a list of Crazyflie URIs and ROS server parameters, and it will connect to multiple Crazyflies through multiple Crazyradios. This configuration YAML file also contains all the logging and parameters that need to be initialized within the Crazyflie ecosystem. It will then convert those specific logging and parameters to their ROS 2 equivalent and prepare them in proper topics and parameter types.

Additionally, the server converts any control topics to their Crazyflie framework equivalent through the commander structure, which is used for both the pitch/roll/yaw and velocity/position commands for control in real time. The Crazyflie's high-level commander framework is accessed through ROS 2 services, which only need to be called upon once and the Crazyflie will execute the command fully onboard. Moreover, services also exist to enable logging/parameters upon runtime, as well as an emergency service that will shut down the Crazyflie for safety.

The Crazyflie server includes three different backends: (1) the Crazyflie C++ library, (2) the Crazyflie Python library, and (3) the simulation backend. A fourth backend that is written in Rust and uses the official Rust library by the vendor is currently work in progress. (1) is a C++-based library that was developed alongside the original Crazyswarm project and reimplemented for Crazyswarm2. In time, (2) the Crazyflie Python library was added and made almost feature complete with (1). Cflib (Crazyflie Python Library) is the officially maintained communication library by Bitcraze AB, the developers of the Crazyflie.

The simulation backend acts as a gateway to the hybrid software-in-the-loop (SITL) simulation. The hybrid SITL consists of wrappers that convert the original Crazyflie's C-based firmware into callable Python functions, which can be called from the ROS 2 node. The simulation backend also has various physics and visualization sub-backends to choose from. The physics backends consist of various options, like a simple quadcopter dynamics based on the Python library NumPy and a dynamics library called dynobench. The visualization backends consist of libraries like the ROS 2 native RViz2, or Blender for high-level rendering purposes for camera sensing.

Finally, the Crazyswarm2 architecture also consists of a separate ROS 2 package that is a Python library. The main purpose of this library is to provide a simplified interface for users to control their Crazyflies as a layer above the full ROS 2 interface. Instead of writing service calls and topic publishers, they can call simple functions per Crazyflie entity in this library, which handles the ROS 2 calling on the backend.

Additionally, the Crazyswarm2 architecture supports integration with motion capture systems which provide positioning data into the Crazyflie server. To help users get started with the framework, Crazyswarm2 also includes a collection of example scripts that demonstrate common use cases, ranging from simple "Hello World" demonstrations to more complex multi-trajectory coordination scenarios.

Compared to Crazyswarm (Preiss et al., 2017), there are two key differences: 1) the motion capture support is not tightly integrated and instead properly separated; 2) the simulation supports the full ROS interface and physics (including inter-robot interaction forces).

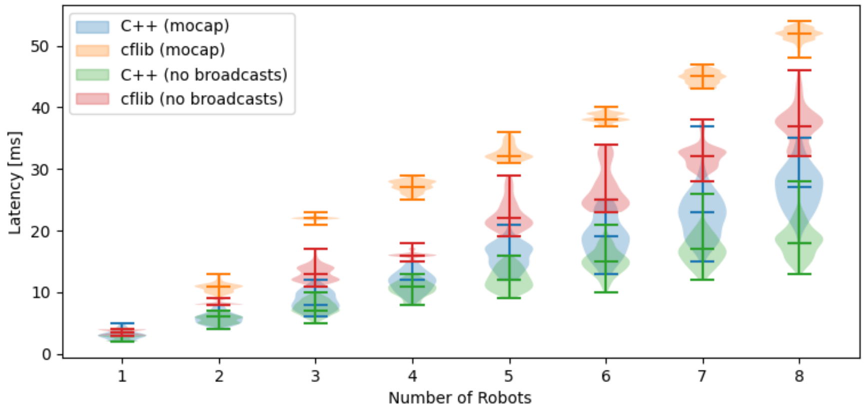

Figure 3: Communication Latency of Crazyswarm2.

Crazyswarm2 continuously measures the latency of the radio communication by sending the current timestamp to an "echo"-service and recording the timestamp once the packet is returned. The latency is the time difference between those two timestamp. The latencies for the two backends are shown in Figure 3. Here, we show the distribution of all latencies, i.e. stacked over all robots for a case with motion capture information being transmitted at 100 Hz.

We consider two cases: 1) external localization, e.g., using a motion capture system. In that case, the position/pose information of all robots is computed centrally and sent to the robots. The cpp-backend uses broadcast messages, while the cflib uses unicast messages, explaining the big difference especially for larger team sizes. 2) self-localization, e.g., by using the LightHouse localization system or on-board sensors. Here, both backends only rely on unicast messages and the difference between the two backends is less pronounced.

Research Impact Statement

Crazyswarm2 has already been used by several researchers, mostly to validate novel algorithms on physical hardware. The following is a non-exhaustive list, demonstrating the usefulness of the package in different areas of robotics.

- Exploration / Active Sensing (Dong et al., 2025; Liu & Ren, 2025; Pagano et al., 2025; Yongce Liu, 2025)
- Novel hardware design / SW frameworks(Boëgeat et al., 2025; Chiun et al., 2024)
- Motion Planning (Khan et al., 2024; Li et al., 2025; Moldagalieva et al., 2024; Toumieh & Floreano, 2024; Wahba et al., 2024; Wahba & Hönig, 2024, 2025)
- Online learning (Cobo-Briesewitz et al., 2025; Lorentz et al., 2025; Tseng et al., 2025)
- Controls (Aram & Bekmez, 2023; Engl, 2024; Karasahin, 2025; Yan et al., 2024)
- Perception (Moldagalieva & Hönig, 2023)
- Integration with ROS 2 (Llanes et al., 2024)

Conclusion and Future Work

Crazyswarm2 is a ROS 2 software stack that allows researchers to simulate and physically validate (multi-)robot algorithms. Compared to other ROS 2 solutions, it supports more (low-level) features, similar to the vendor's official software stack. Compared to the official software, we add ROS 2 integration, we improve the scalability and usability for larger teams by using broadcast communication, and we include an easy way to simulate robots with the same (ROS 2) API.

In the future, we plan to add a Rust backend as the default option, which will allow us to reuse more of the official vendor's software stack without sacrificing performance for larger teams. We are also planning to improve the swarm management tools to better support distributed operation use cases.

Conflict of Interest

Kimberly N. McGuire started her work on Crazyswarm2 while being employed at Bitcraze AB, Sweden, the vendor of the robots supported in this stack. She is currently an independent roboticist with no financial relationship to Bitcraze AB.

Acknowledgements

The work was supported by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Grant 448549715 and the Federal Ministry for Research, Technology and Aeronautics Germany (BMFTR) under Grant 16ME1000.

We would also like to thank one of the original authors of the predecessor project, Crazyswarm, namely James A. Preiss as well as the contributors of Crazyswarm2.

AI Usage Disclosure

No generative AI tools were used in the development of this software, the writing of this manuscript, or the preparation of supporting materials. Light editing for improving wording, spelling, and grammar was done with Claude sonnet 4.5.

References

Aram, K., & Bekmez, A. (2023). Leader-follower based formation control of heterogeneous UAV-AGV multi-agent system. *European Journal of Technique (EJT)*, *13*(2), 241–248. https://doi.org/10.36222/ejt.1350641

Baca, T., Petrlik, M., Vrba, M., Spurny, V., Penicka, R., Hert, D., & Saska, M. (2021). The MRS UAV system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles. *Journal of Intelligent & Robotic Systems*, *102*(1), 26. https://doi.org/10.1007/s10846-021-01383-5

Boëgeat, S., Bonnel, A., Mattos, L. P., Pouthier, F., Chibane, H., & Durand, S. (2025). Innovative and frugal design of a modular and reconfigurable system of UAVs. *26e Congrès Français de Mécanique, Laboratoire d'Etude Des Microstructures Et de Mécanique Des Matériaux (LEM3 - UMR CNRS 7239)*.

Chiun, J., Tan, Y. R., Cao, Y., Tan, J., & Sartoretti, G. (2024). STAR: Swarm technology for aerial robotics research. *International Conference on Control, Automation and Systems (ICCAS)*, 141–146. https://doi.org/10.23919/ICCAS63016.2024.10773308

Cobo-Briesewitz, E., Wahba, K., & Hönig, W. (2025). *Neural-augmented incremental nonlinear dynamic inversion for quadrotors with payload adaptation*. https://doi.org/10.48550/arXiv.2503.09441

Dimmig, C. A., Silano, G., McGuire, K., Gabellieri, C., Hönig, W., Moore, J., & Kobilarov, M. (2024). Survey of simulators for aerial robots: An overview and in-depth systematic comparisons. *IEEE Robotics & Automation Magazine*, 2–15. https://doi.org/10.1109/MRA.2024.3433171

Dong, D. E., Berger, H., & Abraham, I. (2025). Time-optimal ergodic search: Multiscale coverage in minimum time. *The International Journal of Robotics Research*, *44*(10–11), 1664–1683. https://doi.org/10.1177/02783649241273597

Engl, M. (2024). *Coordinated control of ground and aerial vehicle during takeoff and landing* [Master's thesis]. TU Wien.

Fernandez-Cortizas, M., Molina, M., Arias-Perez, P., Perez-Segui, R., Perez-Saura, D., & Campoy, P. (2023). *Aerostack2: A software framework for developing multi-robot aerial systems*. https://doi.org/10.48550/arXiv.2303.18237

Karasahin, A. T. (2025). Trajectory tracking with zero-shot reinforcement learning on aerial robots: A real-world evaluation. *International Symposium on Innovative Approaches in Smart Technologies (ISAS)*, 1–4. https://doi.org/10.1109/ISAS66241.2025.11101903

Khan, R. A., Zafar, M., Batool, A., Fedoseev, A., & Tsetserukou, D. (2024). SwarmPath: Drone swarm navigation through cluttered environments leveraging artificial potential field and impedance control. *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 402–407. https://doi.org/10.1109/ROBIO64047.2024.10907517

Kimberly McGuire, M. F. C., Wolfgang Hönig. (2024). *Aerial swarm tools and applications: Half-day tutorial and workshop at Robotics Science and Systems (RSS)*. https://imrclab.github.io/workshop-aerial-swarms-rss2024/

Li, J., Long, T., Sun, J., & Zhong, J. (2025). *TRUST-planner: Topology-guided robust trajectory planner for AAVs with uncertain obstacle spatial-temporal avoidance*. https://doi.org/10.48550/arXiv.2508.14610

Liu, Y., & Ren, Z. (2025). Multi-robot ergodic trajectory optimization with relaxed periodic connectivity. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Llanes, C., Kakish, Z., Williams, K. A., & Coogan, S. (2024). CrazySim: A software-in-the-loop simulator for the crazyflie nano quadrotor. *IEEE International Conference on Robotics and Automation (ICRA)*, 12248–12254. https://doi.org/10.1109/ICRA57147.2024.10610906

Lorentz, V., Wahba, K., Auddy, S., Toussaint, M., & Hönig, W. (2025). *CrazyMARL: Decentralized direct motor control policies for cooperative aerial transport of cable-suspended payloads*. https://doi.org/10.48550/arXiv.2509.14126

Macenski, S., Foote, T., Gerkey, B., Lalancette, C., & Woodall, W. (2022). Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics*. https://doi.org/10.1126/scirobotics.abm6074

Moldagalieva, A., & Hönig, W. (2023). Virtual omnidirectional perception for downwash prediction within a team of nano multirotors flying in close proximity. *International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, 64–70. https://doi.org/10.1109/MRS60187.2023.10416772

Moldagalieva, A., Ortiz-Haro, J., Toussaint, M., & Hönig, W. (2024). Db-CBS: Discontinuity-bounded conflict-based search for multi-robot kinodynamic motion planning. *IEEE International Conference on Robotics and Automation (ICRA)*, 14569–14575. https://doi.org/10.1109/ICRA57147.2024.10610999

Pagano, F., De Carli, N., Restrepo, E., Marino, A., & Giordano, P. R. (2025). Distributed multi-robot active-sensing of a diffusive source. *IEEE Robotics and Automation Letters*, *10*(10), 10807–10814. https://doi.org/10.1109/LRA.2025.3606376

Pichierri, L., Testa, A., & Notarstefano, G. (2023). CrazyChoir: Flying swarms of crazyflie quadrotors in ROS 2. *IEEE Robotics and Automation Letters*, *8*(8), 4713–4720. https:

//doi.org/10.1109/LRA.2023.3286814

Preiss, J. A., Hönig, W., Sukhatme, G. S., & Ayanian, N. (2017). Crazyswarm: A large nano-quadcopter swarm. *IEEE International Conference on Robotics and Automation (ICRA)*, 3299–3304. https://doi.org/10.1109/ICRA.2017.7989376

Toumieh, C., & Floreano, D. (2024). High-speed motion planning for aerial swarms in unknown and cluttered environments. *IEEE Transactions on Robotics*, *40*, 3642–3656. https://doi.org/10.1109/TRO.2024.3429193

Tseng, K.-Y., Shamma, J. S., & Dullerud, G. E. (2025). Hybrid gradient-based policy optimization for sample-efficient policy learning in autonomous systems. *American Control Conference (ACC)*, 3035–3040. https://doi.org/10.23919/ACC63710.2025.11107826

Vinzenz Malke, S. Rossi. (2025). *Dynamic swarms crazyflies*. https://dynamicswarms.github.io/ds-crazyflies/

Wahba, K., & Hönig, W. (2024). Efficient optimization-based cable force allocation for geometric control of a multirotor team transporting a payload. *IEEE Robotics and Automation Letters*, *9*(4), 3688–3695. https://doi.org/10.1109/LRA.2024.3351001

Wahba, K., & Hönig, W. (2025). Pc-dbCBS: Kinodynamic motion planning of physically-coupled robot teams. *IEEE Robotics and Automation Letters*. https://doi.org/10.1109/LRA.2025.3607267

Wahba, K., Ortiz-Haro, J., Toussaint, M., & Hönig, W. (2024). Kinodynamic motion planning for a team of multirotors transporting a cable-suspended payload in cluttered environments. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 12750–12757. https://doi.org/10.1109/IROS58592.2024.10802794

Yan, B., Ni, J., Zhong, Y., Yu, D., & Wang, Z. (2024). Collision-free formation control for heterogeneous multiagent systems under DoS attacks. *IEEE Transactions on Cybernetics*, *54*(10), 6244–6255. https://doi.org/10.1109/TCYB.2024.3418973

Yongce Liu, Z. R. (2025). A probabilistic measure of multi-robot connectivity and ergodic optimal control. *Robotics Science and Systems (RSS)*.