Crazyswarm: A Powerful Framework for Aerial Swarms in Research and Education

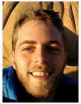
Bitcraze Awesome Meetup (BAM) days

October 21, 2021

Wolfgang Hönig

Technische Universität Berlin

In collaboration with James A. Preiss



Once Upon a Time... (2015)

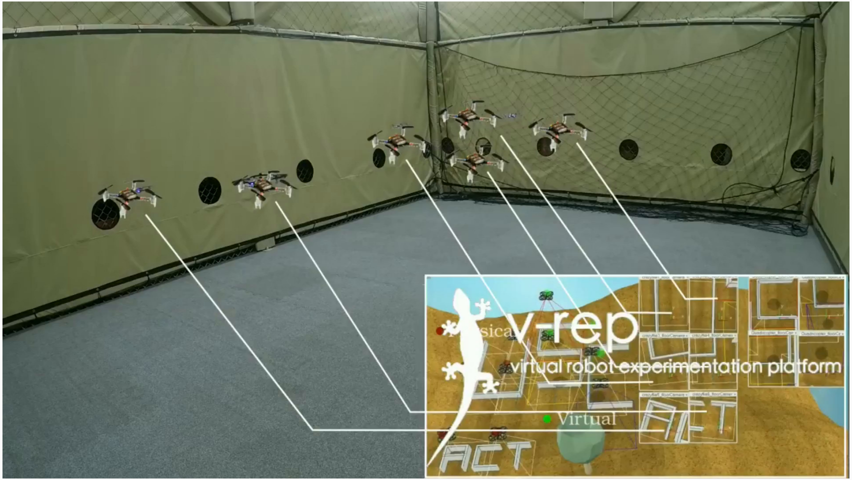
- Very few (research) labs were able to operate more than 10 robots
  - High (single-robot) cost
  - High engineering effort
  - Limited reliability

90 %

(independent success probability)

35 %

Video

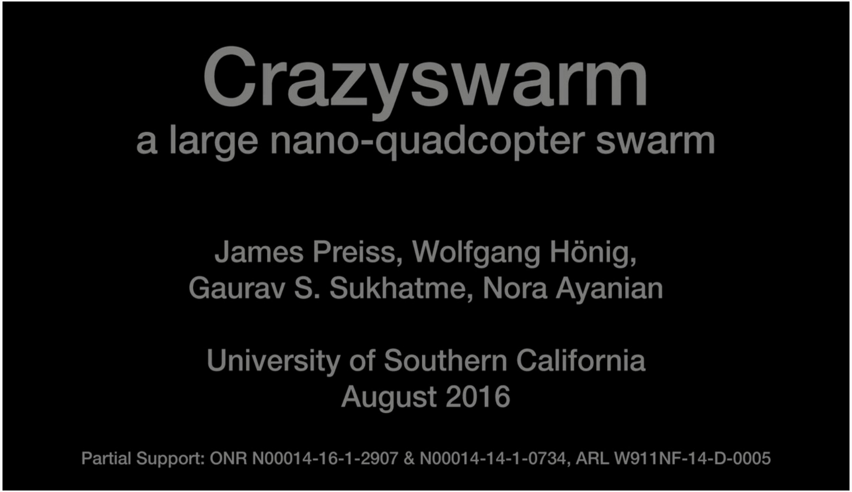


https://youtu.be/px9iHkA0nOI

Once Upon a Time... (2016)

- What would it take to fly one order of magnitude more?

Video



https://youtu.be/D0CrjoYDt9w

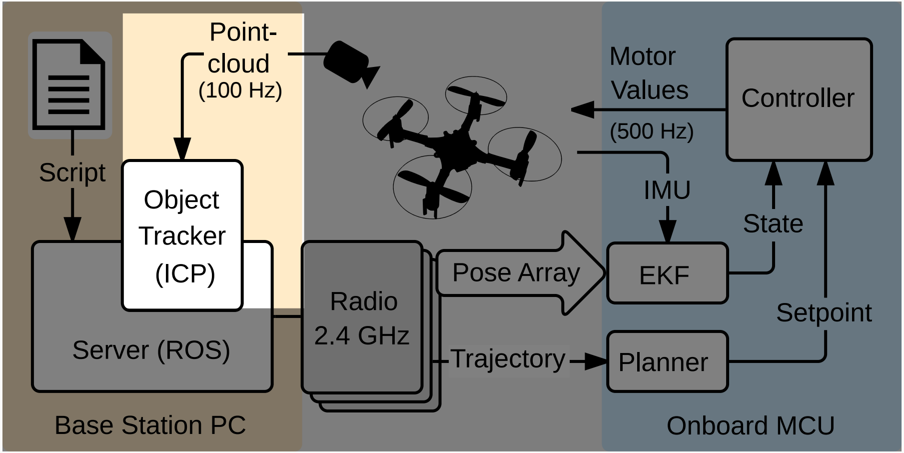
Overview

1. How Crazyswarm Works
2. Use Cases and Users
3. Outlook

4. Crazyswarm Tutorial
5. Conclusion

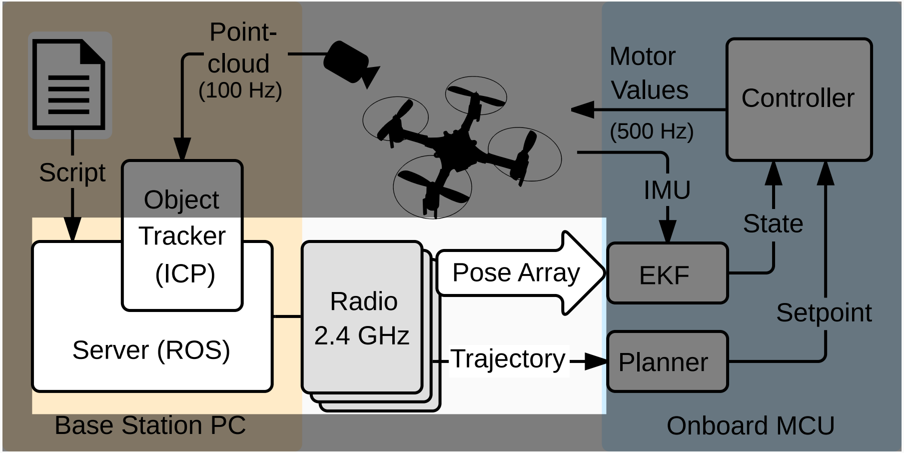
How Crazyswarm Works

System Architecture



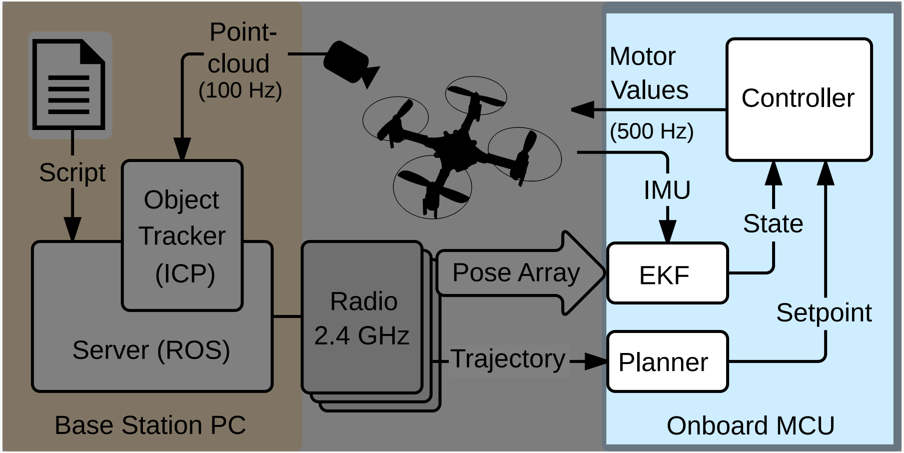
- Tight integration with motion capture systems (Vicon, OptiTrack, Qualisys)
- Custom, robust object tracker (a single marker per CF is sufficient)

System Architecture



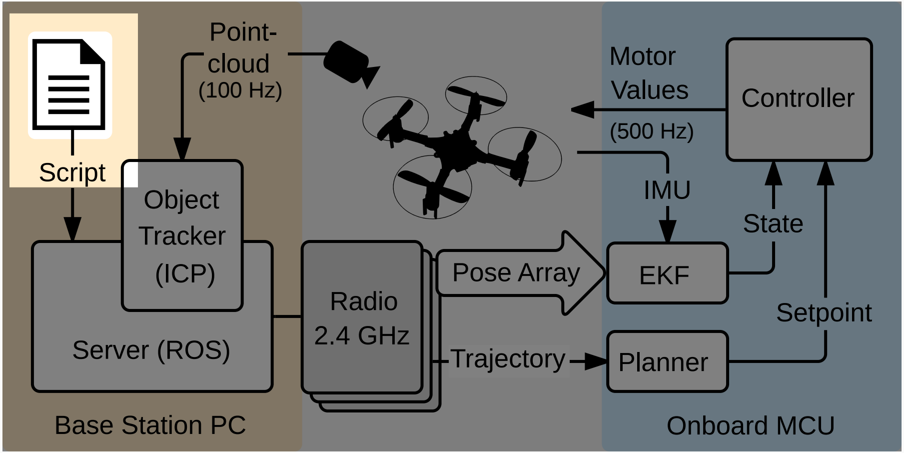
- One-way data flow (broadcast communication)
- Native (C++) backend optimized for low latency

System Architecture



- Relies on on-board autonomy (state estimation, planning, control)
- Works with the official firmware

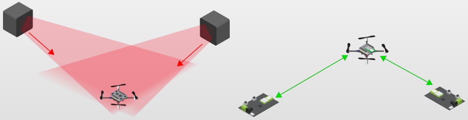
System Architecture



- High-level Python scripts, optimized for swarms
- Tools for swarms: simulation, visualization, battery check, reboot, etc.

External Localization

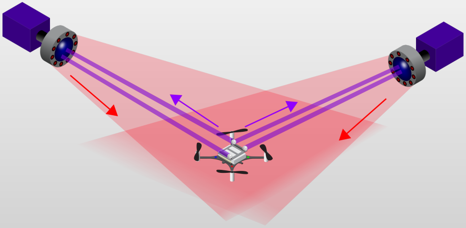
LightHouse and LPS



Low infrastructure cost; decentralized

Supported, but not primary use-case

Motion Capture



Low per-robot cost; sub-mm accuracy

Primary focus

Motion Capture Challenges

1. A centralized approach
    - Single native (C++) application tightly integrating: motion capture API, object tracking, radio communication for all Crazyflies (anti-ROS pattern)
    - Low-level communication optimization (broadcasts; special radio mode; compressed radio packets)
2. Tracking of many tiny rigid bodies
    - Commercial motion capture designed for tracking: a) humans (skeleton, face), or b) a few, large rigid bodies
    - We developed libobjecttracking for custom frame-by-frame tracking

Custom Rigid Body Tracking

Pose (Position + Orientation)

Iterative Closest Point (ICP); Greedy

Disadvantage: Less robust; needs more markers

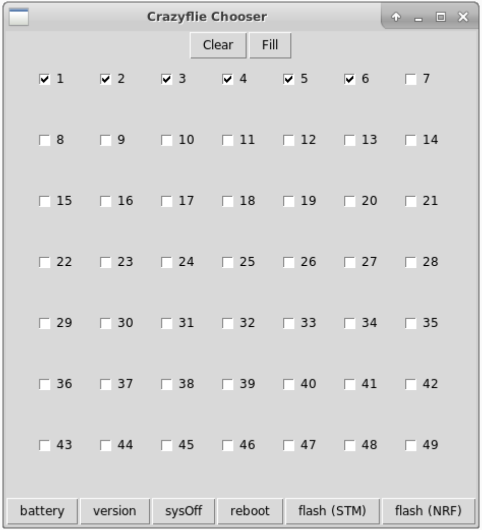
Position

Optimal Assignment

Disadvantage: requires *xy* movement to recover yaw

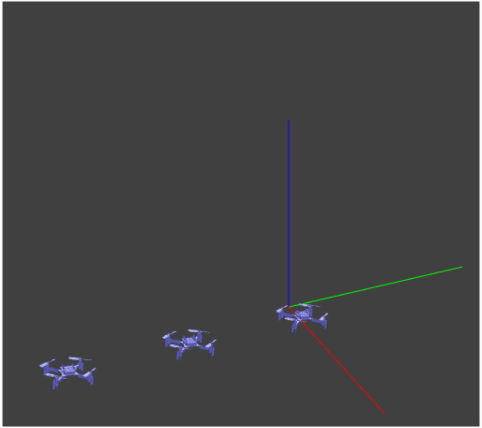
Both: Dynamics filter (max a, v, ω); initial position guess required

Tools: Swarm Management



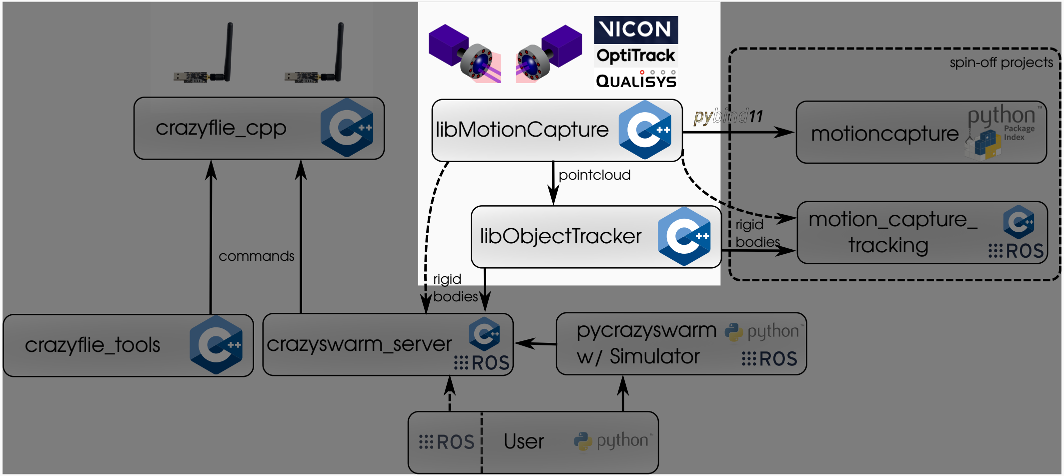
- Visualization of initial position
- Enable/Disable crazyflies
- Battery check
- Reboot

Tools: Simulation

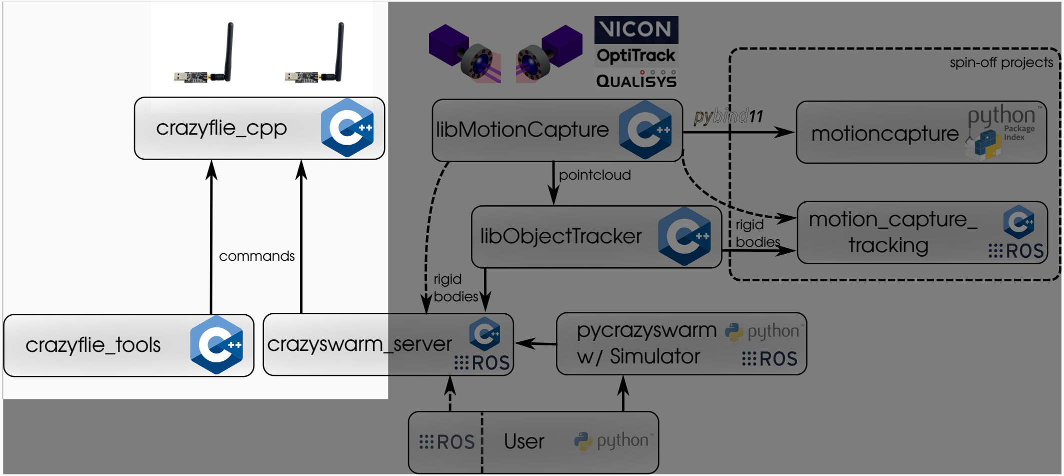


- Same script as used for physical robots (`--sim` flag)
- No ROS dependency
- Linux and Mac support
- Software-in-the-loop (SIL) via Python bindings to firmware
- Fast (no (aero)dynamics; use CrazyS if you need that)

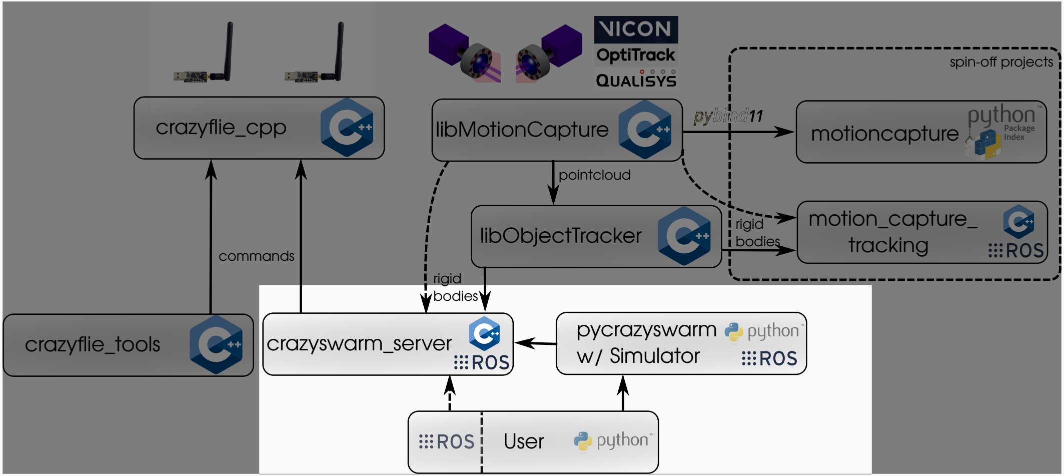
Software Architecture



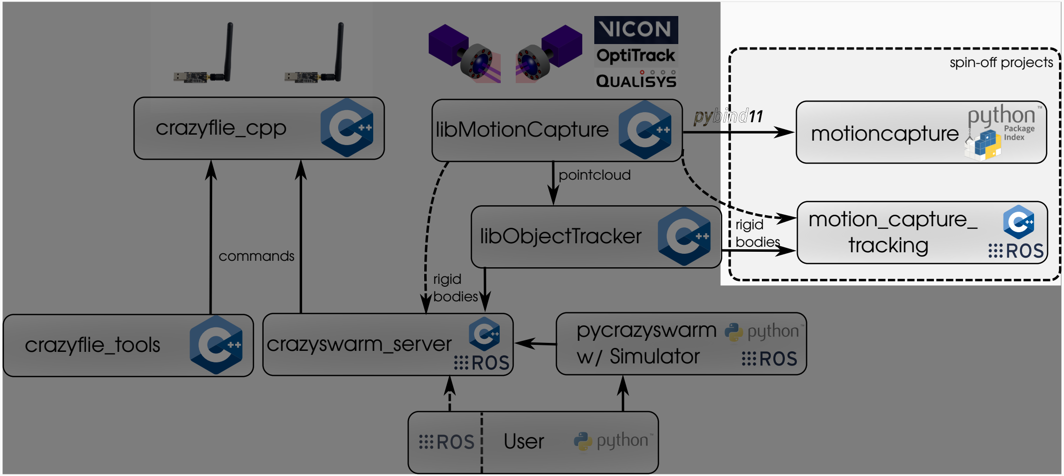
Software Architecture



Software Architecture



Software Architecture



Use Cases and Users

Google Scholar

Crazyswarm: A large nano-quadcopter swarm

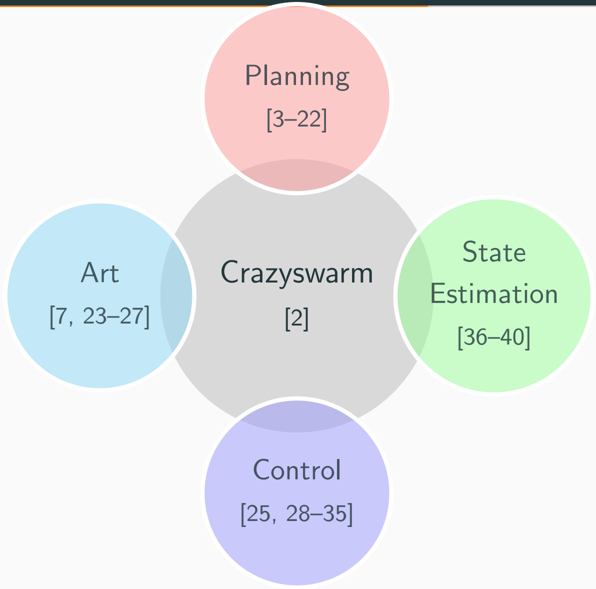
JA Preiss, W Honig, GS Sukhatme… - 2017 IEEE International …, 2017 - ieeexplore.ieee.org

We define a system architecture for a large swarm of miniature quadcopters flying in dense formation indoors. The large number of small vehicles motivates novel design choices for state estimation and communication. For state estimation, we develop a method to reliably …

Cited by 199 Related articles All 6 versions TU-Service TU-Service (check Print?)

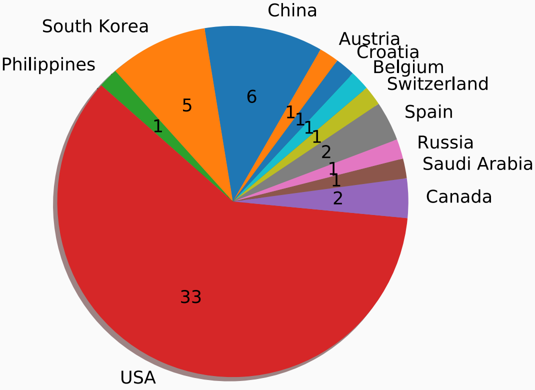
- 57 actually using the Crazyswarm
    - 11 of which are co-authored by James or myself
- Crazyswarm partially enabled 46 (or 12 per year) novel research contributions

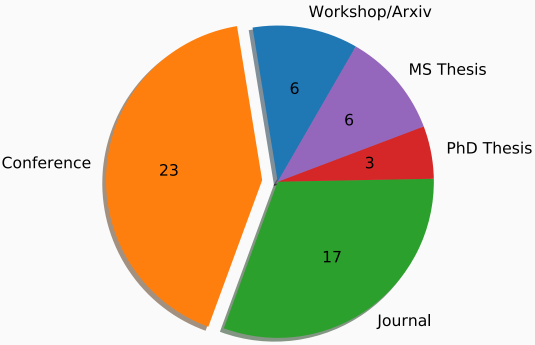
Research Projects By Topic



- Machine Learning: [34, 35, 41–43]
- Software Engineering: [44, 45]
- VR: [46, 47]
- Optimization: [48–51]
- Target Tracking: [6, 28]
- Exploration: [52–56]
- Teleoperation: [57]
- Resilience: [3, 58]

Research Projects by Country and Publication Type





Example: Formation Control / Art [25]

Fast and In Sync: Periodic Swarm Patterns for Quadrotors

Xintong Du, Carlos E. Luis, Marijan Vukosavljev, and Angela P. Schoellig

***Abstract*— This paper aims to design quadrotor swarm performances, where the swarm acts as an integrated, coordinated unit embodying moving and deforming objects. We divide the task of creating a choreography into three basic steps: designing swarm motion primitives, transitioning between those movements, and synchronizing the motion of the drones. The result is a flexible framework for designing choreographies comprised of a wide variety of motions. The motion primitives can be intuitively designed using a few parameters, providing a rich library for choreography design. Moreover, we combine and adapt existing goal assignment and trajectory generation algorithms to maximize the smoothness of the transitions between motion primitives. Finally, we propose a correction algorithm to compensate for motion delays and synchronize the motion of the drones to a desired periodic motion pattern. The proposed methodology was validated experimentally by generating and executing choreographies on a swarm of 25 quadrotors.**

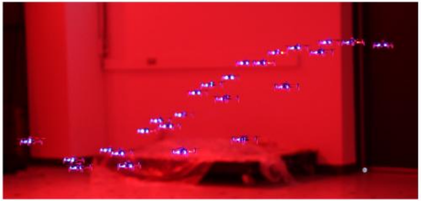
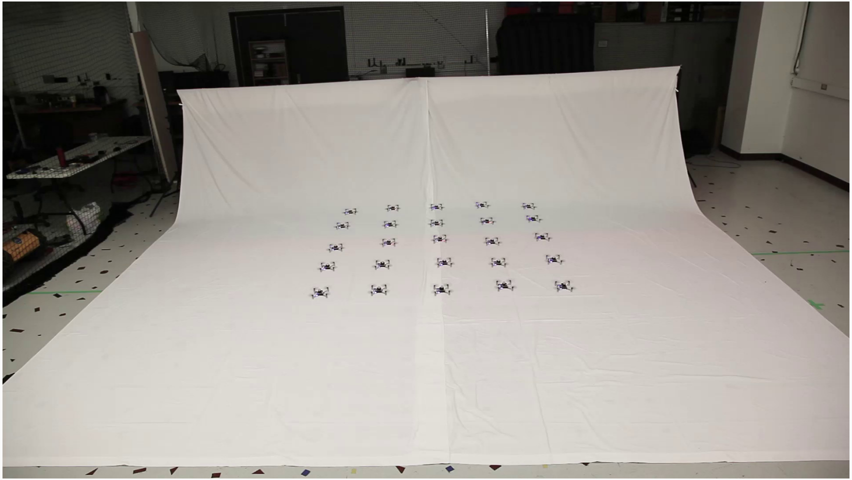


Fig. 1. Twenty-five quadrotors perform a periodic wave motion in the vertical direction. A video of a full performance is available at `http://tiny.cc/fast-periodic`

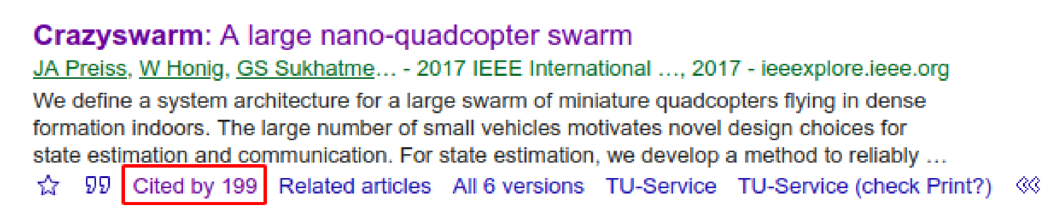
Nominated for the Best Paper Award on Multi-Robot Systems at ICRA 2019

Video



https://youtu.be/Iw8mwt3l0RE

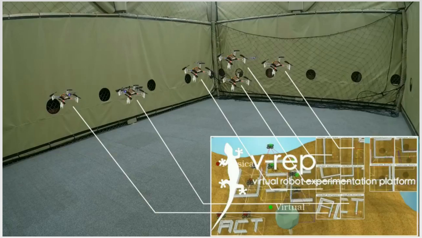
Google Scholar — revisited



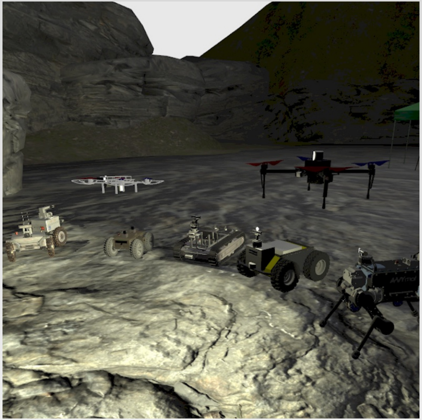
- 199 - 57 = 142
- Some survey papers
- Some unrelated papers
- Many citations for unrealistic/centralized (“bad”) approach

Outlook

Common Research/Engineering Motivation



Manual operation in highly predictable lab setting



Autonomous operation in highly unpredictable and unknown environments

Crazyswarm Focus

Robustness

- More on-board autonomy

Distributed Operation

- Improved documentation (we already support LightHouse, distributed execution)
- Swarm Monitoring/Logging/Management (e.g., Peer2Peer)

Heterogeneity

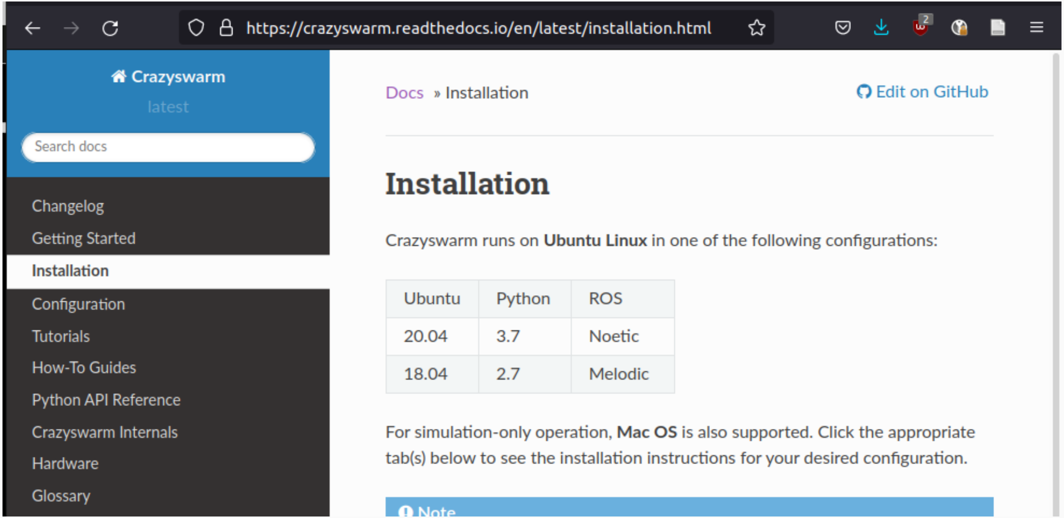
- Better ROS-simulation to allow heterogeneous teams
- Optional physics-based simulation w/ inter-drone interactions

Short-term Next Steps

1. Switch to crazyflie-link-cpp for communication robustness
2. New ROS-based simulation for distributed algorithms
3. ROS2 port (this will break compatibility with config files)
4. Physics-based simulation w/ aerodynamic interactions
5. Use in the classroom

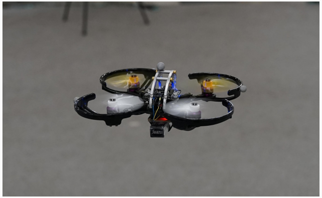
Crazyswarm Tutorial

Getting Started: Installation



Getting Started: Define your UAV Type

90 mm, 33 g



120 mm, 124 g

210 mm, 491 g

crazyflieTypes.yaml

```yaml
default:
  bigQuad: False
  batteryVoltageWarning: 3.8  # V
  markerConfiguration: 0
  # ...
```

Getting Started: List all UAVs

allCrazyflies.yaml

```yaml
crazyflies:
  - id: 1
    channel: 100
    initialPosition: [1.5, 1.5, 0.0]
    type: medium
  - id: 40
    # ...
```

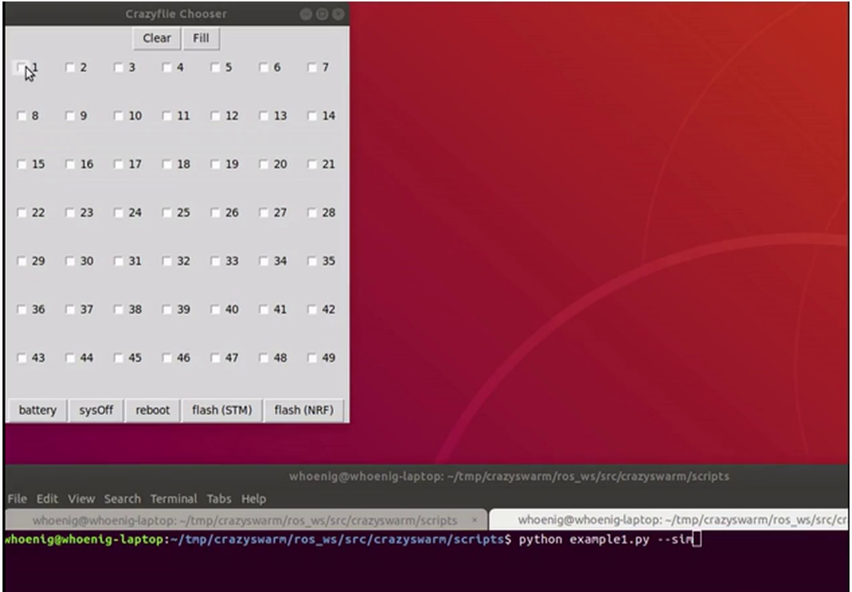
- Initial position for frame-by-frame tracking and simulation

Example 1: Hello World

example1.py

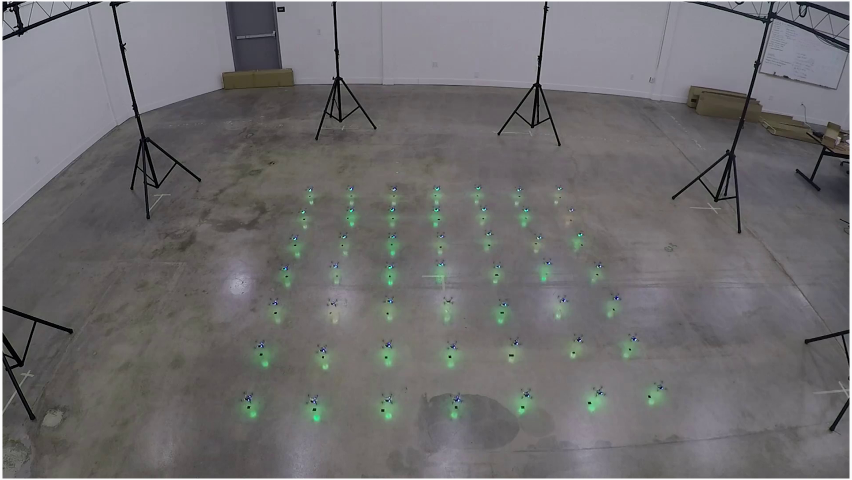
```python
from pycrazyswarm import *

swarm = Crazyswarm()
swarm.allcfs.takeoff(targetHeight=0.5, duration=2.0)
swarm.timeHelper.sleep(3.0)
swarm.allcfs.land(targetHeight=0.0, duration=2.0)
```

Video



Crazyflie Chooser
Clear
Fill
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31 32 33 34 35
36 37 38 39 40 41 42
43 44 45 46 47 48 49
battery
sysOff
reboot
flash (STM)
flash (NRF)
whoenig@whoenig-laptop: ~/tmp/crazyswarm/ros_ws/src/crazyswarm/scripts
File Edit View Search Terminal Tabs Help
whoenig@whoenig-laptop: ~/tmp/crazyswarm/ros_ws/src/crazyswarm/scripts
whoenig@whoenig-laptop:~/tmp/crazyswarm/ros_ws/src/crazyswarm/scripts$ python example1.py --sim

Video



Example 2: Trajectory Tracking

example2.py

```
from pycrazyswarm import *

swarm = Crazyswarm()
for cf, fname in zip(cfs, fnames):
  traj = uav_trajectory.Trajectory()
  traj.loadcsv(fname)
  cf.uploadTrajectory(0, 0, traj)

swarm.allcfs.takeoff(targetHeight=0.5, duration=2.0)
swarm.timeHelper.sleep(2.0)
allcfs.startTrajectory(0)
# ...
```

Video

http://youtu.be/7KIa9FlmbRc

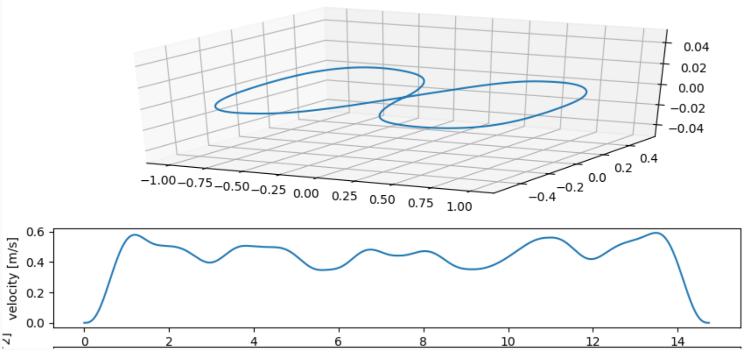
Example 2: Trajectory Generation (1)

Trajectories can be generated from higher-level data using
`https://github.com/whoenig/uav_trajectories`

- Sequence of waypoints and desired maximum velocity/acceleration

figure8.csv

```
0.0, 0.0, 0.0
0.5, -0.5, 0.0
1.0, 0.0, 0.0
0.5, 0.5, 0.0
0.0, 0.0, 0.0
-0.5, -0.5, 0.0
...
```



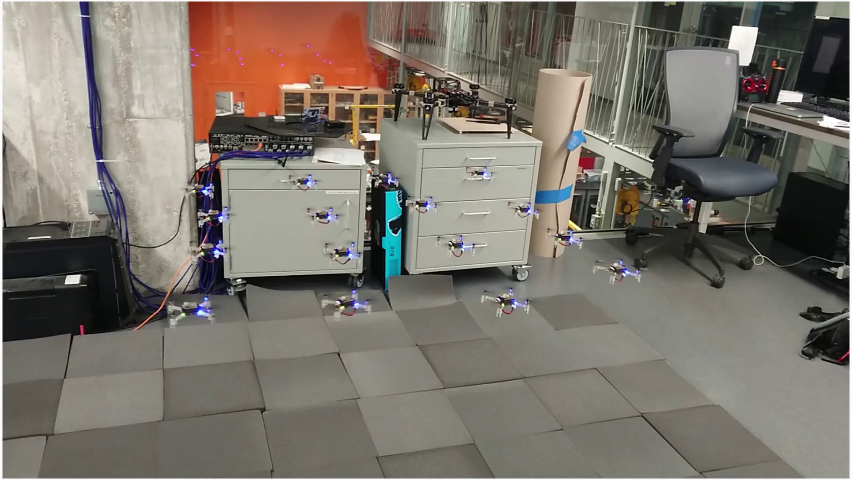
Example 2: Trajectory Generation (2)

- Sequence of (time, waypoint) pairs

genCircle.py

```python
for t in linspace(0, T, 100):
  f.write("{},{},{},{}\n".format(
    t,
    r * cos(t + phase),
    r * sin(t + phase),
    height))
```

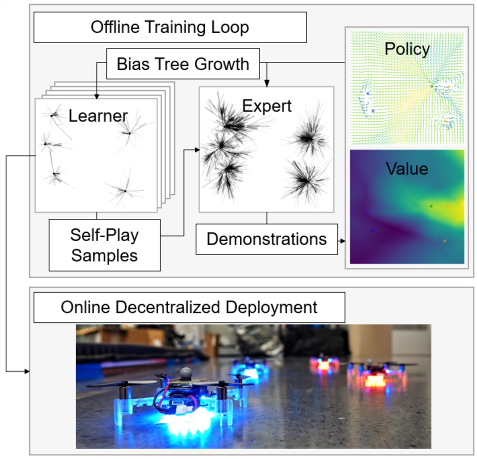
Video



Example 2: Trajectory Generation (3)

- More advanced techniques is active area of research [4, 5, 9, 14, 20]
- Some of which are open-source, e.g,
    - `https://github.com/USC-ACTLab/rlss`
    - `https://github.com/mjdebord/smoothener/tree/cylinders`

Example 3: Distributed Execution



- Verification of a new distributed algorithm
- Implementation in firmware might be time-consuming or even impossible
- Here: Monte-Carlo Tree Search with multiple neural networks [43]
- Run one process per CF on a host computer, only using local/relative information

Example 3: Distributed Execution: Script

example3.py

```
import rospy
from pycrazyswarm.crazyflie import Crazyflie

rospy.init_node("CrazyflieDistributed", anonymous=True)
# ... (logic to identify which Crazyflie should be served)
cf = Crazyflie(cfid, initialPosition, tf)

while not rospy.is_shutdown():
  # heavy computation, e.g., optimization that
  # can rely on own and neighbors state
  pos = complicatedFunction()
  cf.cmdPosition(pos)
```

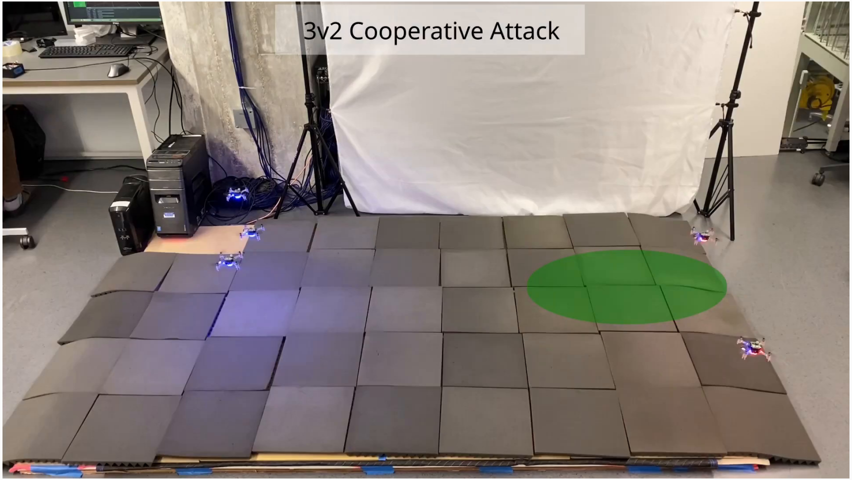
Example 3: Distributed Execution: Launch File

run.launch

```xml
<launch>
  <!-- ... -->
  <node name="cf1" pkg="crazyswarm" type="example3.py">
    <param name="cfid" value="1" />
  </node>
  <node name="cf2" pkg="crazyswarm" type="example3.py">
    <param name="cfid" value="2" />
  </node>
  <!-- ... -->
</launch>
```

Can use `roslaunch crazyswarm run.launch sim:=True` for simulation

Video



https://youtu.be/mklbTfWl7DE

Conclusion

The Crazyswarm is ...

- Versatile, open-source framework for aerial heterogeneous swarms
- Supports centralized and (some) decentralized operation and many different localization methods (motion capture, LightHouse, LPS, ...)
- Widely used to validate research in many domains

Not what you are looking for?

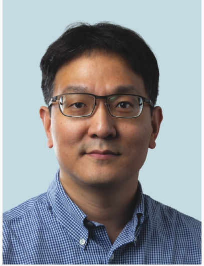
- Tracking: libmotioncapture, pip motioncapture, libobjecttracker, standalone ROS Stack
- Simulation/Development: Firmware Python bindings
- Matlab: Use the ROS Toolbox

Acknowledgments


James A. Preiss


Nora Ayanian


Gaurav S. Sukhatme

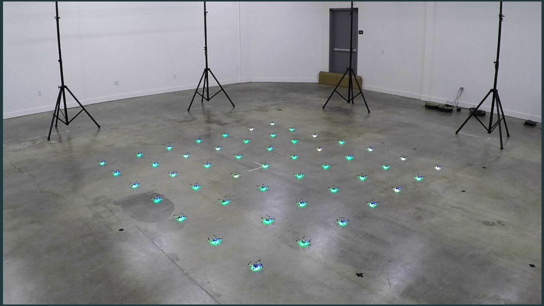

Soon-Jo Chung

bitcraze

USC University of Southern California

Caltech

Crazyswarm: A Powerful Framework for Aerial Swarms in Research and Education



More Information:

http://crazyswarm.readthedocs.io

Contact:

hoenig@tu-berlin.de

[1] Wolfgang Hönig, Christina Milanes, Lisa Scaria, Thai Phan, Mark T. Bolas, and Nora Ayanian. "Mixed reality for robotics". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 5382–5387. DOI: 10.1109/IROS.2015.7354138.

[2] James A. Preiss*, Wolfgang Hönig*, Gaurav S. Sukhatme, and Nora Ayanian. "Crazyswarm: A large nano-quadcopter swarm". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3299–3304. DOI: 10.1109/ICRA.2017.7989376.

[3] Ragesh K. Ramachandran, James A. Preiss, and Gaurav S. Sukhatme. "Resilience by Reconfiguration: Exploiting Heterogeneity in Robot Teams". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Nov. 2019, pp. 6518–6525. DOI: 10.1109/IROS40897.2019.8968611.

[4] Jungwon Park and H. Jin Kim. "Fast Trajectory Planning for Multiple Quadrotors Using Relative Safe Flight Corridor". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 6, 2019.

References ii

[5] Wolfgang Hönig, James A. Preiss, T. K. Satish Kumar, Gaurav S. Sukhatme, and Nora Ayanian. "Trajectory Planning for Quadrotor Swarms". In: *IEEE Transactions on Robotics* 34.4 (Aug. 2018), pp. 856–869. DOI: 10.1109/TRO.2018.2853613.

[6] John J. Gainer Jr., Jeremy J. Dawkins, Levi D. DeVries, and Michael D. M. Kutzer. "Persistent Multi-Agent Search and Tracking with Flight Endurance Constraints". In: *Robotics* 8.1 (1 Mar. 2019), p. 2. DOI: 10.3390/robotics8010002.

[7] Ellen A. Cappo, Arjav Desai, Matthew Collins, and Nathan Michael. "Online Planning for Human–Multi-Robot Interactive Theatrical Performance". In: *Autonomous Robots* 42.8 (Dec. 1, 2018), pp. 1771–1786. DOI: 10.1007/s10514-018-9755-0.

[8] Ellen Cappo. "Data-Driven Multi-Robot Planning for Online User-Directed Coordination". thesis. Carnegie Mellon University, Apr. 14, 2021. DOI: 10.1184/R1/14402270.v1.

[9] Mark Debord, Wolfgang Hönig, and Nora Ayanian. "Trajectory Planning for Heterogeneous Robot Teams". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Oct. 2018, pp. 7924–7931. DOI: 10.1109/IROS.2018.8593876.

[10] Benjamin Gravell and Tyler Summers. "Centralized Collision-Free Polynomial Trajectories and Goal Assignment for Aerial Swarms". In: *Control Engineering Practice* 109 (Apr. 1, 2021), p. 104753. DOI: 10.1016/j.conengprac.2021.104753.

[11] Shuang Guo, Bo Liu, Shen Zhang, Jifeng Guo, and Changhong Wang. *Continuous-Time Gaussian Process Trajectory Generation for Multi-Robot Formation via Probabilistic Inference*. July 30, 2021. arXiv: 2010.13148.

[12] Jungwon Park. "Trajectory Planning for Multiple Quadrotors Using Relative Safe Flight Corridor and Relative Bernstein Polynomial". MS. Seoul National University, 2020.

[13] Paul Ladinig, Bernhard Rinner, and Stephan Weiss. "Time and Energy Optimized Trajectory Generation for Multi-Agent Constellation Changes". In: ().

References iv

[14] Jungwon Park, Junha Kim, Inkyu Jang, and H. Jin Kim. "Efficient Multi-Agent Trajectory Planning with Feasibility Guarantee Using Relative Bernstein Polynomial". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. May 2020, pp. 434–440. DOI: 10.1109/ICRA40945.2020.9197162.

[15] Jungwon Park, Dabin Kim, Gyeong Chan Kim, Dahyun Oh, and H. Jin Kim. *Online Distributed Trajectory Planning for Quadrotor Swarm with Feasibility Guarantee Using Linear Safe Corridor*. Sept. 18, 2021. arXiv: 2109.09041.

[16] Jungwon Park and H. Jin Kim. "Online Trajectory Planning for Multiple Quadrotors in Dynamic Environments Using Relative Safe Flight Corridor". In: *IEEE Robotics and Automation Letters* 6.2 (Apr. 2021), pp. 659–666. DOI: 10.1109/LRA.2020.3047786.

[17] Ryan Peterson, Ali Tevfik Buyukkocak, Derya Aksaray, and Yasin Yazıcıoglu. "Decentralized Safe Reactive Planning under TWTL Specifications". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2020, pp. 6599–6604. DOI: 10.1109/IROS45743.2020.9341624.

References v

[18] Ryan Peterson, Ali Tevfik Buyukkocak, Derya Aksaray, and Yasin Yazıcıoğlu. "Distributed Safe Planning for Satisfying Minimal Temporal Relaxations of TWTL Specifications". In: *Robotics and Autonomous Systems* 142 (Aug. 1, 2021). DOI: 10.1016/j.robot.2021.103801.

[19] Ryan James Peterson. "Safe Multi-Agent Planning under Time-Window Temporal Logic Specifications". MS Thesis. University Of Minnesota, 2020.

[20] James A. Preiss, Wolfgang Hönig, Nora Ayanian, and Gaurav S. Sukhatme. "Downwash-Aware Trajectory Planning for Large Quadrotor Teams". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 2017, pp. 250–257. DOI: 10.1109/IROS.2017.8202165.

[21] Wolfgang Hönig. "Motion Coordination for Large Multi-Robot Teams in Obstacle-Rich Environments". PhD thesis. University of Southern California, 2019. URL: https://www.proquest.com/openview/d0b33193c67f560a034258b9aaa9619b/1?pq-origsite=gscholar&cbl=2026366&diss=y.

[22] Yasin Yazıcıoğlu, Raghavendra Bhat, and Derya Aksaray. "Distributed Planning for Serving Cooperative Tasks with Time Windows: A Game Theoretic Approach". In: *Journal of Intelligent & Robotic Systems* 103.2 (Sept. 11, 2021), p. 27. DOI: 10.1007/s10846-021-01477-0.

[23] Kejia Ren and Paul G Kry. "Single Stroke Light Painting with a Quadrotor Robot". In: ().

[24] Ryan Cotsakis, David St-Onge, and Giovanni Beltrame. "Decentralized Collaborative Transport of Fabrics Using Micro-UAVs". In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019 International Conference on Robotics and Automation (ICRA). May 2019, pp. 7734–7740. DOI: 10.1109/ICRA.2019.8793778.

[25] Xintong Du, Carlos E. Luis, Marijan Vukosavljev, and Angela P. Schoellig. "Fast and In Sync: Periodic Swarm Patterns for Quadrotors". In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019 International Conference on Robotics and Automation (ICRA). May 2019, pp. 9143–9149. DOI: 10.1109/ICRA.2019.8794017.

[26] Ryan Cotsakis, Ying Gao, David St-Onge, and Giovanni Beltrame. "Clothes Design Using Micro-UAVs". In: ().

[27] Justin C. Rooney. "Visualizing Physical Phenomena Using Swarming Nano Quadrotors". MS Thesis. The Cooper Union for the Advancement of Science and Art, 2020. URL: https://www.proquest.com/openview/0abda4161f26d043e0d9c0faa5391fc8/1?pq-origsite=gscholar&cbl=51922&diss=y (visited on 10/04/2021).

[28] Levi DeVries and Jeremy Dawkins. "Multi-Vehicle Target Tracking and Formation Control in Non-Inertial Reference Frames". In: *2018 Annual American Control Conference (ACC)*. 2018 Annual American Control Conference (ACC). June 2018, pp. 4026–4031. DOI: 10.23919/ACC.2018.8430831.

[29] Kaveh Fathian. "Distributed Formation Control of Autonomous Vehicles via Vision-Based Motion Estimation". PhD thesis. University Of Texas At Dallas, 2018.

[30] Kaveh Fathian, Sleiman Safaoui, Tyler H. Summers, and Nicholas R. Gans. "Robust 3D Distributed Formation Control With Collision Avoidance And Application To Multirotor Aerial Vehicles". In: *2019 International Conference on Robotics and Automation (ICRA)*. May 2019, pp. 9209–9215. DOI: 10.1109/ICRA.2019.8794349.

[31] Bryan Convens, Kelly Merckaert, Bram Vanderborght, and Marco M. Nicotra. “Invariant Set Distributed Explicit Reference Governors for Provably Safe On-Board Control of Nano-Quadrotor Swarms”. In: *Frontiers in Robotics and AI* 8 (June 22, 2021), p. 663809. DOI: 10.3389/frobt.2021.663809.

[32] Talha Kavuncu, Ayberk Yaraneri, and Negar Mehr. *Potential ILQR: A Potential-Minimizing Controller for Planning Multi-Agent Interactive Trajectories*. July 10, 2021. arXiv: 2107.04926.

[33] Yang Lyu, Jinwen Hu, Ben M. Chen, Chunhui Zhao, and Quan Pan. “Multivehicle Flocking With Collision Avoidance via Distributed Model Predictive Control”. In: *IEEE Transactions on Cybernetics* 51.5 (May 2021), pp. 2651–2662. DOI: 10.1109/TCYB.2019.2944892.

[34] Guanya Shi, Wolfgang Hönig, Xichen Shi, Yisong Yue, and Soon-Jo Chung. “Neural-Swarm2: Planning and Control of Heterogeneous Multirotor Swarms Using Learned Interactions”. In: *IEEE Transactions on Robotics* (2021), pp. 1–17. DOI: 10.1109/TRO.2021.3098436.

[35] Guanya Shi, Wolfgang Hönig, Yisong Yue, and Soon-Jo Chung. "Neural-Swarm: Decentralized Close-Proximity Multirotor Control Using Learned Interactions". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. May 2020, pp. 3241–3247. DOI: 10.1109/ICRA40945.2020.9196800.

[36] Zhiyuan Shi, Hanbo Li, Hezhi Lin, and Lianfen Huang. "A Nano-Quadcopter Formation Flight System Based on UWB Indoor Positioning Technology". In: *2018 13th International Conference on Computer Science Education (ICCSE)*. Aug. 2018, pp. 1–4. DOI: 10.1109/ICCSE.2018.8468720.

[37] Hongzhe Yu, Weifan Zhang, Xinjun Sheng, and Wei Dong. "State Estimation for Swarm UAVs Under Data Dropout Condition". In: *Intelligent Robotics and Applications*. Vol. 10984. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 81–91. DOI: 10.1007/978-3-319-97586-3_7.

[38] J.M. Amador, J.R. Martinez de Dios, J.L. Paneque, and A. Ollero. "An Agile Low-Cost Testbed for Multi-Drone Target Tracking". In: *2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)*. Cranfield, United Kingdom: IEEE, Nov. 2019, pp. 344–351. DOI: 10.1109/REDUAS47371.2019.8999718.

[39] Vicko Prkačin, Ivana Palunko, and Ivan Petrović. "State and Parameter Estimation of Suspended Load Using Quadrotor Onboard Sensors". In: *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. Sept. 2020, pp. 958–967. DOI: 10.1109/ICUAS48674.2020.9213840.

[40] Ira B. Schwartz, Victoria Edwards, and Jason Hindes. *Interacting Swarm Sensing and Stabilization*. June 3, 2021. DOI: 10.14339/STO-MP-SCI-341-08-PDF. arXiv: 2106.01824 [nlin].

[41] Artem Molchanov, Tao Chen, Wolfgang Hönig, James A. Preiss, Nora Ayanian, and Gaurav S. Sukhatme. "Sim-to-(Multi)-Real: Transfer of Low-Level Robust Control Policies to Multiple Quadrotors". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nov. 2019, pp. 59–66. DOI: 10.1109/IROS40897.2019.8967695.

[42] Benjamin Rivière, Wolfgang Hönig, Yisong Yue, and Soon-Jo Chung. "GLAS: Global-to-Local Safe Autonomy Synthesis for Multi-Robot Motion Planning With End-to-End Learning". In: *IEEE Robotics and Automation Letters* 5.3 (July 2020), pp. 4249–4256. DOI: 10.1109/LRA.2020.2994035.

[43] Benjamin Rivière, Wolfgang Hönig, Matthew Anderson, and Soon-Jo Chung. "Neural Tree Expansion for Multi-Robot Planning in Non-Cooperative Environments". In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 6868–6875. DOI: 10.1109/LRA.2021.3096758.

[44] Enric Cervera. "Try to Start It! The Challenge of Reusing Code in Robotics Research". In: *IEEE Robotics and Automation Letters* 4.1 (Jan. 2019), pp. 49–56. DOI: 10.1109/LRA.2018.2878604.

[45] Chijung Jung, Ali Ahad, Jinho Jung, Sebastian Elbaum, and Yonghwi Kwon. "Swarmbug: Debugging Configuration Bugs in Swarm Robotics". In: (2021).

[46] Thai Phan, Wolfgang Hönig, and Nora Ayanian. "Mixed Reality Collaboration between Human-Agent Teams". In: *IEEE Conference on Virtual Reality and 3D User Interfaces, (VR)*. 2018, pp. 659–660. DOI: 10.1109/VR.2018.8446542.

[47] Evgeny Tsykunov, Roman Ibrahimov, Derek Vasquez, and Dzmitry Tsetserukou. "SlingDrone: Mixed Reality System for Pointing and Interaction Using a Single Drone". In: *25th ACM Symposium on Virtual Reality Software and Technology*. Parramatta NSW Australia: ACM, Nov. 12, 2019, pp. 1–5. DOI: 10.1145/3359996.3364271.

[48] Yifan Zhang. "Experimental Implementation of Distributed Time-Varying Optimization Algorithms Using Crazyflie Platform". University Of California Riverside, 2019.

[49] Ben Remer. "Optimization of Last Mile Delivery With Unmanned Aerial Vehicle Assistance". In: (), p. 52.

[50] Shan Sun, Yifan Zhang, Peng Lin, Wei Ren, and Jay A. Farrell. "Distributed Time-Varying Optimization With State-Dependent Gains: Algorithms and Experiments". In: *IEEE Transactions on Control Systems Technology* (2021), pp. 1–10. DOI: 10.1109/TCST.2021.3058845.

[51] Jie Xu. "Experimental Implementation of A Distributed Time-Varying Constrained Optimization Algorithm by Using UAVs". MS Thesis. University Of California Riverside, 2020.

[52] Wei Dong, Sensen Liu, Ye Ding, Xinjun Sheng, and Xiangyang Zhu. "An Artificially Weighted Spanning Tree Coverage Algorithm for Decentralized Flying Robots". In: *IEEE Transactions on Automation Science and Engineering* 17.4 (Oct. 2020), pp. 1689–1698. DOI: 10.1109/TASE.2020.2971324.

[53] Jinho Kim, Charles D. Eggleton, Stephen A. Wilkerson, and S. Andrew Gadsden. "Cooperative Sensor-Based Selective Graph Exploration Strategy for a Team of Quadrotors". In: *Journal of Intelligent & Robotic Systems* 103.2 (Oct. 2021), p. 24. DOI: 10.1007/s10846-021-01485-0.

[54] Mark Roberts, Laura M. Hiatt, Vivint Shetty, Benjamin Brumback, Brandon Enochs, and Piyabutra Jampathom. "Goal Lifecycle Networks For Robotics". In: *The International FLAIRS Conference Proceedings* 34 (Apr. 18, 2021). DOI: 10.32473/flairs.v34i1.128553.

[55] Mark Lester F. Padilla, Pakpong Chirarattananon, Argel A. Bandala, Ryan Rhay P. Vicerra, Renann G. Baldovino, and Elmer P. Dadios. "Formation-Based 3D Mapping of Micro Aerial Vehicles". In: *2019 IEEE/SICE International Symposium on System Integration (SII)*. Jan. 2019, pp. 342–345. DOI: 10.1109/SII.2019.8700377.

[56] Faten Aljalaud and Heba A. Kurdi. "Autonomous Task Allocation for Multi-UAV Systems Based on Area-Restricted Search Behavior in Animals". In: *Procedia Computer Science* 191 (2021), pp. 246–253. DOI: 10.1016/j.procs.2021.07.031.

[57] Matteo Macchini, Thomas Havy, Antoine Weber, Fabrizio Schiano, and Dario Floreano. "Hand-Worn Haptic Interface for Drone Teleoperation". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. May 2020, pp. 10212–10218. DOI: 10.1109/ICRA40945.2020.9196664.

[58] Ragesh K. Ramachandran, Pietro Pierpaoli, Magnus Egerstedt, and Gaurav S. Sukhatme. *Resilient Monitoring in Heterogeneous Multi-Robot Systems through Network Reconfiguration*. Aug. 6, 2021. arXiv: 2008.01321 [cs]. URL: http://arxiv.org/abs/2008.01321 (visited on 09/30/2021).