UNIVERSITÄT KLAGENFURT
Institute of Networked and Embedded Systems

How to Act as Team
Multi-Robot Coordination

Bernhard Rinner

AI-DLDA Summer School, July 7, 2022

Acknowledgements

- Pervasive Computing group
  https://www.bernhardrinner.com

- Dronehub Klagenfurt
  https://uav.aau.at

Team Behavior of Multiple Robots

- Coordinate actions in space and time to achieve common goal
- Key coordination tasks include
  - Sharing of knowledge
  - Joint decision making
  - Resource allocation
- Complex problem with huge design space
  - Various constraints: energy, communication, deadlines, payloads etc.
  - Multiple objectives: mission, QoS, resource-efficiency
  - Different realizations: offline/online, centralized/distributed,
- Highly relevant for many MRS applications, eg.,
  - Entertainment, monitoring/inspection, search&rescue, transportation

Two Examples of Team Behavior

UNIVERSITÄT KLAGENFURT

Multi-drone constellation change

- Collision-free trajectory planning with MPC path following
- Framed as optimization problem with various constraints
- Offline coordination

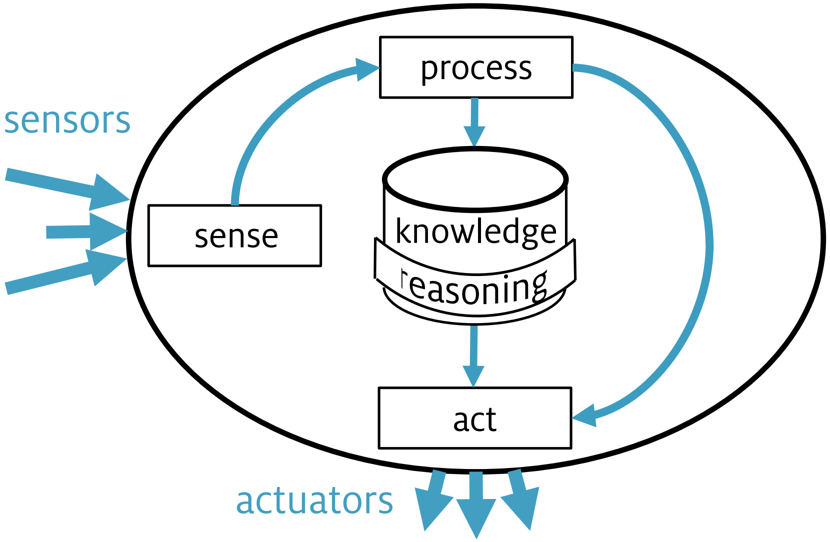
[Ladinig et al. Time and Energy Optimized Trajectory Generation for Multi-Agent Constellation Changes. In *Proc. ICRA*, 2021.]

Swarming and synchronization

- Collective behavior emerges from local processing and interaction
- Following self-organization principles („swarmalators“)
- Online coordination

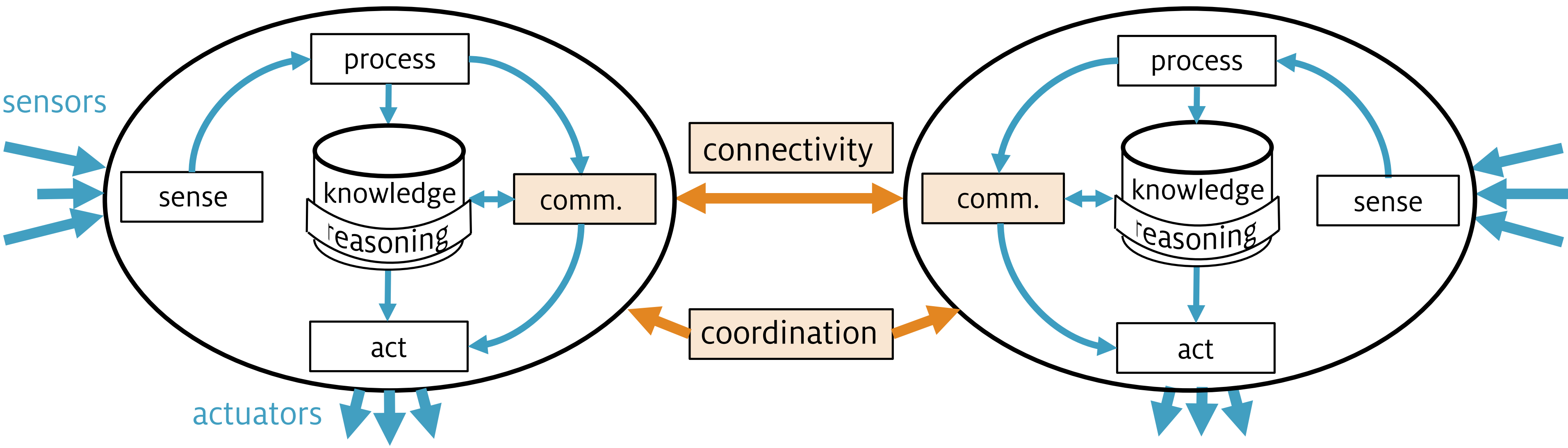
[Barciś, Bettstetter. Sandsbots: Robots that sync and swarm. *IEEE Access*, 2020.]

From single Autonomous Agents



- Traditional architecture of autonomous agent
  - With sense-process-act cycle
  - Maintains knowledge base with reasoning capabilities

To Multi-Robot Architecture



- Expanded data processing of individual robots by
  - Coordination of decision making
  - Robust wireless connectivity to transfer data with different QoS
  - Communication for optimized data distribution (what, when, to whom)

[Rinner, Bettstetter, Hellwagner, Weiss. Multidrone systems: More than the sum of the parts. *IEEE Computer*, 2021.

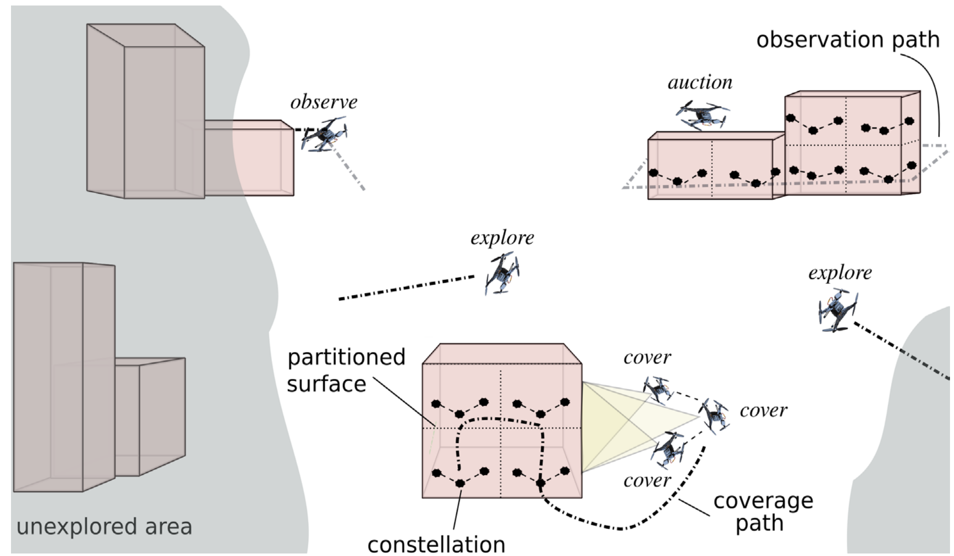
Terminology

- Robot is a special kind of agent (mostly) realized as a mechatronic construct
- Multi-robot system is a group of robots operating in the same environment
- MRS coordination can be classified wrt.
  - Cooperation: Do the robots cooperate to solve a problem?
  - Knowledge: How much knowledge do have robots about each other
  - Coordination: How much joint decision making is enforced?
  - Organization: What kind of decision structure is employed?
- Emphasis on MRS consisting of autonomous robots that collaborate in order to achieve a common global goal

[K.Geihs. Engineering Challenges Ahead for Robot Teamwork in Dynamic Environments. *Applied Sciences*, 2020.]

Multi-robot Application Scenario

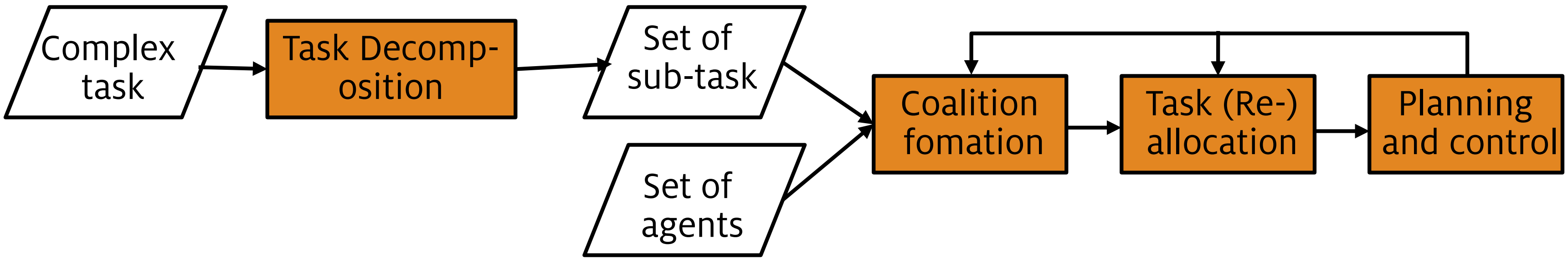
- Drones equipped with cameras collect data about environment
- Drones take on different roles
  - Explore unknown areas
  - Detect and partition objects
  - Analyze partitions in joint constellations
  - Move along observation paths



- Example for generic MRS coordination tasks

[Mazdin, Rinner. Distributed and Communication-Aware Coalition Formation and Task Assignment in Multi-Robot Systems. *IEEE Access*, 2021.]

MRS Work Flow



- Four main design blocks for MRS
  - Task decomposition: How to divide complex tasks into sub-tasks?
  - Coalition formation: How to form teams for sub-tasks?
  - Task allocation: How to assign sub-tasks to agents for execution?
  - Task execution: How to plan and control actions to complete sub-tasks?
- Varying degree of automation for design blocks
  - Human vs. robot
  - Offline vs. online

[Rizk, Awad, Tunstel. Cooperative Heterogeneous Multi-Robot Systems: A Survey. *ACM Computing Surveys*, 2019.]

Task Decomposition

- Task decomposition typically requires domain knowledge and considers robot capabilities
- Complexity, amount and interdependency of sub-tasks has impact on other blocks
  - Multi-agent tasks require collaboration of multiple robots
- Simple decomposition strategies
  - Spatial partitioning: decompose the environment
  - Temporal partitioning: decompose into sequence of sub-tasks
- Decomposition often performed offline and manually

Coalition Formation

- Coalition formation typically casted as (multi-criterion) optimization problem with distinction into
  - Single or multi-robot tasks
  - Static or dynamic teams
  - Deterministic or uncertain robots' behavior; decentralized approaches
- Search algorithms often adapted to coalition formation problems, eg.,
  - Ant colony optimization, particle swarm optimization, evolutionary algorithms
- Combined coalition formation and task allocation
  - Market-based approaches, voting games
  - Reinforcement learning frameworks

Task Allocation

- Task allocation is well-investigated with many approaches, but MRS introduce specific constraints (eg., spatial, temporal, sensing and actuation)
- Examples of MRS-specific task allocation approaches
  - Optimization or approximation travel distance of robots
  - Auctioning algorithms
  - Multi-agent reinforcement learning, distributed constraint optimization
- Combined coalition formation and task allocation typically improves performance at additional computation cost

Planning and Control

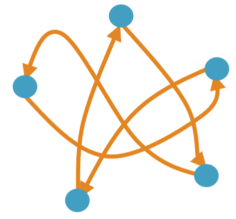
- MRS planning and control (decision making) determines sequence of actions agents should perform to complete their assigned task
- Multiple frameworks for solving decision-making problems, including RL, game theory, swarm intelligence, and graph-theoretic models.

Model	Degree of Scalability	Degree of Heterogeneity	Degree of Communication
Swarm Intelligence	High	Low	Low
Multi-agent MDP	Medium	Medium	Medium
Decentralized MDP	Medium	Medium	Medium
Multi-agent POMDP	Medium	Medium	High
Decentralized POMDP	Medium	Medium	High
Interactive POMDP	Low	Medium	High
Partially Observable Stochastic Games	Low	High	Medium

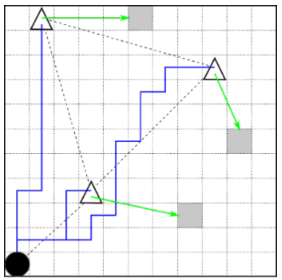
[Rizk, Awad, Tunstel. Cooperative Heterogeneous Multi-Robot Systems: A Survey. *ACM Computing Surveys*, 2019.]

Team Behavior Examples

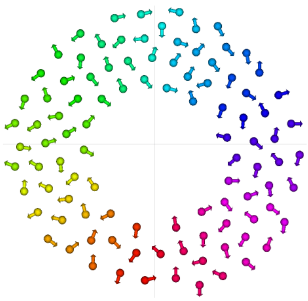
1. Collision-free and efficient path planning
   – Fast and safe constellation changes for many drones
   – Optimize for mission time and energy consumption



2. Cooperative route planning for surveillance
   – Multi robot routes for persistent surveillance with latency, idleness and energy constraints
   – Maintain continuous or intermittent connectivity



3. Swarming and synchronization
   – Emergent, self-organizing team behavior
   – Couple oscillators for joint synchronization and motion



[Dronehub Klagenfurt https://uav.aau.at]

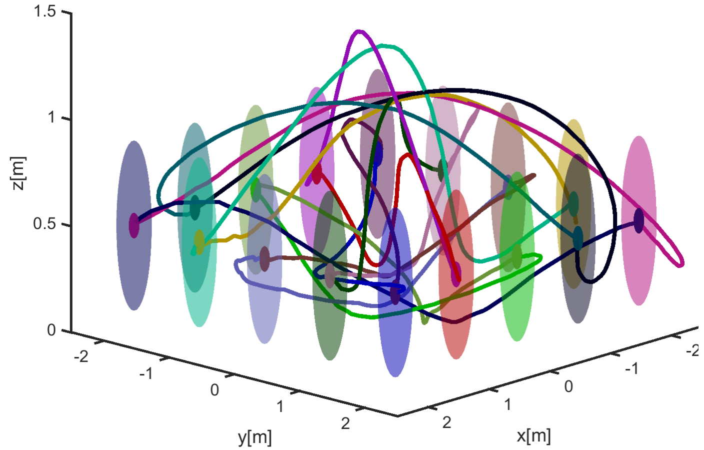
Motion Planning in MRS

- Planning the movement of robots to reach a target
  - Path planning: determine path (trajectory) from A to B
  - Route planning: navigate in environment along multiple (way) point
- Variations in modelling environment and robots, e.g.,
  - Continuous models, grids, graphs
  - Robot dynamics, linear movements models, discrete movement
- Constraints and objectives
  - Collision free movement, space limitations
  - Connectivity (permanent or intermittent)
  - Energy consumption, movement time, …

- Motion planning in MRS as complex coordination problem

Multi-drone Constellation Change

- Many drones simultaneously move from start to end constellation
  - avoiding collisions among each other
  - reducing transition time and energy



- Two-step path planning approach
  "Snap-optimized MPC with potential field extension (SOMPF)"
  - Compute initial, individual trajectories with model predictive control (MPC) and reduce potential collisions with potential fields
  - Generate final trajectories by minimizing snap within dynamic volume constraints

[Ladinig, Rinner, Weiss. Time and Energy Optimized Trajectory Generation for Multi-Agent Constellation Changes. In *Proc. ICRA*, 2021.]

Initial Trajectories

- System dynamics of drone i with discretization step size h

$$\boldsymbol{p}_i[\mathrm{k}+1] = \boldsymbol{p}_i[k] + h\boldsymbol{v}_i[k] + \frac{h^2}{2}\boldsymbol{a}_i[k]$$
$$\boldsymbol{v}_i[k+1] = \boldsymbol{v}_i[k] + h\boldsymbol{a}_i[k]$$

  with physical limits on position $\boldsymbol{p}_i$, velocity $\boldsymbol{v}_i$ and acceleration $\boldsymbol{a}_i$

- Collision constraints at time points k

$$\left\|\Theta^{-1}\left(\boldsymbol{p}_i[k] - \boldsymbol{p}_j[k]\right)\right\|_2 \geq r_{min} + \varepsilon_{ij}, \qquad \forall i,j,k|i \neq j$$

  with scaling matrix Θ and relaxation factor ε_{ij}

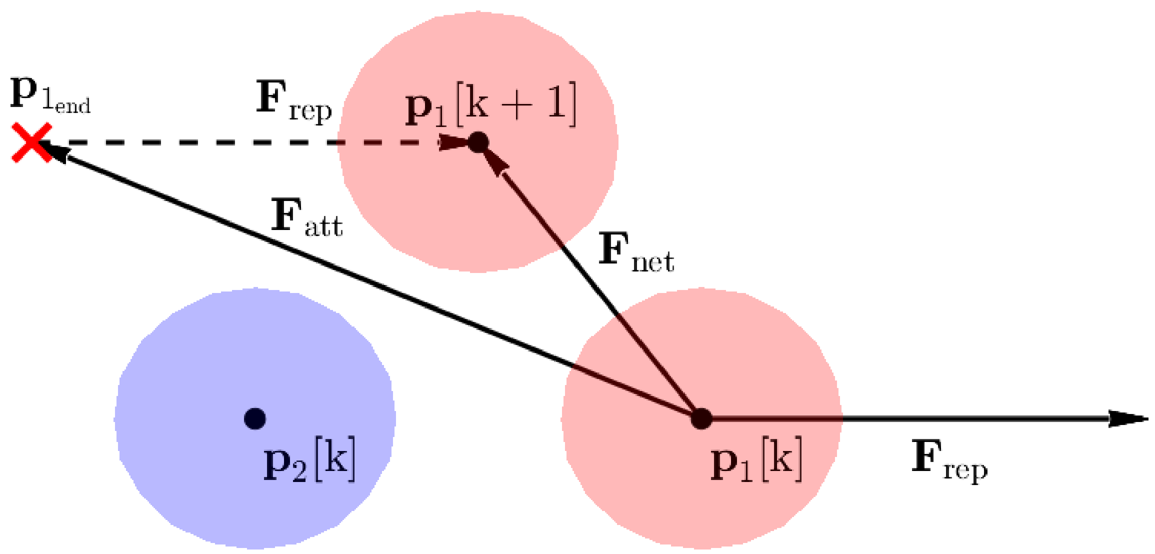
- Compute the predicted accelerations $\boldsymbol{U_i}$ over MPC time horizon by solving a quadratic program

$$\min_{U_i} \boldsymbol{U}_i^T \boldsymbol{H} \boldsymbol{U}_i + \boldsymbol{f}^T \boldsymbol{U}_i, \forall i$$
$$s.t. \boldsymbol{A}\boldsymbol{U}_i \leq \boldsymbol{b}$$

  Cost functions ($\boldsymbol{H}$ and $\boldsymbol{f}$) governs transition of each agent

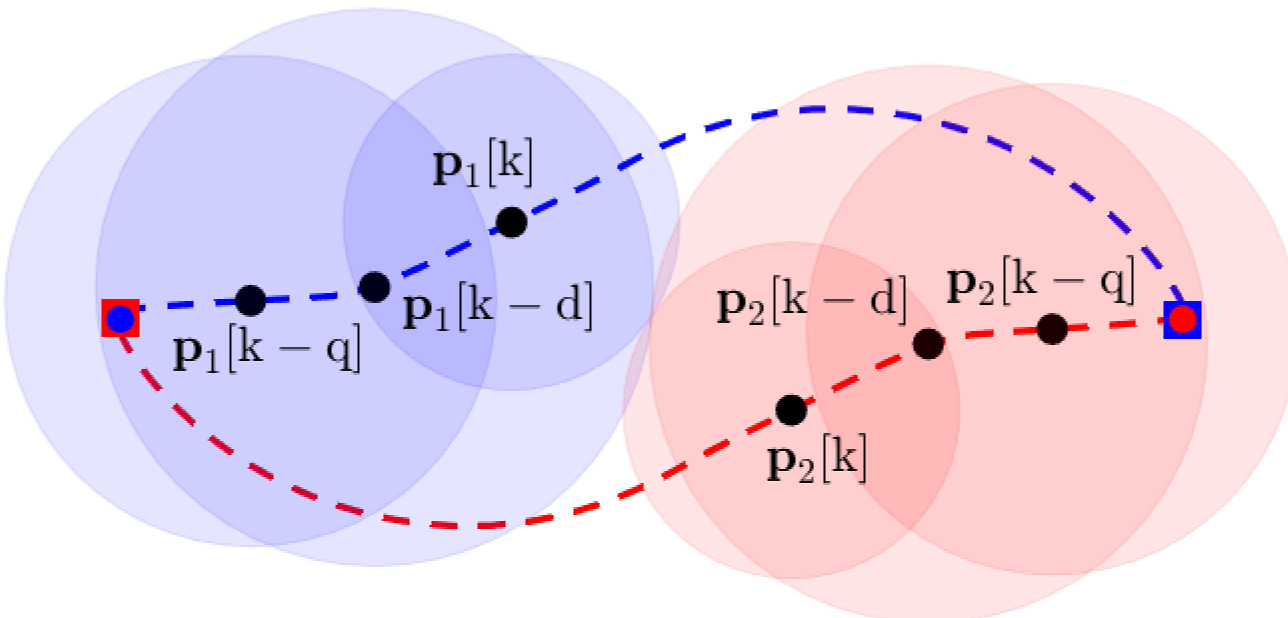
Collision Avoidance

- If collision at newly propagated $\boldsymbol{p}_i[k+1]$ is detected apply potential fields to bypass collision



Trajectory Refinement

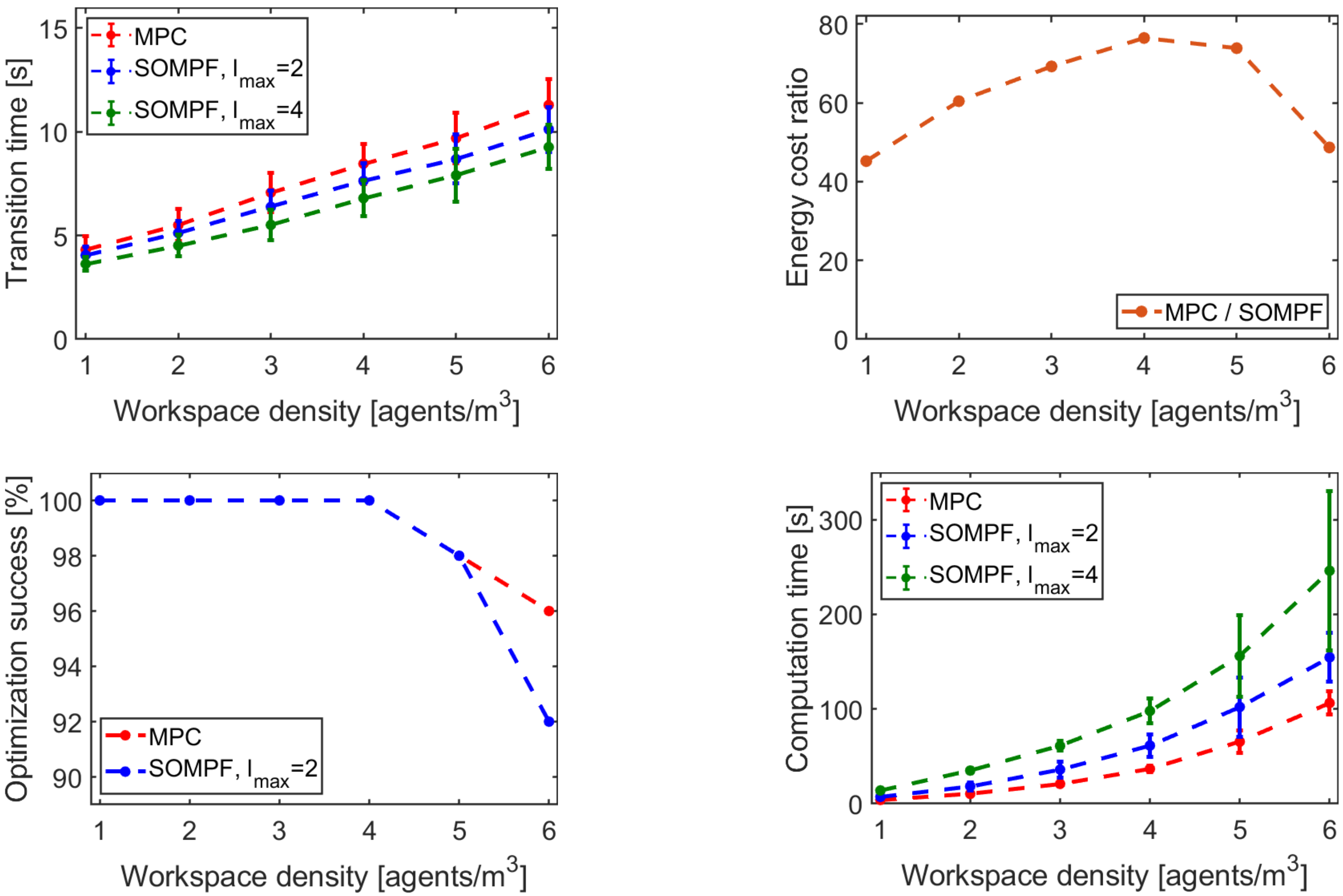
- Perform snap minimization with dynamic volume constraints



- Compute the maximum intersection-free corridor (spheres) at each time step
- Refined trajectories are faster and more energy-effient

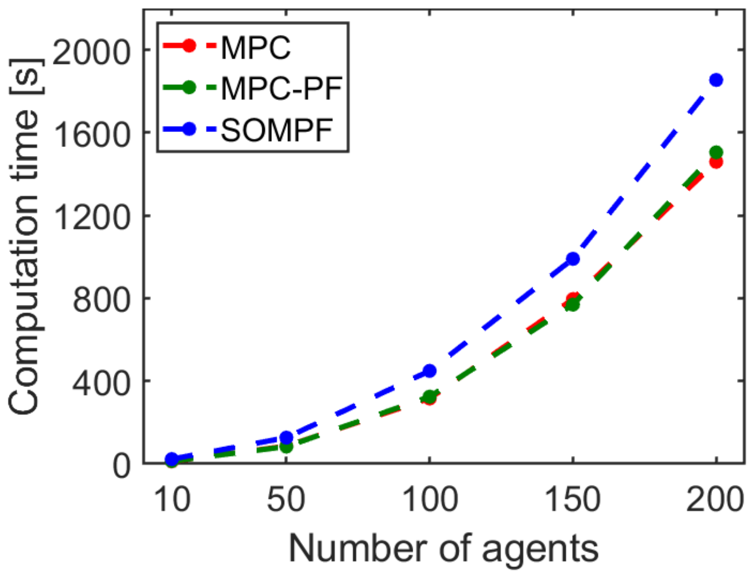
Performance Evaluation

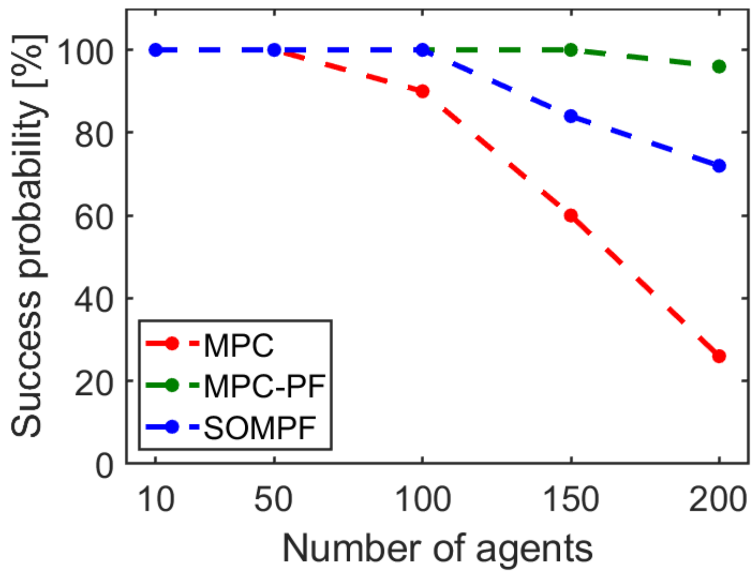
- Constellation changes within a fixed flight volume and varying agent density



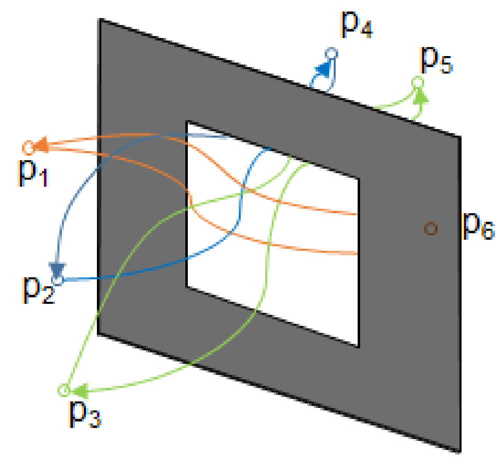
Performance Evaluation (2)

- Constellation changes with varying agents and constant agent density (1 agent/m3)

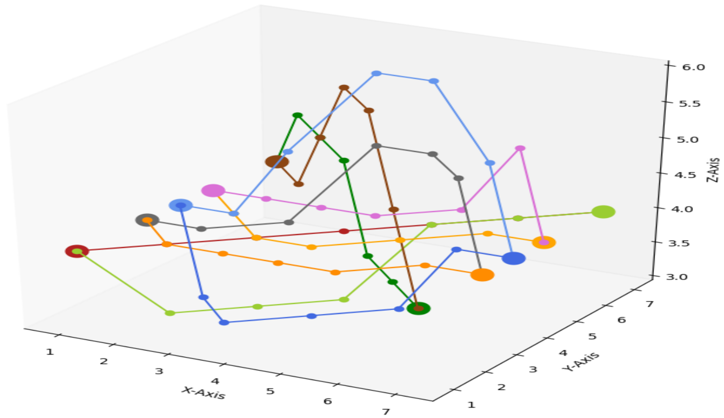


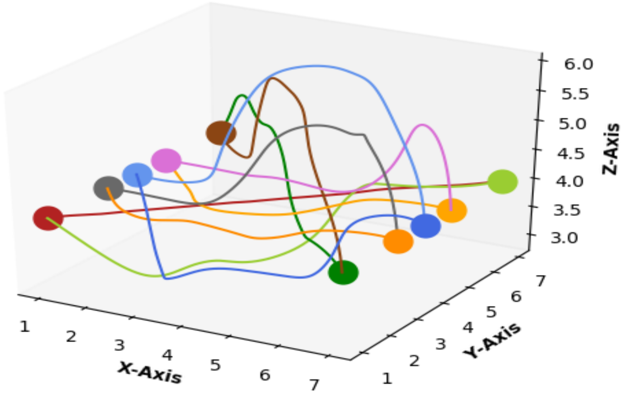


Constellation Change with Obstacles



- N drones simultaneously move in confined environments
- Path planning approach
  - Apply discrete multi-agent path planning (enhanced conflict-based search)
  - Generate trajectories within volume of discrete path (following snap minimization)

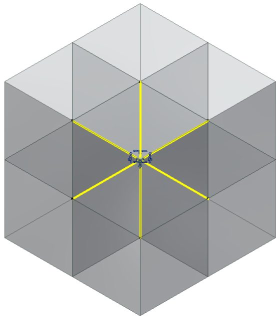


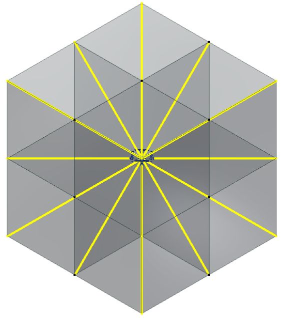


[Beyoglu, Weiss, Rinner. Multi-Agent Path Planning and Trajectory Generation for Confined Environments. In *Proc. ICUAS*, 2022.]

Waiting Events to Avoid Collision

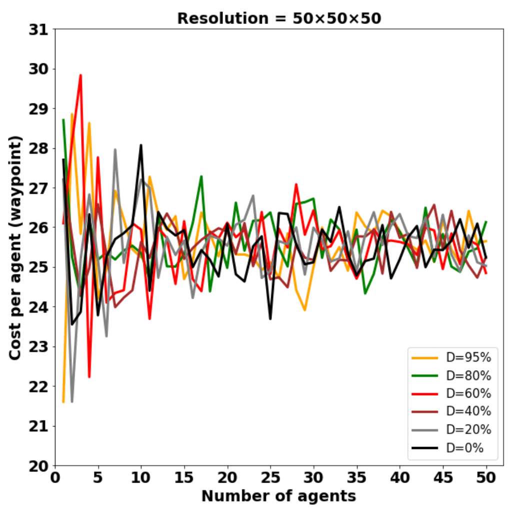
- Waiting events are problematic with discrete planning
  - If no conflict-free movement to any neighboring cell is possible, agents need to wait (“waiting event”)
  - Waiting events impede smooth and energy-efficient trajectory generation (deceleration, hovering and acceleration)
- Extend the available movement actions at each discrete planning step
  - Diagonal movement results in 26 potential movement actions within discretized 3D grid



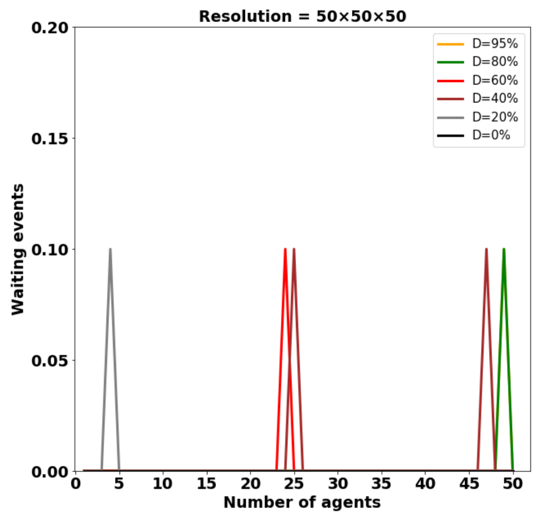


Simulation Results

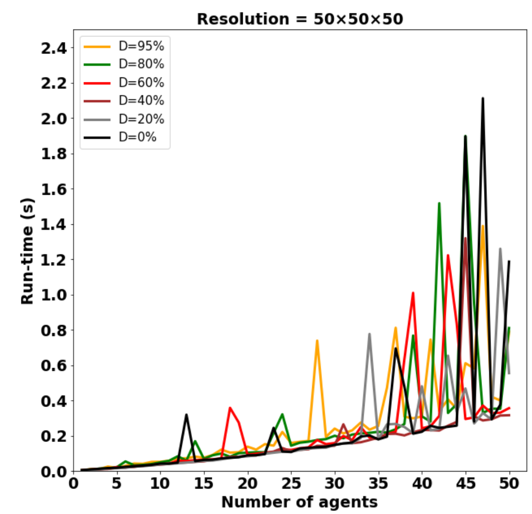
- Performance in environments with randomly placed obstacles and fixed spatial resolution
  - Mean values from 10 simulation runs



Average path length



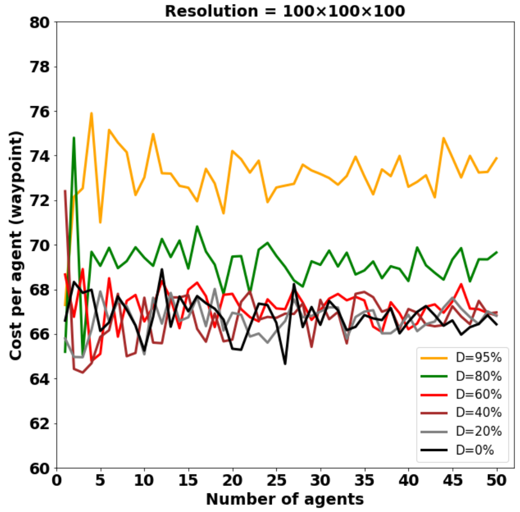
Introduced waiting events



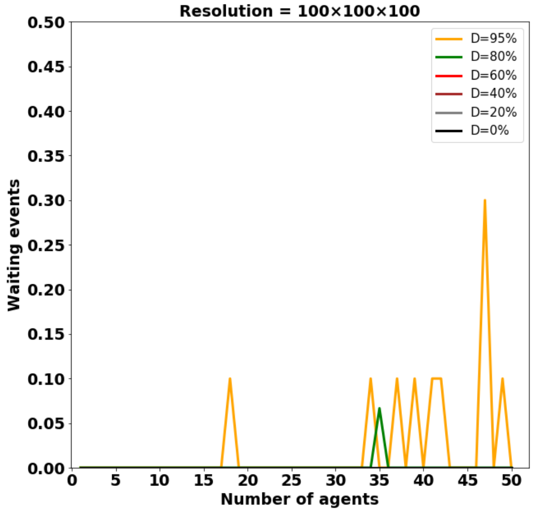
Planning time

Simulation Results (2)

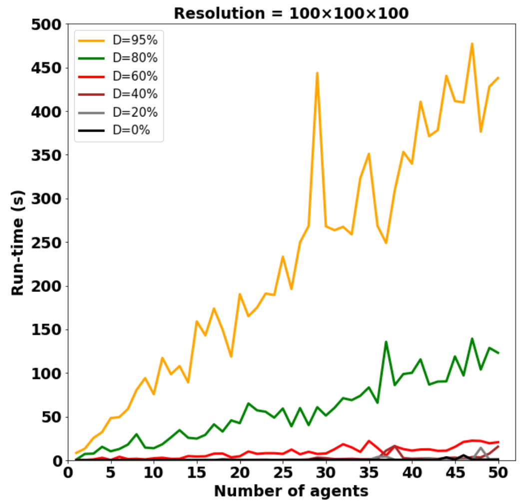
- Performance in environments with varying window border size and fixed spatial resolution
  - Mean values from 10 simulation runs



Average path length



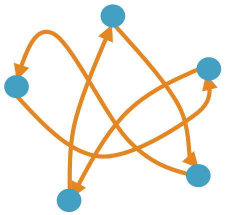
Introduced waiting events



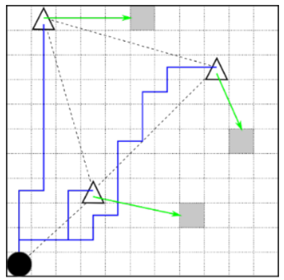
Planning time

Team Behavior Examples

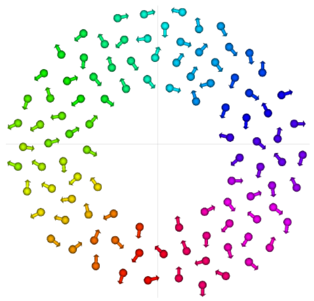
1. Collision-free and efficient path planning
   - Fast and safe constellation changes for many drones
   - Optimize for mission time and energy consumption



2. Cooperative route planning for surveillance
   - Multi robot routes for persistent surveillance with latency, idleness and energy constraints
   - Maintain continuous or intermittent connectivity

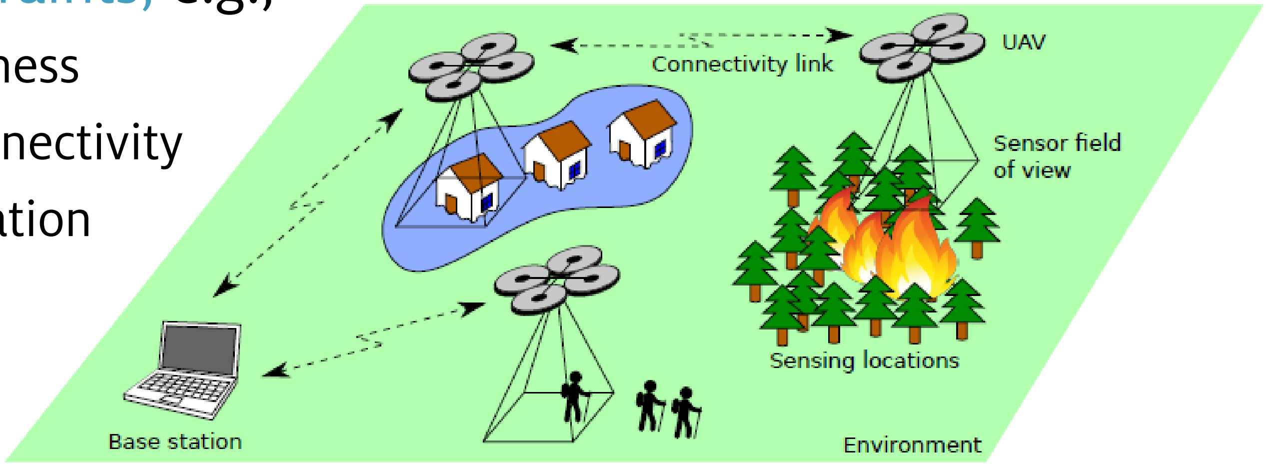


3. Swarming and synchronization
   - Emergent, self-organizing team behavior
   - Couple oscillators for joint synchronization and motion



Motion Planning for Surveillance

- Continuously monitor environment while maintaining multiple constraints, e.g.,
  - Sensing idleness
  - Network connectivity
  - Energy limitation

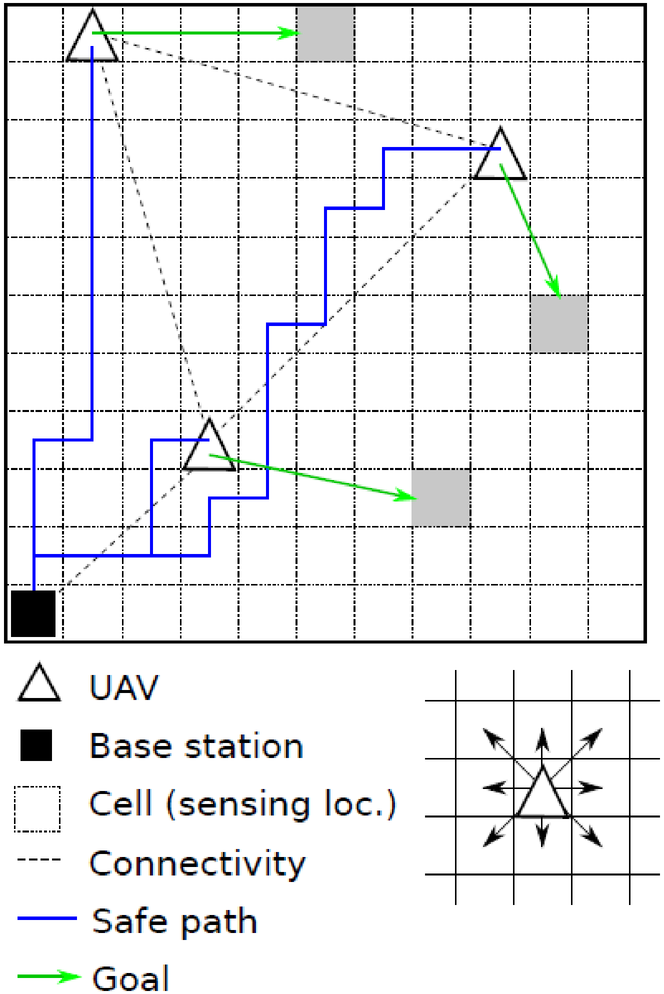


- Persistent surveillance (PS) or patrolling problem
  - Requires coordination of robots' movement in space and time
  - Fundamental problem for various MRS applications

[Scherer, Rinner. Multi-robot persistent surveillance with connectivity constraints. *IEEE Access.* 2020]

PS with Persistent Connectivity

- Plan paths that minimize worst idleness and keep network connected
  - Grid-based environments (partitioned into convex areas) and synchronous movement
  - Intermediate (relay) robots for maintaining connectivity (distance between robots)
  - If energy is limited return periodically to BS for recharging (along safe path)
  - Path planning is NP hard (proof by transformation to 3SAT)
- Investigate new motion planning heuristics with different planning horizon and coordination
  - Short vs. full horizon planning
  - Individual vs. cooperative planning



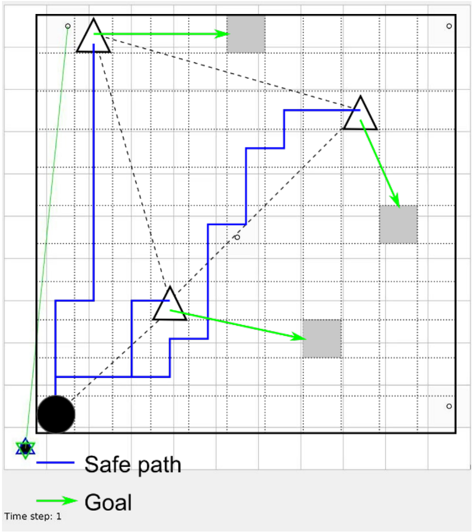
Short Horizon Planning (SH)

- Adopted goal-based heuristic [Nigam 2012]
  - Each drone u moves to cell c with highest $A(u,c)$

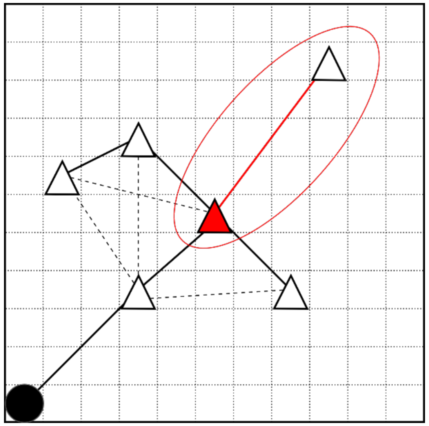
$$A(u,c) = I(c) + w_0 \cdot dist(u,c) + w_1 \cdot \min_{v \neq u} dist(v,c)$$

Idleness — Distance to cell — Minimum distance to other drones v

  - Weights w_0 and w_1 initially determined
    $A(u,c)$ values updated at each iteration



- Extended by safe paths for each drone to guarantee return to base station
  - with remaining energy without loosing connectivity
  - If one drone reaches its energy limit, all drones move along safe paths back to base

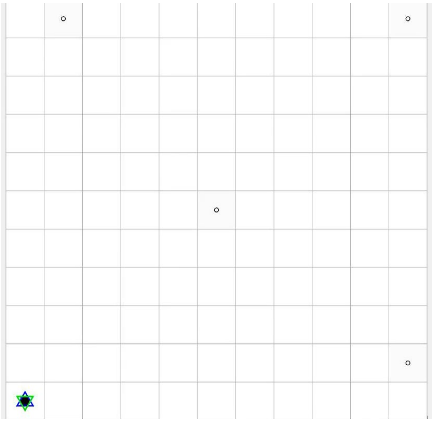


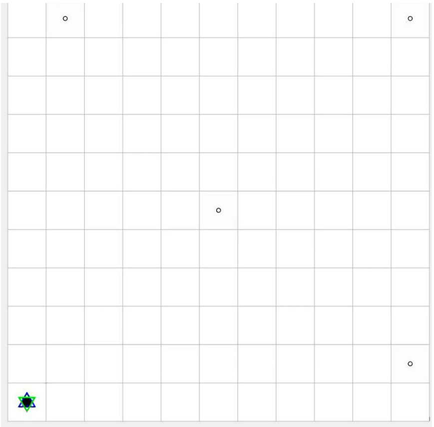
[Nigam et al. Control of multiple UAVs for persistent surveillance: Algorithm and flight test results. *IEEE Trans. on Control Systems Technology.* 2012]

Cooperative Planning

UNIVERSITÄT KLAGENFURT

- Avoid limitations of non-cooperative SH planning, e.g.,
  - Deadlocks (oscillations and mutual blocking)
- Short horizon cooperative (SHC) planning
  - Select robots for next sensing locations
  - Coordinate movement by graph matching (from current to next sensing configuration)
- Full horizon (FH) planning
  - Leader robot traverses tour through all sensing locations, other robots relay data to BS
  - If many robots available, several leader robots (partitioning)
  - Planning cycle for complete environment

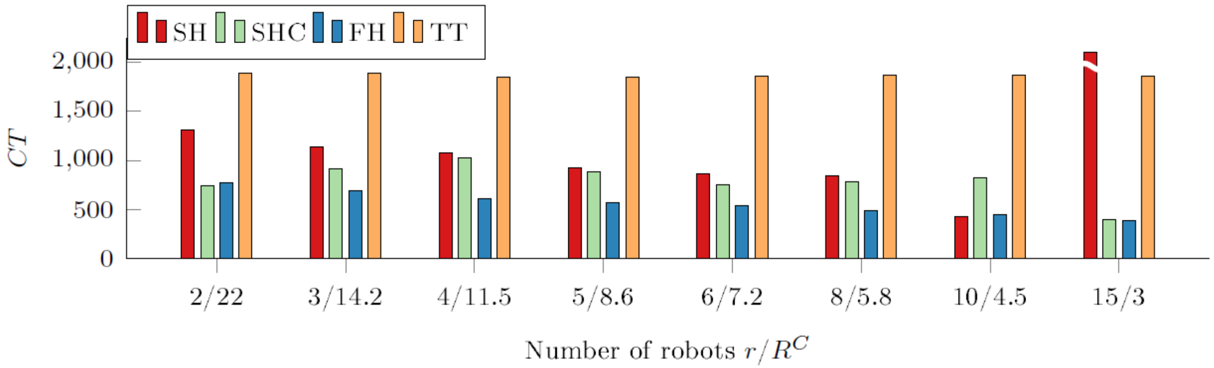




[Scherer, Rinner. Short and Full Horizon Motion Planning for Persistent multi-UAV Surveillance with Energy and Communication Constraints. In *Proc IROS*. 2017]

Comparison of Planning Algorithms

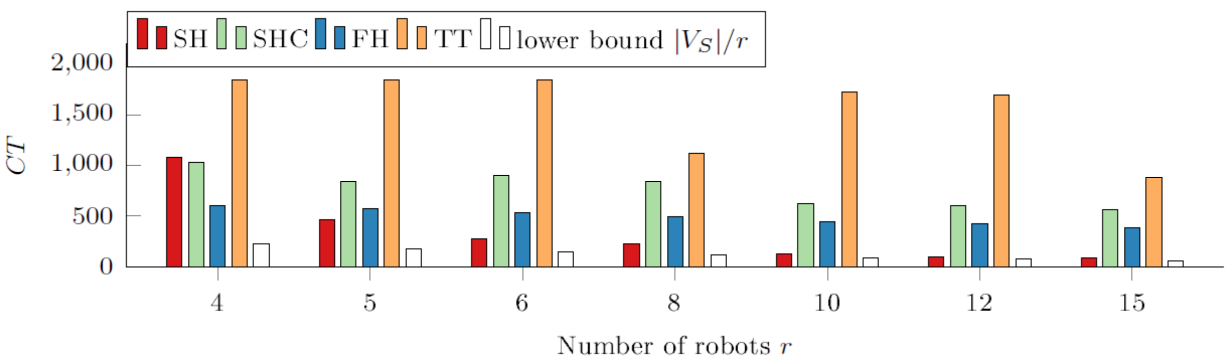
- Simulation study on 30 × 30 cells
  - First coverage time (CT) of entire area
    Varying number of robots with decreasing and fixed comm. range R^C



Strong comm. constraints

(all robots required for remote cells)

- FH outperfoms others
- SH fails to cover area for 15 robots
- TT tree traversal for partitioned areas as reference



Weak comm. constraints

(4 robots required for remote cells)

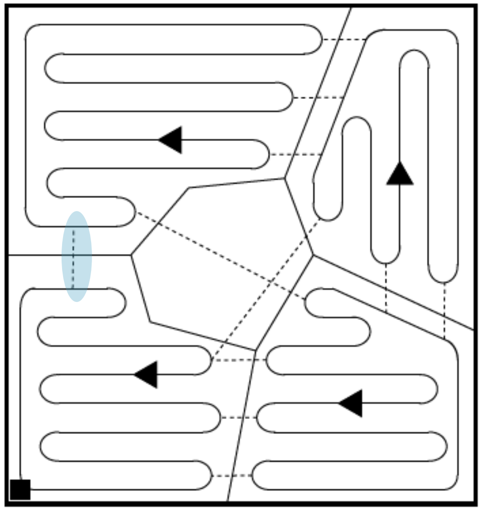
- SH approaches lower bound
- TT tree traversal for partitioned areas as reference

[Scherer, Rinner. Multi-robot persistent surveillance with connectivity constraints. *IEEE Access*. 2020]

PS with Intermittent Connectivity

UNIVERSITÄT KLAGENFURT

- Schedule movement along given tours to minimize worst idleness and data delivery latency
  - Data transfer only a meeting points (store-and-forward)
  - Who should meet when and where
  - Minimum delay scheduling is NP hard (proof by transformation to 3SAT)

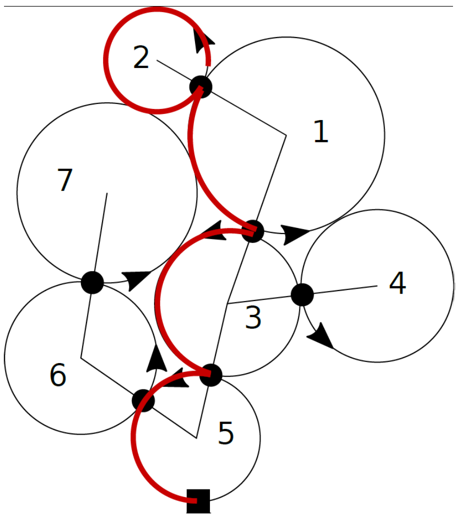


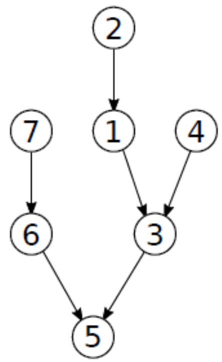
- Investigate new robot scheduling algorithms for given tours
  - Selecting travel direction and meeting points
  - Executing schedules onboard robots

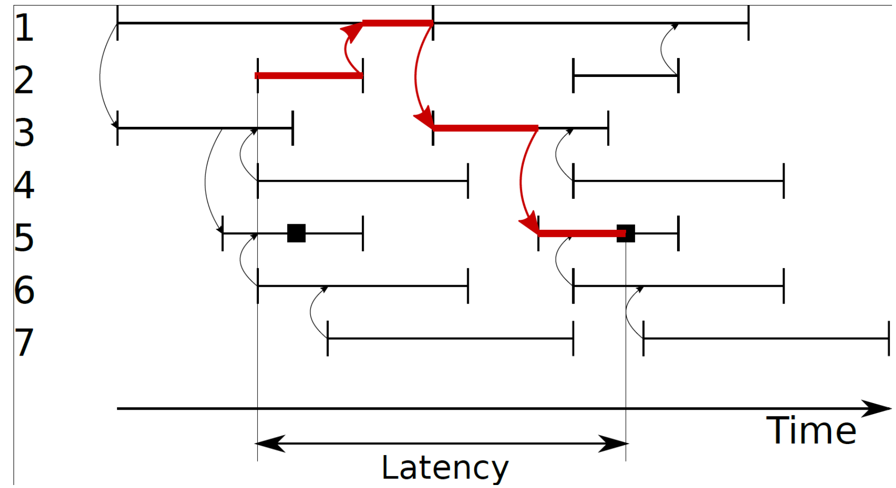
[Scherer, Rinner. Multi-Robot Patrolling with Sensing Idleness and Data Delay Objectives. *J. Intelligent & Robotic Systems.* 2020]

Scheduling based on Tour Graphs

- Model data transfers among patrolling robots as tour







- Robots patrol environment along given tours
- Data transfer via meeting points represented as graph (here tree)

- Determine for each robot when to move, wait and transfer
- Move along longest tour without waiting
- Construct schedule beginning with base station

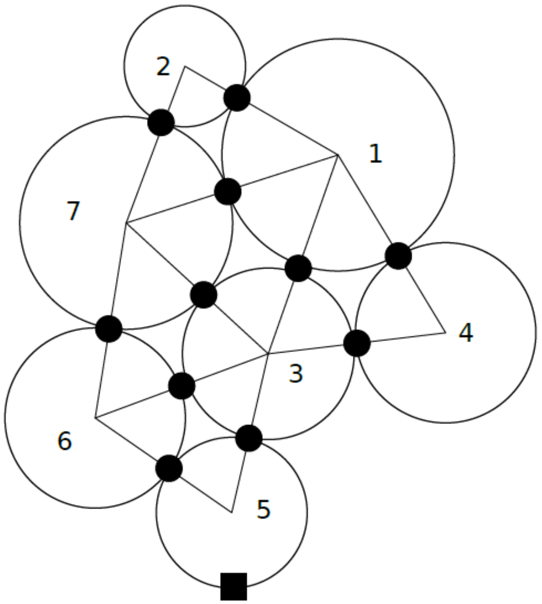
[Scherer, Rinner. Multi-Robot Patrolling with Sensing Idleness and Data Delay Objectives. *J. Intelligent & Robotic Systems.* 2020]

Various Problem Instances

UNIVERSITÄT KLAGENFURT

Minimize latency for given tours and bounded idleness

1. Select tour directions
   - Input: tour tree
   - Output: directions, schedule
2. Select tree (NP hard)
   - Input: tour graph, directions
   - Output: tree, schedule
3. Select tree and tour directions (NP hard)
   - Input: tour graph
   - Output: tree, directions, schedule
4. Select tree, meeting points and tour direction (NP hard)
   - Input: tour multi-graph (multiple meeting points between tours)
   - Output: tree, directions, schedule

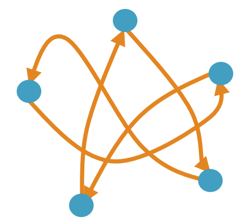


Minimize idleness for given tours and latency constraint

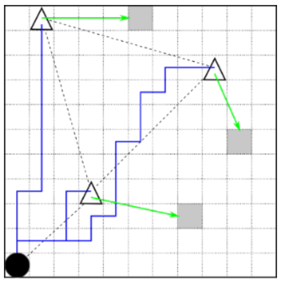
[Scherer, Rinner. Multi-UAV Surveillance with Minimum Information Idleness and Latency Constraints. *IEEE Robotics and Automation Letters.* 2020]

Team Behavior Examples

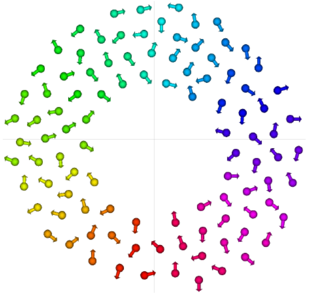
1. Collision-free and efficient path planning
   - Fast and safe constellation changes for many drones
   - Optimize for mission time and energy consumption



2. Cooperative route planning for surveillance
   - Multi robot routes for persistent surveillance with latency, idleness and energy constraints
   - Maintain continuous or intermittent connectivity



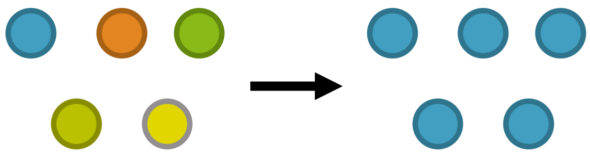
3. Swarming and synchronization
   - Emergent, self-organizing team behavior
   - Couple oscillators for joint synchronization and motion



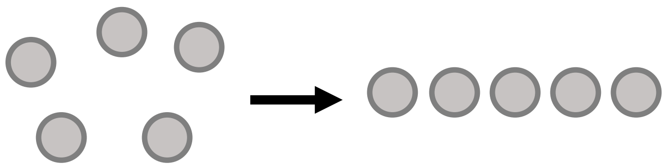
Selforganization for Drone Swarms

[lead by c. Bettstetter]

- Synchronization
  - Coordination of robots to achieve order in time domain
  - Adjustment of phases ϕ_i



- Swarming
  - Coordination of robots to achieve order in space domain
  - Adjustment of locations x_i



- Swarmalators
  - Unified model where robot's phase and location are mutually coupled
  - Emergent space-time patterns

[O'Keeffe, Hong, Strogatz. Oscillators that sync and swarm. *Nature Communications.* 2017]
[Barcis, Bettstetter. Sandsbots: Robots That Sync and Swarm. *IEEE Access.* 2020]

Swarmalators

Synchronization

The phases ϕ_i of robots influence each other.

E.g.: Phases synchronize to a common value, or "desynchronize" to differing values (splay states).

Swarming

The locations x_i of robots influence each other.

E.g.: Robots physically attract or repel each other based on their distance.

Swarmalators: bidirectional coupling between sync and swarming

The phases ϕ_i influence the movements $\dot{x}_i$, and
the positions x_i influence the phase dynamics ϕ_i.

E.g.: Entities with similar phases may attract or repel each other stronger, and close-by entities may synchronize faster.

Swarmalator Model

- N nodes indexed by i
- Location x_i and distance x_{ij}
- Phase ϕ_i and phase diff ϕ_{ij}

- Phase-dependent movement

$$\dot{\mathbf{x}}_i = \frac{1}{N} \sum_{j \neq i}^{N} \left[\mathbf{I}_1(\mathbf{x}_{ij}) \, F(\phi_{ij}) - \mathbf{I}_2(\mathbf{x}_{ij}) \right] \quad \text{with} \quad F(\phi_{ij}) = 1 + J \cos \phi_{ij}$$

Attraction Repulsion

- Location-dependent synchronization

Behavior is governed by two parameters J and K.

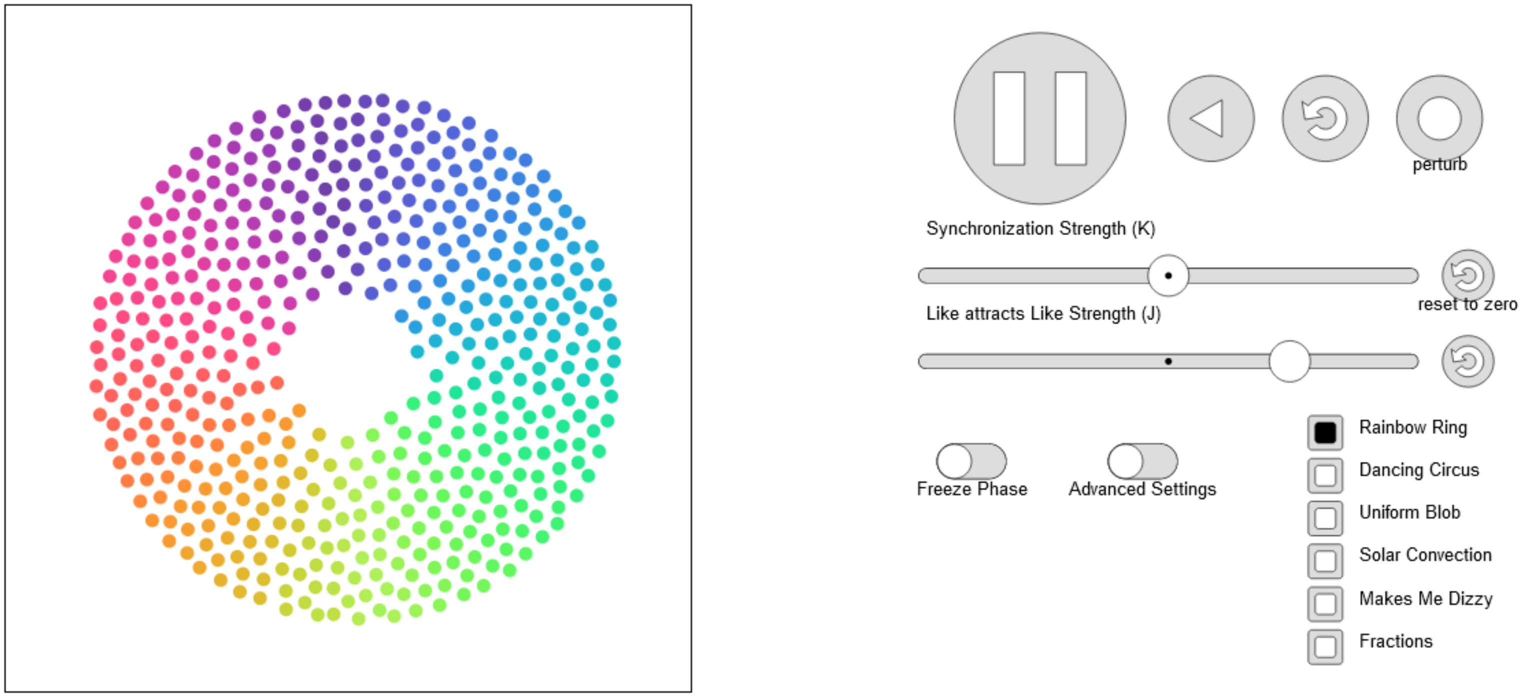
$$\dot{\phi}_i = \frac{K}{N} \sum_{j \neq i}^{N} H(\phi_{ij}) \, G_\phi(\mathbf{x}_{ij}) \quad \text{with} \quad G_\phi(\mathbf{x}_{ij}) = \frac{1}{\|\mathbf{x}_{ij}\|}$$

Attraction

[O'Keeffe, Hong, Strogatz. Oscillators that sync and swarm. *Nature Communications.* 2017]

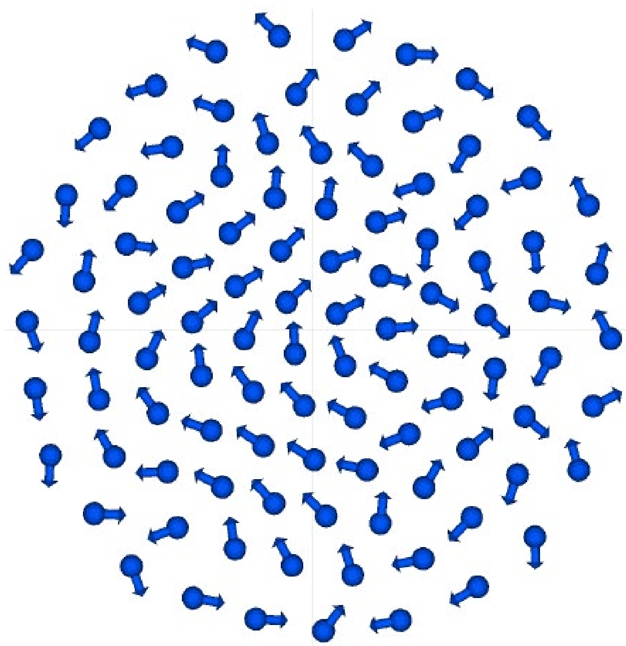
Complexity explorable “Swårmalätørs”

[by D. Brockmann]



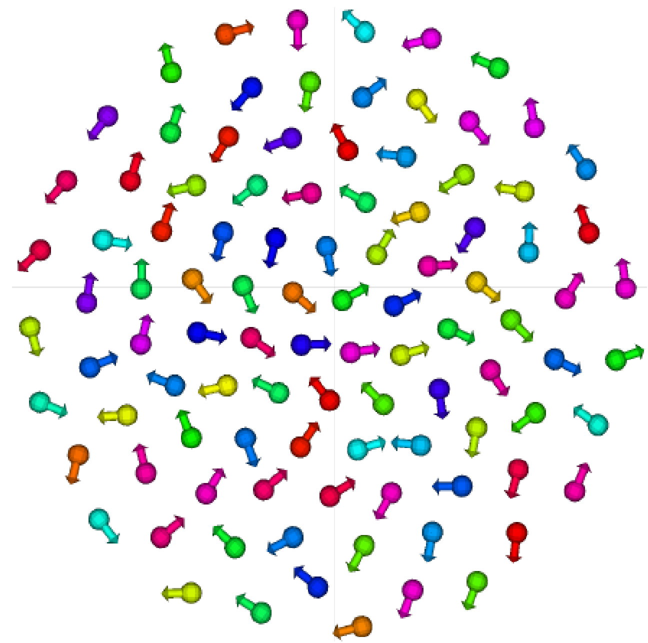
[Brockmann: Complexity explorable: Swårmalätørs - Pattern that emerge when collective motion and synchronization entangle, complexity-explorables.org, 2021.]

Swarmalator Patterns



Static sync

$(J, K) = (0.1, +1)$

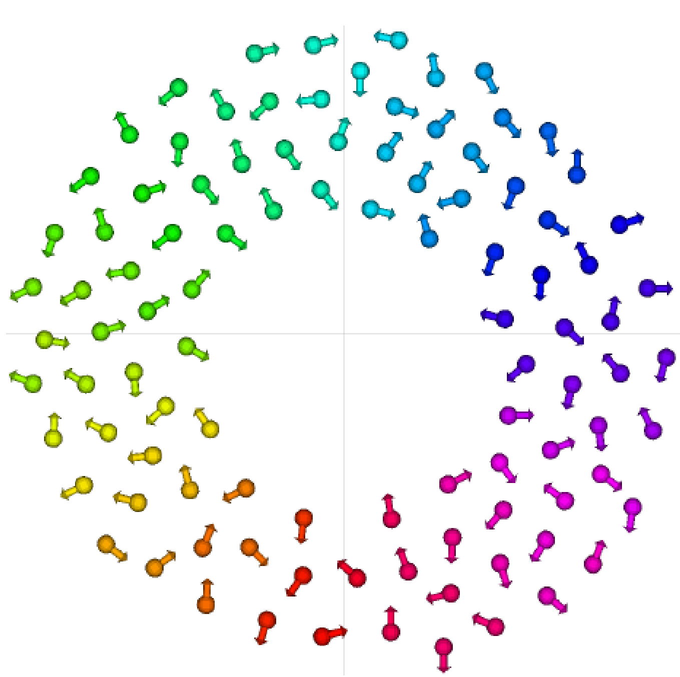


Static async

$(J, K) = (0.1, -1)$

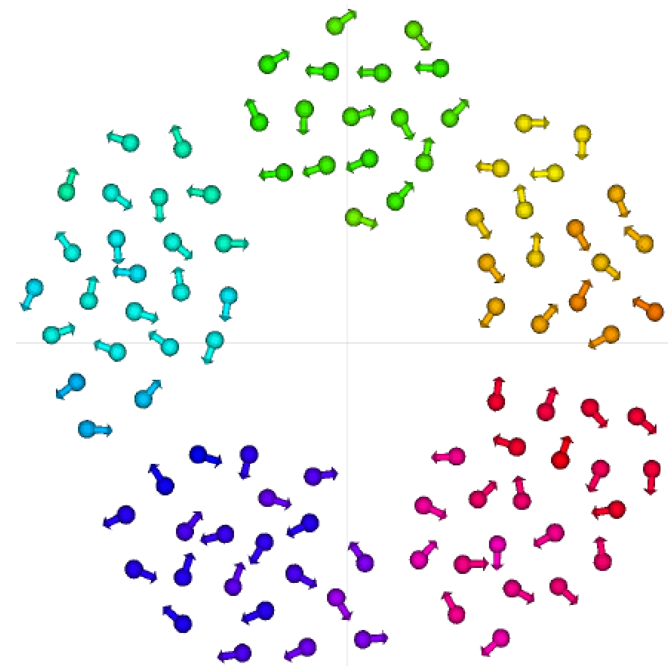
[O'Keeffe, Hong, Strogatz. Oscillators that sync and swarm. *Nature Communications.* 2017]

Swarmalator Patterns (2)



Static phase wave

(J, K) = (1, 0)



Splintered phase wave

(J, K) = (1, −0.1)

[O'Keeffe, Hong, Strogatz. Oscillators that sync and swarm. *Nature Communications.* 2017]

Swarmalators for Drone Swarms

[lead by c. Bettstetter]

- From theory to practice
  - How to adapt the swarmalator model for use in robotics?
  - Does the adapted model lead to identical patterns?

- Limitations of real multi-robot systems
  - Movement constraints and speed limitations
  - Collision avoidance
  - Interactions at discrete times via messages
  - Message loss and delay
  - Limited communication range

[Schilcher, Schmidt, Vogell, Bettstetter: Swarmalators with stochastic coupling and memory. Proc. IEEE Intern. Conf. on Autonomic Computing and Self-Organizing Systems (ACSOS), 2021.]

[Barciś, Barciś, Bettstetter: Robots that sync and swarm: A proof of concept in ROS 2. Proc. IEEE Intern. Symp. on Multi-Robot and Multi-Agent Systems (MRS), 2019.]

Deployment on Drones

Ground robot deployment: [Barcis et al. Robots that Sync and Swarm: A Proof of Concept in ROS 2. In *Proc MRS.* 2019]

Lessons Learnt

UNIVERSITÄT KLAGENFURT

- Complexity
  - Multi-robot systems perform highly interdependent tasks
  - Interdisciplinary research for tackling complexity
- Methodology
  - Experimental research with multi-robot systems is challenging
  - Interesting feedback from engineering work to basic research
- Non-technical issues
  - Safety, legal, ethical and regulatory aspects need to be considered
- Perseverance

Conclusion

- Two step path planning for free and confined environments
- Heuristic route planning for surveillance with connectivity constraints
- Swarmalators for emerging synchronization and swarming

- Coordination problems are challenging and fundamental for several MRS applications
  – Aim for efficient heuristics with performance bounds
  – Rely on various (simplifying) assumptions
  – Provide still many open research questions

Drone Research at Klagenfurt

- Started in 2008
- Developed into a key research area
  - 8 Profs
  - >20 PhDs & PostDocs
  - Dedicated doctoral school
- Covering various research topics
  - Autonomous navigation & coordination
  - Mission & path planning
  - Wireless communication
  - Interaction & various applications
- Opening Europe‘s largest drone hall
  - > 1000 m3 flight space
  - Motion capturing & 5G connectivity

Acknowledgements

- Pervasive Computing group
  https://www.bernhardrinner.com

- Dronehub Klagenfurt
  Profs. Bettstetter, Hellwagner, Weiss
  https://uav.aau.at