IEEE Transactions on Robotics, resubmission of paper **IEEE T-RO 20-0271**

“Multi-Robot Field of View Control with Adaptive Decentralization”

by Matteo Santilli, Pratik Mukherjee, Ryan Williams, and Andrea Gasparri

Reply to the Editor

Dear Prof. Billard,

Please find enclosed our point-by-point reply to each of the Reviewers’ comments. For the sake of clarity, the main changes in the revised version of the manuscript are highlighted in blue.

Best regards,
Dr. Matteo Santilli
On behalf of the authors

Comment:

> The topic of this manuscript is very relevant and timely for the TRO audience. However, in its current form the contribution is not strong enough for publication, as theoretically, the assumptions should be clarified and backed up wherever possible by stronger argumentation and stronger practical experimentation. The reviewers and editorial board would like to offer the authors the chance to respond to these in a revised manuscript.

Reply:

We thank the Editor for handling our manuscript and providing suggestions for improvement.

In the revised version of the manuscript, as well as in this reply, we have addressed the key areas of concern detailed in the following:

- The state of the art review section has been updated to include many more recent works;
- The description of our system model has been further detailed to improve its clarity;
- The roles of the three graphs involved in our framework, i.e., the sensing graph, the communication graph, and the control graph, have been better detailed and a Figure has been added to improve understanding;
- The assumptions required for our framework have been stated more clearly and with additional practical justification;
- An intuition of the control graph topologies that yield stable coordinated motion is now given;
- The description of the Critical and Planning states of the finite-state machine have been rewritten to improve their clarity;
- The description of the behaviour of the finite-state machine as well as the procedure that the robots use to interact with each other and with the centralized planner have been rewritten to improve their clarity;
- Finally, the experimental validation has been significantly improved. Due to space limitations, the manuscript highlights the results of a single experiment. However, we have included all new experimental results in the Appendix at the end of this document as well as in the new submitted video.

Beyond the above improvement areas, all major and minor concerns of the Reviewers have been addressed.

This document is organized as follows:

- Reply to Reviewer #2: page 3;
- Reply to Reviewer #5: page add page ;
- Reply to Reviewer #6: page add page ;
- Reply to Reviewer #7: page add page ;
- Appendix with the results of 5 experimental validations of our framework: page addpage .

We would like to thank the Editor and Reviewers for their comments and suggestions that allowed us to further improve, in our opinion, the quality of the original manuscript.

Reply to Reviewer #2

Comment:

> I would like to start by saying that I know experiments are hard and it takes a lot of work to do experiments on networked air vehicles.
> Overall, the paper itself is well written; however, key content to understand the framework is lacking and it appears that the simulations and experiments are not using the controller described in the paper.

Reply:

We thank the Reviewer for their feedback on our manuscript. As detailed below in replies to specific comments of the Reviewer, in the revised version of the manuscript we made substantial additions and changes in order to significantly improve clarity. Furthermore, the results of a new set of experiments have been added in the validation section to clearly demonstrate the entirety of our proposed framework. Finally, all minor issues pointed out by the Reviewer have been addressed.

Comment:

> Furthermore, the claimed contribution was in a limited FOV sensing neighbors, but this is contradicted in the paper given somehow agents can share state information if they are going to collide. This implies there is a centralized planner all the time not just when a switch occurs as the paper claims.

Reply:

We thank the reviewer for pointing out this apparent discrepancy. We realized that the explanation of the finite-state machine was not clear enough regarding the issues the Reviewer raised. In the revised version of the manuscript, Sections VI.A, VI.B, and VI.C have been significantly rewritten to improve their clarity. Figures 6 and 8 have also been updated to reflect the changes in the text.

We now provide an in-depth illustration of our revisions and describe how they address the Reviewer's concern. First, it is important for us to differentiate between state information shared via communication and sensing. In our framework the robots *do not* receive state information through communication with a centralized entity. Instead, states are *sensed locally* by a sensor with a field of view (the primary sensor, e.g., a camera), or a near-field sensor for avoiding collisions (the secondary sensor, e.g., ultrasonic detectors). Most importantly, the FOV sensor is the only source of *distributed team-level* control as it has a much larger effective range than the near-field sensor (practically this assumption holds on our DJI Matrice platform where ultrasonic collision sensors are effective only when vehicles are in very close proximity). Thus, when the system is operating with distributed control, only FOV sensors are used and there is no constant connection to a centralized entity. Critical interactions that may lead to collisions are only recognized when vehicles enter into the near-field sensor's much smaller radius. To further clarify, the information shared through distributed communication consists only of i) requests for disconnections enforced by the conditions in eq. (74) (which are reported below for the sake of completeness) and ii) of information about the neighborhood of each robot in order to distributively evaluate the topology conditions required by the Reactive state ($\overline{\mathcal{L}}_p \succeq 0$).

$$\mathcal{N}_i^{c}(p') = \{j \in \mathcal{V} : (i,j) \text{ is a critical interaction}\}, \tag{74a}$$

$$\mathcal{N}_i^{+}(p') = \emptyset, \tag{74b}$$

$$\mathcal{N}_i^{-}(p') = \emptyset. \tag{74c}$$

Specifically, each robot that detects a critical interaction (with their near-field sensing) informs their communication neighbors (i.e., robots that are inside the communication radius ρ_{co}) that they are going to ignore all interactions that are not considered critical. Thus, in our framework communication is *event-triggered* and serves only to communicate information when a transition in the finite-state machine occurs.

For example, consider the scenario depicted in Figure R.1 here. Notably, the proposed scenario would not require the usage of the finite-state machine since any configuration with 3 robots is stable according

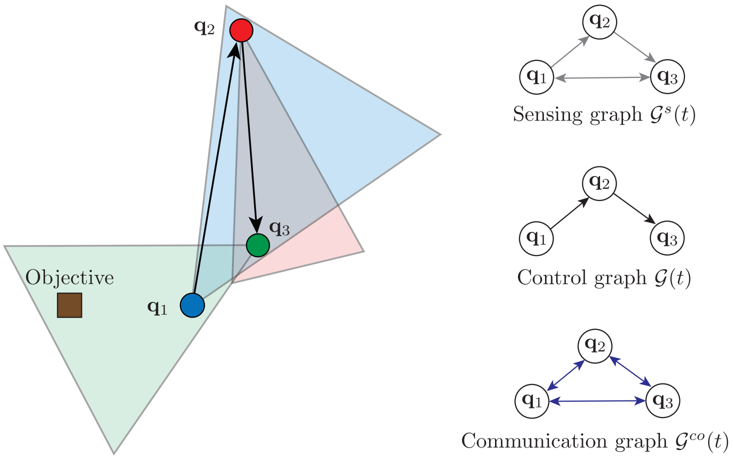

q2
q3
Objective
q1
q2
q1
q3
Sensing graph $\mathcal{G}^s(t)$
q2
q1
q3
Control graph $\mathcal{G}(t)$
q2
q1
q3
Communication graph $\mathcal{G}^{co}(t)$

Figure R.1: Scenario with critical interaction between robots q_1 and q_3.

to the results of Theorem 1. For the sake of presentation, however, we assume here that that is not the case and we illustrate a possible evolution of the system.

Three robots q_1, q_2, and q_3 are performing some collaborative objective which requires the following operations: i) robot q_3 must reach the brown "Objective" box, ii) robot q_2 must follow and not lose interaction with robot q_3, and iii) robot q_1 must follow and not lose interaction with robot q_2. This collaborative objective, however, implies that robot q_3 must transit close to robot q_1. In doing so, the two robots may collide with each other.

To avoid this potentially unstable (and therefore dangerous) configuration, the two robots able to sense each other and recognize that their interaction fits the criteria to be considered critical, start the procedure to transition to the critical state. In order to so, they send a broadcast message within their communication radius ρ_{co} (not shown in Figure R.1 but represented by the communication graph $\mathcal{G}^{co}(t)$) informing their neighbors that they are going to focus on the critical interaction only, disconnecting themselves from the rest of the network. Note that by Assumption 3, ρ_{co} is greater than the sensing range (this again matches our practical implementation where long-range Wifi provides much wider coverage than FOV cameras used for control). After the disconnection occurs, the two robots, relying on local measurements only, perform collision avoidance and reach a safe configuration. After the collision has been averted, the multi-robot system can either go back to the reactive state (if its requirements are met) or ask for help (in an event-triggered manner) from the centralized planner which will guide them back into a configuration for which the reactive state (and distributed control) can be enabled again.

Suppose for this case that after the critical interaction has been addressed, robot q_2 is still sensing robot q_3, and robot q_3 is still sensing q_1. In this case, the system could transition immediately into the Reactive state since the robots know that this topology is stable. Before doing so, the robots rely on the communication graph $\mathcal{G}^{co}(t)$ to inform the rest of the network that they can safely reconnect to the old neighbors. After all robots have been informed that everyone can connect back with the old neighbors, the system switches into the Reactive state.

In a more complex scenario, the robots may find themselves with different *sensed* neighbors. In that case, before enabling the transition to the Reactive state, the robots must collaboratively verify that the new topology p' is able to guarantee stability, i.e., that the related matrix $\overline{\mathcal{L}}_p$ is positive semi-definite. To do this the robots could utilize a distributed finite-time eigenvalues estimator as the one presented in [4]. In particular, the distributed evaluation in [4] could be carried out in our context by exploiting the communication graph where each robot informs the others on which neighbor it is able to sense, allowing every robot to compute the matrix $\overline{\mathcal{L}}_p$ and then start the process to locally estimate its eigenvalues.

If the evaluation fails, i.e., the new topology would not guarantee stability, then the system is forced to go through the Planning state in which the centralized planner will relocate the robots in a configuration that enables the Reactive state. Similarly, if the team does not have the capabilities to carry out the distributed evaluation of a topology's stability (e.g., an unexpected failure has occurred), then the system is forced through the Planning state after the resolution of any critical interaction.

In summary, when the robots address a critical interaction, e.g., a potential collision, the robots do not share state information with each other but rely only on local sensing to avoid the dangerous situation.

Finally, we believed that the usage of the three graphs involved in our framework was not clear, hence an explanatory paragraph has been added before the start of Section IV.A to clarify this topic.

References:

[4] P. Mukherjee, A. Gasparri, and R. K. Williams, "Stable motion and distributed topology control for multi-agent systems with directed inter- actions," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec 2017, pp. 34553460.

Comment:

```
Furthermore, it appears that the state of each agent is always measurable, and
it seems some localization system (motion capture or GPS) is used to directly
measure the state implying local feedback is not used anywhere.
```

Reply:

We thank the Reviewer for the comment. We apologize for the lack of clarity. The robots are indeed able to measure their own state but the control laws in eqs. (7) and (82) require only local measurements to be computed.

In our experimental validation, we use the Pozyx localization system to obtain the pose information for each robot. The detailed description of the pozyx localization system is provided in Appendix B. Once the pose information is obtained for each robot, it is then fed to the local controller that is running on the on-board computer of each robot.

To reflect this assumption we added the following paragraph in the description of our system:

> *We assume that the robots share a common reference frame or have access to absolute positions by GPS.*

Comment:

```
Video: The simulations and experiment videos are a nice addition, but it is
hard to tell what is going on. It seems like the authors are showing the top
down of the agents FOV while the agents move around. It is not clear what the
goal formation is. Additionally, are the agents only using one neighbor for
localization? That appears to be the objective for the decentralized controller.
```

Reply:

We thank the Reviewer for the comment and we regret the lack of clarity. The objectives of the simulations and experiments are to demonstrate the following:

1. limited FOV control of a multi-robot system;
2. the functionality of the adaptive decentralization mechanism of the machine states $\mathfrak{R}, \mathfrak{C}, \mathfrak{P}$.

The objective for the robot team itself is to execute a classical leader-follower formation with six robots in the case of Gazebo simulation, and with four robots for the real experiments. A detailed video is submitted along with the revised version of the manuscript which shows the top view of the actual robots as well as a synchronized MATLAB plot created with the data collected from the Gazebo simulation and the real experiments. This MATLAB plot gives the reader a perspective of how the robots are able to maintain their respective neighbors in the assigned perimeter of the FOV. We ask the Reviewer to refer to Appendix C of this document for a detailed analysis of our new real-world experiments as well as the new video submitted with the revised manuscript and improved explanation in the manuscript itself. These experiments, video, and explanations provide a much deeper illustration of our framework that we hope alleviates any lack of clarity.

With respect to the question on localization using one neighbor, our FOV controller is fed the localization information directly from on-board Pozyx system measurements which uses distributed ultrawide-band (UWB) time-of-flight information for global localization. A local FOV controller runs on the on-board computers of each robot where it receives its own localization information as well as the localization information of its neighbor which is in the FOV based on the Pozyx UWB localization system. Then the on-board FOV controller computes the control law local to each robot and generates lower-level commands (i.e. velocity and yaw rate, or position and yaw). Thus, the robots' positions are not determined by the neighboring agent via a FOV sensor, but are received directly from the Pozyx localization sensors that are rigidly attached to the robots. To summarize, we simulate FOV sensors for use in distributed motion control based on time-of-flight Pozyx sensors for localization. We have chosen this practical implementation as we have found that experimentally relative camera systems are still not sufficiently robust for real-time cooperative control of teams of robots outdoors. We therefore see our work as a stepping stone to future real-world implementations where the FOV sensors themselves are used for both localization and control (e.g., as vision algorithms and sensors continue to improve). We also point out that our Pozyx localization system itself has documented performance variability over a FOV, and thus we can also interpret our real-world experiments as an initial illustration of controlling robots with onboard FOV sensors that provide information for both localization and control.

Comment:

> Losing the line of sight is a challenging problem and I understand space requirements to perform these experiments; however, I recommend using videos that show the camera feeds for each agent enabling the viewer to see what is happening more clearly. Additionally, it is not clear what approach is used to find a neighbor in the camera FOV. Is the relative position measurement simulated using the localization system, that is, it seems like each agent can measure its pose and are they sharing that using a simulated FOV? What information is shared between agents? I assume that no information is directly shared given the graph is directed but what measurements are simulated?

Reply:

We thank the Reviewer for the comment. Yes, the reviewer has correctly pointed out that we have space limitation when it comes to the real-world experiments as the perimeter of the experiment is decided by the limited number of Pozyx anchors that we have set in the environment and the size of our outdoor drone facility. We typically conduct experiments in an area of $30m \times 20m$. The primary reason we do not take camera feeds from on-board cameras is because it is computationally expensive and as explained above the feeds are still not robust enough or real-time multi-robot control in outdoor environments. Our pozyx localization system is accurate to only $10cm$ (claimed by manufacturer) and it has an update rate of about $20Hz$, thus we cannot afford computationally to run a camera feed in real-time with the experiments. As on-board computation, localization systems, vision algorithms, and cameras continue to improve, we hope to remedy these sacrifices in experimental setup.

In our experimental setup, the robots gain information about the relative position of the their neighbors through the Pozyx localization system (described above) and simulate the presence of a FOV only virtually, i.e., if the relative position of a potential neighbor j is inside the simulated FOV of the robot i then robot i considers robot j as its neighbor. Hence, no state information is shared among the robots.

TO DO

discussion about identification / how agent recognize neighbors within FOV as in the paper

Comment:

> Also, what information is shared when the centralized planner takes over? How does each agent know that the centralized planner is going to take over? The video and paper should explain these types of detail more clearly to really improve the presentation.

Reply:

We thank the Reviewer for the comment. We realized that the explanation of the Planning state as well as the description of the finite-state machine were not very clear. The related sections and paragraphs of the revised version of the manuscript have been edited to improve their clarity.

In the Planning state a subset of robots $\mathcal{S} \subseteq \mathcal{V}$ is being driven by a centralized planner out of an undesired configuration towards a configuration in which the multi-robot system can switch back to the

Reactive state and resume normal operations. The subset of robots not involved in this operation, i.e., the robots $i \in \mathcal{V} \setminus \mathcal{S}$, during the Planning state continue following the prescribed control law as they were doing in the Reactive state.

Indeed, although the whole multi-robot system is associated to a single global finite-state machine, each robot possesses its own local version of the finite-state machine which dictates the actions of the robot itself. The robots involved in the operations led by the planner are in the local Planning state, i.e., $\mathfrak{P}_i = 1$ with $i \in \mathcal{S}$, whereas the rest of the team is in the local Reactive state, i.e., $\mathfrak{R}_i = 1$ with $i \in \mathcal{V} \setminus \mathcal{S}$, and continue pursuing their objective. Hence, at a given time, only a subset of robots interacts with the planner.

As detailed in the revised version of Assumption 4, the centralized planner, when activated, is able to access the latest state information of the whole team of robots. If for a certain subset of robots these information are not available, e.g., communication problems or delays, then the subset of robots stays in idle and waits for the moment in which the connection becomes active.

The robots are able to autonomously switch into the Planning state and as soon as they enter it, they wait for instruction from the planner. If at a certain time the connection with the centralized entity is lost, then the robot simply waits in idle. While in the Planning state, each robot $i \in \mathcal{S}$ receives instruction on a trajectory to follow which will lead to a configuration it needs to reach before being able to switch again into the Reactive state. After such configuration has been reached, the centralized planner informs the robot with an acknowledgement (ACK in Figure 8 of the manuscript) that it is in a safe configuration and the transition towards the Reactive state can be activated.

Comment:

> Section 1: This paper is an extension to their previous work for directed networks. Interactions are dictated by sensors with limited FOVs. This paper shows how a fully decentralized framework may not achieve formation and they provide an adaptive switching framework they refer to as adaptive decentralization which generates a switching signal to take the network from decentralized to centralized depending on the state of the system. They claim this systematic approach enables stability and avoids issues in critical interactions such as inter-robot collisions. They claim the explicit extensions over their previous papers is modeling directed and undirected interactions with limited FOV, the adaptive decentralization switching framework, and experimental results to demonstrate performance.

Reply:

We thank the Reviewer for their detailed summary of our extensions.

Comment:

> Section 2: The literature review includes a decent body of work with some detail of their previous work. They reference several of their previous papers but only detail a few of those references. They reference several recent papers but some of the most recent referenced are their own papers. I would like to see references to other work in the past 3 years outside of their own publications. Vijay Kumar, Davide Scaramuzza, Warren Dixon, and Jon How have many publications in 2019 and the start of 2020 alone related to this topic. They end the section with explaining their framework as one that can be specialized to ensure stability in tasks that inherently require agents to interact asymmetrically because many sensing modalities have finite FOVs.

Reply:

We thank the Reviewer for the suggestion. In the revised version of the manuscript we expanded the previous work section with the addition of more recent works. The revised paragraphs are reported hereinafter:

In [23] and [24] the authors address the leader-follower formation control problem in which the robots exhibit directed interactions. In [25] the authors address the formation control problem under the presence *of temporary node failures in directed settings and provide necessary and sufficient conditions on the connectivity of the graph in order to achieve the desired objective.*

[...]

Several works can be found at the state of the art that experience limited FOV in varying application settings, ranging from formation control to multi-target tracking [27]–[34]. In [27] the authors proposed a potential-based control law to asymptotically achieve formation control while maintaining FOV based interactions. In [28] the authors developed a method to generate trajectories in order to enclose and track a moving target while the robots are subjected to limited FOV sensing. The tracking problem under FOV sensing has also been addressed in [29] where a nonholonomic robot is able to track in finite-time another nonholonomic robot. FOV sensors such cameras have been used at the state of the art for different objectives such as localization [30], target tracking [31],[32], and safety enhancement [33]. In [34] a model predictive control approach is adopted to develop a collision avoidance system that allows quadrotors to fly outside the perception of their FOV with guaranteed safety.

Specifically, the following works were added:

[24] T. Endo, R. Maeda, and F. Matsuno, "Stability analysis for heterogeneous swarm robots with limited field of view," in *2019 12th International Conference on Developments in eSystems Engineering (DeSE)*, 2019, pp. 27–32.

[27] X. Li, Y. Tan, I. Mareels, and X. Chen, "Compatible formation set for uavs with visual sensing constraint," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 2497–2502.

[29] S.-L. Dai, K. Lu, and J. Fu, "Adaptive finite-time tracking control of nonholonomic multirobot formation systems with limited field-of-view sensors," *IEEE Transactions on Cybernetics*, pp. 1–14, 2021.

[30] T. Nguyen, K. Mohta, C. J. Taylor, and V. Kumar, "Vision-basedmulti-mav localization with anonymous relative measurements using coupled probabilistic data association filter," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3349–3355.

[31] A. Carrio, J. Tordesillas, S. Vemprala, S. Saripalli, P. Campoy, and J. P. How, "Onboard detection and localization of drones using depth maps," *IEEE Access*, vol. 8, pp. 30 480–30 490, 2020.

[32] C. G. Harris, Z. I. Bell, R. Sun, E. A. Doucette, J. Willard Curtis, and W. E. Dixon, "Target tracking in the presence of intermittent measurements by a network of mobile cameras," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 5962–5967.

[33] S. Sun, G. Cioffi, C. de Visser, and D. Scaramuzza, "Autonomous quadrotor flight despite rotor failure with onboard vision sensors: Frames vs. events," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 580–587, 2021.

[34] B. T. Lopez and J. P. How, "Aggressive collision avoidance with limited field-of-view sensing," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 1358–1365.

Comment:

> Section 3B: This work applies to planar 3DOF holonomic systems and they say it can extend easily to 4DOF model where the addition of altitude is considered but only the yaw axis. Each robot has a communication radius and communication is undirected while the sensing FOV is a triangular region fixed to body frame described as a directed sensing graph.

Reply:

We thank the Reviewer for their accurate summary.

Comment:

> Section 3C: Based on the previous comment extending 3DOF to 4DOF, how would you change the triangle in the 4DOF scenario? Would you use a tetrahedron?

Reply:

We thank the Reviewer for the comment. As suggested at the end of Section III-C, a pyramid could be used to model the triangular FOV in the case of 4DOF.

Specifically, the FOV could be described by the following five points:

$$
\begin{aligned}
&v_{i,1} = x_i, \\
&v_{i,2} = x_i + l_i\,\bar{R}(\eta_i)\begin{bmatrix}\cos(\theta_i+\alpha_i)\\ \sin(\theta_i+\alpha_i)\\ \sin(\beta_i)\end{bmatrix}, \quad v_{i,3} = x_i + l_i\,\bar{R}(\eta_i)\begin{bmatrix}\cos(\theta_i+\alpha_i)\\ \sin(\theta_i+\alpha_i)\\ -\sin(\beta_i)\end{bmatrix}, \\
&v_{i,4} = x_i + l_i\,\bar{R}(\eta_i)\begin{bmatrix}\cos(\theta_i-\alpha_i)\\ \sin(\theta_i-\alpha_i)\\ -\sin(\beta_i)\end{bmatrix}, \quad v_{i,5} = x_i + l_i\,\bar{R}(\eta_i)\begin{bmatrix}\cos(\theta_i-\alpha_i)\\ \sin(\theta_i-\alpha_i)\\ \sin(\beta_i)\end{bmatrix}.
\end{aligned}
$$

where, as shown in Figure R.2, α_i is the width of the field of view as in the 3DOF version, β_i is the altitude of the field of view, l_i is the length of the side of the field of view, and $\bar{R}(\eta_i) \in \mathbb{R}^3$ is the diagonal block matrix composed of $R(\eta_i)$ and 1. For the sake of simplicity in Figure R.2 the orientation θ_i and the offset angle η_i are both equal to 0.

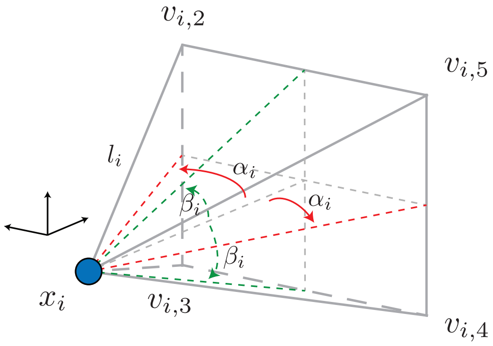


Figure R.2: Field of view modelled by a pyramid when also the vertical axis is considered.

Comment:

Section 3D and Section 4: The gradient control law, assumptions, and analysis imply you are doing a type of setpoint regulation, but it seems that a trajectory tracking control scheme would be more appropriate for this type of problem, for example tracking a trajectory and including a feedforward term in the control design.

Reply:

We thank the Reviewer for the comment. The robots evolve following update law $\dot{q}_i(t) = u_i(t)$ in which the control input $u_i(t)$ is equal to

$$u_i(t) = -\sum_{j\in\mathcal{N}_i^+} \nabla_{q_i} V_{ij}^d(q_i,q_j) - 2\sum_{j\in\mathcal{N}_i} \nabla_{q_i} V_{ij}^u(q_i,q_j)$$

in the case of the decentralized coordination framework, and equal to

$$u_i(t) = -\sum_{j\in\mathcal{N}_i^+(p)} \nabla_{q_i} V_{ij}^d(q_i,q_j) - 2\sum_{j\in\mathcal{N}_i(p)} \nabla_{q_i} V_{ij}^u(q_i,q_j) - \sum_{j\in\mathcal{N}_i^{\mathfrak{c}}(p)} \nabla_{q_i} V_{ij}^{\mathfrak{c}}(q_i,q_j)$$

in the case of the adaptive decentralization framework.

In both cases the potential terms $V_{ij}^d(q_i,q_j)$, $V_{ij}^u(q_i,q_j)$, and $V_{ij}^{\mathfrak{c}}(q_i,q_j)$ can be computed by the robot i using only local measurements. Hence, each robot is continuously computing a time-varying velocity reference using local information only.

As commonly done in literature for integrator systems [1]–[3], these velocity references are then fed to low level controllers which are in charge of controlling the real dynamics of the robot. This approach allows us to decouple the actual physical dynamics of the robots from the velocity references of the desired coordination objectives, hence abstracting away from the specific kind of robot that implements the proposed framework.

References:

[1] J. Hu, P. Bhowmick, I. Jang, F. Arvin and A. Lanzon, “A Decentralized Cluster Formation Containment Framework for Multirobot Systems,” in *IEEE Transactions on Robotics (Early Access)*, doi: 10.1109/TRO.2021.3071615.

[2] J. Son and H. Ahn, “Formation Coordination for the Propagation of a Group of Mobile Agents via Self-Mobile Localization,” in *IEEE Systems Journal*, vol. 9, no. 4, pp. 1285-1298, Dec. 2015, doi: 10.1109/JSYST.2014.2326795.

[3] H. Wei, Q. Lv, N. Duo, G. Wang, and B. Liang, “Consensus Algorithms Based Multi-Robot Formation Control under Noise and Time Delay Conditions” in *Applied Sciences*, 2019, vol. 9, no. 5, doi: 10.3390/app9051004.

Comment:

> Furthermore, the control design assumes both the state of the agent and its neighbor are directly measurable which seems counter to some of the wording used to this point. You need to add an additional assumption or make it clearer that an agent can measure its own state and that the agent can measure the neighbor state while it is in the FOV.

Reply:

We thank the Reviewer for the comment. We apologize for the lack of clarity in our manuscript. The robots are indeed able to measure their own state with respect to a common reference frame, or have access to GPS (absolute positions). The following sentence has been added to Section III.B to clearly state this requirement:

We assume that the robots share a common reference frame or have access to absolute positions by GPS.

Furthermore, as suggested by the Reviewer, we also assume that the relative position of each neighbor inside the FOV of a robot can be measured by the robot itself. The following sentence has been added as well to Section III.B:

We assume that if $(i,j) \in \mathcal{E}^s(t)$, then robot i can recognize its neighbor j and measure its relative position $x_{ij}(t)$.

The robots do not require to sense the orientation of their neighbors $\theta_j(t)$ since the pairwise potential fields are built according to Assumption 1 and Assumption 2, i.e., they do not depend on the orientation of the robot j.

Clearly, since the robots share a common reference frame and are able to measure their own state $q_i(t)$, the absolute position of their neighbors can also be computed using the relative information $x_{ij}(t)$.

Comment:

> Furthermore, given the control design and later the centralized planner, it seems that some of the challenges of FOV constraints in networks is avoided by the communication radius and the ability to communicate state directly. This limits the contribution of this work significantly. A large body of research has focused on intermittent sensing and communication which is assumed away in this result. Relaxing these assumptions will greatly increase the impact of this result.

Reply:

We thank the Reviewer for the comment. We believe that the presentation of our work may have lead the Reviewer into believing that the computation of the local control laws require communication with neighbors. We apologize for our lack of clarity.

Indeed, to achieve distributed coordinated motion when asymmetric (and symmetric) interactions are considered, no communication is required among the robots since each robot is able to compute its own control law using local measurements only, as has been seen extensively in the literature.

In the case of our adaptive decentralization framework, in order to guarantee safety for the team of robots, communication is used *only* when transitions in the finite-state machine occur. Specifically, each robot communicates to the neighbors in its communication radius ρ_{co} that a critical interaction is about to be addressed (the topology gets disconnected) or has just been addressed (the topology may have changed). As described in a response above, all other distributed control actions exploit sensing, not communication.

An informal way to interpret our framework is that we have exploited communication only as a means of getting out of sticky situations, when distributed sensing is simply not enough. Theoretically, our paper shows that for asymmetric topology control such sticky situations simply cannot be avoided.

In this work we focused on providing a distributed coordination framework (which does not require communication) and a mechanism to guarantee the safety of the team when dangerous situations may occur. Among other improvements, future works will investigate the possibility of alternative methods that could replace the presence of an omnidirectional communication graph among the robots.

In the revised version of the manuscript a new Figure, i.e., Figure 1, and a related paragraph have been added in Section III.B to explain the differences between the three graphs in play in our framework. Specifically, the following paragraph describes Figure 1:

Figure 1 depicts a team of three robots in which the three graphs involved in our system, i.e., the sensing graph $\mathcal{G}^s(t)$, the control graph $\mathcal{G}(t)$, and the communication graph $\mathcal{G}^{co}(t)$, are shown. Notice how the control graph edges are a subset of the edges of the sensing graph. Furthermore, as it will detailed later, in our framework, the communication is event-triggered and it is not used by the robots to share state information.

Furthermore, after the introduction of control law in eq. (7) the following paragraph has been added to explain which information are needed to compute the update law:

Note that eq. (7) *can be computed by each robot exploiting local information only. Specifically, each robot i needs to collect through local sensing the state information of the control out-neighbors $\mathcal{N}_i^+$ and of the control undirected neighbors $\mathcal{N}_i$. Hence, to implement eq.* (7)*, the robots do not require to share any information using the communication graph $\mathcal{G}^{co}(t)$. The communication graph will be required only later for the safety framework described in Section VI. Notice also that the neighbors $\mathcal{N}_i^+$, $\mathcal{N}_i$ are the control neighbors, i.e., a subset of neighbors selected from the* sensed *neighbors, whose interactions are guaranteed to be yield stable coordinated motion according to the results of Theorem 1.*

Similarly, the following sentence has also been added after the introduction of eq. (82):

Similarly to eq. (7)*, eq.* (82) *can be computed using local measurements only and no communication is needed among the robots to implement it.*

Comment:

> Section 4: Example graphs are shown to illustrate configurations that result in unstable systems which I believe is beneficial for explaining the contribution of this work. A Lyapunov-based stability analysis is presented for examining the stability of the decentralized controller design. Additionally, the remark at the end of the proof summarizes the theoretical benefits and limits of this approach well.

Reply:

We thank the Reviewer for the positive comments.

Comment:

> Section 5: The case study provided illustrates the approach but a field that would turn the FOV would be a better example to demonstrate the approach. The potential fields described in the section seem like they only translate the FOV around which is not practical in many scenarios where the vehicle may rotate unintentionally or is required to track a moving object such as a neighbor. Most tasks require the vehicles to rotate. I realize these are meant to illustrate examples of the design; however, I believe the addition of an orientation term would greatly improve the example.

Reply:

We apologize for the lack of clarity in the paper. We agree with the Reviewer that the presence of an orientation term is a pivotal element in the considered scenario. Indeed, the state of our robots consist of position $x_i(t)$ and orientation $\theta_i(t)$ which are both controlled by the control input $u_i(t)$. Each robot is hence able to rotate the FOV if the situation requires it.

As a representative example we discuss potential field Ψ_{ij} introduced in eq. (62). Since our update law drives the robots along the negative gradient of the potential fields, each robot i will change its orientation θ_i (and position x_i) to improve the quality of the interactions according to the potential field Ψ_{ij}. Specifically, the control on the orientation θ_i originates from the following

$$\nabla_{\theta_i}\Psi_{ij} = 2\,A\,l_i^\star\big(a(-x_{i,1}+x_{j,1})\sin\zeta + b(x_{i,2}-x_{j,2})\cos\zeta\big)\,e^{-\mu}$$

where $\zeta = \theta_i + \eta_i$ and $\mu = a\,(l_i^\star\cos\zeta + x_{i,1} - x_{j,1})^2 + b\,(l_i^\star\sin\zeta + x_{i,2} - x_{j,2})^2$ as defined in the manuscript (eq. (65)). Clearly then we have that $\dot{\theta}_i = -\nabla_{\theta_i}\Psi_{ij}$.

We constructed the potential such that the Guassian mean encoded by $x_i^\star$ is at the center of the FOV, i.e., at an orientation equal to $\theta_i + \eta_i$. Clearly, an angle offset $\beta_i^\star$ could also be included in $x_i^\star$ to model shifted desired points as depicted in Figure R.3.

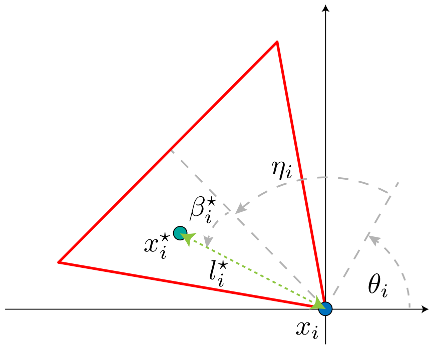


Figure R.3: Representation of an offset angle β_i on desired point $x_i^\star$ for the potential field Ψ.

Comment:

> Section 6: It seems that the switching conditions are based on a global observation of the network, that is, the state of all the agents must be somehow known for them to detect the network is approaching an unstable configuration and switch from directed to undirected. If this is not the case, then the approach to determine the graph configuration must be explained more clearly but I am not aware of any method to determine the complete configuration using only local measurements in a directed graph.

Reply:

We thank the Reviewer for the comment. We apologize for the lack of clarity. All the conditions involved in the transitions of the finite-state machine can be computed locally by each robot except for one, which is $p \in \mathcal{P}^s$. This condition requires the evaluation of the eigenvalues of the matrix $\overline{\mathcal{L}}_p$ which contains information on the structure of the whole controlled network.

As now described in the revised Section VI.B of the manuscript, the robots can evaluate this condition distributively in finite-time with an estimator as the one presented in [4] exploiting the communication graph $\mathcal{G}^{co}(t)$. In doing so, the robots would need to communicate only the topological structure of their neighborhoods, allowing each robot to compute the matrix $\overline{\mathcal{L}}_p$. Hence, no state information is exchanged by the robots.

Note also that this evaluation is required only when transitioning from the Critical to the Reactive state. In the other two cases, i.e., i) Planning to Reactive and ii) Reactive to Critical, either i) the centralized planner guarantees that $p' \in \mathcal{P}^{st}$ or ii) it holds by construction that $p' \in \mathcal{P}^s$ since critical (undirected) interactions do not affect the positive (semi-)definiteness of the matrix $\overline{\mathcal{L}}_p$ as shown in Lemma 3.

References:

[4] P. Mukherjee, A. Gasparri, and R. K. Williams, "Stable motion and distributed topology control for multi-agent systems with directed inter- actions," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec 2017, pp. 34553460.

Comment:

> If global observations are assumed available, then it seems like the state of all the agents in the network is always observable using the communication network. This would significantly reduce the contribution of this approach since that would imply that it does not matter if the neighbor is in the FOV or not which is a major part of the contribution.

Reply:

We thank the Reviewer for the comment and we apologize for the lack of clarity. The communication graph $\mathcal{G}^{co}$ is used by the robots only in two cases: i) to inform their neighbors about the disconnection enforced by the Critical state as detailed in eq. (74), and ii) to evaluate with the rest of the network the condition $p \in \mathcal{P}^s$ which only requires the structure of the neighborhood of each robot (and not their state).

Each robot is always capable of measuring its own state whereas it is able to access the neighbors' data only through local sensing, as now described in the revised Section III.B of the manuscript. The idea is that while sensing is continuous in time and it is used for computing the local update laws in eqs. (7) and (82), communication between the robots occurs at isolated instants of time when a robot needs to switch to a different state of the finite-state machine.

To reduce confusion in the revised version of the manuscript Figure 1 has been added in Section III.B to explain the differences between the three graphs used in our framework. Furthermore, after the definition of the control laws in eqs. (7) and (82) it has been made clearer that communication is not needed to compute the local update laws and the robots do not utilize the communication graph to share state information.

Comment:

> Assumption 3 implies you are using communication network to share agent state making the FOV constraints unnecessary. This is a major assumption and significantly reduces the impact of the paper since the point of the paper is to not require global information to be shared between agents and instead use local measurements through a limited FOV sensor. There are already many published articles on obstacle avoidance if the entire system is known. Based on this, it is not clear where there is a novel contribution in this result.

Reply:

We thank the Reviewer for the comment. We agree with the Reviewer that the idea of the paper is to not share global information among the robots and instead rely on local measurements only. Indeed, the proposed frameworks rely on local measurements only.

Specifically, we first propose a distributed coordination framework that considers both asymmetric and symmetric interactions. Notably, the proposed control law (eq. (7)) can be computed exploiting information obtained by the robot i using local sensors only and no communication is required between the robots.

Then we propose an adaptive decentralization framework which does not require communication to compute the update law introduced in eq. (82). However, in order to be able to guarantee the safety of the team of robots, communication is used by the robots when a transition in the finite-state machine involving the Critical state, occurs. Specifically, the robots communicate with their neighbors at isolated time instants either to inform them that a critical interaction is about to be addressed (and hence disconnection of the control graph $\mathcal{G}_p(t)$ is about to occur) or to evaluate collaboratively the condition $p \in \mathcal{P}^s$ required to transition into the Reactive state. After any of these two operations has been concluded, communication is halted and will be triggered again by the robots only when one of these situations reoccurs.

To clarify the above points the following paragraphs has been added in the revised version of the manuscript:

Note that eq. (7) can be computed by each robot exploiting local information only. Specifically, each robot i needs to collect through local sensing the state information of the control out-neighbors $\mathcal{N}_i^+$ and of the control undirected neighbors $\mathcal{N}_i$. Hence, to implement eq. (7), the robots do not require to share *any information using the communication graph $\mathcal{G}^{co}(t)$. The communication graph will be required only later for the safety framework described in Section VI. Notice also that the neighbors $\mathcal{N}_i^+$, $\mathcal{N}_i$ are the control neighbors, i.e., a subset of neighbors selected from the* sensed *neighbors, whose interactions are guaranteed to be yield stable coordinated motion according to the results of Theorem 1.*

Similarly to eq. (7)*, eq.* (82) *can be computed using local measurements only and no communication is needed among the robots to implement it.*

Furthermore, the description of the states and transitions of the finite-state machine has been further elaborated to improve clarity.

Comment:

> Section 7: It is not clear what potential field was used for the simulations and experiments. Up to this point, all of the fields were only controlling translation, but the simulation and experiment appear to rotate the vehicle FOV.

Reply:

We thank the Reviewer for the comment. We apologize for the lack of clarity. In the numerical and experimental validations the following following potential fields have been adopted: $\Phi(\cdot)$ to maintain the connectivity among the robots, $\Psi(\cdot)$ to maximize the quality of the FOV interactions between the robots, and $\Gamma(\cdot)$ to avoid collision when needed.

The rotation of the FOV is not controlled by our robots but their orientation θ_i is. Consequently, the FOV of the robots rotates as well, being it rigidly attached to the robots. Specifically, the dynamics of our system is defined as $\dot{q}_i(t) = u_i(t)$ where $q_i(t) = [x_i(t)^T, \theta_i(t)]^T$ is the state of robot i composed of position $x_i(t) = [x_{i,1}(t), x_{i,2}(t)]^T \in \mathbb{R}^2$ and orientation $\theta_i \in (-\pi, \pi]$. Since our control laws defined in eqs. (7) and (82) consist in negative gradient descent of the potential fields $\Phi(\cdot)$, $\Psi(\cdot)$, and $\Gamma(\cdot)$, the robots are allowed to change their orientation θ_i and with it their FOV.

Overall the final control law that is being used by the robots in both numerical and experimental validation is the following:

$$\dot{q}_i = -\sum_{j \in \mathcal{N}_i^+(p)} \nabla_{q_i} \left(\Phi_{ij}(q_i, q_j) + \Psi_{ij}(q_i, q_j) \right) - \sum_{j \in \mathcal{N}_i^c(p)} \nabla_{q_i} \Gamma_{ij}(q_i, q_j).$$

TO DO

Pratik

we should add the above equation in the paper and appendix if hasn't been done already

Comment:

> It appears that you are not even using the potential field method to control orientation given you say that "Figure 11 shows the orientation sub-controller is taking action to optimize FOVs". What does this statement mean? Optimize in what sense? For example, what is the cost function and why did you not use a potential field method that you described throughout the entire paper?

Reply:

We thank the Reviewer for the comment. We apologize for the confusion. We do not optimize anything with respect to the FOV. In fact, in the experimental validation we used the potential fields that have been described in Section V. Specifically, we adopted the following potential fields: $\Phi(\cdot)$ to maintain connectivity among the robots, $\Psi(\cdot)$ to encode the best possible interactions between each pair of robots, and $\Gamma(\cdot)$ to perform collision avoidance, only when needed.

Comment:

> Furthermore, the authors show the position change, but it is unclear what the error is and what the desired configuration is. These experiments need to discuss topics such as the design of the potential fields used, the desired configuration, how information is shared, what information is known, how the vehicles determine whether an object is in the FOV, the error throughout the experiments, and when switches occur. Overall, the experimental section weakly explains the framework.

Reply:

We thank the Reviewer for the comments. We apologize for the lack of clarity in the presented results. In the revised version of the manuscript these information are all been given in the validation section. Furthermore, to corroborate our theoretical findings we have conducted a set of five different real-world experiments whose result can be found at the end of this document.

For space limitation, the updated manuscript contains the results only of the real-world experiment which validates the entirety of the proposed framework.

In all the experiments that we have conducted, the collaboration objectives are the maintenance of the initial topology and the maximization of the quality of the FOV interaction while performing a leader-following scenario in which one of the four UAVs, i.e., robot 4 is the leader-robot.

The coordination objectives are encoded by the potential fields $\Phi(\cdot)$ and $\Psi(\cdot)$ as illustrated in Section V. In case a critical interaction may occur, we resort to potential field $\Gamma(\cdot)$ to perform collision avoidance.

All robots are equipped with a near field collision avoidance sensor for safety concerns. If this is not included we must add it in the paper/appendix The robots compute and run locally their own controller which is defined by the following law:

$$\dot{q}_i = - \sum_{j \in \mathcal{N}_i^+(p)} \nabla_{q_i} \left(\Phi_{ij}(q_i, q_j) + \Psi_{ij}(q_i, q_j) \right) - \sum_{j \in \mathcal{N}_i^c(p)} \nabla_{q_i} \Gamma_{ij}(q_i, q_j).$$

In our experimental setup the robots are not able to individually recognize and measure the relative positions of their neighbors. For this reason we resort to the Pozyx localization system described in Appendix B to obtain the relative positions of the robots. The robots, using this information, simulate the presence of a FOV and are able to detect if a robot j is effectively a neighbor.

In the revised version of the manuscript as well as in the reports in the Appendix C at the end of this document, all the necessary information to assess the quality of the conducted experiments, such as coordination objective, evolution of the state of the robots, switches of the finite-state machine, etc., are given.

TO DO address identification robots

We would like to thank the Reviewer for all the useful comments and suggestions that, in our opinion, helped us to further improve the quality of the revised manuscript with respect to the initial submission.

Reply to Reviewer #5

Comment:

```
The paper proposes an adaptive decentralization coordinating motion algorithm
for a team of mobile robots with the field of view constraint. A decentralized
potential-based coordinating motion framework for multi-robot systems is
firstly designed, wherein the FOV constraint is modeled as a triangular. In
order to cope with the collision due to the limited field of view in the
formation transformation, which is called ``critical interactions'', a
switching control mechanism is then introduced by applying a finite-state
machine. Rigorous theoretical analysis proves the stability of the proposed
framework. Simulation and experimental results are presented to validate the
effectiveness of the adaptive decentralized algorithm.
```

Reply:

We thank the Reviewer for the detailed summary of our manuscript.

Comment:

```
The paper is a theoretical paper more than a practical one. The robot model and
sensing model are over simplified, and some assumptions are too abstract so
that it is difficult to judge if such assumptions can be satisfied in practical
experiments. The experimental part is too simple to support the main
contributions. Detailed comments are listed as follows:
```

Reply:

We thank the Reviewer for the feedback. As detailed hereinafter in the replies to the comments of the Reviewer, in the revised version of the manuscript we clarified the meaning of the required assumptions and significantly improved the experimental validation of our framework.

Comment:

```
1.      The robot model and sensing model are over simplified. For robot model,
nonlinear dynamics are full ignored as well as physical constraints including
the underactuated property, nonholonomic constraints, maximum velocity and
torque constraints, etc. Such simplification generates large gap between the
ideal particle robot and realistic robots, making those fragile analysis less
convincing when facing realistic robots. In the reviewer's opinion, this paper
is more suitable for a control-oriented journal like IEEE TAC, rather than
robotics-oriented journal.
```

Reply:

We thank the reviewer for the comment. Integrator control laws have been largely used in the literature for coordinating teams of robots [1]–[5]. The idea behind this approach is to generate time-varying velocity references for the low-level controllers that move the robot. In this way, the solution of the coordination problem and the physical control of the robot are decoupled and can be handled separately. In doing so, this kind of approach maximizes the scope of the proposed framework since it abstracts from the physical platform performing coordinated motion.

Indeed, as described in the revised Section VII, the aforementioned approach has been validated by our experimental setup which involved a team of four quadrotors in a real-world scenario with disturbance such as wind, communication delays, and localization inaccuracies. Specifically, during the experimental validations each of our drones run a low level controller that was in charge of handling the speed of the vehicle as specified by the control inputs $u_i(t)$. An extensive report on our set of experiments can be found at the end of this document as well on Github.com at the address ...

On a different note, studies at the state of the art such as [6],[7], have focused on demonstrating also that coordination objectives developed using potential-based frameworks and gradient-descent control laws can be adapted to consider physical constraints such as maximum velocity or nonholonomic constraints.

References:

[1] G. Zhao and M. Zhu, “Pareto optimal multi-robot motion planning,” in *IEEE Transactions on Automatic Control*, doi: 10.1109/TAC.2020.3025870.

[2] H. Wei, Q. Lv, N. Duo, G. Wang, and B. Liang, “Consensus Algorithms Based Multi-Robot Formation Control under Noise and Time Delay Conditions” in *Applied Sciences*, 2019, vol. 9, no. 5, doi: 10.3390/app9051004.

[3] J. Son and H. Ahn, “Formation Coordination for the Propagation of a Group of Mobile Agents via Self-Mobile Localization,” in *IEEE Systems Journal*, vol. 9, no. 4, pp. 1285-1298, Dec. 2015, doi: 10.1109/JSYST.2014.2326795.

[4] W. Luo and K. Sycara, “Minimum k-Connectivity Maintenance for Robust Multi-Robot Systems,” *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 7370-7377, doi: 10.1109/IROS40897.2019.8968058.

[5] J. Hu, P. Bhowmick, I. Jang, F. Arvin and A. Lanzon, “A Decentralized Cluster Formation Containment Framework for Multirobot Systems,” in *IEEE Transactions on Robotics (Early Access)*, doi: 10.1109/TRO.2021.3071615.

[6] S. Zhao and Z. Sun, “Defend the practicality of single-integrator models in multi-robot coordination control,” *2017 13th IEEE International Conference on Control & Automation (ICCA)*, 2017, pp. 666-671, doi: 10.1109/ICCA.2017.8003139.

[7] S. Zhao, D. V. Dimarogonas, Z. Sun and D. Bauso, “A General Approach to Coordination Control of Mobile Agents With Motion Constraints,” in *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1509-1516, May 2018, doi: 10.1109/TAC.2017.2750924.

Comment:

> 2. Fig. 2 is not easy to understand. It is suggested to clearly give the physical meaning underlying Theorem 1. The physical meaning of those assumptions as well as those conditions in the proposed theorems should be clearly explained, especially those abstract conditions on the varying matrices.

Reply:

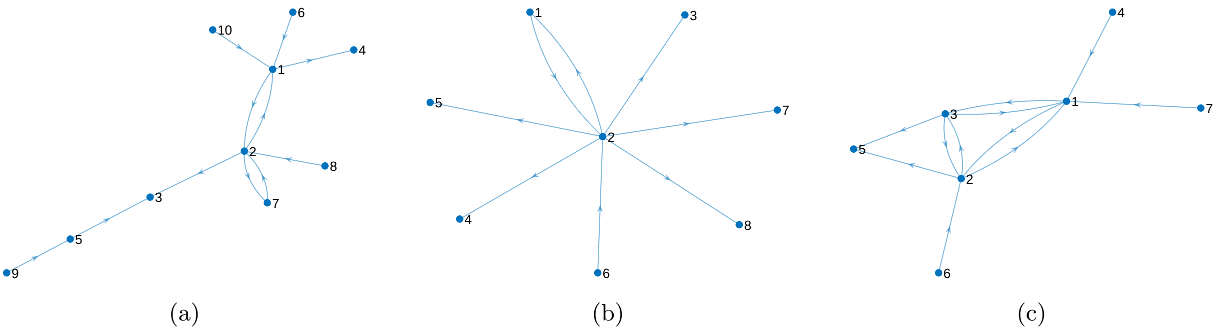
We thank the Reviewer for the suggestion. The paragraph before the statement of Theorem 1 and the description of Figure 3 have been edited to improve their clarity.

The updated paragraph and the description of the Figure are reported hereinafter for the sake of clarity:

Nevertheless, a class of stable topologies can be inferred from the structure of the symmetric part of the matrix $\overline{\mathcal{L}}$, where stability is intended in the Lyapunov sense, i.e., $\dot{V}(\mathbf{q}) = -\xi^T\overline{\mathcal{L}}\,\xi \leq 0$. Figure 3 depicts four multi-robot graph topologies, where the top topologies correspond to positive semi-definite $\overline{\mathcal{L}}$ and the bottom topologies to indefinite and negative semi-definite $\overline{\mathcal{L}}$. Notably, spanning trees with possibly bidirectional edges and non-tree structures such as minimal directed cycles can yield stable coordinated motion under asymmetry (Figure 3a), however even simple leader-follower structures (Figure 3b, right) can be unstable if interactions are chosen inappropriately. Based on the positive (semi-) definiteness of the matrix $\overline{\mathcal{L}}$, general coordination with limited field of view can be achieved as stated in the following result.

Two stable (unstable) graph topologies for Theorem 1, i.e., $\overline{\mathcal{L}}^+ \succeq 0$ ($\overline{\mathcal{L}}^+ \not\succeq 0$).

Theorem 1 requires the positive (semi-)definiteness of the symmetric part of the $\overline{\mathcal{L}}$ matrix. From a topological standpoint, this requirement implies that directed spanning trees, minimal cycles, spanning trees with undirected edges (as the one shown in Figure R.4a), star-like structure (as the one shown in Figure R.4b), and spanning trees with a single directed cycle (as the one shown in Figure R.4c) can yield a stable coordinated motion.



(a)

(b)

(c)

Figure R.4: Example of class of graphs that are able to yield a stable motion according to the results of Theorem 1.

The assumptions required for the structure of the potential fields involved in the control laws (7) and (82) are the classical assumptions required by potential-based control methodologies. Potential fields do not always encode a physical meaning but rather they are mathematical functions built to model coordination objectives to achieve an emerging collective behavior. From a practical standpoint, Assumptions 1 and 2 guarantee the presence of certain level of symmetry between the interactions of the robots.

Comment:

> 3. The introduction about experimental setup is not clear. For example, the angle of the camera field of view, the communication and sensing radius mentioned in Assumption 3.

Reply:

We thank the Reviewer for the comment. We have modified the experimental section in the paper to include details on the experimental setup. The details on the FOV parameters are also discussed in this document in Appendix C. With respect to the communication and the sensing radius, we apply Assumption 3 to imply that the bi-directional (undirected graph in nature) communication between robots is larger than the sensing (asymmetric or directed graph) between robots which is defined by the perimeter of the FOVs of each agent. This is a reasonable assumption because clearly sensing equipment like cameras have lower range than robots communicating over Wi-Fi for instance. TO DO

Comment:

> 4. The experimental results are too few to support your theoretical contributions.

Reply:

We thank the Reviewer for the comment. We refer the reader to Appendix C where we present results from five new experiments:

1. A fast speed leader robot sweep across a 20 m field (Section C.1);
2. Variant leader trajectory (Section C.2);
3. Demonstration of Machine State transition (Section C.3);
4. Robots with rotating FOV (Section C.4).
5. An alternate demonstrations of Machine State transition (Section C.5);

The media file submitted with the paper contains the video analysis of each of the five experiments and relevant results. For the sake of brevity, we only include and discuss in detail the results of Experiment-3 (Appendix C.3) in the revised manuscript.

Comment:

> 1) For typical quadrotor UAVs, the translational dynamics is coupled with the rotational dynamics, i.e., the underactuated property. Hence, the model in (2) is not proper for quadrotors. In addition, the manuscript only considers the FOV on yaw angle, which is too simple. To the reviewer's knowledge, roll and pitch FOV constraints have even more impact when there is a velocity change.

Reply:

We thank the Reviewer for the comment. We agree with the Reviewer that the consideration of more complex and suited dynamics as well as the addition in the state variables of roll and pitch of the robots would improve the scope of the proposed framework. However, this work can be considered as the first step in addressing generic asymmetric (and symmetric) coordination for multi-robot system equipped with limited FOV sensing, given also that, to the best of our knowledge, no prior work has addressed such problem.

Hence, as a first step, we believe that the usage of a single integrator model and the adoption of a state that is composed of position and yaw only, are able to model a sufficient amount of different scenarios which may attract the community. Indeed, the proposed control laws could be used as velocity references for any kind of robotic platform, allowing our framework to be adopted by as many users as possible.

Clearly, future works will focus on the development of distributed generic coordination frameworks for more complex dynamics that consider orientation terms composed of roll, pitch, and yaw rather than yaw only.

Comment:

> 2) No comparative experiments are provided to highlight the advantages of the proposed adaptive decentralized framework with respect to the existing works;

Reply:

We thank the Reviewer for the comment. The goal of the experimental validation was to corroborate our theoretical findings and to experimentally validate the robustness of the proposed framework against unmodeled factors such as wind disturbances and localization inaccuracies. In addition, to the best of our knowledge, there are no prior works at the state of the art that are able to achieve safe distributed coordinated motion under both asymmetric and symmetric interactions, while maintaining desired topologies with FOV sensors, hence, for this reason, no comparison was given.

Comment:

> 3) The process of coping with the critical interactions using the proposed switching mechanism needs to be illustrated specifically in the experiment;

Reply:

We thank the Reviewer for the comment. In the revised version of the manuscript as well as in the Appendix C.3, the results now clearly state when a transition in the finite-state machine occur.

Comment:

> 4) How to show the switching mechanism in Fig. 10? The centralized phase, switching phase and the decentralized phase should be marked in the figure.

Reply:

We thank the Reviewer for the comment. We have added new figures in the paper for both the ROS Gazebo simulation as well as the real-world experiments with the machine state transitions marked. The position plots of robot 4 in Figure R.18 show the initiation of the planning state whereas the distance plots for the real-world experiments are also shown in Figures R.21 and R.22 of this document in Appendix C.3.1 with the machine state transitions demarcated. These plots specifically show the “switching phase” or the machine state transitions.

Comment:

> 5) It seems that the experimental setting is much simpler than that in the theoretical analysis. For example, the secondary potential field defined in (63) is not reflected in the experiment.

Reply:

We thank the Reviewer for the comment. The potential fields introduced in Section V have been introduced to illustrate the capabilities of the proposed framework as well as to show how potential fields that satisfy Assumptions 1 and 2 could be built.

In our experimental setup we have used the potential fields that we believe better demonstrate the strength of the proposed framework. The original submission included only experiments focusing on the generic asymmetric coordination framework. In the revised version of the manuscript the simulations as well as the experiments corroborate both the generic asymmetric coordination framework and the adaptive decentralization one.

For space limitations, the manuscript contains the results of a single experiment (Experiment-3 in Appendix C.3) even though different scenarios have been considered for the resubmission. The results of all the conducted experiments can be found in Appendix C.

TO DO

Comment:

```
5.      Multi-robot coordination with FOV is not new, for example, X. Li, Y.
Tan, I. Mareels and X. Chen, "Compatible Formation Set for UAVs with Visual
Sensing Constraint," 2018 Annual American Control Conference (ACC), Milwaukee,
WI, 2018, pp. 2497-2502.
```

Reply:

We thank the Reviewer for the suggestion. We apologize for the lack of clarity, we did not intend to imply that multi-robot coordination with FOV is new but rather that multi-robot coordination for asymmetric interactions, of which FOV-based interactions are a great illustrative example, is an active topic of research that gained interest especially in the last five or so years.

The suggested work however does not fit the criteria of coordination under asymmetric interactions since it assumes that the interactions between the robots are symmetric. In our case the complexity arises from the need to consider *asymmetric* interactions which introduce the possibility of degeneracies coordinated motion control schemes when multiple objectives (like leading-following and collision avoidance) are considered.

Nevertheless, the suggested work was added to the paragraph in which we review generic applications of limited FOV sensing.

We would like to thank the Reviewer for all the useful comments and suggestions that, in our opinion, helped us to further improve the quality of the revised manuscript with respect to the initial submission.

Reply to Reviewer #6

Comment:

> This paper deals with the coordination of multi-agent systems under various operational specifications. The problem is pertinent to the TRO journal and it is very well motivated in the introductory section of the paper along with a thorough literature review.

Reply:

We thank the Reviewer for the positive comments on our manuscript.

Comment:

> However, the organization of the work is extremely confusing and the manuscript is unnecessarily lengthy from a technical point of view (a reader would expect more results that validate the theoretical findings than repeating technical stuff in several points). In general, the reviewer suggests shortening the paper down to a technical note by presenting specific technical points (limited field of view, directed operational specifications) and not as a generic strategy. Additionally, certain technical issues need to be fixed:

Reply:

We thank the Reviewer for the comment. In the revised version of the manuscript many edits have been made to improve the clarity and the presentation of the two frameworks. Specifically, i) the required assumptions are clearly stated; ii) the roles of the three graphs involved in our framework are better detailed; iii) the description of the adaptive decentralization framework has been shortened and made clearer; and iv) the experimental validation has been significantly improved to increase practical value. For space limitations reasons, the results of all the experiments conducted to validate our frameworks have not been included in the manuscript but have been appended at the end of this document (Appendix A–D).

Regarding the suggestion to shorten the paper down to a technical note, we simply cannot conceive of a manner to fit deep theoretical analysis with extensive multi-robot experimentation in outdoor environments into such a short format. We argue that the theory cannot be given without the experimentation and visa versa. Indeed, the theory makes assumptions that must be realized in the real-world to show practical value, while experiments can lose impact without a theoretical underpinning. Thus, without significantly diminishing the quality and understanding of our contribution, we have kept the regular paper format in the revision.

Comment:

> -) Please adopt "limited field of view" and not "limited fields of view" as it is written in multiple places throughout the manuscript.

Reply:

We thank the reviewer for pointing this out. In the revised version of the manuscript we adopted the suggested "limited field of view(s)".

Comment:

> -) In the robot modeling (1) why the velocity command that corresponds to the rotational part should lie within (-pi,pi]? It makes no sense and I wonder if the proposed solution respects such specification.

Reply:

We thank the Reviewer for pointing this out. It was a typo. In the revised version of the manuscript we changed the two velocity commands vector definitions to $u_i(t) \in \mathbb{R}^3$ and $\mathbf{u}(t) \in \mathbb{R}^{3n}$, respectively.

Comment:

```
-) In the modeling of the field of view, since the sensing adopted in
directional (visual, infrared, acoustic) then a significant issue that arises
concerns the occlusion that an agent may create between two other agents (e.g.,
an agent lies between an agents that senses another one (occluded) on the other
side of the first one). Such issue is critical and is not referred in the
problem formulation section.
```

Reply:

We thank the Reviewer for the comment. We agree with the Reviewer that potential occlusions can be critical in multi-robot systems. This phenomena, however, can be modeled in our framework as loss of sensing, for example, a potential field that goes to infinity when a control neighbor j is suddenly lost, i.e., its state is not measurable anymore, would be enough to trigger the switch to the Critical state. Furthermore, since the occluded robot j would not be part of the critical neighborhood $\mathcal{N}_i^c$ since $(i,j) \notin \mathcal{E}^s$, the critical interaction would be immediately considered as solved.

The robot could then transition into the Reactive state if it possesses other neighbors that would allow him to pursue its objective (given that all the necessary conditions are checked, i.e., energy-finiteness and topological stability as described in the revised Section VI of the manuscript) or otherwise it could transition into the Planning state and ask the centralize planner for help, i.e., it could ask to the planner where it should be relocating to pursue its coordination objective.

Other strategies involving the "rejection" of a sensing edge with robot j, if j is close to another sensed neighbor k, could also be investigated. Similar to above, however, the robot would be forced to go through the Critical state and possibly Planning state before being able to switch back to the Reactive state and continue the coordinated motion.

Discuss. This potential field would jump to infinity, i.e., it is not continuous

Comment:

```
-) Assumptions 1 and 2 mention that all directed and undirected objectives
should be independent of the orientation of each agent (see (13) and (16)).
However, this is not clearly the case for the limited field of view
specification if we take a quick look in (3). Apparently, the points v_{i,2}
and v_{i,3} depend heavily on the orientation of agent i. For example, if we
consider a camera attached on an agent then if the agent rotates then its field
of view changes affecting the corresponding operational specification. This is
a critical issue in the paper as FoV is presented as one of the major
contributions of this work and for this contribution apparently in the proof of
Lemma 1 (and Lemma 2) in the middle of the right column the argumentation is
not correct. The distances d^1_{ij}, d^2_{ij}d^3_{ij} depend on the orientation
of the agent i and therefore Assumption 1 does not hold.
```

Reply:

We thank the Reviewer for the question. Eq. (16) (now eq. (12)) defines the characteristics that the undirected potential fields must possess. Specifically, the pairwise undirected potential field V_{ij}^u must not depend on the orientation variables $\theta_i(t)$ and $\theta_j(t)$. Similarly, eq. (13) (now eq. (9)) defines the characteristics of the directed potential fields. Specifically, it is required that the pairwise directed potential field V_{ij}^d does not depend on the orientation variable $\theta_j(t)$ only. Hence, it is not required for the directed potential fields to not depend on the orientation $\theta_i(t)$ of the robot i.

As suggested by the Reviewer, this is indeed the case in our directed potential fields $\Phi_{ij}(\cdot)$ and $\Psi_{ij}(\cdot)$. However, since Assumption 1 allows the directed potential fields to depend on the orientation $\theta_i(t)$ of the robot i, the conditions are verified and Assumption 1 does hold for both $\Phi_{ij}(\cdot)$ and $\Psi_{ij}(\cdot)$.

Comment:

```
-) In Section IV, Theorem 1 dictates that convergence to the global minimum is
achieved asymptotically without imposing any structural assumptions on the
```

selection of the potential functions V^d_{ij} and V^u_{ij}. The reviewer is really skeptic about the results of Theorem 1 regarding the asymptotic convergence to the global minimum since the unwanted local minima in potential fields coordination schemes is the main drawback. Superimposing potential function inevitably introduces critical (even stable) points (other than the desired one) where the controller vanishes and thus the agents may be trapped. A vast number of works study the Morse properties of the underlying potential functions in order to alleviate such unwanted critical points. Yet in Theorem 1, the invariant set theorem is invoked to establish asymptotic convergence to the global minimum. However, based on the aforementioned analysis there may exist certain other critical points with non-trivial region of attraction that may act as a counterexample to Theorem 1, which is critical for the theoretical findings of the whole paper.

Reply:

We thank the Reviewer for pointing this out. We apologize for the error, it was a typo and in the revised version of the manuscript we changed the term "minimum points" with "critical points". Theorem 1 proves that the critical points of energy encoding the desired coordination objective $\mathbb{O}^\star$ are asymptotically reached since the system converges to the largest invariant set of $\{\xi : \dot{\mathbf{q}} = \mathbf{0}_{3n}\}$.

Comment:

-) In Theorem 1, the symmetric part of the matrix \bar{L} is mentioned. Intuitively (or through an example) what does it mean (from the graph point of view) to have the symmetric part of this matrix positive semi-definite?

Reply:

We thank the Reviewer for the comment. We apologize for the lack of clarity. In our case, as shown in Figure 3 of the manuscript, small edges differences to similar structure can lead to significative differents

In the revised version of the manuscript before the statement of Theorem 1 the following paragraph has been added:

In fact, it is not possible to infer class of topologies that would able to yield a stable coordinated motion, where by stability we intended the stability in the Lyapunov sense, i.e., $\dot{V}(\mathbf{q}) = -\xi^T \overline{\mathcal{L}}\,\xi \leq 0$*. Consider the graphs depicted in Figure 3 where the top topologies correspond to positive semi-definite* $\overline{\mathcal{L}}$ *and the bottom topologies to indefinite and negative semi-definite* $\overline{\mathcal{L}}$*. This example shows how similar class of topologies can differ fundamentally from a stability standpoint when even a single edge is added. Hence, stability can only be guaranteed by the positive (semi-)definiteness of the matrix* $\overline{\mathcal{L}}$ *as shown in the next result.*

We would like to thank the Reviewer for all the useful comments and suggestions that, in our opinion, helped us to further improve the quality of the revised manuscript with respect to the initial submission.

Reply to Reviewer #7

Comment:

```
I think the major contributions of this paper is related to the generalization
of the problem common problem in this area (multi-robots coordination, limited
angle of view, etc) using network theory that can be executed smoothly using
graph theory. Beside the unique modelling abstraction, the notable contribution
is the novel way from the authors to combine the Robot+network and FOV as a
complete model equipped with topological properties.
```

Reply:

We thank the Reviewer for the positive feedback on our manuscript.

Comment:

```
Regardless of those contribution, however, it takes me significant time to
understand the paper. For a novice reader, the organization of the paper can be
made better by giving extra illustrations. Figure 1 and 2 are perfect
illustration for explaining FOV and Stable/unstable graph respectively. It
could be great if the authors can also add illustration related to the
important parts, for example, Problem 1. In fact, Figure 5 could be use as
illustration purpose by giving sufficient explanation after Problem 1
statement. I think the authors can be better in serving the reader to improve
the readability of the paper.

I have no objection to other parts of the paper. Thank you.
```

Reply:

We thank the Reviewer for the comment. In the revised version of the manuscript significative efforts have been made into improving the clarity and the presentation of the proposed frameworks. Specifically, the following major improvements were made: i) the required assumptions are clearly stated; ii) the roles of the three graphs involved in our framework are better detailed with a new Figure, i.e., Figure 1; iii) the description of the adaptive decentralization framework has been shortened and made clearer; and iv) the experimental validation has been significantly improved.

In addition, Figure 1 has been added to improve the description and to highlight the differences of the three graphs involved in our framework, i.e., the sensing graph $\mathcal{G}^s(t)$, the control graph $\mathcal{G}(t)$, and the communication graph $\mathcal{G}^{co}(t)$.

Furthermore, Figures 6, 8, 9, 10, and 11 have all been updated to improve their clarity.

We would like to thank the Reviewer for all the useful comments and suggestions that, in our opinion, helped us to further improve the quality of the revised manuscript with respect to the initial submission.

A Introduction

In this report, we directly address the concerns of the Reviewers with respect to the limited FOV experiments. We will specifically address the following:

- Clarity regarding the experimental set-up and the framework;
- Experiments conducted with different *stable* graph configurations;
- Variety of experiments conducted to showcase the full spectrum of functionalities of the adaptive limited FOV controller (i.e. transitions of machine states);
- Provide extensive experimental results for different scenarios;
- Aligning the experiments with the theory produced in the manuscript;
- Limitations of the experimental set-up;
- Advantages of equipping the *Portable Multi-Robot Testbed* with the adaptive mechanism of the *Finite State Machines* ;

We provide experimental results from five different experiments:

1. A fast speed leader robot sweep across a 20 m field (Section C.1);
2. Variant leader trajectory (Section C.2);
3. Demonstration of Machine State transition (Section C.3);
4. Robots with substantial, fast FOV rotation (Section C.4).
5. An alternate demonstration of Machine State transition (Section C.5);

The rest of the Appendices is outlined as follows: In Section B, we will discuss the details of the different hardware and software components of the *Portable Multi-Robot Testbed*; In Section C, we provide the results from five different outdoor experiments we conducted using the *Portable Multi-Robot Testbed*. Note that these appendices act as a detailed explanation of the improved (but condensed) material added in the revised manuscript.

B Portable Multi-Robot Testbed

The idea of the *portable multi-robot testbed* is to enable us to carry out experiments in different environments, whether indoors or outdoors, such that we can test our algorithms in widely varying conditions. Using this *portable multi-robot testbed* we intended to conduct field trials to evaluate our directed topology control concept. Therefore, the primary objective was to develop an infrastructure for operating a team of unmanned aerial vehicles (UAVs). Since, our work pertains to topology control, it was vital that we test the behaviour of our controller in different conditions which will help us understand the controller and help us make critical decisions to make these controllers more robust. Therefore, a multi-robot testbed that consists of a team of state of the art UAVs and a localization system was developed to aid the purpose of research in the field of multi-robot stable topology control.

We developed a framework for multi-robot systems with various features: (1) a base station for UAV command; (2) stable UAV flight from high-level commands; (3) UAV-to-UAV interaction. The Robot Operating System (ROS) is used to implement a multi-robot base station for flight operations. The base station software operates ROS nodes for certain computation (i.e. ROS nodes for obtaining localization information from Pozyx). However, we have a separate local controller running on the on-board computers of the robots that compute the control law for each robot and pass it on as commands (velocity, yaw rate or position and yaw angle) to the lower level controller of the robots. Currently, we deploy a team of four DJI Matrice 100s (M100s) to evaluate our directed topology control framework. Before we conducted any experimental evaluation, we first tested our framework in a Gazebo simulation environment. Only after realising the behavior of our directed topology control framework in Gazebo did we move towards validating the controller with the M100s.

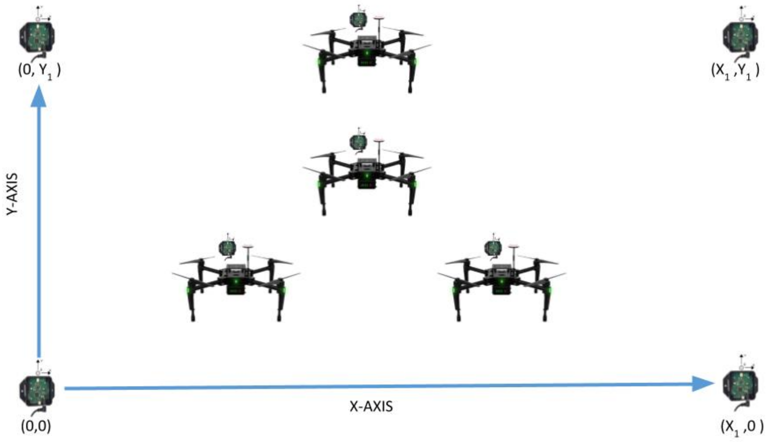


Figure R.5: Pozyx anchors are set at known positions(Cartesian coordinate system) to define the perimeter of the experimental environment and Pozyx tags are rigidly attached to M100s to obtain IMU and localization data.

B.1 Hardware

With the above objectives, we developed the *portable multi-robot testbed* that enables us to experiment with a team of four M100s in both indoor and outdoor settings. The M100 is a state-of-the-art quadcopter that is equipped with a visual inertial system, Guidance, that enables it to fly outdoors with GPS availability as well as indoors where GPS is denied. The Guidance visual inertial navigation system is equipped with five pairs of stereo cameras and sonar sensors, that also enables obstacle avoidance. DJI, the manufacturer of M100s, provides a sophisticated Software Development Kit (SDK) that is compatible with ROS and runs on $C++$ and Python. The lower level controller for the M100s is well-tuned and provides extremely stable flights for the UAVs under autonomous conditions. As a user, we are only concerned about the higher level control inputs that can be in the form or position or velocity.

For localization of these UAVs during an indoor or outdoor experiment, we use the Pozyx positioning system. It is this Pozyx positioning system that provides the feature of portability to this multi-robot testbed. The Pozyx system consists of a set of anchors and tags. The anchors are placed in the environment at known positions and the tags are rigidly attached to the M100s. The only difference between the anchors and tags are that the tags also provide IMU data which is not available from the Pozyx anchors. However, using the anchors, we can define the perimeter of the experimental environment which represents a 2-D Cartesian coordinate system as shown in Figure R.5. Typically, we use a rectangle of dimensions $30m \times 20m$ to conduct our experiments due to size limitations of Virginia Tech's outdoor drone facility. The perimeter of this rectangle experimental set-up is defined by the anchors.

Finally, to enable on-board computation, each M100 is equipped with a Nvidia Jetson Tx1 or Tx2. The Nvidia Jetson Tx1 or Tx2 are state of the art computation platforms that come with GPU and CPU. They run the ARM architecture with Linux operating system. The Jetsons are installed with the DJI SDKs that help us fly the M100s remotely and autonomously.

B.2 Software Architecture for Multi-Robot Experimental Setup

With the required hardware described, we will now describe the underlying software architecture of the multi-robot testbed. The multi-robot tesbed is fundamentally based on the ROS architecture. As already mentioned earlier, the DJI SDK that helps control the lower level controller of the M100s is compatible with ROS and runs purely on $C++$. On the other hand, the software for Pozyx is written in Python. However, because of ROS's compatiblity with both $C++$ and Python, integrating the M100s and the Pozyx at a software level was seamless. Specifically for this paper, we have written a FOV controller in C++ that runs on the on-board computers (Nvidia Jetsons) of the robots.

C Outdoor Experiments

In the following experiments, we demonstrate limited Field of View (FOV) control with four DJI Matrice 100 UAV in an outdoor setting with approximately 10mph wind speeds. UAV 4 is chosen to be the leader robot and is given a prescribed velocity. The UAVs $1, 2, 3$ are supposed to follow their neighbor robot maintaining the interaction through their limited FOV. The triangular field of view $\mathcal{T}_i$ with side length $l_i = 8m$ attached to robot i in which the angle $\alpha_i = 45°$ has been used for each robot. All the five different experiments depict a typical leader-follower scenario where the robot 4 is always chosen to be the leader with a prescribed trajectory. In the following sub-sections we provide information related to each of the five experiments with a basic description of the experimental setup, brief discussion of the results and a brief conclusion of the outcomes. Finally, we end this section with a brief discussion of the experimental outcomes in the context of the reviews we have received.

C.1 Experiment 1 - Sweep across a 20m field

In this experiment, we provide a fast prescribed velocity to the leader robot 4, which moves from its original position to sweep a distance of $20m$ and then return back to its position. The follower robots are shown to be able to keep up with the fast pace of the leader robot through out the experiment.

C.1.1 Results

The initial and final configurations of the team of UAVs are shown in Figures R.6 and R.7.

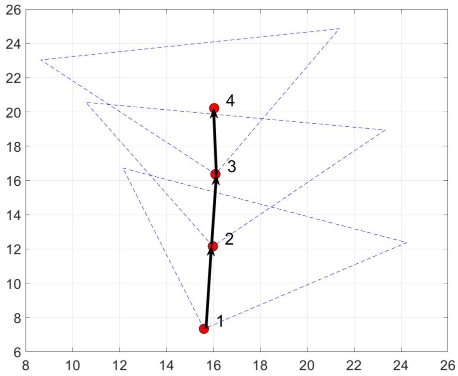


Figure R.6: Experiment 1 - Initial position.

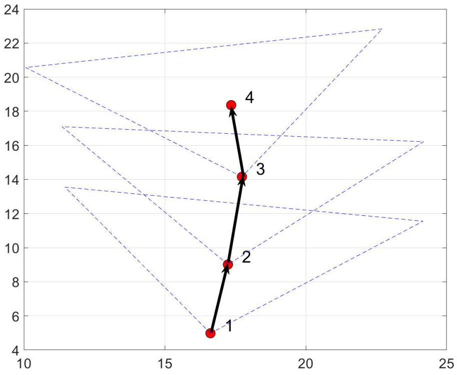


Figure R.7: Experiment 1 - Final position.

The evolution of each UAV's position throughout the complete experiment is given in Figure R.8. The evolution of each UAV's yaw is given in Figure R.9. The measured velocity for each UAV is given in Figure R.10.

C.1.2 Conclusion

In this experiment, we demonstrate the following:

- UAVs are moving much faster, in the range 0.4–0.6 m/s in comparison to the experiments initially submitted where the UAVs had measured velocities in the range 0.1–0.3 m/s;
- We provide a more complex trajectory to the leader robot 4, which has to travel longer distance at higher velocities for longer time;
- There is diversity in the trajectory of the follower robots as they try to keep up with the leader moving at a faster pace.

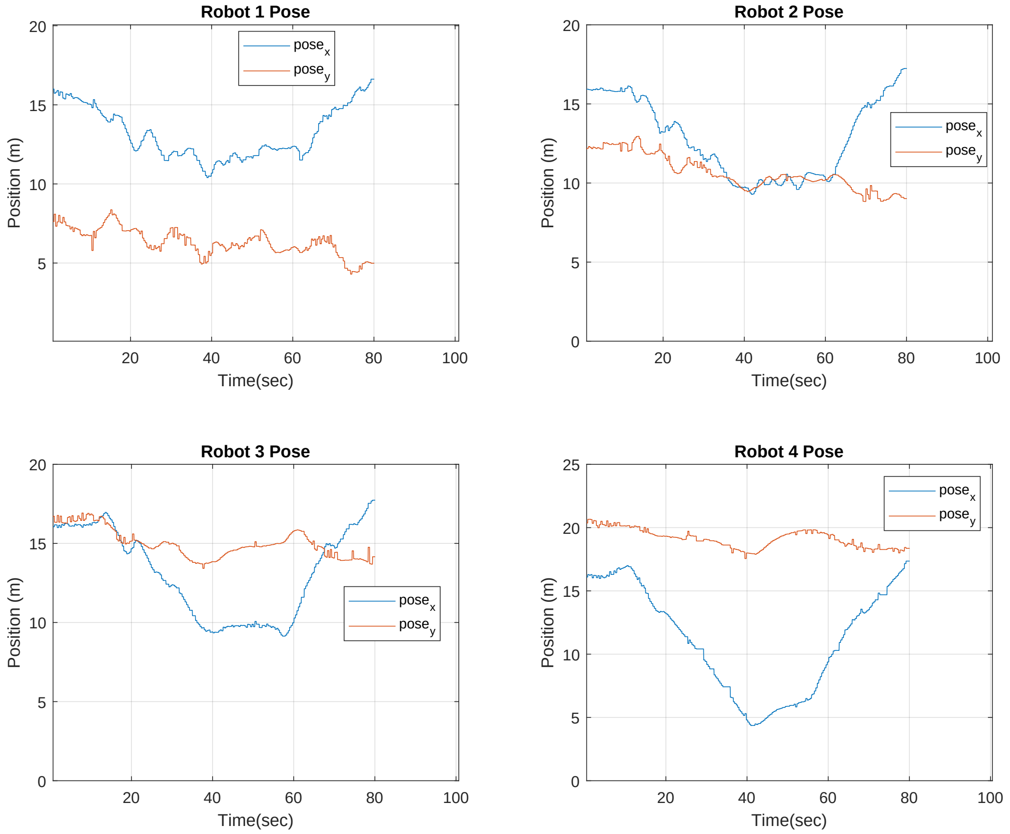
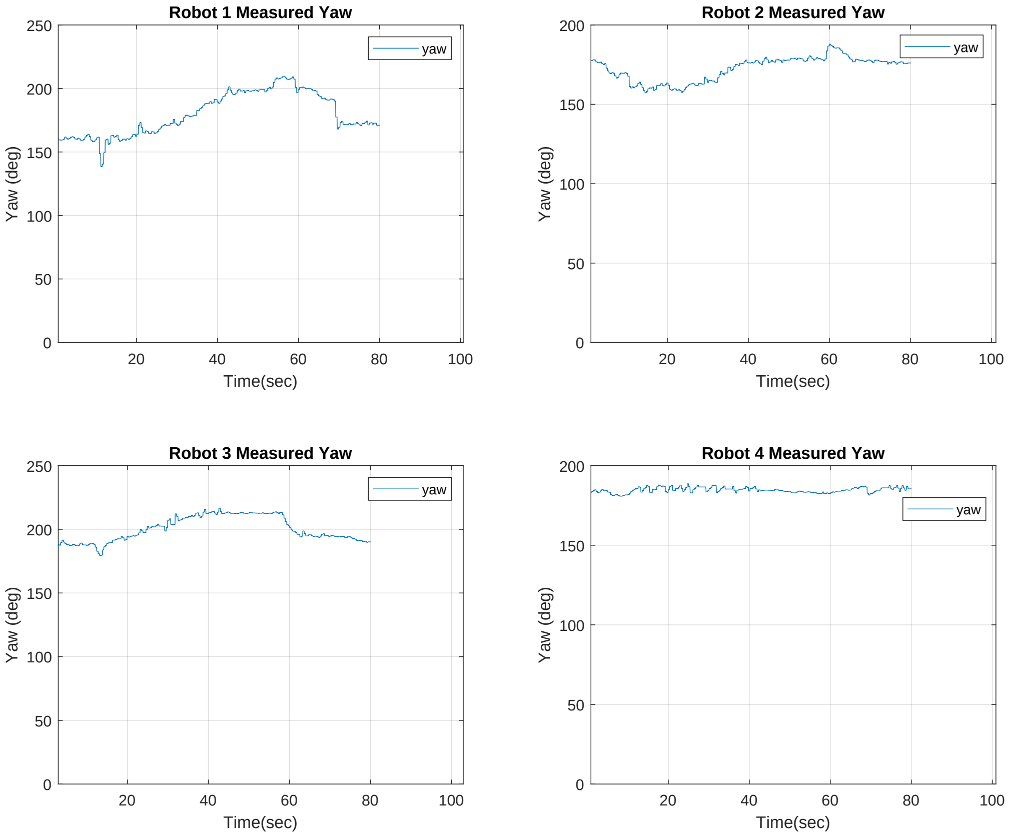


Figure R.8: Experiment 1 - Position data for all robots.


Robot 1 Measured Yaw

Robot 2 Measured Yaw

Robot 3 Measured Yaw

Robot 4 Measured Yaw

Figure R.9: Experiment 1 - Yaw data for all robots.

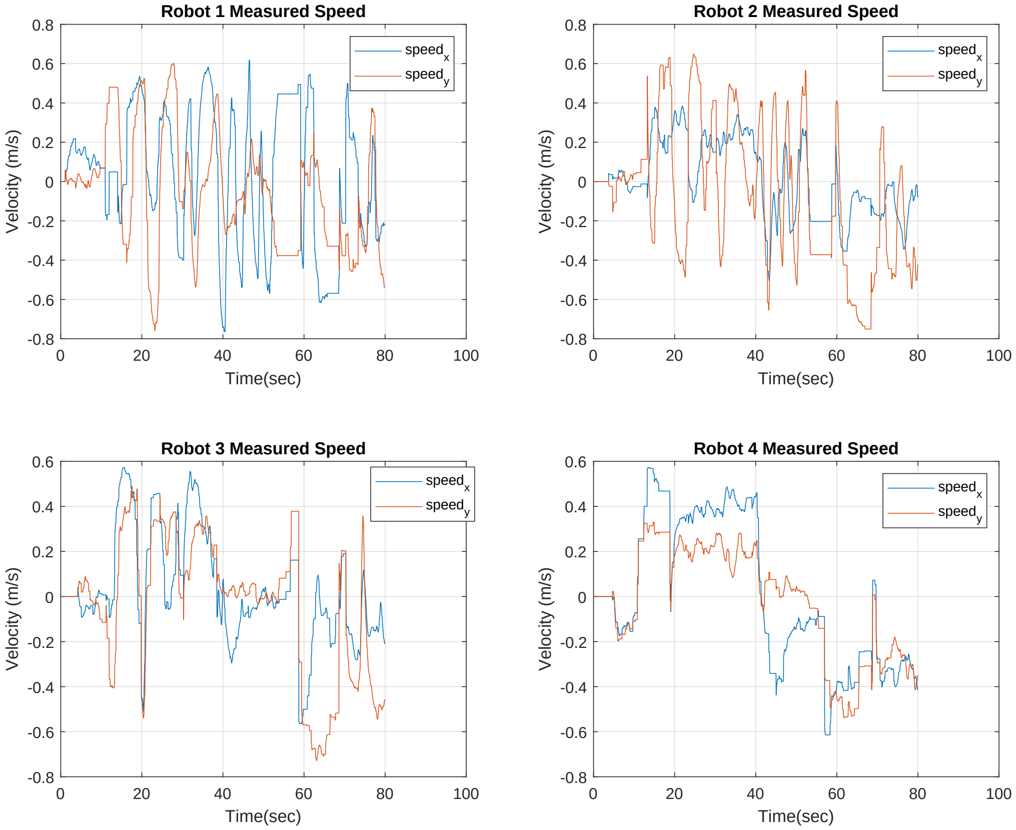


Figure R.10: Experiment 1 - Velocity data for all robots.

C.2 Experiment 2 - Variant Leader Trajectory

In this experiment, we make the leader robot 4 move abruptly with sudden stops or sudden change in direction at higher velocities. This allows us to observe the behavior of the follower robots and their ability to adapt to the sudden maneuvers executed by the leader robot 4.

C.2.1 Results

The initial and final configurations of the team of UAVs are shown in Figures R.11 and R.12.

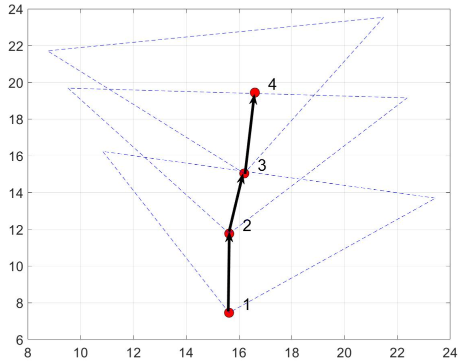


Figure R.11: Experiment 2 - Initial position.

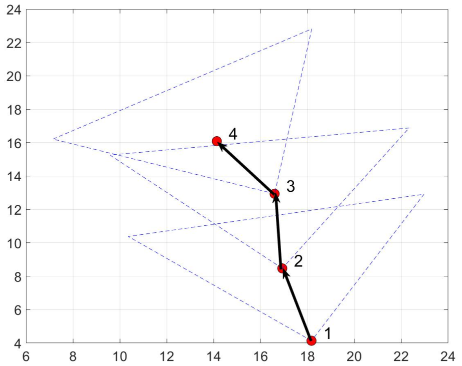


Figure R.12: Experiment 2 - Final position.

The evolution of each UAV's position throughout the complete experiment is given in Figure R.13. The evolution of each UAV's yaw is given in Figure R.14. The measured velocity for each UAV is given in Figure R.15.

C.2.2 Conclusion

In this experiment, we demonstrate the following:

- Follower robots $1, 2$ and 3 are able to adapt to the sudden changes in the trajectory of the leader robot, as evident in the vigorous velocity profiles shown in Figure R.15 ;
- Each follower is able to maintain the same interaction topology, Figure R.11 , which was at the initial time, at the final time as shown in Figure R.12;

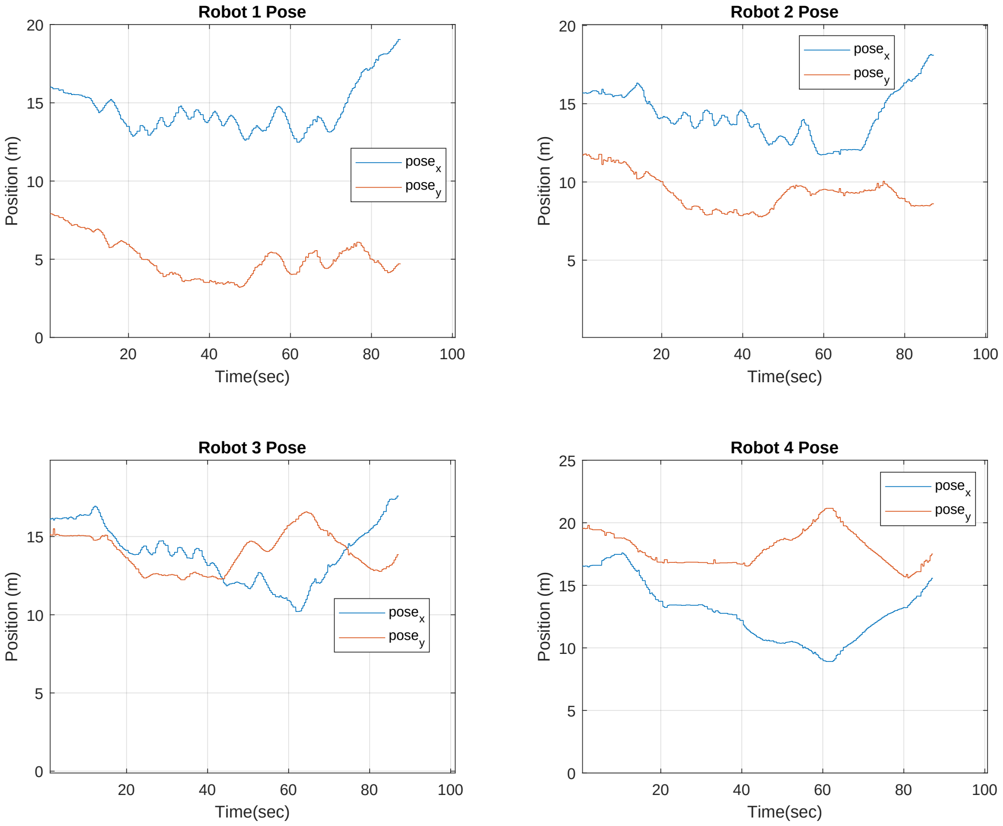


Figure R.13: Experiment 2 - Position data for all robots.

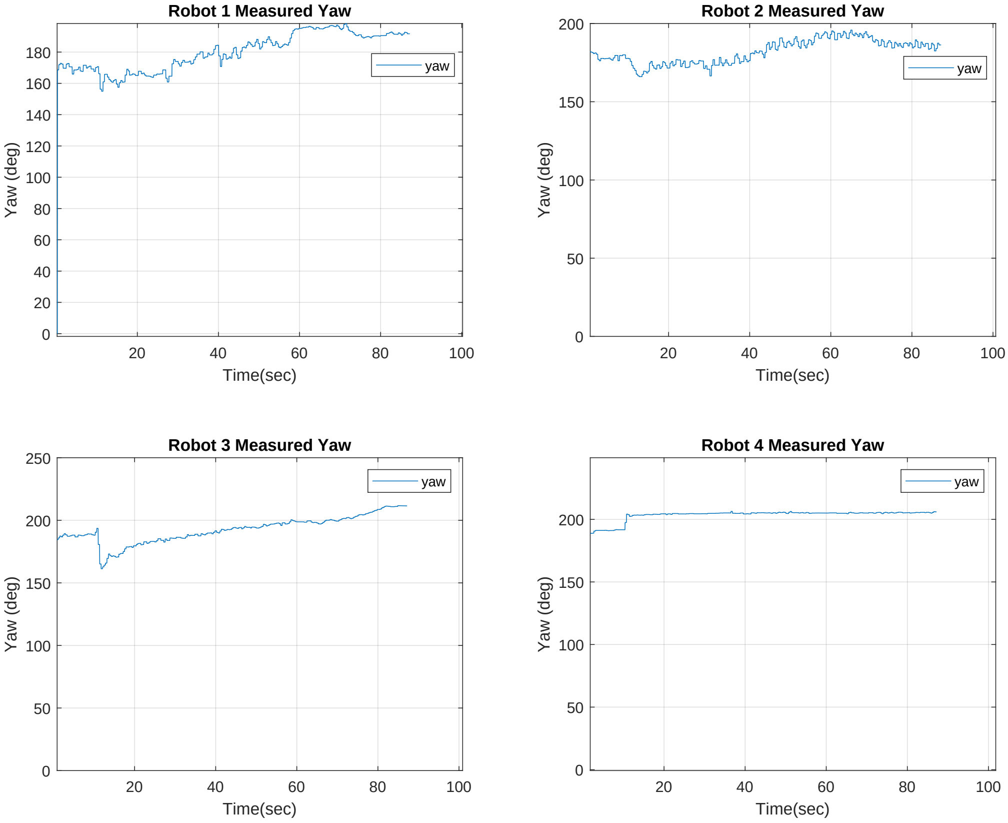


Figure R.14: Experiment 2 - Yaw data for all robots.

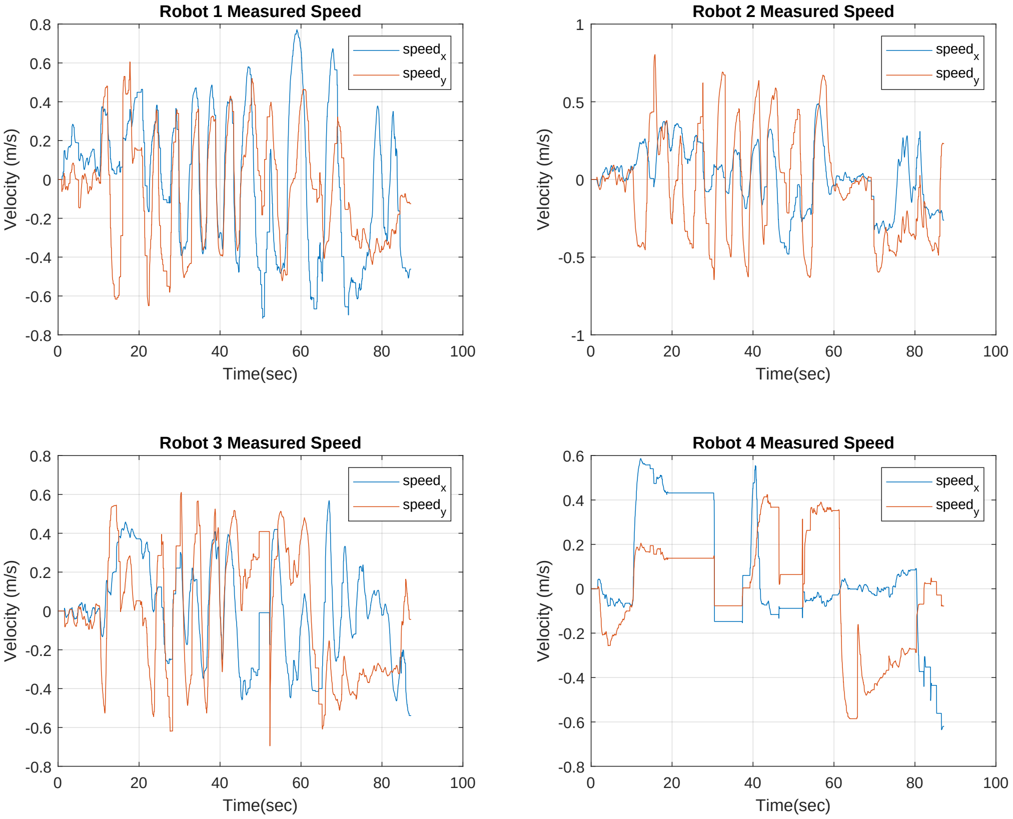


Figure R.15: Experiment 2 - Velocity data for all robots.

C.3 Experiment 3 - Demonstration of Machine State $\mathfrak{R} \rightarrow \mathfrak{C} \rightarrow \mathfrak{P} \rightarrow \mathfrak{R}$

In this experiment, we change the graph configuration as shown by the initial interaction graph in Figure R.16. Using this configuration, we will demonstrate the transition of all the machine states: 1) reactive, 2) critical, 3) planning. Specifically, robots 1 and 2 will be going into the critical state, as they will approach each other very closely. At the same time, robot 4, with a prescribed trajectory will escape the FOV of its neighboring robot 3 (simulating for example a sensor or motion failure that disturbs the topology maintenance) to move the multi-robot system to machine state $\mathfrak{P}$. The centralized planner then produces a trajectory for robot 4 that will return back to re-enter the FOV of robot 3 bringing the multi-robot system to machine state $\mathfrak{R}$ or the reactive state.

C.3.1 Results

The initial and final configurations of the team of UAVs are shown in Figures R.16 and R.17.

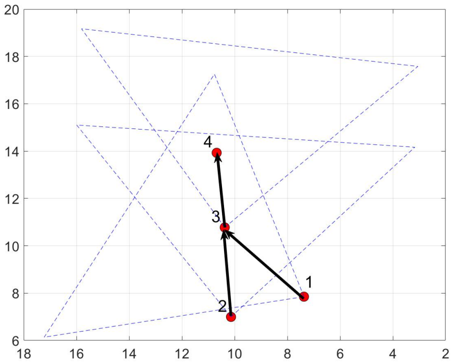


Figure R.16: Experiment 3 - Initial position.

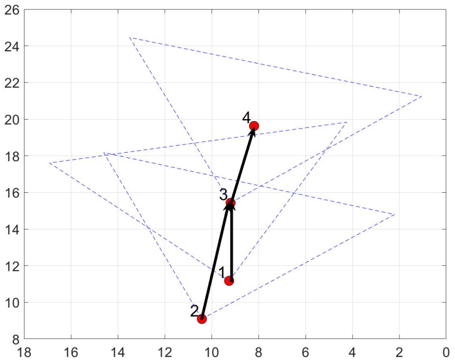


Figure R.17: Experiment 3 - Final position.

The evolution of each UAV's position throughout the complete experiment is given in Figure R.18. The sinusoidal characteristic observed in the position data profile in Figure R.18 , both in x and y for robots 1 and 2 occur because of significant wind disturbances during the experiment coupled with the fact that both robots are trying to exit the critical state. In this phenomenon, once the robots are in the critical state, in other words they approach each other closely, a repulsive force is applied to move the robots far away from each other so that they can exit the critical state. However, because of the graph configuration, the robots keep moving in and out of the state. At the same time, at about $50sec$, robot 4 moves out of the FOV of robot 3 moving the system to machine state $\mathfrak{P}$. Robot 4 moves back into the FOV of robot 3 at about $80sec$, re-initiating the reactive machine state $\mathfrak{R}$. The evolution of each UAV's yaw is given in Figure R.19. The measured velocity for each UAV is given in Figure R.20. Finally, we will discuss the machine state transitions. In Figures R.21 and R.22, we show the distance plot between robot 1 and 3 and 2 and 3 in comparison to the critical distance $4.5m$. The significance of the critical distance is that when any two robots come close enough to be at a distance less than the critical distance, the multi-robot system transitions to critical state. From the graphs in Figures R.21 and R.22, we can see that at about $t = 30s$ robots $1, 2$ and 3 enter the critical state from the reactive state as both robots $1, 2$ have a distance with robot 3 below the critical distance threshold. Once these robots, $1, 2$ and 3 are in critical state, robot 4, which is the leader with prescribed trajectory, keeps following its trajectory and leaves the FOV of its following robot which is robot 3. This is evident in the video that we have submitted with this document. Now because robot 4 leaves the FOV of robot 3, the graph is disconnected and hence the machine state transits to the planning state. With the initiation of the planner robot 4 is made to re-locate itself and modify its trajectory such that it re-renters the FOV of robot 3 to connect the graph. In the meantime the robots $1, 2$ and 3 are attempting to move out of the critical distance, and consequently the critical state. This is evident from the oscilatory distance profile in Figures R.21 and R.22 between $t = 30s$ and $t = 84s$. Finally, at $t = 84s$, the multi-robot system goes back to its reactive state, with its initial stable graph configuration to carry on with its assigned coordination task, which is to conduct a leader-follower scenario. Also, note that robot 1 crosses the critical distance threshold with robot 3 again at about $t = 110s$, which will again be taken care of by our finite machine state adaptive mechanism. At this point our experiment finishes.

C.3.2 Conclusion

In this experiment, we demonstrate the following:

- We are able to demonstrate the transition of machine states $\mathfrak{R} \rightarrow \mathfrak{C} \rightarrow \mathfrak{P} \rightarrow \mathfrak{R}$ with real robots in a new graph configuration;

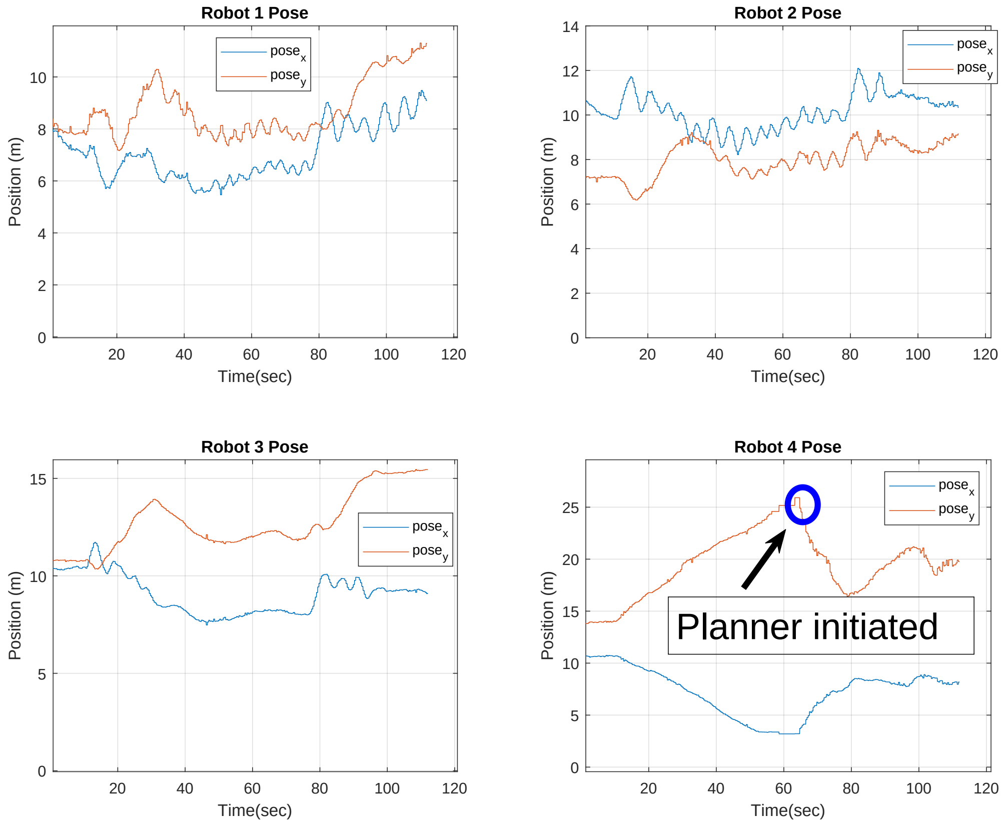


Figure R.18: Experiment 3 - Position data for all robots.

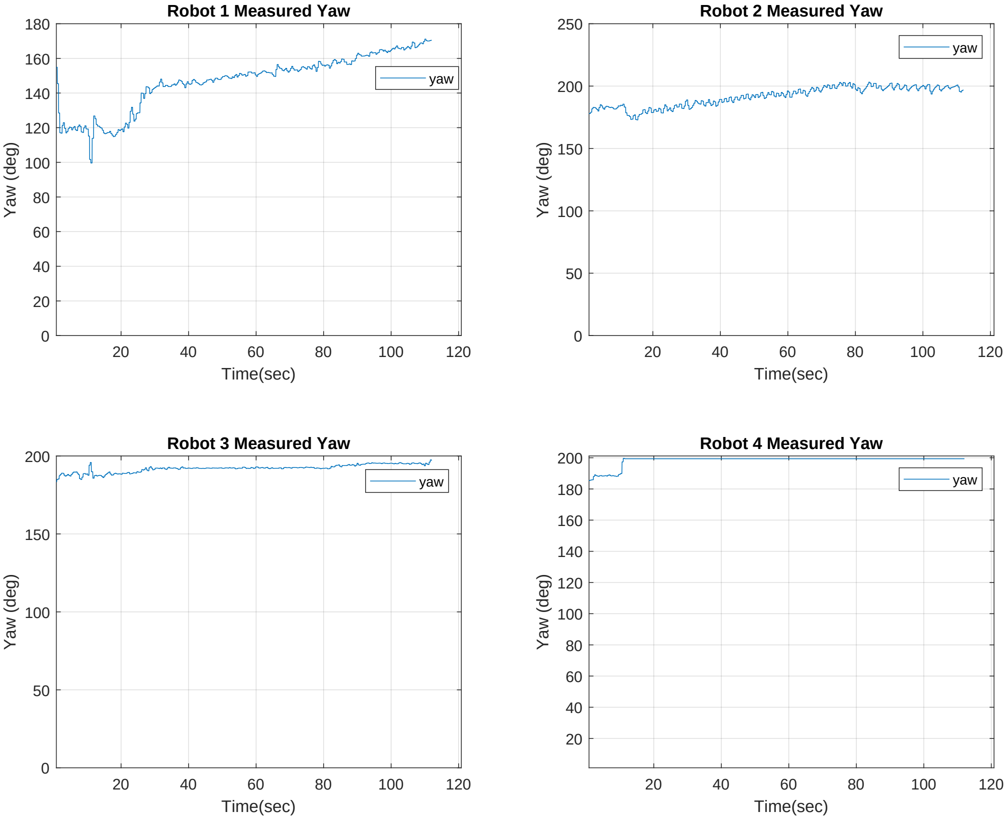


Figure R.19: Experiment 3 - Yaw data for all robots.

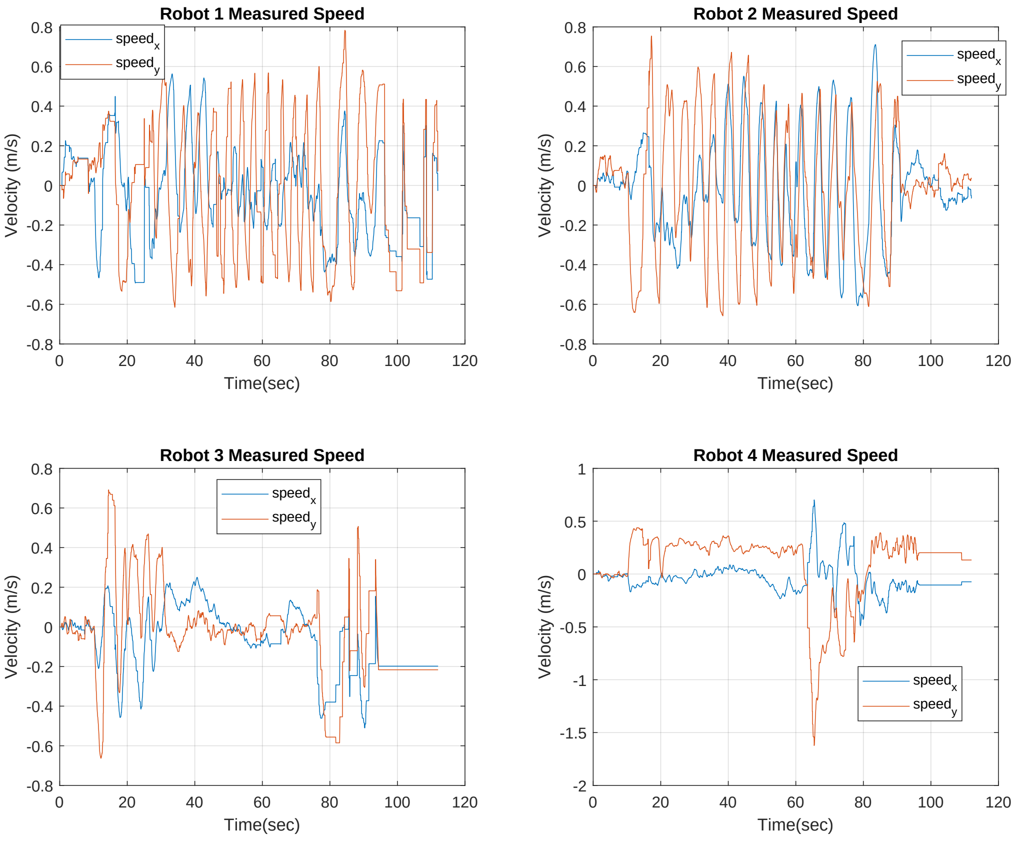


Figure R.20: Experiment 3 - Velocity data for all robots.

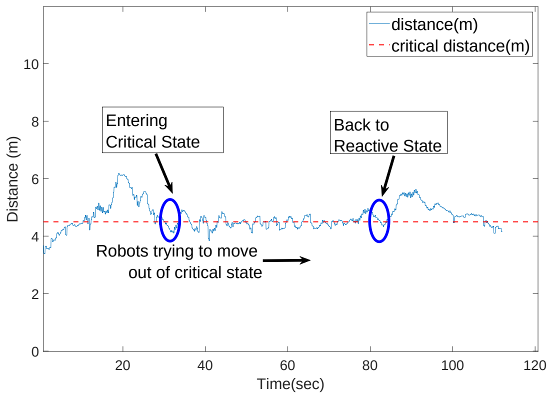


Figure R.21: Experiment 3 - Distance (blue line) between robots 2 and 3 in comparison to critical distance (red dotted line) of 4.5m.

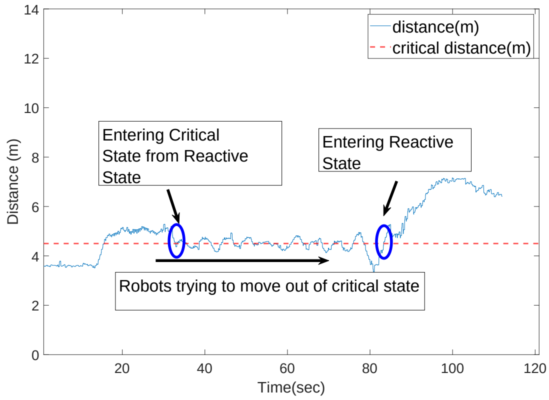


Figure R.22: Experiment 3 - Distance (blue line) between robots 2 and 3 in comparison to critical distance (red dotted line) of 4.5m.

C.4 Experiment 4 - Demonstration of Rotation of FOV

This experiment is a simple experiment, to demonstrate that our limited FOV controller is able to tackle situations that call for significant FOV rotation. We manually fly the leader robot 4 using the remote controller, and the FOV controller computes the control law for all the other robots $1, 2$ and 3. We can observe robot 1 rotating by almost $120°$ to keep its neighboring robot 3 in the FOV.

C.4.1 Results

The initial and final configurations of the team of UAVs are shown in Figures R.23 and R.24.

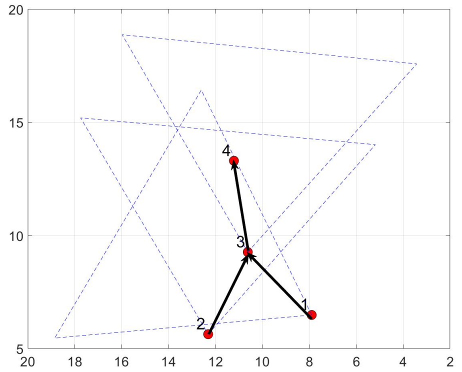


Figure R.23: Experiment 4 - Initial position.

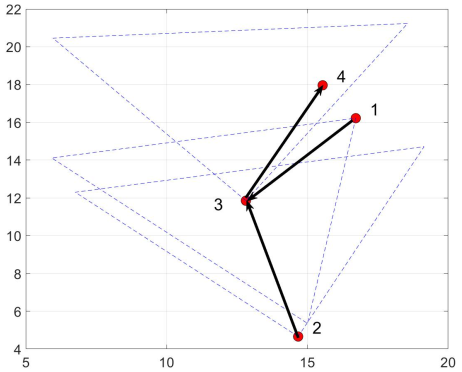


Figure R.24: Experiment 4 - Final position.

The evolution of each UAV's position throughout the complete experiment is given in Figure R.25. The evolution of each UAV's yaw is given in Figure R.26.Note that robot 4 rotates by almost $120°$ The measured velocity for each UAV is given in Figure R.27.

C.4.2 Conclusion

In this experiment, we demonstrate the following:

- We demonstrate limited FOV control of four robots, in the reactive state, where one of the robots rotates significantly to keep its neighboring robot in the FOV.

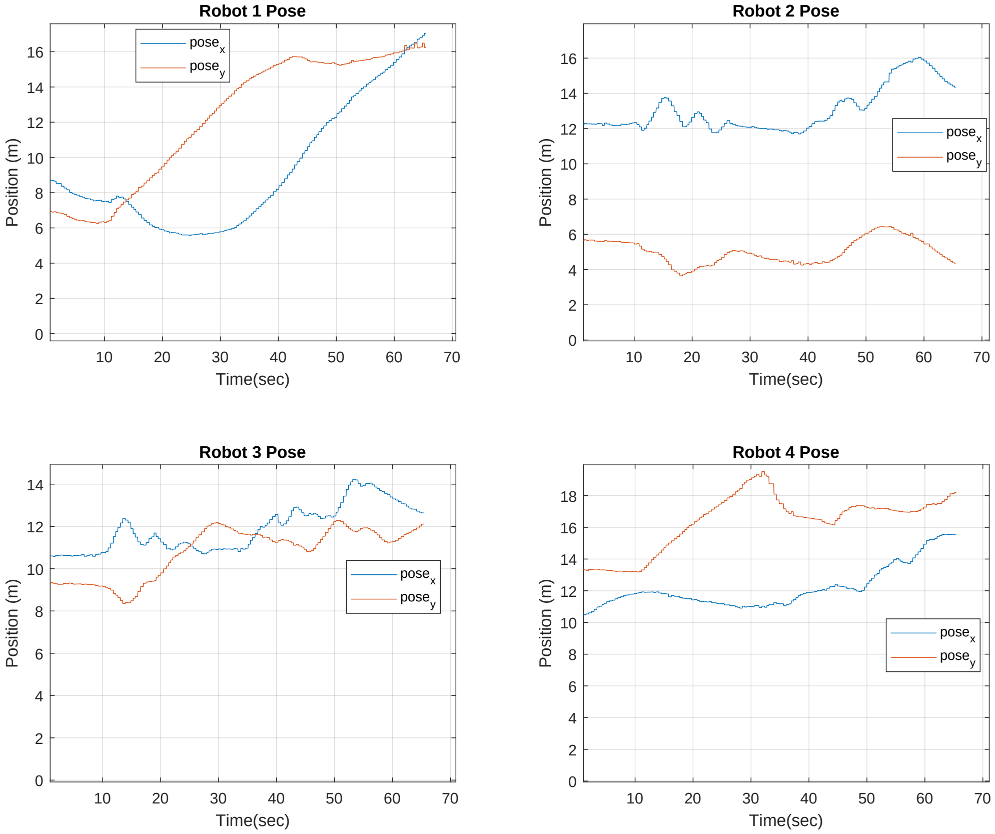


Figure R.25: Experiment 4 - Position data for all robots.

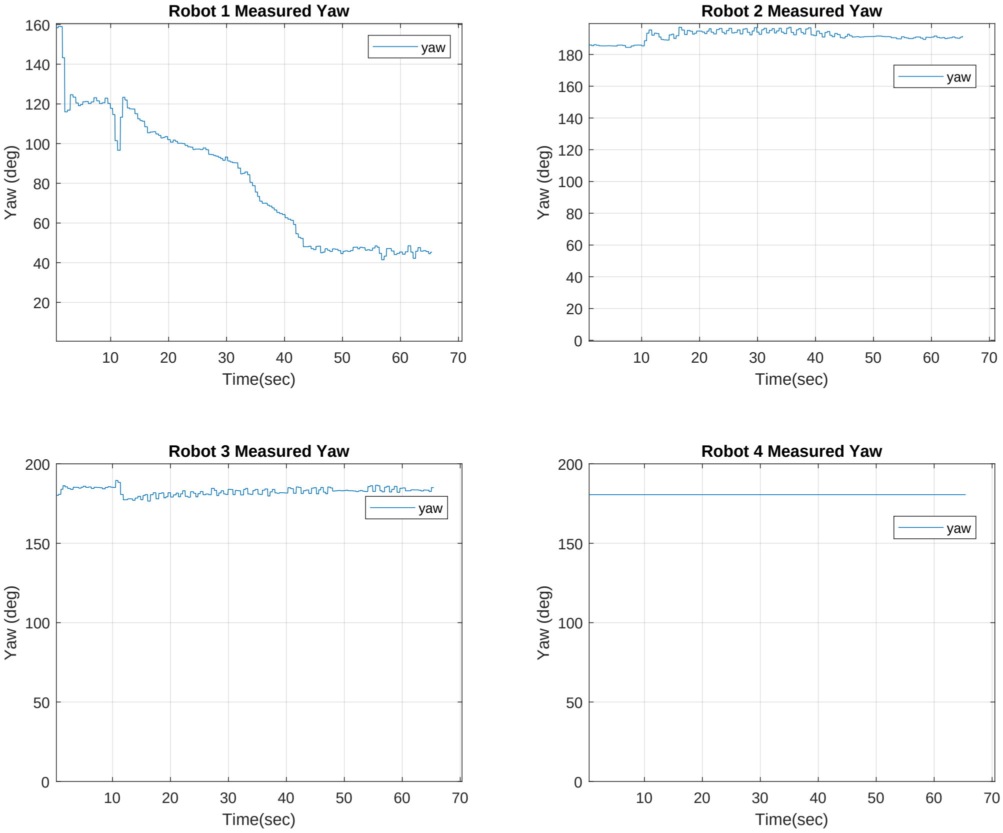


Figure R.26: Experiment 4 - Yaw data for all robots.

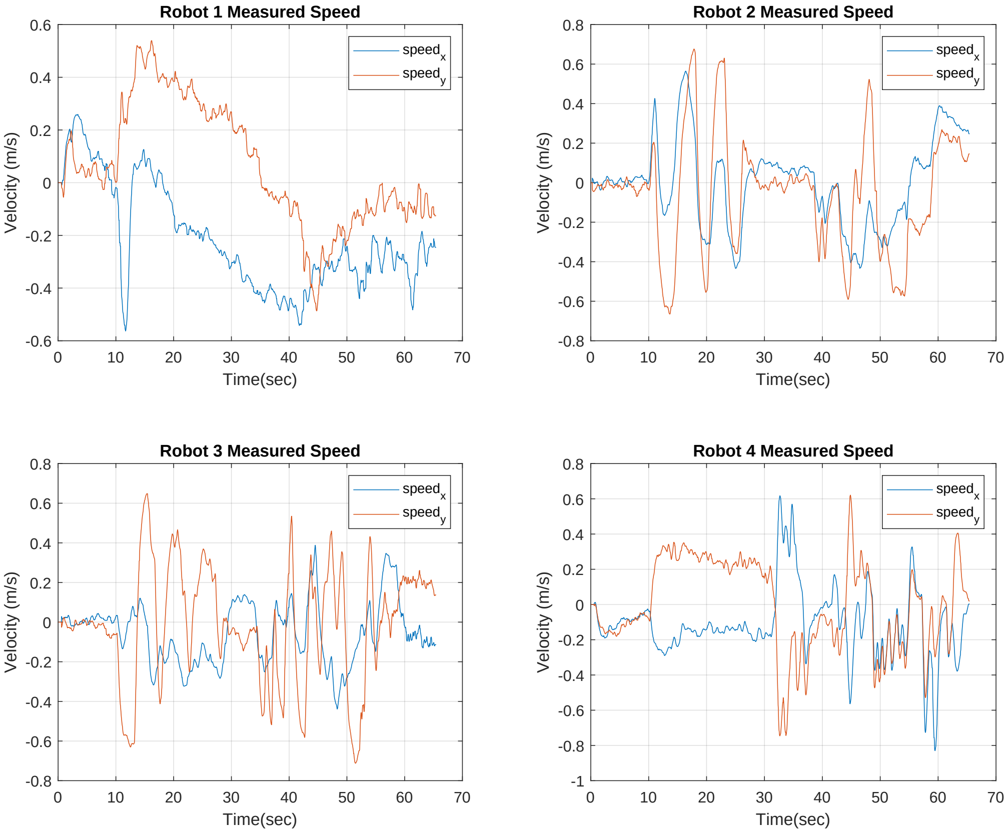


Figure R.27: Experiment 4 - Velocity data for all robots.

C.5 Experiment 5 - Demonstration of Machine State $\mathfrak{P} \to \mathfrak{R}$

In this experiment, we demonstrate the transition of machine states from $\mathfrak{P} \to \mathfrak{R}$. First, the leader robot is provided a velocity to sweep across a $20m$ field, move forward in y direction for $20sec$, move backward for $20sec$ in y direction and sweep back to its original position as also evident in the pose plot of robot 4 given in Figure R.30. The leader robot then flies a trajectory to simulate an escape from the FOV of follower robot 3, disconnecting the interaction graph and leading the system to move from machine state $\mathfrak{P} \to \mathfrak{R}$. Once the leader robot completes it prescribed trajectory and returns back to its position, it re-enters the FOV of robot 3 making the machine state transition $\mathfrak{P} \to \mathfrak{R}$, from the *planning* state to the *reactive* state.

C.5.1 Results

The initial and final configurations of the team of UAVs are shown in Figures R.28 and R.29.

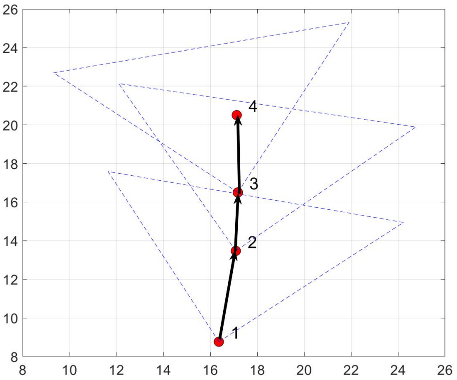


Figure R.28: Experiment 5 - Initial position.

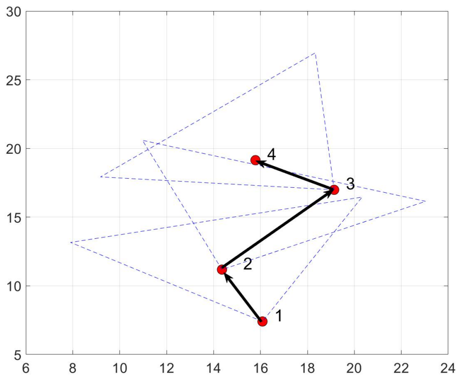


Figure R.29: Experiment 5 - Final position.

The evolution of each UAV's position throughout the complete experiment is given in Figure R.30. It is evident from the position plots of robots $1, 2$ and 3 that the machine state transition has occurred. The pose plot in Figure R.30 for robot 3 shows that after about $20sec$ the leader robot 4 has escaped its FOV because after this point robot 3 is more or less in the same position until about $90sec$ when leader robot 4 re-enters its FOV. This is also evident from the measured velocity plot of robot 3 shown in Figure R.32. In the meantime that the leader robot 4 has escaped the FOV of robot 3, robots 1 and 2 are just trying to keep their respective neighbors, robot 2 for robot 1 and robot 3 for robot 2, in their respective FOV. The evolution of each UAV's yaw is given in Figure R.31. The measured velocity for each UAV is given in Figure R.32. Note that at about $t = 25s$ robot 4 simulates an escape of the FOV of robot 3. This leads the multi-robot system to transition to machine state $\mathfrak{P}$. Then the planner is initiated to bring back robot 4 into robot $3's$ FOV to bring the system back into the reactive state. This occurs at about $t = 90s$ as can be seen in the velocity profile of robot 3 which exhibits a sudden jump. When the robots are in the planning state, especially robot 3, they do not have any robot in their FOV, thus they just hover and hence the velocity profile is close to zero between $t = 25s$ and $t = 90s$.

C.5.2 Conclusion

In this experiment, we demonstrate the following:

- In the instance that a robot, in this case the leader robot 4, escapes the FOV of its neighboring robot and disconnects the graph, the machine state transits from $\mathfrak{R} \to \mathfrak{P}$.
- Since the leader robot's prescribed trajectory was to move back to its original position, the leader robot was able to re-enter the FOV of robot 3 and execute the machine state transition $\mathfrak{P} \to \mathfrak{R}$. This clearly emphasizes the advantage of having our multi-robot system equipped with the adaptive mechanism of the *Finite State Machines* because without the mechanism aid, the system would be disabled due to the disturbance from external factors like wind which lead to the leader robot escaping the FOV of its follower robot;

- This experiment demonstrates a simple machine state transition, which can be made more sophisticated with an online planner when the multi-robot system is in machine state $\mathfrak{P}$. However, because our paper does not focus on the planner details, we here show the functionality of the state transitions with real experiments.

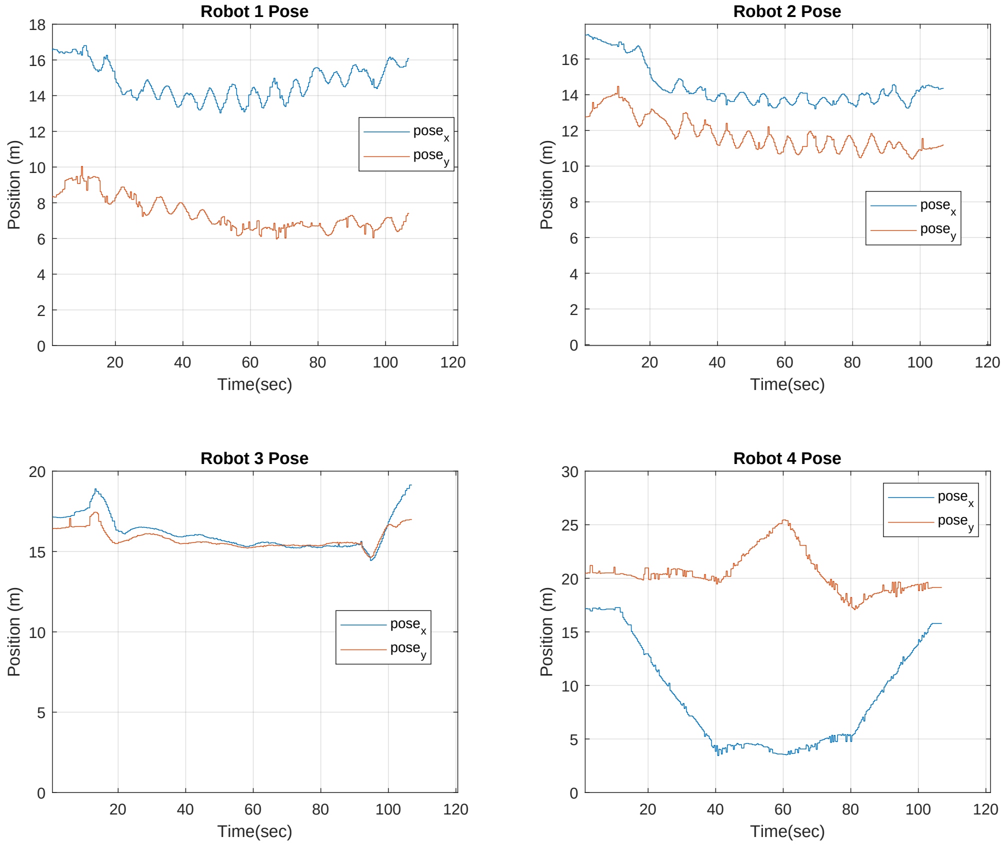


Figure R.30: Experiment 5 - Position data for all robots.

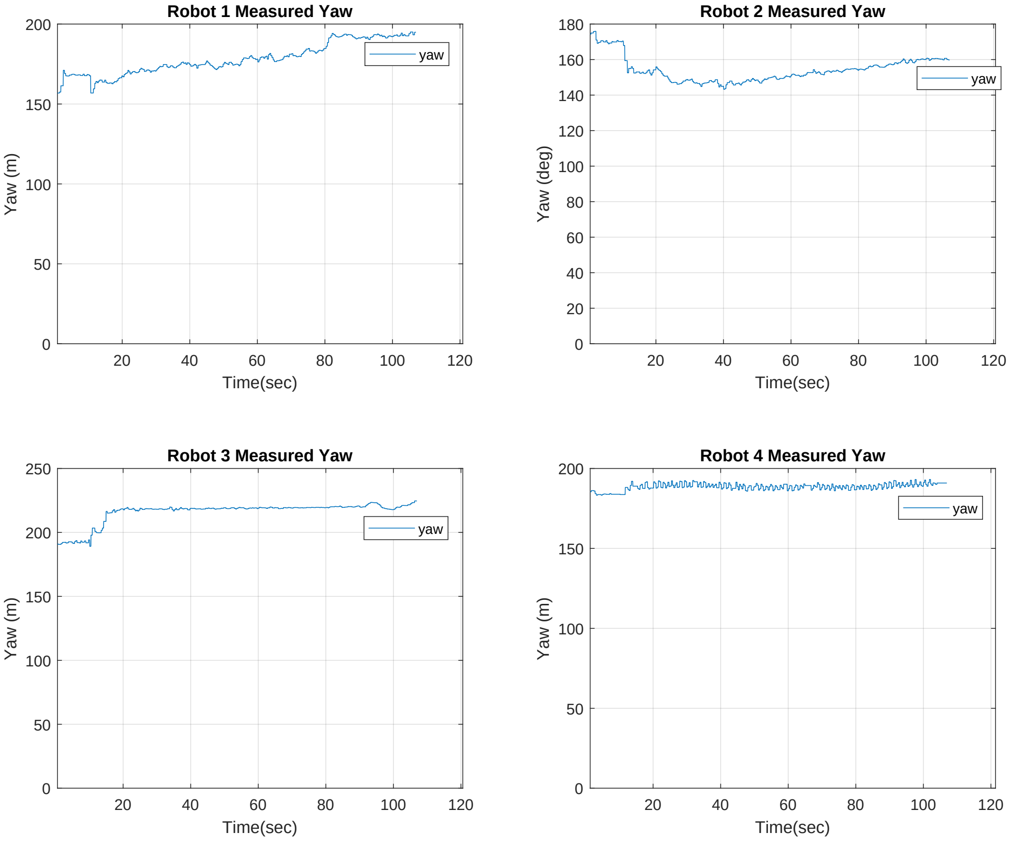


Figure R.31: Experiment 5 - Yaw data for all robots.

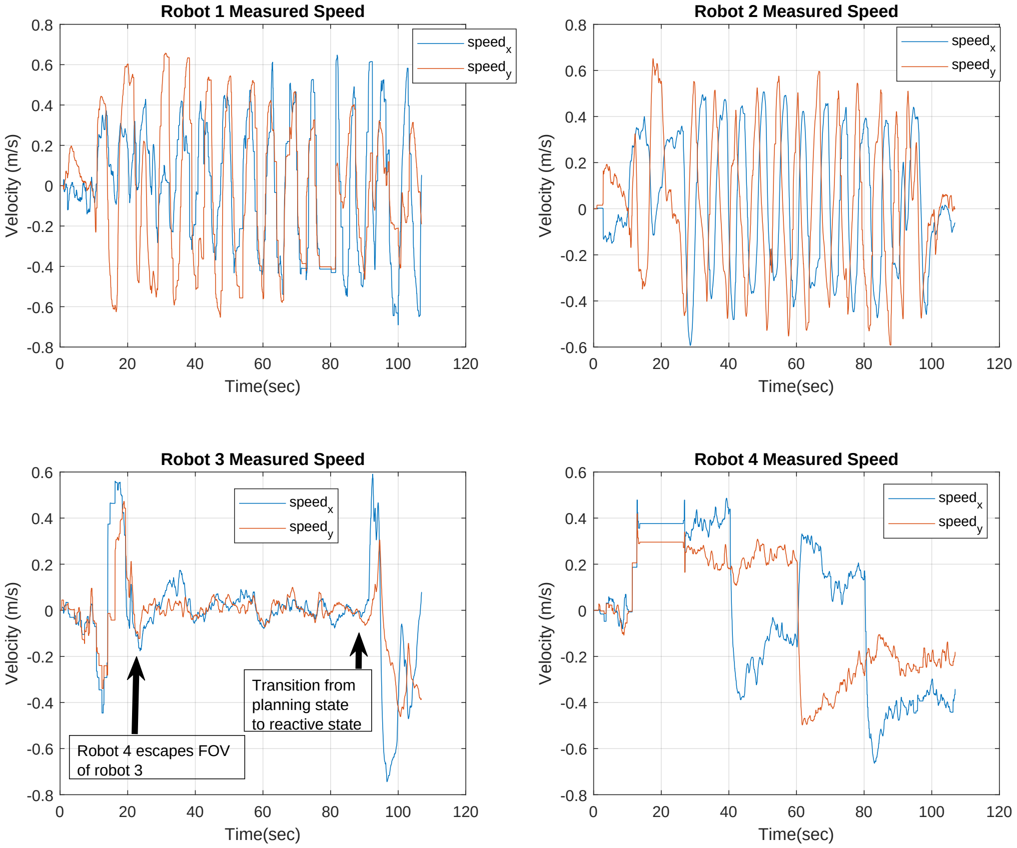


Figure R.32: Experiment 5 - Velocity data for all robots.

D Discussion

From this report, we intend to showcase the versatility of our limited FOV controller. We have presented experimental results from five different experiments where each experiment achieves different objectives. We intend to showcase that our controller works with real robots, in an outdoor environment for the following scenarios

- Different graph configurations;
- Higher velocities;
- Varying trajectories of the leader robot;
- Transitioning between different machine states;
- Rotation of the robots to maintain FOV;

Now that we have discussed the objectives we wanted to fulfill with the above results from the five different experiments, we will like to provide the reviewer some insight into the limitations of the experimental set-up that is the *portable multi-robot testbed*. Currently, we are limited in the number of equipment that we possess. Firstly, we have only four robots to experiment with. Our localization system is also limited which limits the size of the experimental setup we can work with. For instance, with six Pozyx anchor nodes we can only set up an optimum size of $30m \times 20m$ to conduct our experiments. For us to emulate our Gazebo simulation, we will need more anchor nodes, to increase the size of our experimental setup. Related to the accuracy of the localization system, with the current Pozyx kit that we possess, the manufacturers claim that we can obtain accuracy up to $10cm$ which is much higher than indoor localization systems like VICON which gives accuracy up to mm. The update rate of the Pozyx tags, that are attached rigidly to the robots, are also down to about $20Hz$ which is not ideal for real-time multi-robot control applications. Finally, we conduct experiments in outdoor setting, at the Virginia Tech Drone Park, where wind speeds are upwards of $10mph$.

We definitely know that to improve our experiments we will have to improve our equipment, especially the localization system for the robots which must have more accuracy as well as a suitable update rate for real time execution. However, our intention, by showcasing theses results, in not to highlight the deficiency in our equipment and claim that we can simply improve our experiments with better equipment such as implementing a robust motion capture system such as VICON. We believe, that the results that we have showcased are promising because we are able to show the functionality of our controller with limitations in our equipment on real robots. We believe these results are a stepping stone for the next part of our project where we intend to implement an on-board multi-robot robot to robot relative localization system where we use on-board cameras to localize neighboring robots and map our current limited FOV controller to this system to conduct multi-robot coordination. To the best of our knowledge, there still does not exist any state-of-the-art relative robot to robot localization system using vision for multi-robot coordination specifically because it is a very difficult and challenging feat to achieve. However, we in this paper, exhibit a controller that works with non-ideal equipment and still performs well. Currently, we have preliminary work done in [1] to develop a multi-robot robot to robot relative localization system. Our next objective is to implement the work in [1] with our *portable multi-robot testbed.*

Overall, we believe our experimental results exhibit promising results of a limited FOV controller for multi-robot coordination with off-the shelf equipment such as the DJI Matrice 100 UAVs in realistic outdoor settings which can be further enhanced with a state-of the art robot-robot localization system.

References

[1] S. R. Yellapantula. *Synthesizing realistic data for vision based drone-to-drone detection.* PhD thesis, Virginia Tech, 2019.