IEEE Transactions on Robotics, Extended Analysis of Experiments **IEEE T-RO 20-0271**

"Multi-Robot Field of View Control with Adaptive Decentralization"

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

1 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 3.1);
2. Variant leader trajectory (Section 3.2);
3. Demonstration of Machine State transition (Section 3.3);
4. Robots with substantial, fast FOV rotation (Section 3.4).
5. An alternate demonstration of Machine State transition (Section 3.5);

The rest of the report is outlined as follows: In Section 2, we will discuss the details of the different hardware and software components of the *Portable Multi-Robot Testbed*; In Section 3, we provide the results from five different outdoor experiments we conducted using the *Portable Multi-Robot Testbed*. Note that these reports act as a detailed explanation of the improved (but condensed) material added in the revised manuscript.

2 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

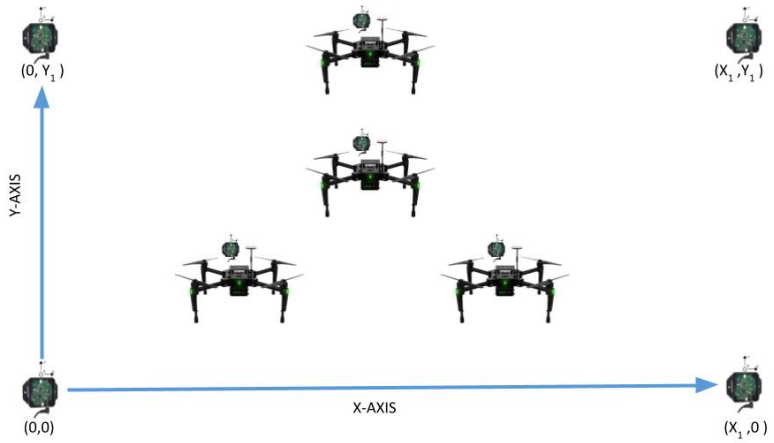

Figure R.1: Pozyx anchors are set at known positions(Cartesian coordinate system) to define the perimeter of the experimental environment and Pozyx tags are rigidly attached to M100s to obtain IMU and localization data.

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.

2.1 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.1. 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.

2.2 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.

3 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. Notably, the described scenario with four UAVs would theoretically not require the use of the adaptive decentralization framework since all the interactions that occur over the evolution of the system yield a stable coordinated motion according to the results of Thereom 1. For the sake of presentation, however, we assume here that that is not the case and we illustrate the evolution of the system in that scenario.

The triangular field of view $\mathcal{T}_i$ attached to robot i is defined by the side length $l_i = 8m$ and the angle $\alpha_i = 45°$. All the robots are also equipped with a near-field sensor for safety reasons which possess a radius of 4.5 m. 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 concerns raised by the Reviewers.

3.1 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.

3.1.1 Results

The initial and final configurations of the team of UAVs are shown in Figures R.2 and R.3.

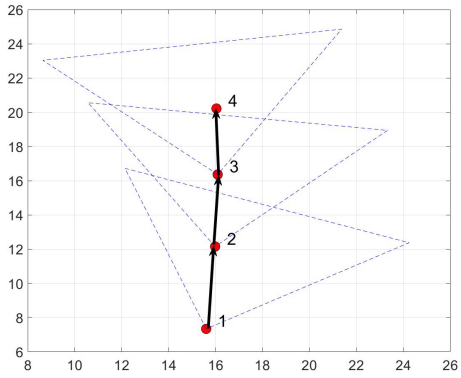

Figure R.2: Experiment 1 - Initial position.

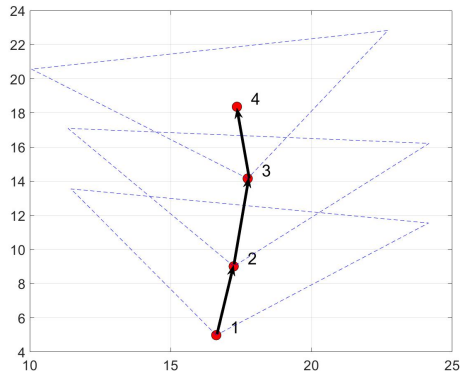

Figure R.3: Experiment 1 - Final position.

The evolution of each UAV's position throughout the complete experiment is given in Figure R.4. The evolution of each UAV's yaw is given in Figure R.5. The measured velocity for each UAV is given in

Figure R.6.

3.1.2 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.

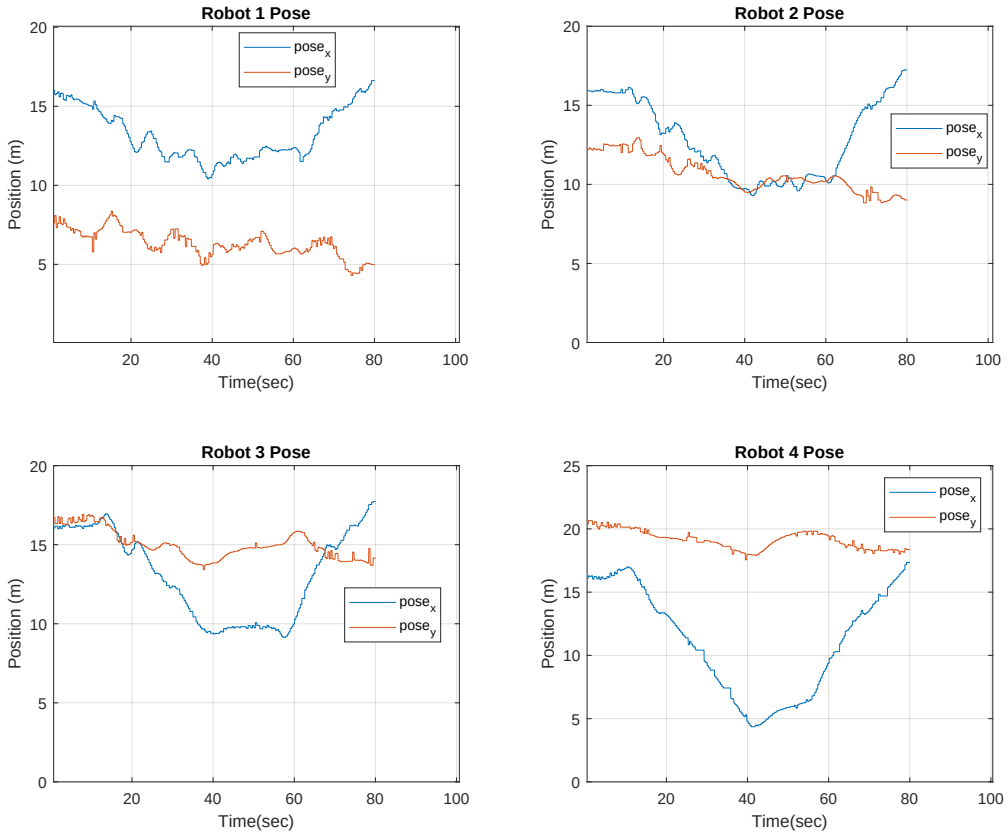

Figure R.4: Experiment 1 - Position data for all robots.

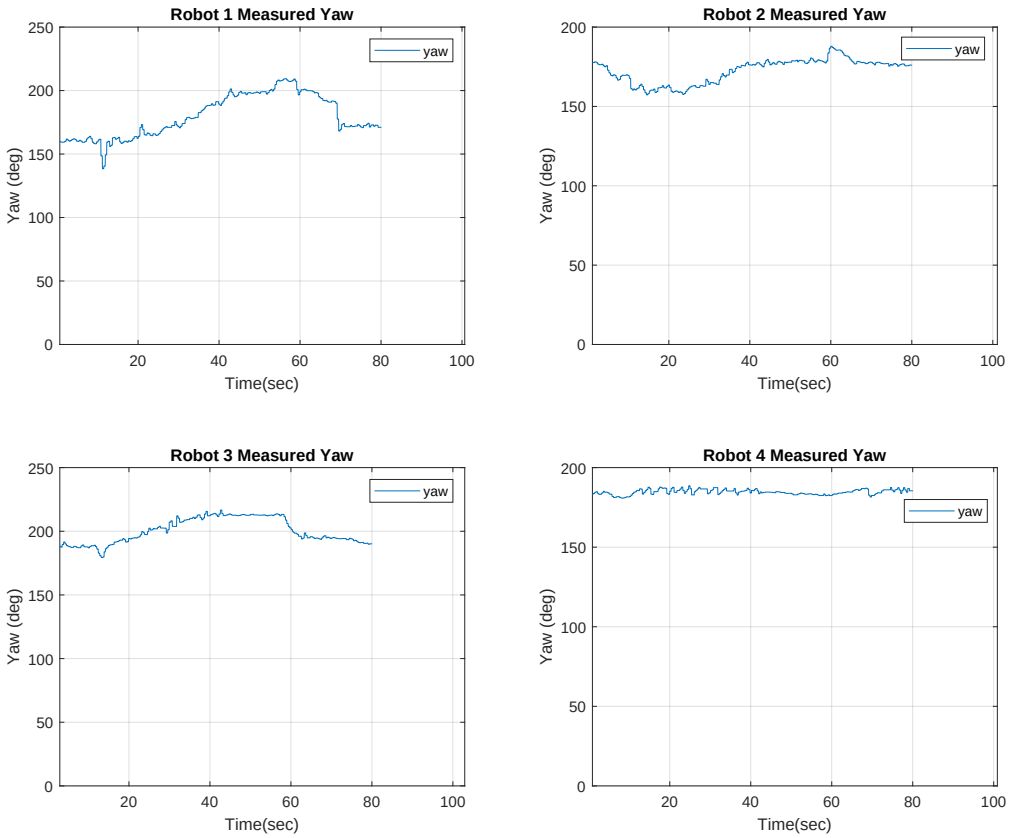

Figure R.5: Experiment 1 - Yaw data for all robots.

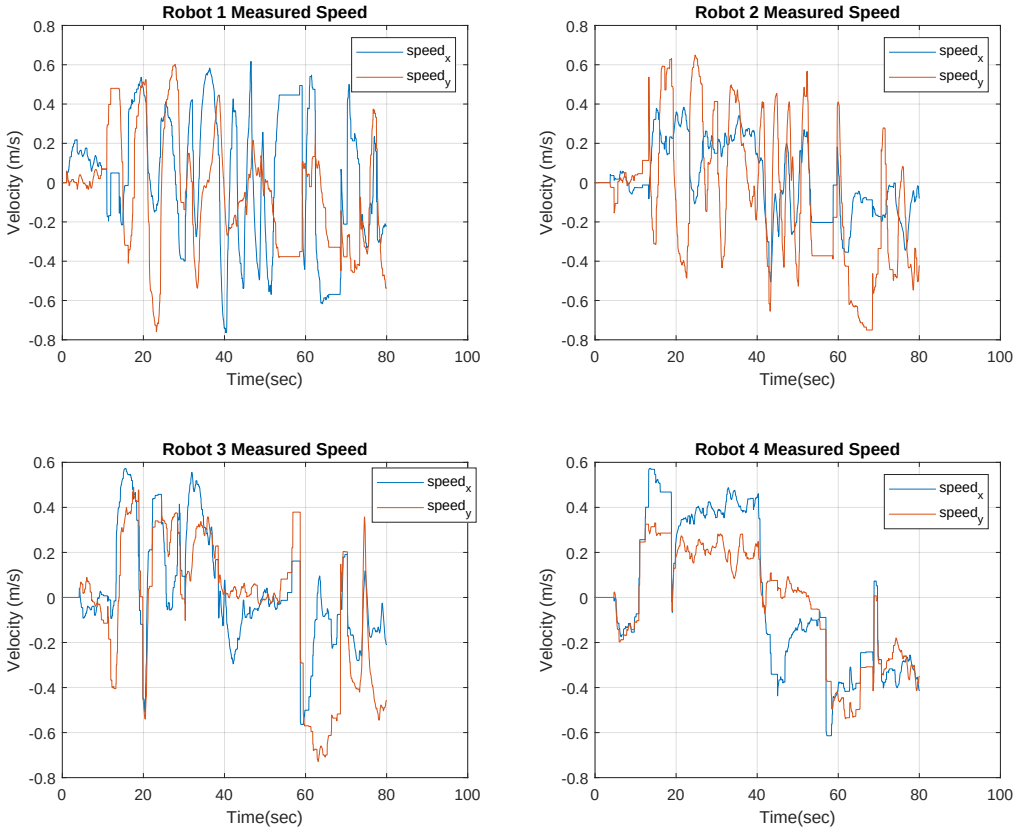

Figure R.6: Experiment 1 - Velocity data for all robots.

3.2 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.

3.2.1 Results

The initial and final configurations of the team of UAVs are shown in Figures R.7 and R.8.

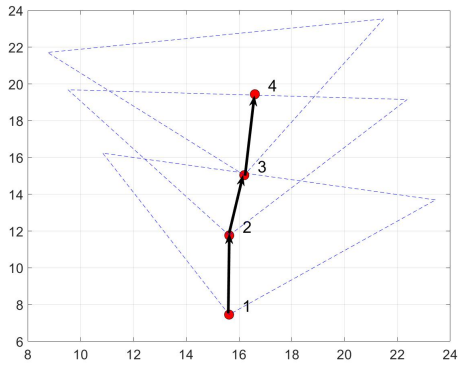

Figure R.7: Experiment 2 - Initial position.

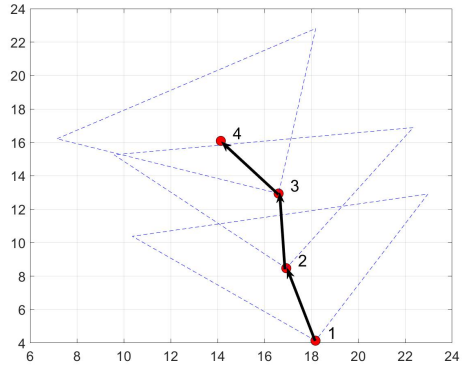

Figure R.8: Experiment 2 - Final position.

The evolution of each UAV's position throughout the complete experiment is given in Figure R.9. The evolution of each UAV's yaw is given in Figure R.10. The measured velocity for each UAV is given in Figure R.11.

3.2.2 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.11 ;
- Each follower is able to maintain the same interaction topology, Figure R.7 , which was at the initial time, at the final time as shown in Figure R.8;

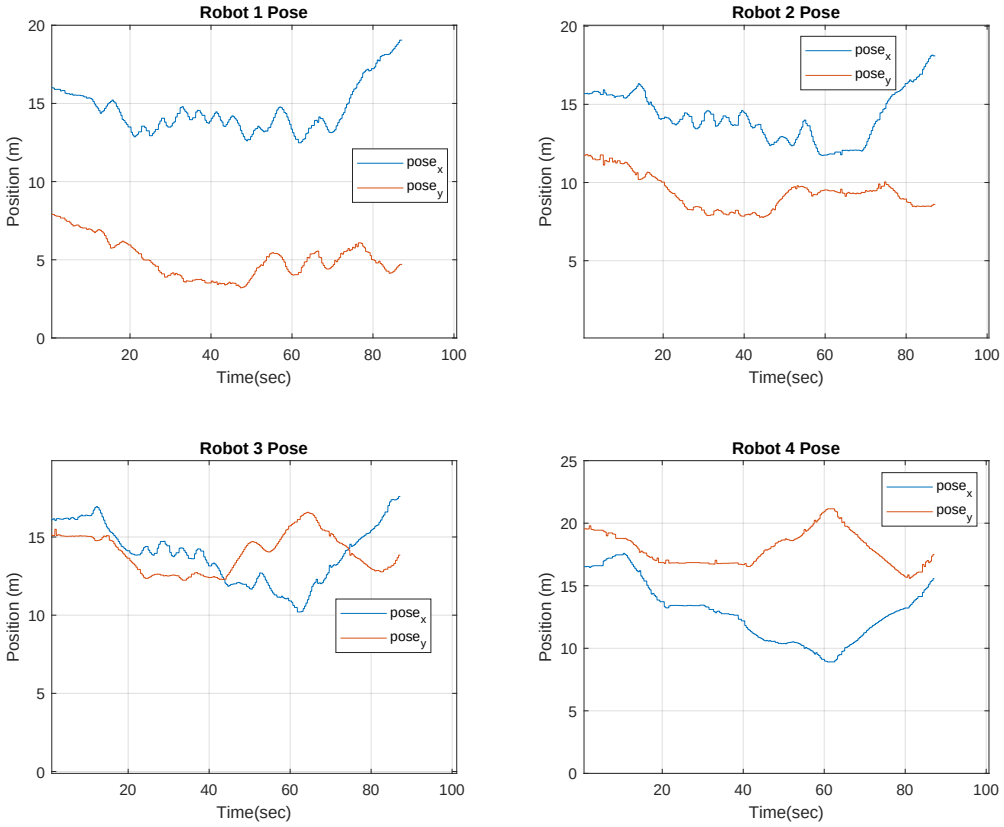

Figure R.9: Experiment 2 - Position data for all robots.

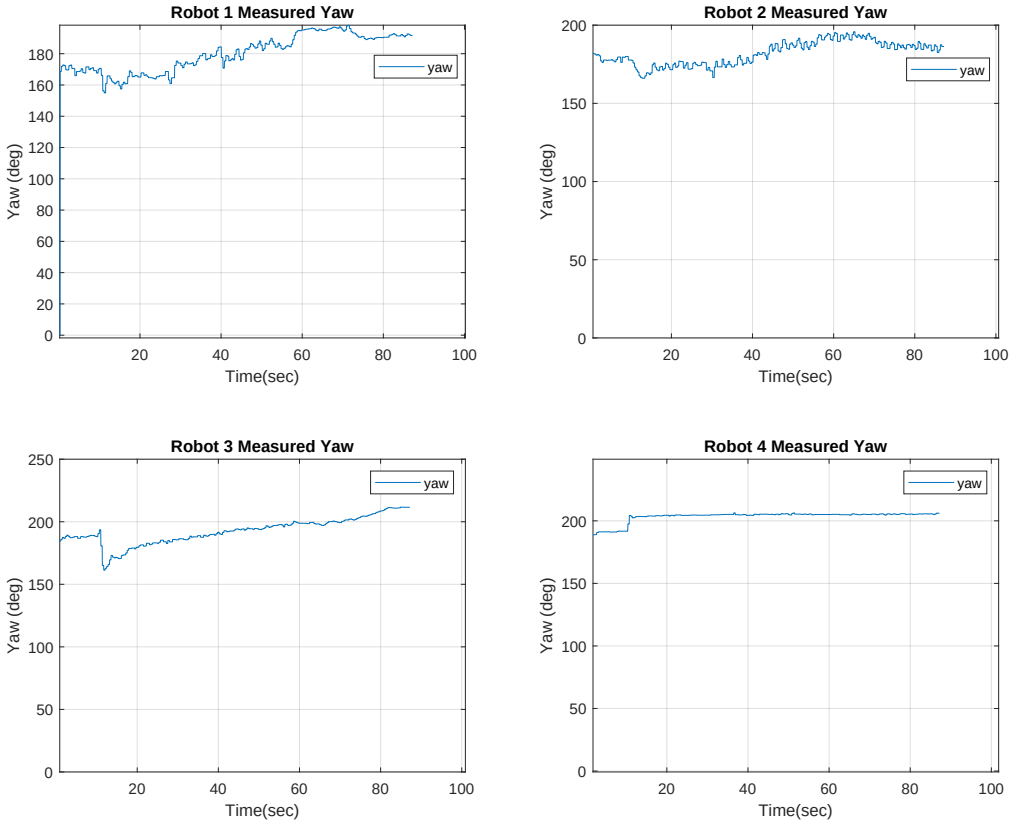

Figure R.10: Experiment 2 - Yaw data for all robots.

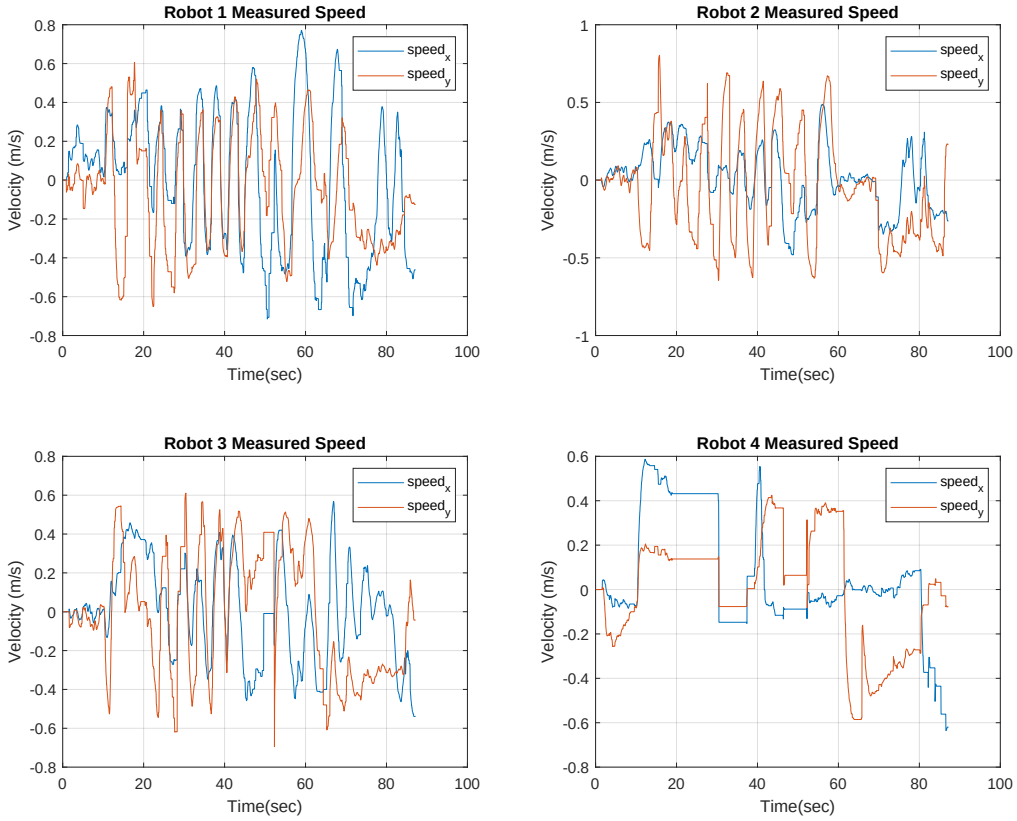

Figure R.11: Experiment 2 - Velocity data for all robots.

3.3 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.12. Using this configuration, we will demonstrate the transition of all the machine states: 1) Reactive, 2) Critical, 3) Planning. Specifically, robots 1, 2, and 3 will be going into the Critical state, as 1 and 2 will approach robot 3 very closely. At the same time, robot 4, continuing its normal operation, i.e., following the prescribed trajectory, escapes the FOV of robot 3 which instead is focused on addressing the potential collisions with robots 1 and 2. After the critical interactions have been addressed, the system switches into the Planning state $\mathfrak{P}$. The centralized planner then produces a trajectory for robot 4 that will relocate it into re-enter the FOV of robot 3 and allowing the multi-robot system to transition back into the Reactive state $\mathfrak{R}$.

3.3.1 Results

The initial and final states of the team of UAVs are shown in Figures R.12 and R.13.

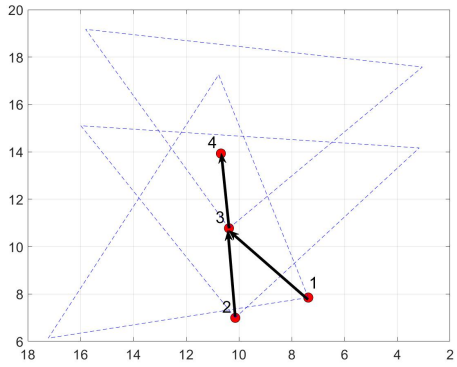

Figure R.12: Experiment 3 - Initial position.

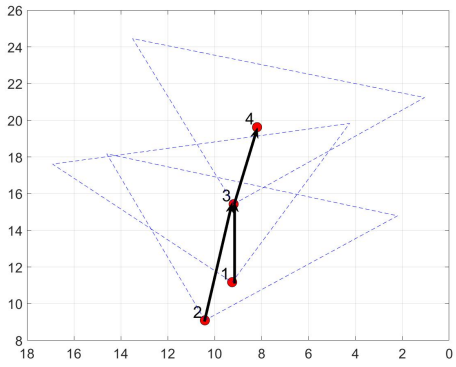

Figure R.13: Experiment 3 - Final position.

The evolution of each UAV's position throughout the complete experiment is given in Figure R.14. The sinusoidal characteristic observed in the position data profile in Figure R.14 , both in x and y axis 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 robots 1 and 2 are in the critical state with robot 3, a repulsive force is applied to move the robots far away from robot 3 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 $65sec$, 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 $84sec$, re-initiating the reactive machine state $\mathfrak{R}$. The evolution of each UAV's yaw is given in Figure R.15. The measured velocity for each UAV is given in Figure R.16. To bring more clarity in the understanding of the state machine transitions we will now study the distance plots of the robots in Critical state, that is the distance of robots 1,2 from robot 3. In Figures R.17 and R.18, 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.17 and R.18, 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.17 and R.18 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.

3.3.2 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;

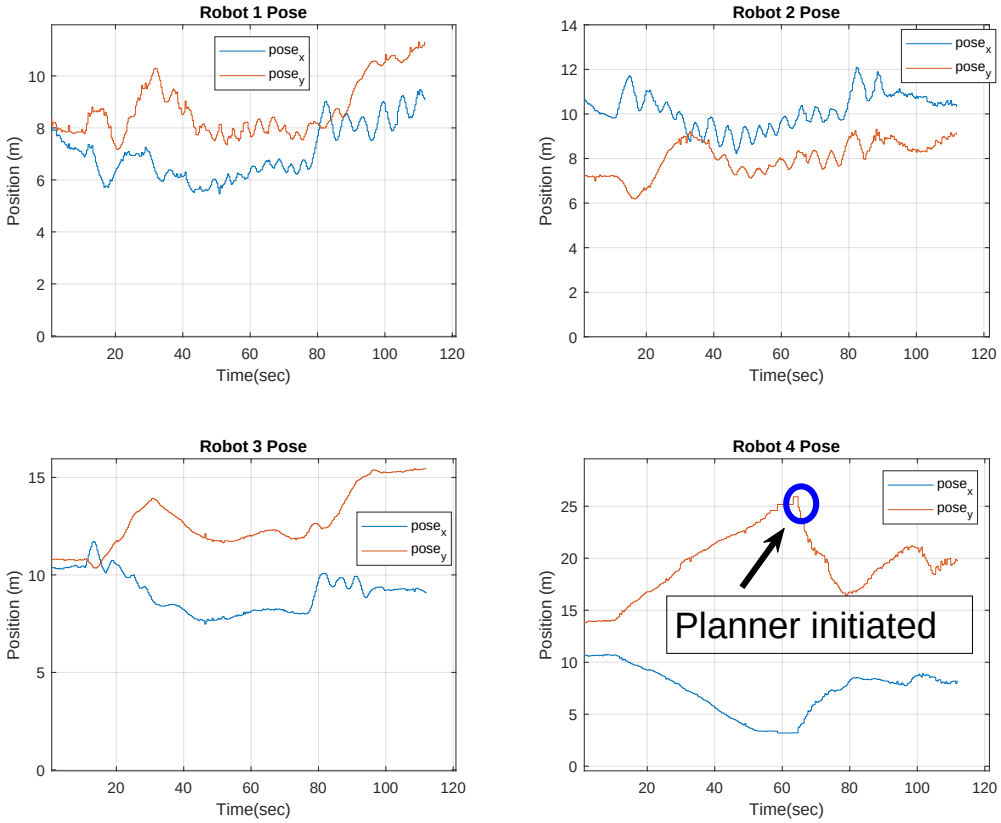

Figure R.14: Experiment 3 - Position data for all robots.

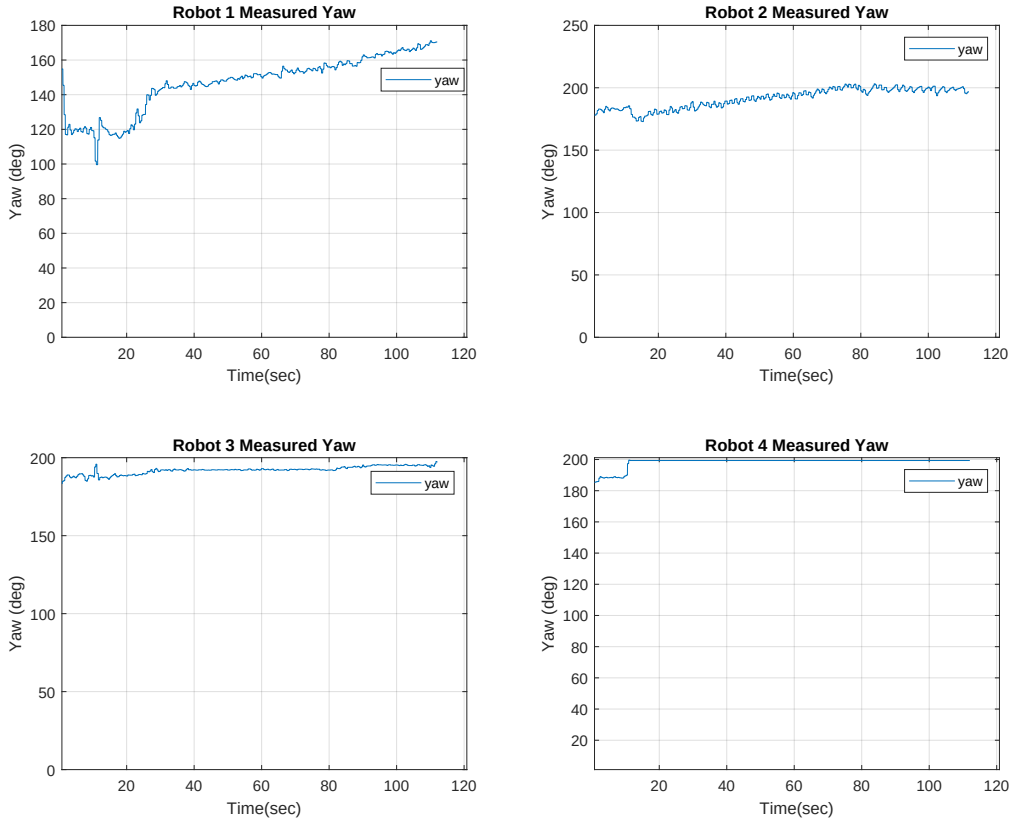

Figure R.15: Experiment 3 - Yaw data for all robots.

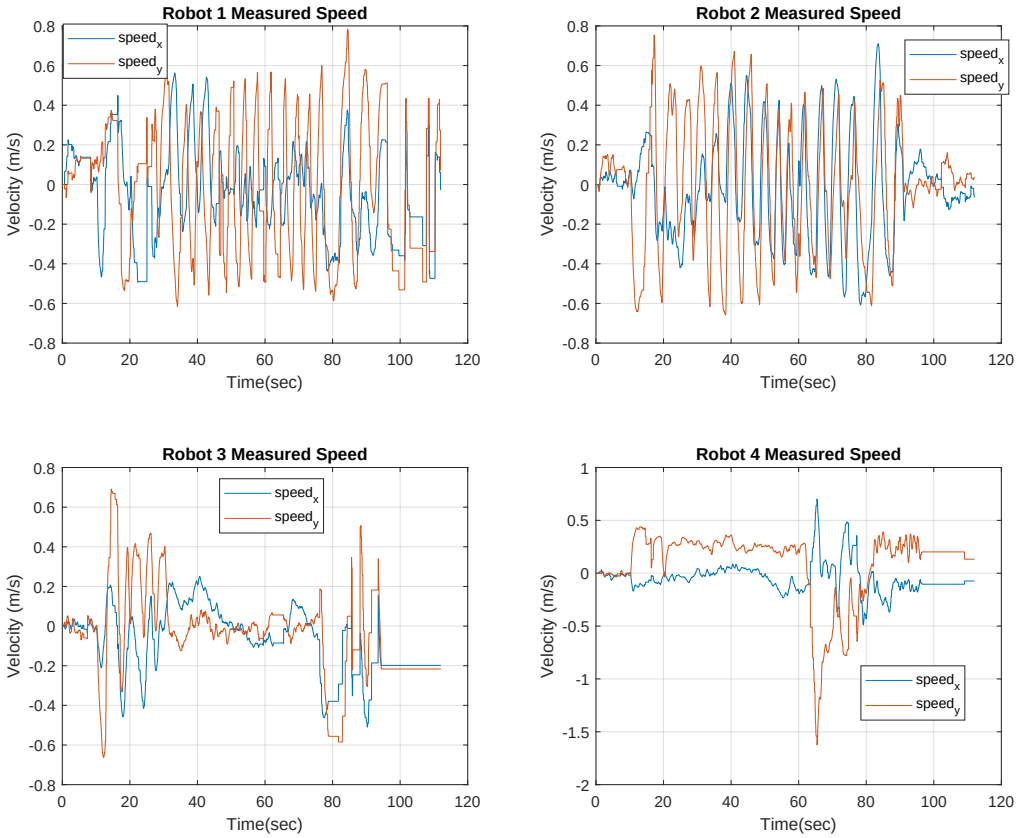

Figure R.16: Experiment 3 - Velocity data for all robots.

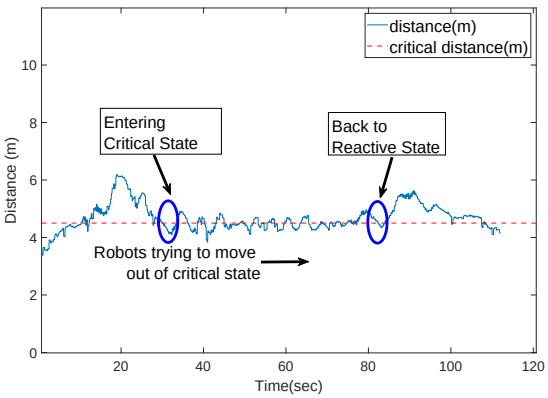

Figure R.17: Experiment 3 - Distance (blue line) between robots 2 and 3 in comparison to critical distance (red dotted line) of 4.5m.

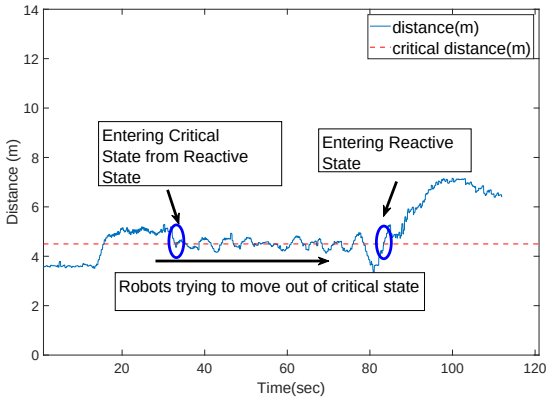

Figure R.18: Experiment 3 - Distance (blue line) between robots 2 and 3 in comparison to critical distance (red dotted line) of 4.5m.

3.4 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.

3.4.1 Results

The initial and final configurations of the team of UAVs are shown in Figures R.19 and R.20.

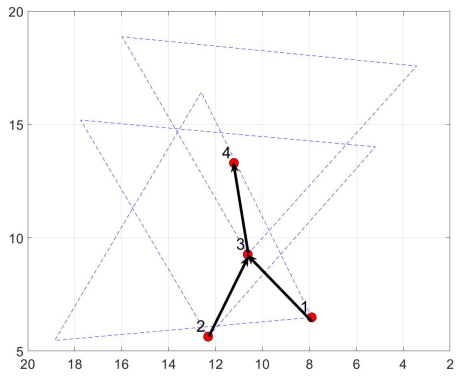

Figure R.19: Experiment 4 - Initial position.

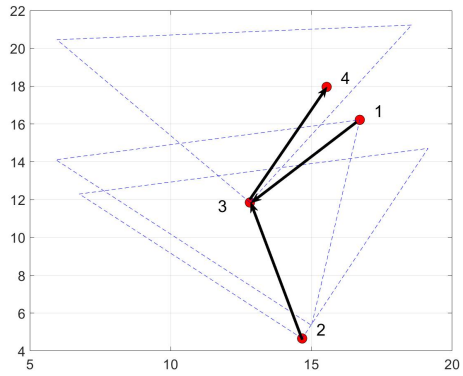

Figure R.20: Experiment 4 - Final position.

The evolution of each UAV's position throughout the complete experiment is given in Figure R.21. The evolution of each UAV's yaw is given in Figure R.22. Note that robot 4 rotates by almost $120°$ The measured velocity for each UAV is given in Figure R.23.

3.4.2 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.

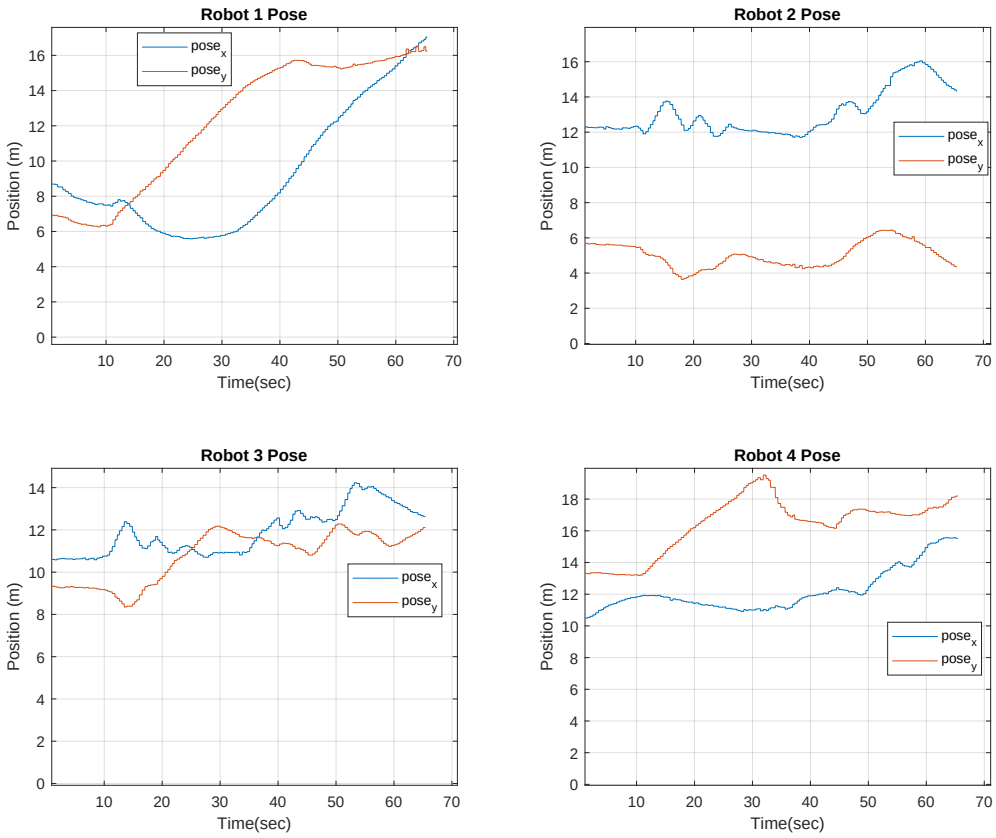

Figure R.21: Experiment 4 - Position data for all robots.

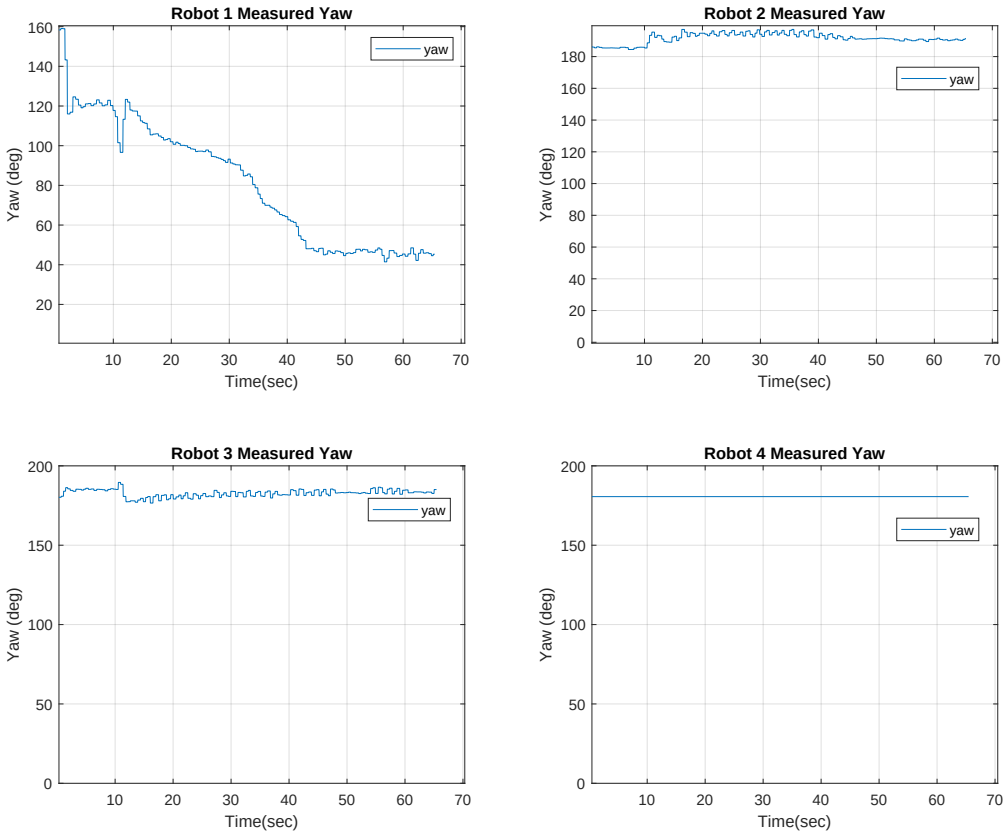

Figure R.22: Experiment 4 - Yaw data for all robots.

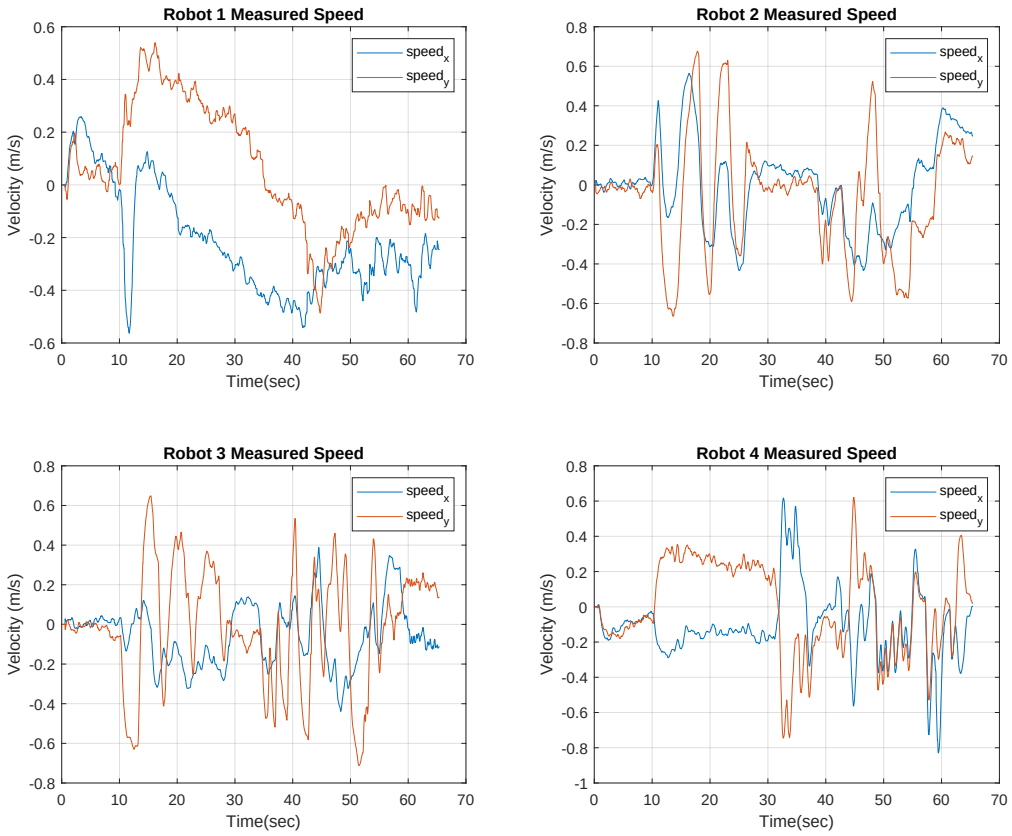

Figure R.23: Experiment 4 - Velocity data for all robots.

3.5 Experiment 5 - Demonstration of Machine State $\mathfrak{P} \rightarrow \mathfrak{R}$

In this experiment, we demonstrate the transition of machine states from $\mathfrak{P} \rightarrow \mathfrak{R}$ to simulate possible scenarios in which the team of robots is deployed in a configuration in which the Reactive state cannot be enabled. In order to do so, we first instruct the leader robot to sweep across a $20m$ field, move forward in y direction for $20sec$, move backward for $20sec$ in y direction, and then sweep back to its original position as also evident in the pose plot of robot 4 given in Figure R.26. The leader robot then flies a trajectory to disconnect himself from robot 3 simulating the deployment of the team of robots. The Reactive state cannot be enabled and the system is forced in the Planning state. The leader robot is then driven to re-enter the FOV of robot 3 allowing the system to transition in the Reactive state, i.e., $\mathfrak{P} \rightarrow \mathfrak{R}$.

3.5.1 Results

The initial and final configurations of the team of UAVs are shown in Figures R.24 and R.25.

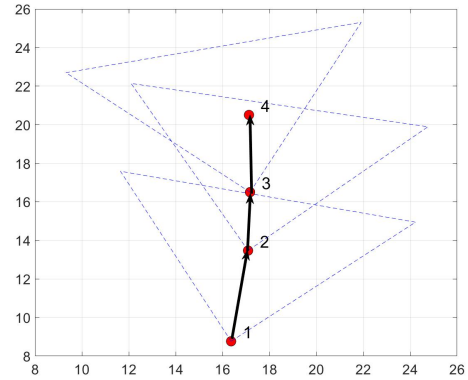

Figure R.24: Experiment 5 - Initial position.

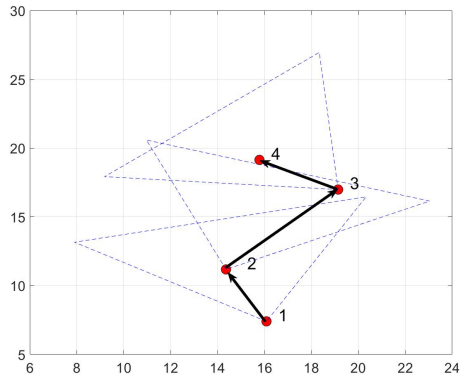

Figure R.25: Experiment 5 - Final position.

The evolution of each UAV's position throughout the complete experiment is given in Figure R.26. 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.26 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.28. 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.27. The measured velocity for each UAV is given in Figure R.28. 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 $t = 90s$.

3.5.2 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, simulating the deployment of a multi-robot system in a configuration such that the machine state cannot be in $\mathfrak{R} = 1$. The finite-state machine hence starts from the Planning state $\mathfrak{P} = 1$;
- 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 $\mathfrak{P} \rightarrow \mathfrak{R}$. 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.

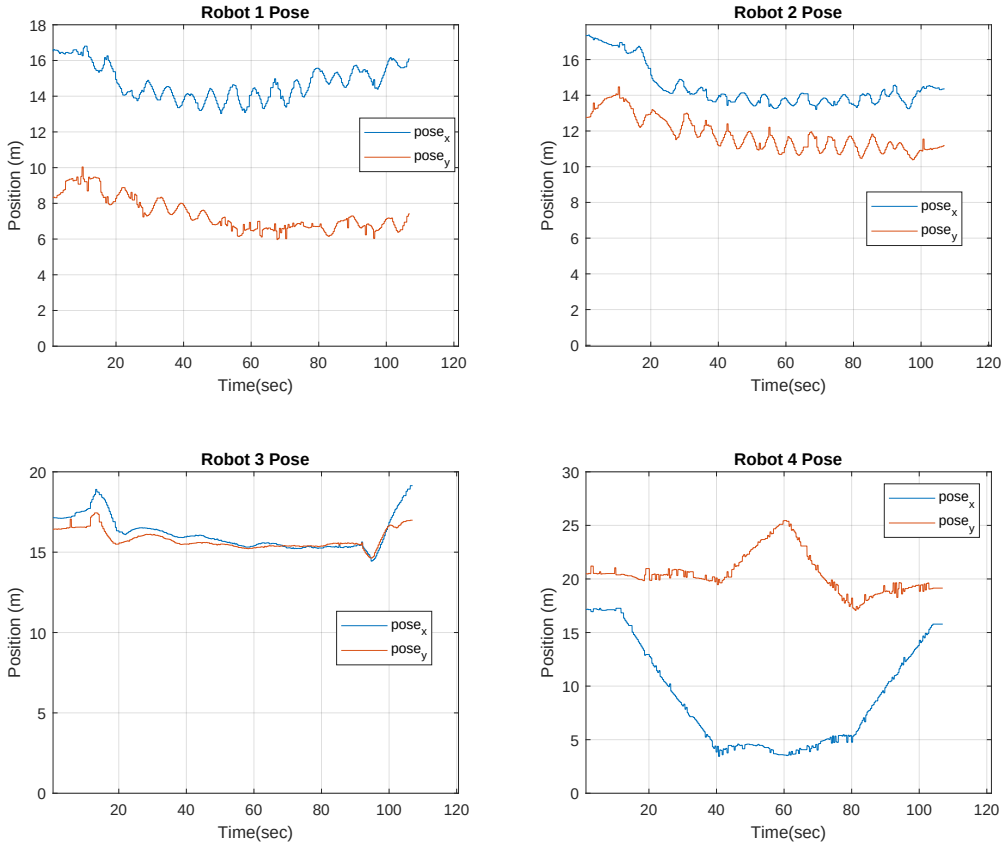

Figure R.26: Experiment 5 - Position data for all robots.

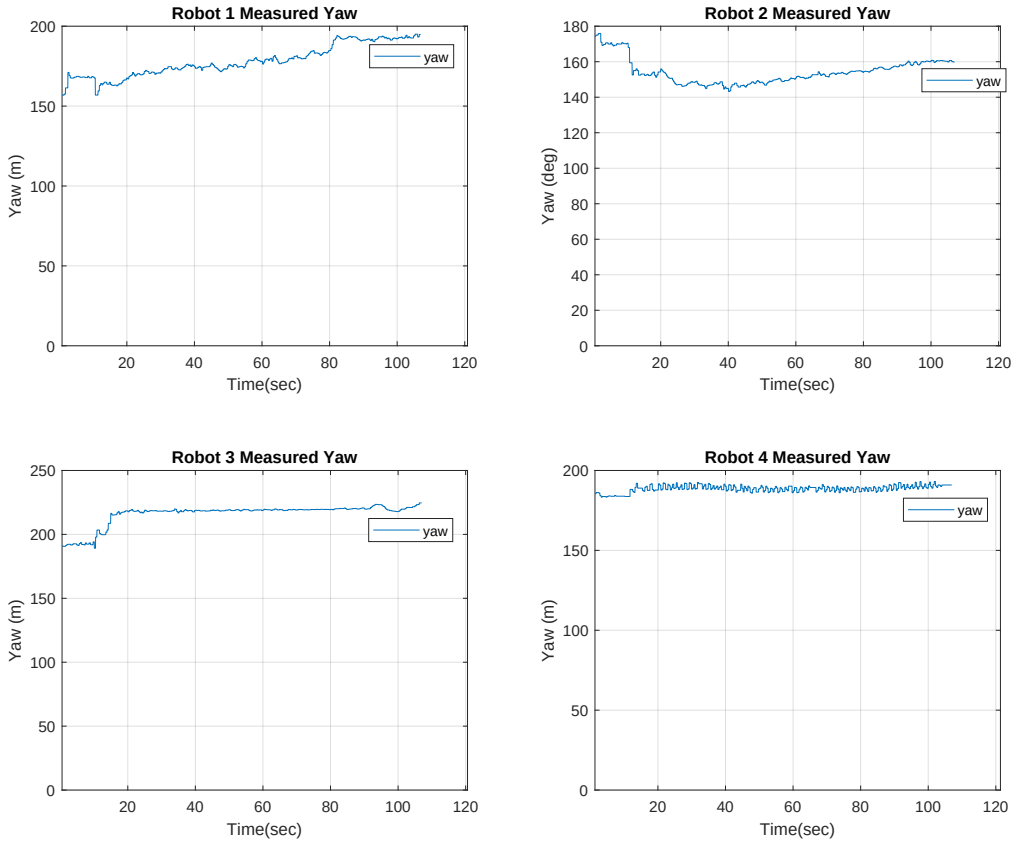

Figure R.27: Experiment 5 - Yaw data for all robots.

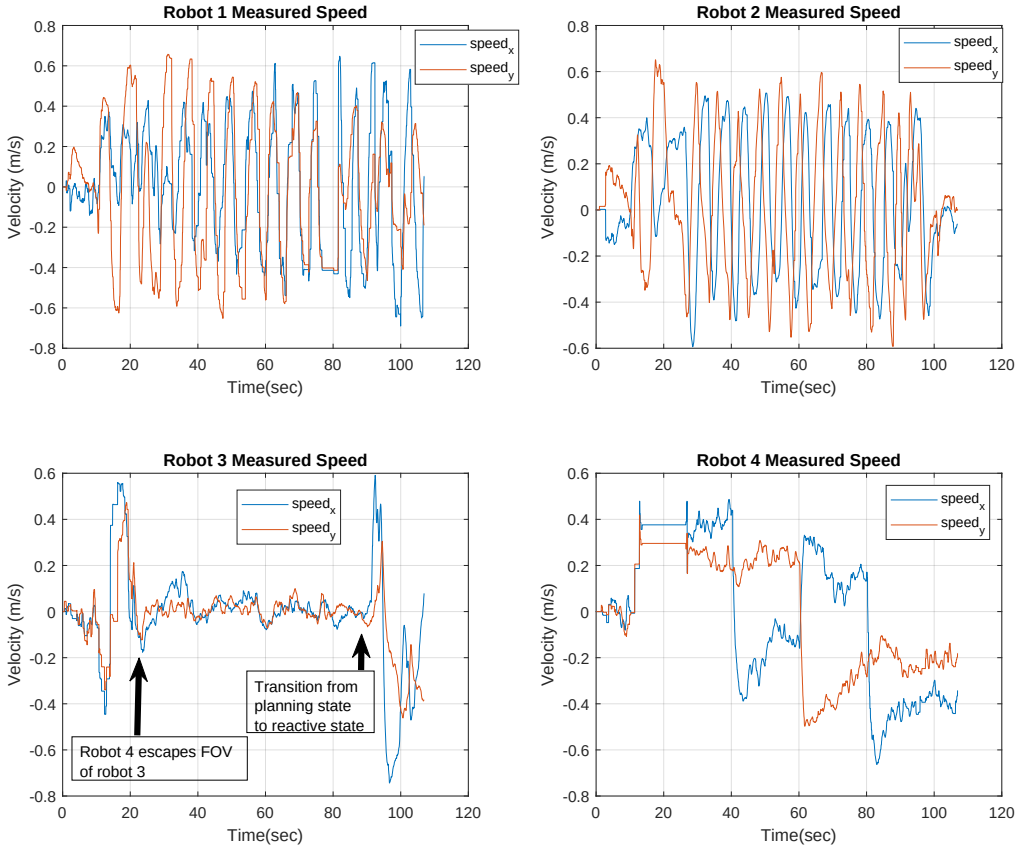

Figure R.28: Experiment 5 - Velocity data for all robots.

4 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;
- High 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 Reviewers some insight into the limitations of the experimental setup 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. Currently, we have preliminary work done in [1] to develop a multi-robot robot to robot relative localization system. Our next objective is to use this 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, 2019. Available at `http://hdl.handle.net/10919/91460`.