Quadrotor Swarm Arena (QuaSAr) Development of a Swarm Control Testbed

Shane T. Stebler
*Embry-Riddle Aeronautical University, steblers@my.erau.edu*

William MacKunis
*Embry-Riddle Aeronautical University, mackuniw@erau.edu*

Follow this and additional works at: [https://commons.erau.edu/beyond](https://commons.erau.edu/beyond)

Part of the Controls and Control Theory Commons, and the Navigation, Guidance, Control and Dynamics Commons

**Recommended Citation**
Available at: [https://commons.erau.edu/beyond/vol2/iss1/6](https://commons.erau.edu/beyond/vol2/iss1/6)
Quadrotor Swarm Arena (QuaSAr) Development of a Swarm Control Testbed

Cover Page Footnote
1 Principal Investigator, Aerospace Engineering Department, steblers@my.erau.edu 2 Faculty Advisor, Physical Science Department, mackuniw@erau.edu This research was supported by the ERAU Undergraduate Research Center, Honors program, and Physical Science Department. The authors would also like to thank Michael Nisip, Logan Turco, Josh Teramae, and Gia Donatella for their contribution to the QuaSAr project as part of their undergraduate coursework.

This article is available in Beyond: Undergraduate Research Journal: https://commons.erau.edu/beyond/vol2/iss1/6
Quadrotor Swarm Arena (QuaSAr)  
Development of a Swarm Control Testbed  

Shane T. Stebler & William MacKunis

Abstract
Swarm control systems are increasingly popular in the robotics industry and academia due to their many potential applications. The goal of the Quadrotor Swarm Arena (QuaSAr) project is to construct a quadrotor swarm control testbed to provide researchers with the tools needed to experimentally investigate this emerging science. This testbed is equipped with a motion capture system, test control station, and numerous quadrotor UAVs. MATLAB-Simulink is utilized for control law development, data processing, and test control. This configuration allows researchers to test developing control law in a ’plug and play’ manner as control development and test control are all completed using the same tools. Thus, the QuaSAr testbed an increasingly valuable tool to a wide set of researchers. Currently, the testbed is undergoing final testing and initial operation. Improved single-agent control methods are continuously being developed and initial swarm control research is underway. The combination of the completed and future work has promising implications for the continued success of the QuaSAr project.

Introduction
Intelligent robotic systems are increasingly popular in personal and commercial applications. These systems range from automatic vacuum cleaners and mowers to commercially operated unmanned aerial vehicles (UAVs) and military platforms. The construction and operation of these systems becomes far more complex as their applications continue to evolve. This increased complexity is often detrimental to mission assurance. Swarm control offers a solution to this mission assurance problem while simultaneously addressing the increase in mission complexity [1].

A swarm describes a set of agents that together form a whole and may also be referred to as a multi-agent system [2]. The primary distinction of swarms from any ordinary group of agents is the concept of useful self-organization [3]. Useful self-organization is considered in this work to describe the emergence of productive inherent global behavior as a consequence of the low-level behavior of a group’s constituents or agents. The control of these systems describes the design of the group’s topology through the use of mathematics and control methods including graph theory, game theory, and various forms agent control. Accomplishing this control implies the use of a closed-loop automatic control system at various levels within the swarm. Closed-loop automatic control refers to the use of sensor feedback (such as the temperature measured by a home thermostat) to make control decisions aimed at achieving a specific set of objectives (i.e. setting the heating power to achieve the desired room temperature).

Current research by the authors is directed toward Advanced Control through Learning in Autonomous Swarm Systems (A-CLASS), which utilizes graph theory to achieve the desired control objective introduced above. This control objective is to maximize the potential for mission success through distributed collaboration of a multi-agent quadrotor system.

Increased fault tolerance is also desired in conjunction with a swarm’s control objective. To achieve this increased fault tolerance, decentralized swarms are considered, which are consequently dominated by local interactions. Thus, the control of such swarms requires that agents be distributed over a topology which permits localized interactions. From an engineering perspective, this topology requirement is decomposed into communication requirements of some form that require agents to conduct two-way communication with nearby agents. Furthermore, these decentralized swarms are scalable and, subsequently, more robust to failure. This property is a direct consequence of the fact that collective behavior unaffected by the number of agents (as long as this number is large enough to still constitute a swarm) [3].

Such swarms are also able to complete much more complex tasks, much like a team of people is able to accomplish more complex tasks than an individual. A group of autonomous quadrotors is a great example of a
Development of a Swarm Control Testbed

Beyond Vol. 2 2017/2018 commons-erau.edu/beyond

Swarm, and is also the focus of this research. A team at the University of Indonesia for example, is investigating the use of quadrotor swarms to autonomously deploy and extend the coverage of available Wi-Fi networks during natural disasters and other emergencies [4]. Projects involving quadrotor swarms, such as that at the University of Indonesia, require a large amount of development and testing, which could ultimately benefit from a reliable swarm control testbed.

The primary contribution of this research project is to make significant progress toward experimental validation of new, multi-agent swarm control strategies using quadrotor UAV agents. This progression was made through the development of a testbed capable of supporting a scalable number of quadrotor UAVs with a sophisticated data collection system. Collected data is critical to the experimental testing and validation of emerging swarm control techniques. Additional contributions include a collection of control techniques and applications for individual quadrotor agents. The quadrotor swarm requires controllable agents, thus these techniques for agent control are critical to the performance of swarm. Similarly, reliable communication techniques are required to allow the use of the topologies described above.

**Literature Review**

In the developing industry, there is great demand for this type of capability in technology. There are a variety of applications for quadrotor control theory in both an individual and swarm control basis. This is because unmanned aerial vehicles offer significant advantages over manned vehicles in certain applications. Among these applications are search and rescue missions, border patrol missions, and other surveillance missions [5]. In addition to civilian and defense applications, unmanned aerial vehicle control also has great use in the realm of scientific and atmospheric research. Quadrotors offer the specific advantage of incredible agility over other unmanned aerial vehicles. These increased agility demands sophisticated control strategies and yields valuable results.

Several academic endeavors have been pursued for quadrotor use in search and rescue applications. One of the more popular examples is Stanford University’s and UC Berkley’s STARMAC (Stanford/Berkeley’s Testbed of Autonomous Rotorcraft for Multi-Agent Control), a swarm control testbed that is actually used for several applications other than search and rescue [6]. In addition to academic studies, quadrotors have already been used in real search and rescue/risk assessment applications. For example, in the 2011 Fukushima tsunami, quadrotors were used to assess buildings for the threat they posed to humans due to nuclear radiation [7].

Aside from industry-related applications, quadrotors have also been used for recreational purposes and improving quality of life. In one example, a paralyzed man used a quadrotor to enhance his experience of the world. With a camera on board the vehicle, he was able to experience the outdoors in a new and fairly accessible manner [8].

In addition, quadrotors are an ideal example of unmanned aerial vehicles because they are able to carry a sizable payload and are relatively compact. Their size allows them to fly in close proximity to people and small places otherwise unreachable by standard aerial vehicles, such helicopters. This kind of capability lends itself to many applications and is thus a valuable way to study control theory. Although this project studies swarm control theory via the use of quadrotors, there are many methods to conduct swarm control research.

**Current Systems and Alternate Architectures**

Swarm control theory may be studied using many different approaches. The use of Kilobots for studying supervisory control theory was implemented by University of Sheffield for the purpose of controlling 600 robots [9]. Prior to 2012, University of Stuttgart hosted an open-source swarm robot project. This was comparatively a more basic project than most control theory research endeavors, but it dealt with control theory concepts and swarm control nonetheless [10]. Harvard University conducted swarm control research in

![Figure 1: Kilobot Swarm. The Kilobot testbed at Harvard allows experimenters to mimic the behavior of swarms found in nature [12].](commons-erau.edu/beyond)
conjunction with artificial intelligence when researchers used Kilobots to mimic swarms found in nature, such as ants, birds, and other natural examples of swarms [11]. Figure 1 shows a collection of Kilobots during a swarm control test.

Another area of research is the study of human-swarm interactions. For example, at the University of Houston researchers are investigating potential applications such as the administration of medicine and surgery, all performed by small robots [13]. Such applications have profound implication for the potential of successful swarm control. The research group at Houston also runs crowdsourced experiments in swarm control using short online games where users interact with various swarms running different control algorithms. Figure 2 shows a view of one such game where the user controls a small swarm rather than the individual agents. This is an excellent use of crowdsourcing as users provide a large amount of data for relatively low cost.

Although there are many different methods of studying control theory, quadrotors were chosen for this project because of their vast capabilities and airborne nature. They are inherently tied to the studies at Embry-Riddle Aeronautical University due to the school’s aeronautical focus. For this reason, other research groups at the university will also benefit from using QuaSAr’s testbed. For example, although this project specifically uses the Crazyflie 2.0 nano quadcopter, the testbed can be easily adapted to accommodate larger quadrotors in the future to assist with UAV research as well as control research involving larger bodies and objects.

There are also many other advantages to studying these aeronautical systems. Quadrotors are more capable than ground vehicles in certain missions (such as search and rescue) because they have fewer obstacles to avoid due to their airborne nature and can view greater areas during a given mission. They also allow for a rich study of controls, due to the number of degrees of freedom in the system. This makes the issue of controlling quadrotors more complicated and potentially more rewarding from a research perspective. A positive aspect of the complex nature of quadrotor control is that it directly translates into the quadrotor having more agility and speed than other unmanned aerial vehicles. Packed with capability, a small quadrotor can easily navigate into places that other larger vehicles or humans could not.

The realization of this project is not only complicated from a controls perspective, but also from an integration perspective. Just like many other engineering projects, integration of various systems and subsystems needed to be accomplished successfully in order for this project to work and be of value. This means that although this is primarily a project used to study control theory, there is a need for knowledge of programming, communication protocols, and motion capture systems. All of these elements tie in directly to the solving the control problem at hand, and thus, are equally as crucial as the control theory itself for the current status and ongoing use of the project and testbed.

Methods

System Overview

The QuaSAr system consists of four major subsystems: Quadrotors, Motion Capture System, Communication, and Ground Station. The quadrotor subsystem consists of four stock Crazyflie 2.0 nano quadcopters modified with reflective markers. The Crazyflie 2.0 was selected...
because of its long battery life, compared to other quadrotors of its size, and its communication capacity, which exceeded the requirements of the system. The quadrotors communicate through a Crazyradio included with each Crazyflie package. Data from the suite of on-board sensors is exchanged throughout flight. On-board data from the quadrotors’ sensors, however, this on-board data is not enough for adequate position control. Reflective markers in various patterns must be adhered to the quadrotors in order for the motion capture system to measure their position and orientation.

The motion capture system used in this testbed is comprised of four OptiTrack Flex 13 cameras. These cameras met the system’s requirements for arena size and the rigid body tracking capacity, while staying within the budget. The cameras come in a set along with camera stands, calibration tools, cabling, and OptiTrack’s Motive software. The software is used for calibration, tracking, and data streaming purposes.

The communication subsystem is split into two main paths: the communication between the motion capture system and MATLAB and the communication between the quadrotors and MATLAB. Communication between the motion capture system and MATLAB streams real-time position data for test tracking. The key step between the motion capture system’s Motive software and MATLAB is the Optitrack NatNet SDK software. NatNet is a client/server networking SDK for streaming the motion capture data across networks that Optitrack offers. Commands, external tracking data, and telemetry are streamed between the quadrotors and MATLAB. This path utilizes a Python client as the step between the stock Crazyradio and MATLAB. The two paths of the communication system are tied together...
through the ground station.

The ground station is the heart of this testbed. Every component comes together at this point in the system. The control feedback loop takes place in the ground station. The ground station receives data from Motive and from the Crazyradios. It processes all this data and sends specific information back to the quadrotors. There is a user interface portion of the ground station as well, allowing the user to initialize and manage testing from the ground station. This interface also stores the data collected during each flight to use in later analysis.

Results

Current Progress and Complete Work

The Quadrotor Swarm Arena (QuaSAr) testbed was specifically developed with the intention of conducting research in control theory via the flight of quadrotors. Quadrotors are excellent candidates for control theory research because of their inherent instability. The ultimate goal of this project is to form a completely autonomous quadrotor swarm, complete with individual sensing capabilities as well as full system communication. This is an ideal testbed for student-run university research in control theory because of its easy accessibility.

Note that each quadrotor has its own sensors on board, giving the user to choose any combination of one to several quadrotors to be controlled at any given time. Additionally, the selected Crazyflie 2.0 quadrotors (see Figure 5) are small and relatively safe as compared with larger quadrotors, further increasing the accessibility of the testbed to future students. Information from the flight of the swarm and the health of the system and its individual quadrotors is relayed back to the user via a ground station. This layout allows for full system integration and understanding.

Significant progress toward the final objective was made during the course of this initial research. The major high-level milestones associated with the QuaSAr project’s progression to date include concept definition, design, subsystem assembly, and system integration. Future research and development discussed later in this paper, include system testing and swarm control development.

System testing reveals numerous areas to improve and iteration on minute aspects of the system are actively pursued. In addition, swarm control development is underway, as well as improved agent control. A recent paper, accepted for publication by the peer-reviewed International Conference on Control, Automation, Robotics and Vision, by the authors [15] is aimed toward improved agent control through the implementation of nonlinear control law to track agile maneuvers in the presence of numerous uncertainties present in the testbed. The paper, “Nonlinear Output Feedback Tracking Control of a Quadrotor UAV in the Presence of Uncertainty,” specifically addresses the sensor suite used by the QuaSAr testbed.

The agent control law must rely solely depend on position and attitude measurements because the testbed is inevitably limited to these measurements. The results in [15] show asymptotic altitude and attitude trajectory tracking using output feedback when subject to considerable uncertainty in the system model as well as potential disturbances. This result is very useful for the QuaSAr testbed and is scheduled to be implemented for improved quadrotor performance.

Additionally, work on A-CLASS is underway and scheduled to be presented by the authors at the World Congress on Undergraduate Research located in Doha, Qatar in November 2016. This segment of research is a considerable contribution to item 6 on the milestones list as it is intended to comprise the inaugural experimental testing for swarm control development with the QuaSAr testbed. The combination of the work described above has promising implications for the continued success of the QuaSAr project.

The major result of this project is the QuaSAr testbed. Closed loop control was accomplished with each major component in the loop, performing as described above. This was completed with a single Crazyflie quadrotor. Significant drift of the measured yaw angle was
eliminated using external feedback from the Optitrack motion capture system. Figure 6 shows the output of the ground station after a successful test.

In addition to the primary output of this research project, a closed loop controller was developed, tested, and implemented on the Crazyflie. Development and initial testing was completed using Simulink. The results of this development are also discussed here in detail. The control technique used during this initial phase was found to be very successful for attitude control and future plans for improved techniques are discussed later.

**Simulation Methodology and Results**

A numerical simulation was conducted to test the performance of the selected controller. The high-level control objective was to successfully track position and attitude trajectories, while rejecting disturbances. Note that current and future work includes refined control techniques. This methodology is adequate, however, for initial development and testing of the QuaSAr testbed. This simulation is a stepping stone for experimentation as testing of the selected control design must be compared using both simulation results and experimental results. Furthermore, conducting this simulation ensured the feasibility of the selected design, saving valuable time and resources during initial testing of the control law on the Crazyflie 2.0 quadrotors.

The six degree of freedom (6DOF) simulation is carried out using Simulink. The systems equations of motion are derived from first principles using MATLAB’s symbolic toolbox in the simulation’s initialization script. Simulation parameters are also set in this initialization script, including the simulation start time, stop time, and time step. The quadrotor’s mass, thrust, and torque properties are also included in this initialization file. The Runge-Kutta (ode4) solver is used to solve the 6DOF equations of motion and a time step of 25ms was selected. This time step was found to capture an adequate resolution for the desired results. Additionally, further reduction of the time step yielded the same results and showed that the simulation time step was small enough to capture accurate behavior.

![Figure 6: Test Results. The QuaSAr testbed successfully collected flight data from the Crazyflie as well as the OptiTrak motion capture system during flight.](image1)

![Figure 7: Quadrotor Swarm Arena. The testbed is operational, enclosed by a net for safety. Each of the four cameras is also enclosed in the netting along with a padded cover on the floor of motion are derived from first principles using MATLAB’s symbolic toolbox in the simulation’s initialization script. Simulation parameters are also set in this initialization script, including the simulation start time, stop time, and time step. The quadrotor’s mass, thrust, and torque properties are also included in this initialization file. The Runge-Kutta (ode4) solver is used to solve the 6DOF equations of motion and a time step of 25ms was selected. This time step was found to capture an adequate resolution for the desired results. Additionally, further reduction of the time step yielded the same results and showed that the simulation time step was small enough to capture accurate behavior.](image2)

![Figure 9: Simulation Visualization. The visualization object allows simulation results to be viewed in near-real-time in a virtual lab.](image3)
Figure 9 shows the simulation visualization tool, which updates actively during the simulation. With this tool, the simulation results may be observed while the simulation is running. The body frame, quadrotor plane, and trajectory may be toggled as desired to gain a qualitative perspective of the quadrotor’s performance. Results are exported to MATLAB for additional analysis. This tool was developed in-house, specifically for the QuaSAr project.

The simulation framework is made up of five major parts. These include the system plant, position controller (outer loop), attitude controller (inner loop), sensors, and the command/reference. The quadrotor graphics object as well as a timing tool are also included to allow near-real-time viewing of the simulation. Figure 8 shows the Simulink setup. The system plant was converted to a MATLAB function in the interest of creating an easily modifiable simulation. The initialization file also creates the function (code) used to determine the state derivative from the current state and inputs. This approach also increases simulation speed as the MATLAB function is lean and easily converts to C code via the accelerator.

Initial testing assumes that the system has full state feedback, including angular rates and translational velocity. While this is not the case for most systems, including the quadrotor, such a baseline is useful for comparisons of actual controller performance to ideal performance. The Crazyflie 2.0 has an onboard IMU that measures acceleration along the three body frame axes as well as the vehicle’s attitude. Additionally, a high accuracy barometer is located on-board for measuring altitude. Future work will include the addition of an external motion capture system for additional state feedback.

As discussed previously, the controller is separated into two parts: the outer loop and the inner loop. The outer loop simply passes the heading and altitude commands through without altering the signals. While altitude is clearly not a description of the attitude, it is part of the inner loop (nicknamed the attitude controller).
and is directly controlled by the input $u_1$. The $x$ and $y$ commands, however, are used in the outer loop to create roll and pitch commands. This outer loop has additional translation rate feedback to restrict the maximum speed and maintain stability.

Figure 9 and Figure 10 illustrate the outer and inner loop controllers respectively. Note that the block diagrams for the roll and pitch to position error include a restriction on translation speed. These restrictions are enforced using a saturation block, which is easily translated into hardware compatible code.

A number of tests were completed to assess the effectiveness of the proposed controller. These selections span the expected conditions/maneuvers typically encountered during swarm operations. Test results compare the state commands to the measured positions and attitudes.

Tests include:
- Trajectory Tracking
- Step Commands
- Constant Speed Translation
- Constant Yaw Rate

To examine the system's ability to track a trajectory, a helix command signal was generated. Its diameter is 1.0 m and it climbs approximately 1.5 m. Results are shown in Figure 11. Note that the quadrotor begins at the initial point in the trajectory of (0.0, 0.5, 0.0) m.

This trajectory tracking was found to be successful. A small error of less than 10 cm in the form of phase lag is observed and steady state error is found to diminish to zero. Similarly, the step input resulted in successful behavior. Settling times were found to be approximately

![Figure 11: Inner Loop - Attitude Control. The attitude controller produces system inputs $u_1$, $u_2$, $u_3$, and $u_4$, which are converted to rotor speeds](image)

![Figure 12: Helix Trajectory Results. The quadrotor tracked a constantly changing path, demanding coordination of all six axis controls](image)
Figure 13: Step Input Results. The quadrotor was driven to the desired states within reasonable amounts of time for $x$, $y$, $z$, and $\phi$ step commands.

Figure 14: Constant Speed Translation Results. Here, $\theta$ is used to drive the system to the maximum translation speed of 1.5 m/s.

Figure 15: Constant Yaw Rate Results. Here, $\frac{dp}{dt}$ is set to the maximum allowable rate of 20 degrees/second with an acceptable settling time of approximately 3.5 seconds.

5 seconds for 1.0 m translations, 2 seconds for 1.0 m altitude steps, and 2 seconds for 45° yaw slew maneuvers. Each of these maneuvers were initiated manually in the Simulink simulation. Data are shown for the step responses in Figure 12.

Note that during each of the maneuvers crosstalk occurs between the states. For example, yaw maneuvers cause small spikes in altitude. This is an anticipated artifact of the coupled quadrotor system. Future work will include the construction of a decoupler to mitigate these effects. As shown in Figure 12, however, these effects do not significantly impact simple trajectory tracking, which is expected to be the primary operating mode during swarm operations. For this reason, it is left to future work.
Next, constant translation speed is assessed. In this maneuver, the maximum speed is commanded by saturating the x position controller with a high command. Figure 13 shows the translation rate capture after approximately 3 seconds with a settling time of roughly 10 seconds.

The constant yaw rate maneuver is achieved using the same methodology. The attitude controller is saturated with a large yaw command and the yaw rate is driven to the maximum allowable rate of 20°/second. Rate capture occurs after approximately 0.5 seconds with a final settling time of roughly 3.5 seconds. Figure 15 shows this response.

Lastly, the system's ability to reject disturbances is assessed. This is done by applying external forces and torques to the system. The forces are applied in the form of acceleration pulses or angular acceleration pulses applied directly to the plant output. Note that translational and angular accelerations are simply forces scaled by mass and moment of inertia respectively. Forces were applied to the system over a 0.25 second pulse such that the imparted acceleration was initially 9m/s² and the angular acceleration was 60°/s². Figure 15 shows the response to a disturbance in the positive x direction. Note that the controller returns the quadrotor to its starting position.
Next, a sustained force acts on the body in the positive x direction starting at t=3s. This could resemble an attachment or wind. The controller behaved as expected and rejected the force by pitching as shown in Figure 16. The figure shows that the system compensates for the sustained disturbance as desired.

A similar force was then applied downward on the quadrotor. This could simulate the weight of a payload or the downdraft caused by a nearby quadrotor. This response is very similar to the response seen when the quadrotor simulation starts. Note that the initial rotor thrust is zero and the system must account for the weight of the system as it starts up. The first spike in Figure 17 is the result of initial startup and the second spike (at t=5s) is the introduction of the new force.

Disturbances to the vehicle’s attitude may come from a multitude of sources. These can include the same disturbances that acted on the center of mass discussed previously, such as wind and payload, only with the force acting elsewhere on the structure (not at the center of mass). Figure 18 shows the response to a torque about the $B_x$ axis generating an initial angular acceleration of approximately 20 degrees/second. Disturbances in pitch will exhibit the same behavior as the vehicle is considered to be symmetrical.
Disturbances to yaw were also considered. The yaw control was found to be faster than roll and pitch. An equivalent torque about the $B_z$ axis was almost unnoticeable as the system was able to respond quickly. Additionally, yaw control is only coupled with the altitude control. Figure 19 shows the vehicle’s response to the yaw disturbance.

To summarize, the results of the 6DOF simulation show that a viable controller has been constructed for the MQV. The selected controller is able to maintain stability during trajectory tracking, step commands, constant speed translation, and constant yaw rate maneuvers. Additionally, significant disturbances were introduced to the system and were shown to be manageable. The tuned controller gains will need slight modification when implemented on the actual MQV, but a reasonable starting point has been found.

Discussion

The developed testbed is largely dependent on its various interfaces. The combined effects of the many methods of communication are discussed here. The primary objectives for communication are to transmit commands from the user to the quadrotors and establish feedback from the motion capture system for quadrotor control. These primary objectives yield two functional pathways, which are decomposed into four primary interfaces. These include cooperation between crazyflie-crazyradio, crazyradio-MATLAB, motion capture system - motive, motive-MATLAB, and user-MATLAB. Note that the Simulink package and its cooperation with MATLAB is not considered and interface.

The first interface encountered during operation is the Ground Station graphical user interface. This graphics object utilizes ZMQ server/client functionality to send gathered information to available quadrotors using a connected Crazyradio dongle. The Crazyradio then processed the information and transmits it via RF to available quadrotors. Similarly, information originating from the motion capture system is processed on the Ground Station to automatically generate commands/message for the Crazyflies. These messages are sent though this same path.

As mentioned above, the image data produced by the camera system is processed on the Optihub. Information about traceable reflective markers is then transmitted to the Ground Station where the Motive application processes the traceable data. The results of the Motive tracking algorithms are then sent via NatNet to the MATLAB client for use in control and data processing algorithms.

The performance of these interfaces can be assessed by observing the refresh rate of the system. This describes the time it takes to complete one ‘receive-compute-send’ cycle. In this cycle, data from the cameras is passed to the ground station control algorithm and translated into commands which sent to the quadrotors. Using the developed quadrotor six degree of freedom simulation, various refresh rates were tested. These tests included the introduction of simulated delays in the updated controller output and data feedback. During these delays, the quadrotor system was allowed to evolve dynamically to accurately represent the behavior of

Figure 20: Response to a Disturbance in $\psi$. The torque is applied for 0.25 seconds starting at t=5s about the $B_z$ axis.
Figure 21: 10 Hz Refresh Rate - Step Maneuver Simulation Results. The quadrotor system remains stable at 10 Hz, but performance reaches an unacceptable level with an accuracy threshold of approximately 15 cm.

Figure 22: 10 Hz Refresh Rate - Trajectory Tracking Simulation Results. Trajectory tracking results draw the same conclusions as the step results.

Figure 23: 30 Hz Refresh Rate - Step Maneuver Simulation Results. The quadrotor system remains stable at 30 Hz and retains errors below approximately 2 cm.
a discrete controller. Both step inputs and trajectory inputs were tested using refresh rates as low as 10 Hz. Simulation results are provided in Figures 20 and 21 for the 10 Hz cases.

The constructed system was found to run at approximately 30 Hz, which is well above the 10 Hz described in Figures 20 and 21. The same maneuvers were simulated using this refresh rate (30 Hz) and are shown in Figures 22 and 23. These figures illustrate the conclusion that 30 Hz is adequate for successful control. Low error is maintained and the system is stable.

While the 30 Hz refresh rate was found to be adequate for control, there is significant room for improvement. The communication methodology and speed are the primary limitation of the testbed. More agile maneuvers, for example, require an increase refresh rate. This is the primary topic for future work.

**Future Work**

A suggestion for future improvement is a general renovation of the communication protocol. This includes the reassignment of MATLAB from the role communication data manager to that of a parallel observer. The computational overhead that is introduced by MATLAB and all the necessary protocol for integrating it is detrimental to communication speed. This alteration, however, would not require a complete rebuild of current progress.

The ground station, motion capture system, Crazyflie quadrotors, and associated software would still be useful to the system, motion capture system feedback would only be pulled by MATLAB for observation, rather than passed. This would require the creation of a lean application to collect data from Motive and pass it to the Crazyflie.

In addition to this topography change, multiple quadrotors must also be integrated. This is also the topic of current and future work. Improvement of communication speed will contribute to the expansion of quadrotor capacity as the capability may easily be added during the development of a new application for processing feedback data. This work is scheduled for the next year and is currently under development.

Improved quadrotor control law is the subject of additional future work. Nonlinear methods will be explored to achieve more agile maneuvers. System linearization is not required for these methods, which allows more complex, nonlinear maneuvers to be tracked. Implementation of these methods will greatly improve the performance of a swarm control testbed as this added performance will increase the ability to accommodate more potential of swarm control laws.

**References**


commons.erau.edu/beyond
Development of a Swarm Control Testbed


