

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/378493719>

# Design, Fabrication, and Flight of the Cost- and Risk-Reducing Quadcopter System for GNC Testing

Conference Paper · February 2024

---

CITATIONS

0

READS

162

5 authors, including:



William J. Elke III

National Aeronautics and Space Administration

10 PUBLICATIONS 23 CITATIONS

[SEE PROFILE](#)



Demoz Gebre-Egziabher

University of Minnesota

114 PUBLICATIONS 2,785 CITATIONS

[SEE PROFILE](#)



Ryan Caverly

University of Minnesota

97 PUBLICATIONS 723 CITATIONS

[SEE PROFILE](#)

# **DESIGN, FABRICATION, AND FLIGHT OF THE COST- AND RISK-REDUCING QUADCOPTER SYSTEM FOR GNC TESTING**

**William J. Elke III<sup>\*</sup>, James Johnson<sup>†</sup>, Will Roslansky<sup>‡</sup>, Demoz Gebre-Egziabher<sup>§</sup>, and Ryan J. Caverly<sup>¶</sup>**

This paper presents the steps taken to design, fabricate, and fly the cost- and risk-reducing quadcopter system (CRQS) developed to test and mature novel guidance, navigation, and control (GNC) algorithms. The CRQS testbed features an inverted pendulum and a hanging pendulum mounted on a quadrotor platform to mimic the dynamics of launch vehicles and landing systems, providing a suitable test environment for novel GNC algorithms developed for these applications. The development of CRQS is outlined in this paper, including the design and fabrication of custom hardware, as well as the process of designing flight controllers for the stabilization of both the quadcopter and the inverted pendulum. Results from successful initial flight tests of CRQS with a rigid inverted pendulum are included.

## **INTRODUCTION**

Physical testbeds have long played an important role in the maturation of guidance, navigation, and control (GNC) algorithms for launch vehicles and landing systems. Within the context of launch vehicles and landing systems, the majority of testbeds have focus on advancing GNC algorithms through technology readiness levels (TRLs) 7-9. Examples include NASA's F/A-18 Full-Scale Advanced Systems Testbed that was used to test launch vehicle controllers [1], as well as the Mighty Eagle [2] and other vertical take-off and landing systems developed by Masten Space Systems [3–5]. Although these testbeds play a vital role in the development of GNC algorithms, they are relatively costly and testing with them carries non-negligible cost and safety-of-life risk.

Lower cost and lower risk testbeds are particularly suitable for maturing GNC algorithms through TRLs 4-6, the so-called technology “valley of death.” Although historically there have been limited testbeds developed for this purpose, there has been a recent surge in this area of research [6–13]. In

---

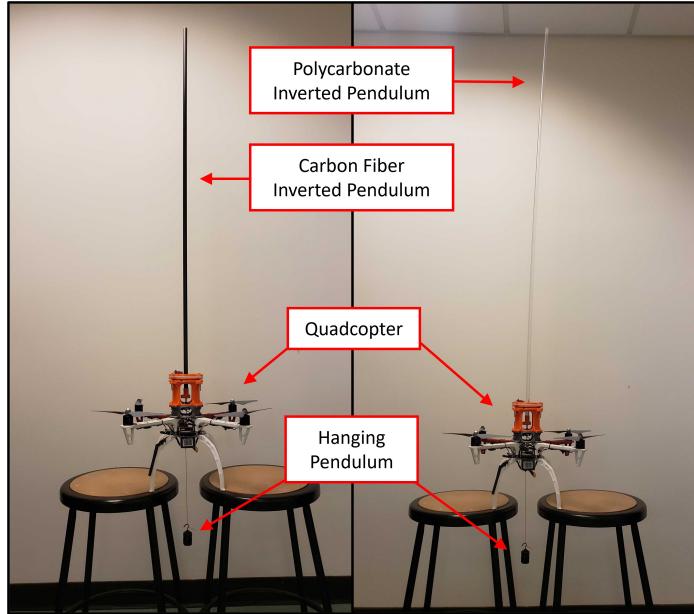
<sup>\*</sup>Graduate Student, Department of Aerospace Engineering & Mechanics, University of Minnesota, 110 Union St SE, Minneapolis, MN 55455.

<sup>†</sup>Undergraduate Student, Department of Aerospace Engineering & Mechanics, University of Minnesota, Minneapolis, MN 55455, USA.

<sup>‡</sup>Undergraduate Student, Department of Aerospace Engineering & Mechanics, University of Minnesota, Minneapolis, MN 55455, USA.

<sup>§</sup>Professor, Department of Aerospace Engineering & Mechanics, University of Minnesota, Minneapolis, MN 55455, USA.

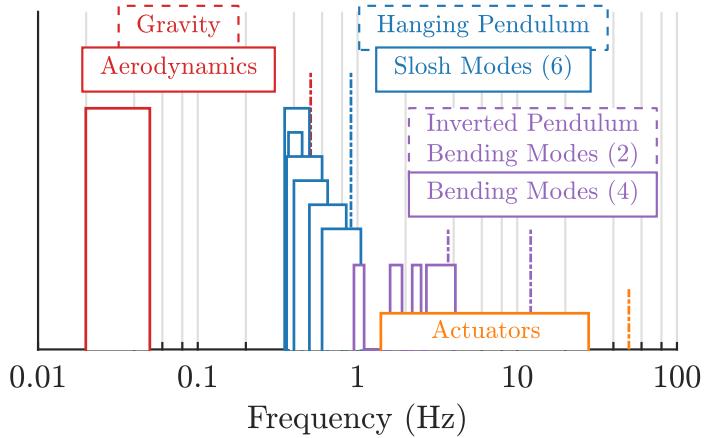
<sup>¶</sup>Assistant Professor, Department of Aerospace Engineering & Mechanics, University of Minnesota, Minneapolis, MN 55455, USA.



**Figure 1:** Prototype of the Cost- and Risk-Reducing Quadcopter System (CRQS), a quadcopter-based test platform with a hanging pendulum and inverted pendulum to capture the relevant dynamics of launch vehicles and planetary landing systems. On the left is a version of the CRQS with a rigid inverted pendulum made of carbon fiber, and on the right is a version with a flexible inverted pendulum made of polycarbonate.

particular, the low-cost testbeds built in [6, 7] recreate the dynamics of a launch vehicle’s first stage, while the testbed in [8] uses model rocketry engines and actuated fins to mimic the dynamics of a descent vehicle. Inspired by concepts developed in [9–11], a quadcopter-based testbed—known as the Cost- and Risk-Reducing Quadcopter System (CRQS, pronounced “circus”)—was proposed in [12] and tested in a numerical simulation environment in [13]. This testbed, shown in Figure 1, features a quadcopter mounted with a flexible inverted pendulum and a hanging pendulum to recreate the dominant features of a launch vehicle or landing system. Specifically, the rigid-body dynamics of the inverted pendulum mimic the unstable aerodynamics of a launch vehicle or the non-minimum phase nature of a landing system. The flexibility of the inverted pendulum models the structural dynamics of the launch vehicle or landing system. The hanging pendulum results in a low-frequency disturbance to the system that relates to the effect of fuel slosh on a launch vehicle or landing system. A unique aspect of CRQS in comparison to the low-cost and low-risk testbeds in [6–8] is that it contains features that account for structural dynamics and fuel slosh effects, both of which can affect the performance of a launch vehicle or landing system.

The mass, length and structural properties of the inverted and hanging pendulums are turning parameters that can be adjusted to change the poles of the CRQS equations of motion of CRQS (i.e., the solutions to its characteristic equation). Thus, in principle, by matching the poles of CRQS with that of some launch vehicle of interest, we have the ability to mimic the dynamics of the

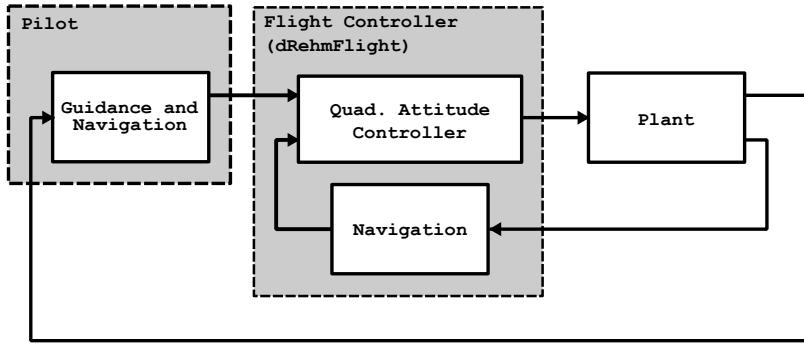


**Figure 2:** A frequency spectrum plot comparing the frequency content of the CRQS to that of the Saturn V. The bars with solid lines represent the frequency spectrum of the Saturn V during first-stage ascent, and the dot-dashed lines represent the frequency content of the CRQS given rough properties of the prototype. Recreated using Refs. [14, 15].

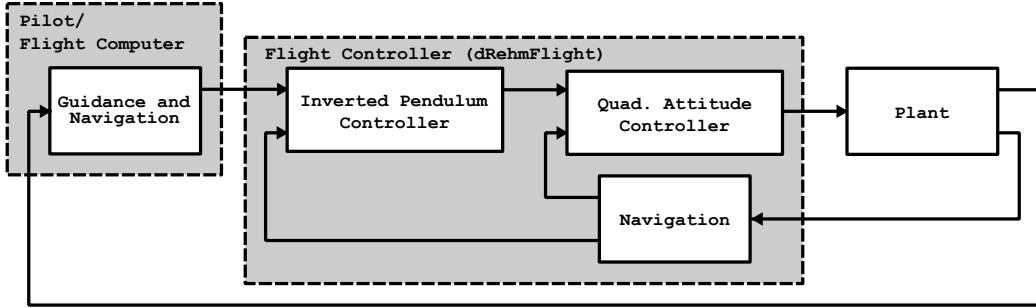
actual vehicle in a laboratory setting. This is not always possible, however, as matching certain poles may require an inverted or hanging pendulums that cannot be realistically accommodated within the size and weight constraints of a quadcopter [12]. Even if we cannot always match poles exactly, we may be able to get close enough to where the insights gained by designing guidance and control algorithms for CRQS will provide useful insight on how the algorithms will perform on a real flight vehicles. For example, Figure 2 plots the frequency spectrum of the dominant features in both the Saturn V launch vehicle (aerodynamics, fuel slosh modes, structural bending modes, and actuators) and CRQS with realistic numerical parameters (gravity, hanging pendulum, inverted pendulum bending modes, and actuators). Although the exact frequencies of these features do not line up, their relative ordering is preserved.

This paper describes work done in the Uninhabited Aerial Vehicle (UAV) Laboratory and the Aerospace, Robotics, Dynamics, and Control (ARDC) Laboratory at the University of Minnesota to design, fabricate, and fly a CRQS prototy whole. Before the CRQS can be used to mature novel GNC algorithms, it must simply be able to perform stable flight. The milestones to achieve this are 1) stable flight of the quadcopter, 2) stable flight of the quadcopter and rigid inverted pendulum (RIP), 3) stable flight of the quadcopter and flexible inverted pendulum (FIP), and 4) utilization of the CRQS for maturing novel GNC algorithms. Milestones 3 and 4 are outside the scope of this paper, and are left to future work. The work presented in this paper, however, was completed with these future goals in mind so all progress in this paper was progress towards these milestones.

The approach to accomplish Milestones 1 and 2 is to first design an inner-loop quadcopter attitude controller to perform stable flight of the quadcopter, then design an outer-loop inverted pendulum controller to stabilize the flying inverted pendulum. The inverted pendulum controller sends commands to the quadcopter attitude controller to successfully balance the inverted pendulum. This



(a) Control loop used to complete Milestone 1 and tune the quadcopter attitude gains for Milestone 2.



(b) Control loop used to complete Milestone 2.

**Figure 3:** Control loops used to complete Milestones 1 and 2. The focus of the work presented here is on the flight controller.

control architecture is shown in Figure 3. Figure 3a shows the architecture used to tune the quadcopter attitude controller, and Figure 3b shows the architecture used to stabilize the flying inverted pendulum. This approach leverages a flight controller, which operates at a very high frequency, to control the quadcopter and balance the inverted pendulum. The flight controller is agnostic to how it receives commands, therefore, a pilot or a flight computer operating at a slower frequency than the flight controller can be used interchangeably to control the position and velocity of the system. This incremental progression is what allows this architecture to be updated to complete Milestones 3 and 4.

This paper outlines the synthesis of the quadcopter attitude and inverted pendulum controllers. This includes discussions on the choice of quadcopter hardware, design of safety and supplemental features in the system, and the construction of a test stand to perform tests in a controlled environment.

The contributions of this paper are summarized as 1) development of a stable flying inverted pendulum for research and pedagogical purposes that can be updated to become the full CRQS, 2) description of the process to realize the functional system, and 3) the presentation of tools to support the design, fabrication, and flight of the physical CRQS. The remainder of this paper proceeds

with a description of the quadcopter hardware and the design of an inner-loop attitude controller, followed by the design of the inverted pendulum and its stabilizing controller, and closing remarks.

## QUADCOPTER HARDWARE AND ATTITUDE CONTROLLER SYNTHESIS

To complete Milestone 2, a quadcopter must first be constructed and flown for Milestone 1. Once the quadcopter is able to perform stable flight, the quadcopter attitude controller is sped up to balance the inverted pendulum, which is discussed in the following section. This section presents the fabrication of the quadcopter, design of the software, and the synthesis of the attitude controller.

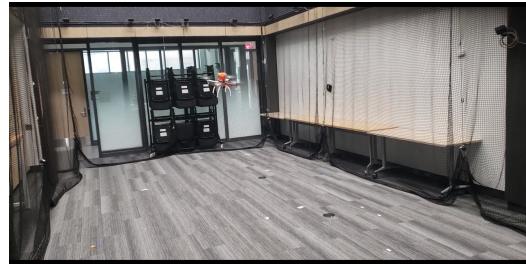
### Quadcopter hardware

There are many options for the flight controller and the software that runs on it—henceforth known as the flight software. The flight software used in this paper is known as dRehmFlight (pronounced like “dream flight”). dRehmFlight was developed by the eponymous Nicholas Rehm to support his Master’s research at the University of Maryland. The code as well as thorough documentation were originally uploaded to GitHub in September of 2020, and is still updated at the time of writing (<https://github.com/nickrehm/dRehmFlight>). Alongside this, tutorial videos to help get the system functioning can be found on YouTube . com [16]. The objectives of this code are to be easy to learn and modify by the user at the source-code level and to be highly modular for use on custom flying machines. The creator emphasizes that the code is to be used for hobby and research applications only, and especially not for human-rated aircraft. Although the creator has implied that the code may not be the most efficient, he and other YouTube channels have demonstrated the successful implementation of dRehmFlight on a wide variety of unique systems [17]. One particularly relevant video is one made by Rehm, which demonstrates the capabilities of the software to balance a flying inverted pendulum [18]. Unfortunately, this code was not made public and no attempt was made to contact Mr. Rehm to share the code. This combination of a simple and open source code, modularity, thorough documentation, successful field tests by first and third parties, and video evidence that this code could be used to complete Milestone 2 is what separated it from the other viable options for flight software. It was judged to be the most efficient way to complete Milestones 1 and 2, and eventually used to complete Milestones 3 and 4.

The selection of the hardware components was driven by the choice of dRehmFlight as the flight software and by what was already available in the University of Minnesota UAV Laboratory or can be easily acquired at a local hardware store. The flight computer selected to run dRehmFlight is a Teensy 4.1 microcontroller, and the inertial measurement unit (IMU) used for sensing is an MPU 6050, as suggested by Rehm [16]. The quadcopter uses a DJI Flamewheel 450 frame with the T Motor F30A 2-4S ESC BLHeli\_32 electronic speed controllers (ESCs), the DJI 2312E motors, and the DJI Z-BLADE 9450 propellers. The system is controlled by a pilot with a Taranis X9D Plus transmitter.



(a) Hovering shortly after takeoff.



(b) Moving back and forth.

**Figure 4:** Frames from a successful flight of the quadcopter once the gains had been tuned.

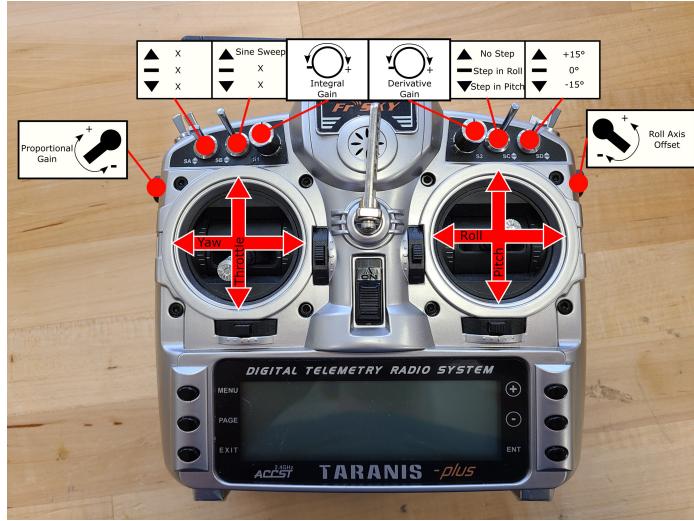
As shown in Figure 3, dRehmFlight serves the role of flight controller for both Milestone 1 and 2. In short, the structure of control loop in the original code is mostly unchanged from the original. The changes include adjustments to the controller gains for the attitude controller as well as the addition of safety and supplemental features. The main features of the original code as it pertains to this work is summarized here, and the additional features are summarized in the following section.

The navigation is completed using a Madgwick filter, a computationally-efficient quaternion-based algorithm for estimating attitude using tri-axis gyroscopes and accelerometers as well as magnetic angular rate and gravity sensor arrays [19]. The Madgwick filter is not modified in this paper as it sufficiently estimated the states to complete Milestones 1 and 2, although a variation that does not make use of magnetometer was used based on the available hardware. The quadcopter attitude controller is a proportional-integral-derivative (PID) controller with separate channels for the roll, pitch, and yaw. The inputs to the controller for the roll and pitch channels are the reference angle and the states from the navigation, and the outputs are desired moments. The inputs to the controller for the yaw channel are the reference and measured yaw rate, and the output is a desired moment. The control mixer takes the throttle and the desired moments from each channel and determines the motor commands written to the ESCs via OneShot125 protocol. As mentioned above, the main change to the code is the alteration of the controller gain values. These are first tuned using the test stand (described below) to get the gains close to the final values as discussed in the subsequent test stand section, then fine tuned mid-flight using the transmitter as discussed in the following section. Figure 4 shows frames from a video of a successful flight of the quadcopter.

### Additional features

The ability to customize dRehmFlight is demonstrated by the features added to the original code for safety and controller synthesis purposes. These features include, but are not limited to

- System kill and reset switches,
- The ability to adjust the controller gains via dials located on the transmitter,
- Live wired-data-streaming during test-stand tests,



(a) Transmitter front.



(b) Transmitter top.

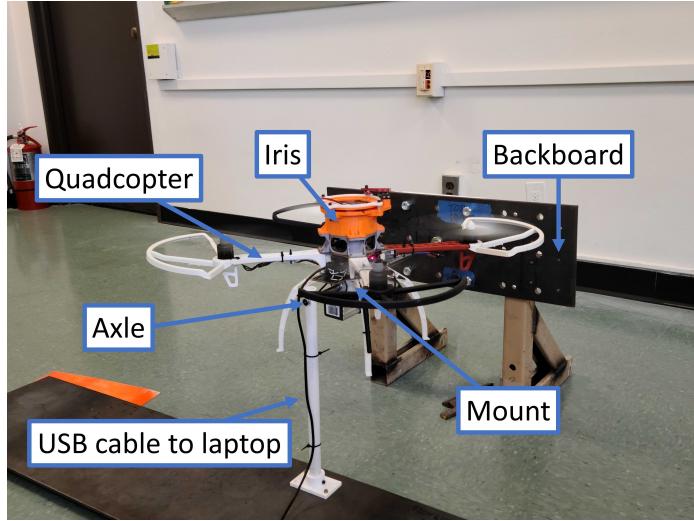
**Figure 5:** Diagram of transmitter with safety, supplemental, and controller synthesis features programmed into the interface.

- Live wireless-data-streaming during flight tests,
- Data recording to an SD card for post-flight analysis,
- Ability to send reference signals via the joysticks or pre-programmed step and chirp inputs on the transmitter,

The adjustments made to the transmitter's various dials and switches are shown in Figure 5. The ability to adjust the controller gains on the fly simplified the synthesis dramatically and reduced the potential for crashes. The gains can be mostly tuned on the test stand using dials on the transmitter and the live-streamed data as discussed in the following section. Then the gains can be fine tuned to their final values during flight testing using the dials.

### Test stand

A test stand, as shown in Figure 6, was constructed to check and tune the system without the

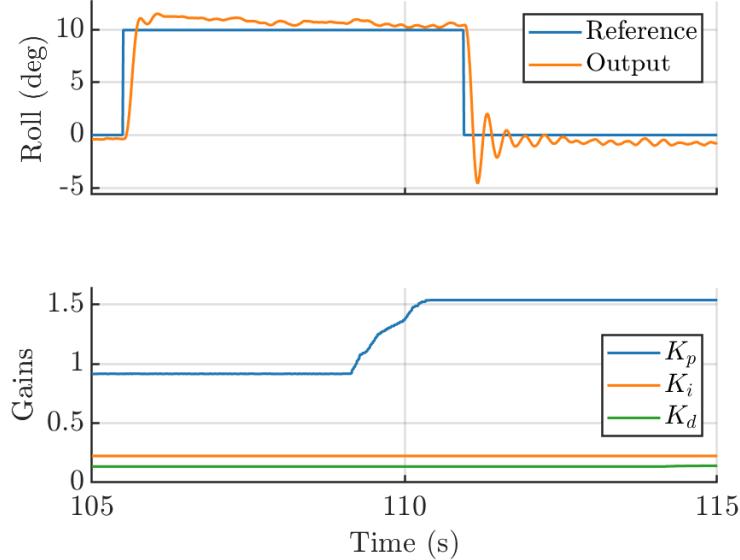


**Figure 6:** Test stand used to perform checks of the system, tune the inner and outer-loop gains, and gather data for system identification.

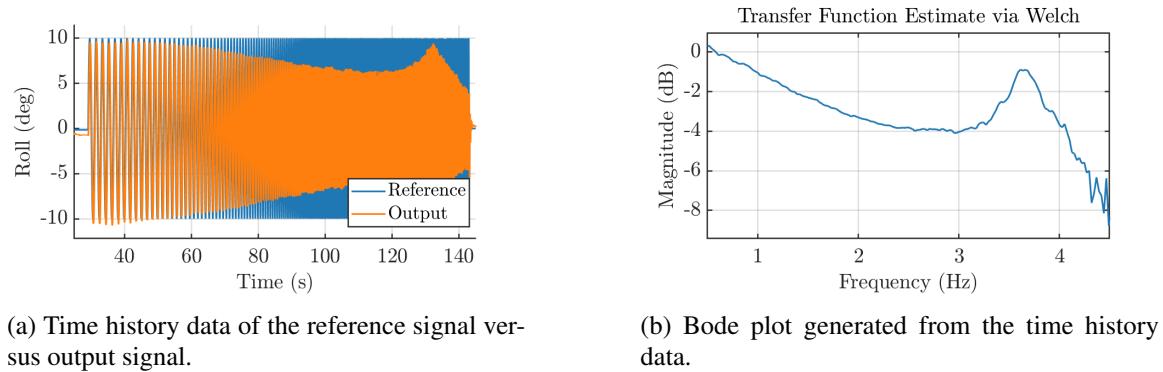
concern of crashing the quadcopter. This includes verifying that the system is constructed and software is coded properly, verifying that the navigation algorithm is performing well, analyzing input-output data, and tuning controller gains. The test stand consists of a stainless steel axle running through a 3D-printed mount attached with a Velcro strap between the battery and main bus of the quadcopter. This holds the quadcopter in place while allowing it to rotate about a single axis at a time. A heavy steel pegboard bolted to the ground, known as the backboard, holds one end of the axle in place. The other end is supported by a weighed-down tube, which has a 12-foot-long USB cable fastened to it that runs from the flight controller to a laptop.

An example of how the test stand can be used to tune the quadcopter attitude controller gains is shown in Figure 7. At  $t = 105$  s, the quadcopter is level to the ground with the proportional gain  $K_p = 0.9 \text{ N}\cdot\text{m}/\text{deg}$ , the integral gain  $K_i = 0.1 \text{ N}\cdot\text{m}/\text{deg}\cdot\text{s}$ , and the derivative gain  $K_d = 0.03 \text{ N}\cdot\text{m}\cdot\text{s}/\text{deg}$ . The user inputs a step input of +10 deg roll using the transmitter at  $t = 105.5$  s and the quadcopter attempts to track the reference. Once the step input is complete, the  $K_p$  gain is then increased using the dial on the transmitter to  $K_p = 1.5 \text{ N}\cdot\text{m}/\text{deg}$ , which results in a faster rise time and higher percent overshoot when the user inputs a step input from +10 deg to 0 deg roll at  $t = 111$  s. The gains were tuned until a satisfactory response was obtained on the test stand. The data plotted here was extracted from the SD card that stored the data, but the user saw this in real time on a plot window in MATLAB that was receiving the live-streamed data. The test stand is not a perfect simulation of the free-flying quadcopter, therefore, some fine tuning must be performed using the transmitter during flight testing.

An example of the test stand being used for the analysis of input-output data is the system identification shown in Figure 8. The quadcopter begins level to the ground at  $t = 35$  s in Figure 8a,

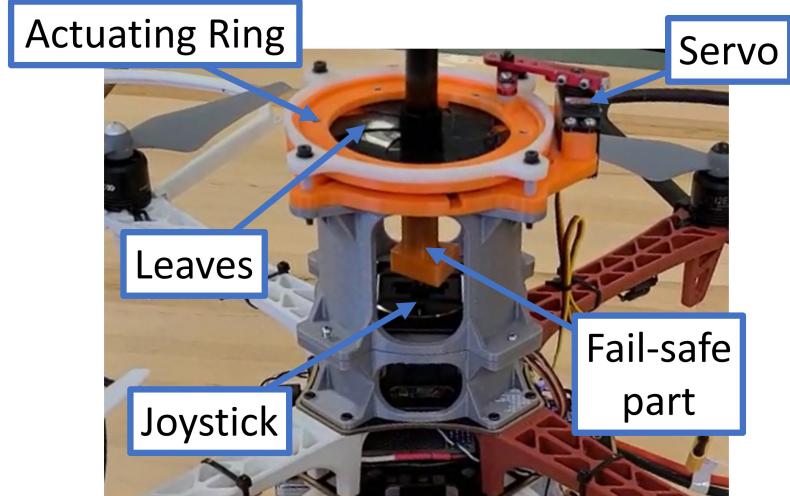


**Figure 7:** An example of live tuning the gains of the quadcopter attitude controller during step inputs while it is on the teststand.



**Figure 8:** Example of use of the teststand to perform a frequency-domain analysis of the quadcopter attitude control system using a chirp reference signal between 0.5 Hz and 4.5 Hz.

then the user inputs a pre-programmed roll chirp that sweeps through the frequencies starting at 0.5 Hz and ending at 4.5 Hz. The chirp spends more time at the lower frequencies to gather more data points in that range. Once complete, the data is extracted from the SD card and loaded into MATLAB, and the `tfestimate` function is used to generate the Bode magnitude plot shown in Figure 8b. Although this feature was not used directly to reach Milestones 1 and 2, its implementation helped eliminate bugs and check the understanding of the system. This feature can also be used for teaching and will need to be used in future work to identify an accurate system model, which will be necessary to complete Milestones 3 and 4.



**Figure 9:** Close up of the interface between the quadcopter and rigid inverted pendulum.

## INVERTED PENDULUM HARDWARE AND CONTROLLER SYNTHESIS

Once the quadcopter is able to perform stable flight, progress towards Milestone 2—a flying inverted pendulum—can be made. The steps to complete Milestone 2 are 1) design the interface between the quadcopter and RIP, 2) select the sensing and estimation for the RIP, and 3) synthesize a controller to stabilize the system. This section presents these steps as well as the design of additional features that improve the operational safety and ease of controller synthesis of the system.

### Quadcopter and rigid inverted pendulum interface

The interface between the quadcopter and rigid inverted pendulum is shown in Figure 9. The rigid inverted pendulum is affixed to the quadcopter via a joystick from a retired transmitter. The interface is designed such that damage to the joystick is mitigated during a crash. A 3D-printed fail-safe part housing a nut is loosely fit onto the end of the inverted pendulum. The nut is then screwed onto the existing threads of the joystick. The fail-safe part went through many design iterations, but the final design is the simplest and most effective. It is designed to have thin walls that act as a failure point so the energy from the crash does not damage the joystick.

At the beginning of a flight when the system is first becoming airborne, the quadcopter cannot simultaneously take off and balance the inverted pendulum. One method to initialize the system is to have a person hold the inverted pendulum steady before the quadcopter can take over—this is the approach taken by Rehm [18]. To eliminate the need for a person to be near the spinning propellers, it was desired to have a remotely-triggered mechanism that rigidly fixes the RIP to the quadcopter at takeoff and releases the RIP during flight. The solution to this is a custom 3D-printed assembly known as “the iris,” and is shown in Figure 9. The iris has a set of curved leaves that open and close like a camera shutter around the RIP as the actuating ring is rotated—this is where the iris gets its

name from. The leaves are made of Polytetrafluoroethylene (PTFE) to reduce friction, as a servo motor, which is connected to the flight controller, rotates the actuating ring. The iris was primarily designed to improve safety at takeoff, but it has also proved that it can mitigate crashes. The relative attitude between the quadcopter and RIP is monitored by the flight controller. If the RIP reaches the limit of its range of motion during flight, which is determined by the diameter of the opened iris, it is considered a balancing failure, and the flight controller closes the iris and returns the quadcopter to level flight. The iris was tested in the lab to make sure it could successfully close when the RIP is away from vertical. Dozens of flights with the open iris have been completed, but, thanks to the automatic closing of the iris, only two crashes were suffered.

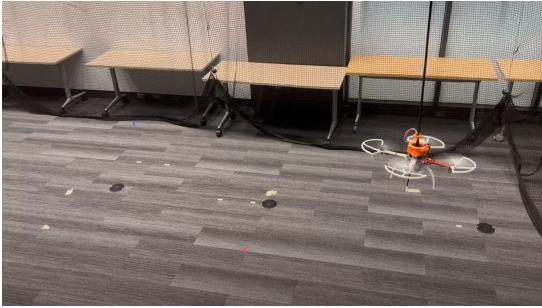
Commands to open and close the iris are sent to the flight controller via a switch on the transmitter. Flipping the switch also indicates to the system whether it should use the control loops from Figure 3a or Figure 3b. When the iris is closed, the control loop from Figure 3a is used, where the flight controller neglects the RIP states and uses the quadcopter attitude controller to track the references coming from the transmitter. When the iris is open, the control loop from Figure 3b is used, where the flight controller uses the inverted pendulum controller to track the RIP attitude references coming from the transmitter. The flight controller automatically switches controllers when the iris closes in the event of a balancing failure.

### Sensing, estimation, and controller synthesis

Another MPU 6050 IMU is mounted to the RIP for attitude estimation. Another Madgwick filter that estimates the roll, pitch, and yaw of the RIP runs in parallel with the Madgwick filter for the quadcopter. The measurements from the joystick were initially used to estimate the states of the inverted pendulum, however, the conclusion was reached that the measurements are too noisy, which was also observed by Rehm [18].

The inverted pendulum controller is a PID controller with separate channels for the roll and pitch. The inputs of the controller for these channels are the reference angle and the states from the navigation, and the outputs are desired roll and pitch of the quadcopter. These outputs are fed into the quadcopter attitude controller to execute. The reference yaw rate from the transmitter is fed directly into the quadcopter attitude controller.

The test stand from Figure 6 was used to verify that the system was behaving as expected and to tune the controller gains. Since the quadcopter is constrained so that it can only rotate, a person has to hold the RIP to stabilize the system. The operator/experimenter can move the pendulum back and forth to check that the control system is exhibiting the expected non-minimum phase behavior characteristic of inverted pendulum systems. While the one experimenter holds the RIP straight up, another person with the transmitter can initiate the step inputs, which are discussed in the previous test stand section, that have been modified to send reference signals into the inverted pendulum controller. This is how the gains for the inverted pendulum controller are initially tuned to stabilize



(a) The moment after the iris opens at  $t = 64$  s to release the inverted pendulum.



(b) The flying inverted pendulum moving around at  $t = 100$  s.

**Figure 10:** Frames from a video taken on August 10<sup>th</sup> of the second successful flight of the flying inverted pendulum. The total flight duration was 1 min 12 s from between when the iris was opened at  $t = 64$  s and closed at  $t = 136$  s by the pilot.

the system before being fine tuned during flight. Additionally, the quadcopter attitude controller gains were tuned further and the end mass on the end of the RIP was increased until there is a successful flight of the flying inverted pendulum.

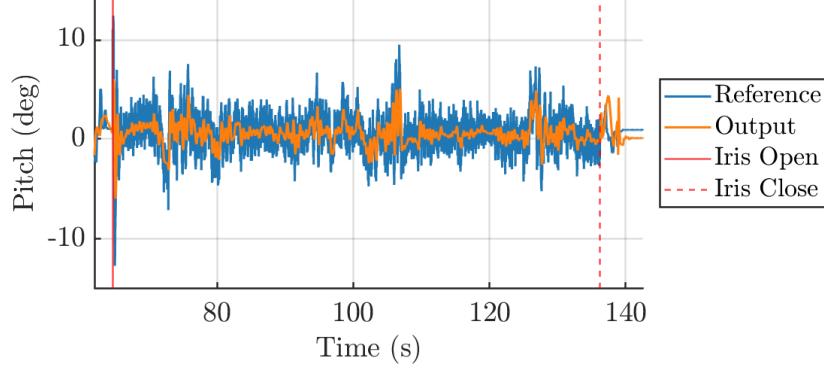
Frames from the video of the second successful flight of the flying inverted pendulum taken on August 10<sup>th</sup>, 2023 are shown in Figure 10. Figure 11 shows some of the data from that flight of the reference and measured pitch of the inverted pendulum as well as the reference and measured pitch of the quadcopter during the same flight.

## CLOSING REMARKS

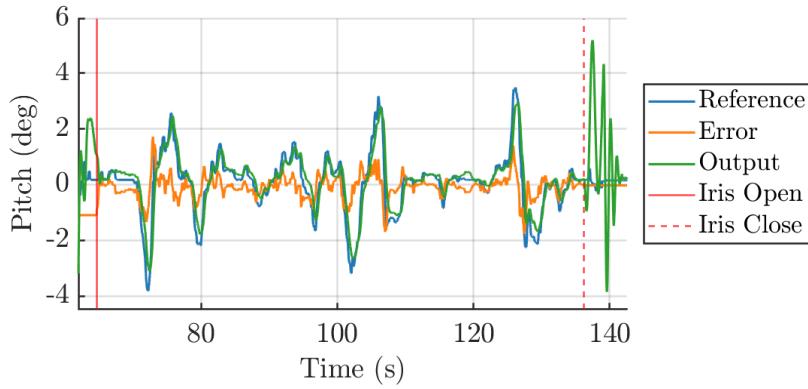
This paper presented an outline of the steps taken to design, fabricate, and fly a flying inverted pendulum. Along the way, we demonstrated the potential of the physical CRQS as a tool for teaching (e.g., dynamic modeling, control design) and research (e.g., maturing novel guidance, navigation, and control algorithms for space vehicles). From the outset of this research effort, the objective was to complete Milestones 2. The reason for this is that a functional flying inverted pendulum can generate enough interest in the project so that the physical CRQS may someday be used to analyze space vehicle trajectories even after the completion of this project.

## ACKNOWLEDGMENTS

This work was supported in part by the NASA Office of STEM Engagement (OSTEM) under grant number 80NSSC19K1672 issued through the NASA Fellowships Program under MUREP funding. This work was also supported in part by the Office of the Vice President for Research at the University of Minnesota through the Grant-in-Aid of Research, Artistry and Scholarship (GIA) program. The authors acknowledge the fellowship support provided by the NASA/Minnesota Space Grant Consortium for some of the undergraduate research assistants (Milan Patel and Grant Hietpas)



(a) The reference and output signals of the quadcopter's pitch.



(b) The reference, error, and output signals of the inverted pendulum's pitch.

**Figure 11:** Plots of flight data collected from second successful flight of the flying inverted pendulum.

who helped design the iris mechanism for CRQS. Finally, the authors would like to thank Dr. Chris Regan (former director of the University of Minnesota UAV Laboratory) for his invaluable research advice and mentoring of the graduate and undergraduate research assistants that worked on CRQS.

## REFERENCES

- [1] C. J. Miller, J. S. Orr, C. E. Hanson, and E. T. Gilligan, “Airborne Simulation of Launch Vehicle Dynamics,” *Annual AAS Guidance and Control Conference*, 2015.
- [2] T. G. McGee, D. A. Artis, T. J. Cole, D. A. Eng, C. Reed, M. R. Hannan, D. G. Chavers, L. D. Kennedy, J. M. Moore, and C. D. Stemple, “Mighty Eagle: The Development and Flight Testing of an Autonomous Robotic Lander Test Bed,” *Johns Hopkins APL Technical Digest*, Vol. 32, No. 3, 2013, pp. 619–635.
- [3] R. Frampton, K. Oittinen, J. M. Ball, P. Ferguson, C. Ake, and S. Mahoney, “Planetary Lander Testbed for Technology Demonstration,” *AIAA SPACE Conference and Exposition*, 2013, p. 5337.
- [4] N. Trawny, J. Benito, B. E. Tweddle, C. F. Bergh, G. Khanoyan, G. Vaughan, J. Zheng, C. Villalpando, Y. Cheng, D. P. Scharf, *et al.*, “Flight Testing of Terrain-Relative Navigation and Large-Divert Guidance on a VTVL Rocket,” *AIAA SPACE Conference and Exposition*, 2015, p. 4418.

- [5] J. M. Carson, C. R. Seubert, F. Amzajerdian, C. Villalpando, C. Bergh, T. V. O’Neal, E. A. Robertson, G. D. Hines, and D. Pierrottet, “COBALT: A Payload for Closed-Loop Flight Testing of Lander GN&C Technologies on Terrestrial Rockets,” *AIAA SPACE Conference and Exposition*, 2016, p. 5592.
- [6] L. Spannagl, E. Hampp, A. Carron, J. Sieber, C. A. Pascucci, A. U. Zgraggen, A. Domahidi, and M. N. Zeilinger, “Design, Optimal Guidance and Control of a Low-Cost Re-Usable Electric Model Rocket,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 6344–6351.
- [7] R. Linsen, P. Listov, A. d. Lajarte, R. Schwan, and C. N. Jones, “Optimal Thrust Vector Control of an Electric Small-Scale Rocket Prototype,” *International Conference on Robotics and Automation*, 2022, pp. 1996–2002.
- [8] N. Bourliatoux, J. M. Riera, and L. R. Lustosa, “Definition of a Landing Strategy for a Model-Scale Reusable Rocket,” *CEAS EuroGNC, Conference on Guidance, Navigation, and Control*, 2022.
- [9] J. Pei and P. Rothhaar, “Demonstration of the Space Launch System Augmenting Adaptive Control Algorithm on Pole-Cart Platform,” *AIAA Guidance, Navigation, and Control Conference*, 2018.
- [10] L. Miller, J. Pei, and P. Rothhaar, “Demonstration of the Space Launch System Augmenting Adaptive Control Algorithm on a Quad-Rotor,” *AIAA Guidance, Navigation, and Control Conference*, 2018.
- [11] J. Pei, A. Puetz, C. Duarte, L. Miller, and P. Rothhaar, “Suppression of Nonlinear Rotary Slosh Dynamics using the SLS Adaptive Augmenting Control System Demonstration on a Quadcopter Testbed,” *AIAA Scitech Forum*, 2019.
- [12] W. J. Elke, J. Pei, R. J. Caverly, and D. Gebre-Egziabher, “A Low-Cost and Low-Risk Testbed for Control Design of Launch Vehicles and Landing Systems,” *IEEE Aerospace Conference Proceedings*, 2021.
- [13] W. J. Elke and R. Caverly, “Dynamics, Guidance, and Control of a Low-Cost Quadcopter-Based Space Vehicle Testbed,” *AIAA SciTech Forum*, 2024, p. 0778.
- [14] J. A. Frosch and D. P. Valley, “Saturn AS-501/S-IC Flight Control System Design,” *Journal of Spacecraft and Rockets*, Vol. 4, 1967, pp. 1003–1009.
- [15] W. Haesserman, “Guidance and Control of Saturn Launch Vehicles,” *AIAA Second Annual Meeting*, 7 1965, pp. 1–70.
- [16] N. Rehm, “dRehmFlight VTOL,” <https://youtube.com/playlist?list=PLTSCov-1GtMax-oA4Pnq80Tx4fTuCrjQ>, 2020.
- [17] N. Rehm *et al.*, “dRehmFlight Vehicles,” <https://youtube.com/playlist?list=PLTSCov-1GtMYf6SePuTPN-6iSvGNNGsI>, 2020–2023.
- [18] N. Rehm, “Flying Inverted Pendulum,” <https://youtu.be/XmYRQi48s-8>, 2021.
- [19] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, “Estimation of IMU and MARG Orientation Using a Gradient Descent Algorithm,” *IEEE International Conference on Rehabilitation Robotics*, 2011.