Fall 2018

Exploration of real-time quadcopter controls

Kirk Rudolph
rudolkir@gmail.com

Follow this and additional works at: https://lib.dr.iastate.edu/creativecomponents

Part of the Controls and Control Theory Commons

Recommended Citation
https://lib.dr.iastate.edu/creativecomponents/99

This Creative Component is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Creative Components by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Exploration of real-time quadcopter controls

by

Kirk Rudolph

A creative component submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Electrical Engineering (Systems and Controls)

Program of Study Committee:
Greg Luecke, Major Professor

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this thesis. The Graduate College will ensure this thesis is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University
Ames, Iowa
2018

Copyright © Kirk Rudolph, 2018. All rights reserved.
DEDICATION

I would like to dedicate this work and the pursuit of my M.S. degree to my family for their endless love and support.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>iv</td>
</tr>
<tr>
<td>NOMENCLATURE</td>
<td>v</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>vi</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>vii</td>
</tr>
<tr>
<td>CHAPTER 1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>CHAPTER 2. BACKGROUND</td>
<td>3</td>
</tr>
<tr>
<td>The Rolling Spider Minidrone</td>
<td>3</td>
</tr>
<tr>
<td>Educational Use</td>
<td>6</td>
</tr>
<tr>
<td>Mathwork’s Contributions</td>
<td>7</td>
</tr>
<tr>
<td>CHAPTER 3. METHODOLOGIES</td>
<td>8</td>
</tr>
<tr>
<td>CAD Design</td>
<td>8</td>
</tr>
<tr>
<td>Physical Characterization</td>
<td>9</td>
</tr>
<tr>
<td>Modeling</td>
<td>11</td>
</tr>
<tr>
<td>CHAPTER 4. RESULTS</td>
<td>13</td>
</tr>
<tr>
<td>Experimental</td>
<td>13</td>
</tr>
<tr>
<td>Simulation</td>
<td>14</td>
</tr>
<tr>
<td>CHAPTER 5. CONCLUSIONS &amp; FUTURE WORK</td>
<td>17</td>
</tr>
<tr>
<td>Simulation Improvements</td>
<td>17</td>
</tr>
<tr>
<td>Educational Potential</td>
<td>18</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>19</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

| Figure 2-1: Rolling Spider minidrone | Page 4 |
| Figure 2-2: iOS UI of the free Rolling Spider minidrone smartphone app | Page 4 |
| Figure 3-1: CAD design of the Rolling Spider minidrone | Page 8 |
| Figure 3-2: 3D printed test stand used for measuring the thrust produced by the quadcopter | Page 10 |
| Figure 3-3: Quadcopter on test stand | Page 10 |
| Figure 3-4: Setup for measuring the rotational velocity of the quadcopter propellers | Page 11 |
| Figure 3-5: Example section of the Rolling Spider Simulink model. This is mapping a single motor throttle command to the predicted thrust produced and rotational velocity of a propeller | Page 12 |
| Figure 4-1: Mapping from throttle to (left) gram force and (right) propeller angular velocity | Page 14 |
| Figure 4-2: Highest level view of the Simulink model | Page 14 |
| Figure 4-3: Simulation environment with unstable quadcopter takeoff | Page 16 |
**NOMENCLATURE**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOF</td>
<td>Degrees of Freedom</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer-Aided Design</td>
</tr>
<tr>
<td>COG</td>
<td>Center of Gravity</td>
</tr>
<tr>
<td>MOI</td>
<td>Moment of Inertia</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processing</td>
</tr>
<tr>
<td>MIT</td>
<td>Massachusetts Institute of Technology</td>
</tr>
<tr>
<td>USD</td>
<td>United States Dollar</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>MPH</td>
<td>Miles per Hour</td>
</tr>
<tr>
<td>MCU</td>
<td>Microcontroller Unit</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

I would like to thank my committee chair, Dr. Greg Luecke, for guidance and support throughout the course of this work. In addition, I would also like to thank my friends, colleagues, the department faculty and staff, without whom, this creative component would not have been possible.
ABSTRACT

Low-cost quadcopters have created new opportunities for the exploration of designing, simulating, and implementing real-time control systems. Here, the Rolling Spider minidrone is exploited in order to demonstrate a powerful workflow usable by most complex embedded applications requiring advanced control design. A review of the evolution and educational impact the Rolling Spider minidrone is given. Finally, a simple control scheme that can be deployed and evaluated in real-time is developed and evaluated.
CHAPTER 1. INTRODUCTION

Quadcopters have exploded in popularity in recent years due to continuously lowering cost and increased awareness, availability, and practical applications. These low-cost quadcopters offer many new and exciting opportunities that previously were not possible. For example, farmers are using quadcopters to monitor fields, first responders are using quadcopters to search for disaster survivors, and hobbyist photographers are able to record aerial photography easily.

The design of quadcopters is of particular interest to controls engineers because of their complex and unique dynamics. Having six degrees of freedom (DOF) and only four actuators, quadcopters are severally underactuated. Quadcopter are also highly non-linear. This increases the difficulty of designing stable and robust controls. Yet, even with these non-trivial complications, quadcopters are still controllable. In order to minimize the challenges of designing a quadcopter control system presents and increase awareness of a new educational tool, this creative component explores a recently developed workflow for learning about quadcopter control systems for professionals, students, and hobbyists alike.

Chapter 2 reviews the history of the Rolling Spider quadcopter and the development of the custom firmware that allows users to create, simulate, and deploy custom algorithms. Since the launch of the commercially available quadcopter in 2014, the Rolling Spider has quickly evolved into a versatile tool for developing advanced control systems. The evolution of using the Rolling Spider minidrone as a method of teaching student controls starting in a class taught at Massachusetts Institute of Technology (MIT) in Spring 2015. Since then, it has evolved through the help of Parrot and Mathworks, Inc into a well-supported hardware extension package connected to MATLAB and Simulink.
Chapter 3 discusses the methodologies utilized to complete the exploration of the quadcopter. A powerful workflow usable by most complex embedded applications requiring advanced control design is demonstrated. Computer-aided design (CAD) software is used to model the quadcopter’s physical properties. Simulations are used in order to design and evaluate robust control systems. Finally, embedded software is generated for use on the Rolling Spider minidrone.

Chapter 4 provides the simulation and experimental results obtained. The developed quadcopter model is explained in detail. Simulation results provide insight into how well the controls are able to stabilize the quadcopter. Experimental results from deploying the custom firmware are also discussed.

Chapter 5 concludes with the key takeaways and future objectives of this work. This including technical future work such as developing more advanced controls, digital signal processing (DSP) filters, and creating higher fidelity models. This also includes suggested future application of the workflow, use in an educational setting, and integration with other work.
CHAPTER 2. BACKGROUND

The Rolling Spider Minidrone

The Rolling Spider minidrone is a small, versatile quadcopter sold by the Parrot Company. The minidrone was first revealed at CES 2014 and started selling in August 2014 at a price of 100 USD [1]. Offered at the lower price range for quadcopters during this time, the commercial product is targeted at average consumers and hobbyists. Figure 2-1 shows the red version of the minidrone that was used in the experimental portion of this creative component.

The Rolling Spider has a small frame with a width of 95 millimeters, length of 96 millimeters, and height of 34 millimeters. This small design allows for both indoor and outdoor flights. The Rolling Spider also comes with a protective guard to prevent damage in the case of a crash when flying indoors. Built with replaceable motors, frame, body, and propellers, the Rolling Spider was designed to be fixed easily. Weighing only ~60 grams, the quadcopters modular lithium-polymer battery can keep the Rolling Spider airborne for ~8 minutes and produce a maximum speed of greater than 11 miles per hour (MPH) according to the included instruction guide.

The Rolling Spider’s hardware design utilizes many common low-cost microcontroller units (MCUs) and sensors. The first sensor, common to most quadcopters, is a 6-DOF inertial measurement unit which measures the x, y, and z accelerations and the which measures the angular velocities of each of the three rotational degrees of freedom. There is also an ultrasonic sensor and a barometric pressor sensor. With the ultrasonic sensor,
Figure 2-1: Rolling Spider minidrone.

Figure 2-2: iOS UI of the free Rolling Spider minidrone smartphone app.
it is possible to accurately measure the minidome’s height above the ground by sending and receiving ultrasonic sound waves. The pressure sensor helps the quadcopter maintain the desired hovering height and improves the stability of the quadcopter. A batter sensor is included to allow the safe landing of the quadcopter when the batteries voltage drops or when the estimated batter life is too low to continue the flight. A camera is also integrated into the Rolling Spider minidrone able to take 0.3-megapixel images. With the help of Mathwork’s hardware support package described later in this report, it is possible to utilize all of these sensors in order to create a robust control system with complex sensor processing and analyze the sensor data in real time.

The Rolling Spider minidrone is controlled over Bluetooth via a free smartphone application available on Apple iOS, Android, and Windows mobile operating systems called FreeFlightMini. Figure 2-2 displays a picture of the user interface (UI) of the app. On the pre-loaded commercial firmware, the left digital joystick controls the yaw angle and altitude while the right joystick controls the pitch and roll allowing for approximate x-y planar movement. Preprogrammed acrobatic maneuvers are also already preloaded on the firmware and can be performed using the buttons on the lower pane.

As of November 2018, the Rolling Spiders can be purchased for as little as 30 USD from large retailers such as Amazon.com [2]. This significant drop in price since the initial 100 USD announcement back in 2014 shows that quadcopter technology is becoming widely available to the average person. This low entry cost has also allowed new and creative uses of the Rolling Spider minidrone outside of the traditional hobbyist such as its use in education.
Educational Use

In Spring 2015, the Aeronautics and Astronautics department at Massachusetts Institute of Technology (MIT) revamped their course 16.30 entitled Feedback Control Systems to include use of the Rolling Spider [3, 4]. This course aims to educate students about the fundamentals of control theory by incorporating the use of palm-size drones. While the course lectures cover the theory, implementing the theory via quadcopters help reinforce understanding and build practical skills.

In the initial offering of the course, each student personally received a Rolling Spider minidrone with which they used to implement the theoretical information gained in lecture. A custom firmware for the Rolling Spider minidrone was developed in order to allow the students to program the control system. Multiple labs conducted throughout the semester incrementally build the control system used to make the quadcopter autonomous. The final included a creative project using the drone to complete a desired task. After the first offering of the redesigned 16.30 course went well, the course has continued to be offered at MIT. ¹

There have been many improvements to the initial class since it started using the Rolling Spider. While the initial custom firmware used to program the Rolling Spider worked well, a more integrated way of developing the control system and flashing the Rolling Spider’s embedded hardware was desired.

¹ The instructions for programing the rolling spider with MIT’s original custom firmware are feely available on GitHub. Also, an earlier version of this course taught in Fall 2010 is freely available through MITOpenCourseware [5].
Mathwork’s Contributions

Between Spring 2015 and Fall 2017, Parrot, MIT, and Mathworks, worked on creating a better environment which optimized the workflow from designing the controls to programing the quadcopter. The results of this work led to Mathworks introducing a hardware support package for the Rolling Spider minidrone in January 2018 (MATLAB R2017b) entitled “Simulink Support Package for PARROT Minidrones” [6]. The hardware support package includes many features that simply the design, testing, and implementation of the control system.

The best feature of the support package is its integration with Simulink. Simulink is a high-level graphical programing language that simplifies control algorithm development. Because the support package is based in Simulink, designing a control system becomes much easier. The filters, control loops, and dynamics of the quadcopter can all be incorporated into the Simulink model and then simulated to predict how the quadcopter would truly behave.

A predefined “Code Generation Template” was added, allowing users to generate code that runs on the quadcopter extremely quickly. Simply by linking the quadcopter over Bluetooth, the new firmware can be deployed. The generated firmware can also be viewed in order to better understand the transfer from the Simulink model to the embedded code.

Another new feature added by Mathworks was a Flight Control Interface. This interface allows quick start and stop commands as well as a safety feature that allows adjusting the total motor gains before the flight. This feature also allows exporting a flight log containing data that can be analyzed post-flight in MATLAB. The hardware support package continued has been continuously upgraded through the current release (R2018b) [7].
CHAPTER 3. METHODOLOGIES

CAD Design

A computer aided design (CAD) software was designed and is used in order to model the Rolling Spider minidrone and to obtain important physical properties such as center of gravity (COG) and the quadcopters moments of inertia (MOI). This CAD model was then implemented into a simulation environment in order to evaluate different control schemes.

Figure 3-1 shows the rolling spider as an STL file already imported into the Simulink environment. The propellers (not shown) were modeled separately and later joined to the Rolling Spider body with a rotational connection. This allows the propellers to rotate and produces a more realistic simulation. The body of the quadcopter was modeled as a single component for simplicity. In future designs, this could be improved by separately modeling the battery and accounting for the different mass distributions. This will slightly improve the estimation of the COG and MOIs.
CAD design was also used to create a test stand required to measure the amount of thrust produced by the quadcopter at different throttles. The test stand is shown in Figure 3-2 and will be further discussed in the next section.

**Physical Characterization**

When designing control systems and simulations for dynamical systems, such as a quadcopter, knowing the physical characteristics of the system is required. The physical properties such as COG and MOIs were estimated through the use of CAD software. However, the electrical motor and aerodynamical properties of the quadcopter are still unknown. Here, the tests used to evaluate the thrust vectors and rotational velocities of the propellers are described.

Because the quadcopter couldn’t be taken apart, the motor dynamics were treated as a black-box. As a simplification, instead of having the motor properties, it was necessary to develop a mapping from an input throttle command to the resulting thrust force applied to the quadcopter.

Figure 3-2 shows a test stand was designed in a CAD software and 3D printed. It was designed to snuggly hold the quadcopter upside-down on top of a scale. By slowly incrementing the throttle command of the motors, a weight can be measured. From this measurement, the thrust force can be calculated. Figure 3-3 shows the quadcopter mounted on the test stand.

In order to account for the energy and inertia stored in the propellers when the quadcopter is in flight, a tachometer was used to estimate the angular velocity for each throttle command. The tachometer that was can be seen next to the Rolling Spider in Figure 3-4. Both the mappings from input to thrust and input to angular velocity were critical.
Figure 3-2: 3D printed test stand used for measuring the thrust produced by the quadcopter.

Figure 3-3: Quadcopter on test stand.
Modeling

Modeling the quadcopter dynamics in Simulink was a significant portion of this creative component. Simscape, a package of Simulink was used to model all of the forces, reference frames, masses, MOIs, and all other physical properties of the quadcopter while classic Simulink signals were used to do all of the calculations and control development. A detailed walkthrough of the model will be given in the results section.

Figure 3-5 shows an example section of the quadcopters overall design. A throttle command for a single motor gets mapped to a thrust force and angular velocity of a propeller. The angular velocity is given to the rotational joint (where the propeller connects with the
Figure 3-5: Example section of the Rolling Spider Simulink model. This is mapping a single motor throttle command to the predicted thrust produced and rotational velocity of a propeller.

quadcopter body through Conn1) which automatically computes the dynamical equations. The thrust force is implemented as an external force acting on the center of the propeller. It can be seen from this simple example that detailed models can be very complicated to create. However, good models allow for the design of better control systems and is a very important when simulating results.
CHAPTER 4. RESULTS

Experimental

Figure 4-1 shows the results from the experiment tests conducted on the quadcopter. On the left, an arbitrary input throttle command ranging from zero to five hundred was given to all four motors simultaneously. The resulting force was read from the scale and was measured in units of grams. By converting grams to SI units, multiplying the recorded value by the value of gravity, and reducing the overall force by a quarter, the thrust of a single motor at each input throttle command can be measured. The resulting mapping can be seen to be mostly linear. The collected data was used in a lookup table within the model to accurately model the thrust of each motor for a given throttle command.

At lower throttle commands, it was much easier to take data because the quadcopter was relatively stable on the test stand. This can be seen by the amount of data collected at different regions of input command. At the higher input commands, the quadcopter and test stand could be seen moving slightly due to the non-perfect resting angle of the quadcopter on the test stand. However, the data collected was a close enough approximation for this simple proof-of-concept experiment.

On the right side of Figure 4-1, the mapping from the arbitrary throttle command to the angular velocity of the quadcopter’s propellers can be seen. The tachometer mentioned above was used to estimate the propeller angular velocity for a given throttle command. A dark mark was made on the end of a propeller to avoid over or under estimating the angular velocity by a factor of two. The results of these measurements are clearly nonlinear. At higher throttle commands however, an approximate linear region appears. These data points were also put in a lookup table and used in the model.
Figure 4-1: Mapping from throttle to (left) gram force and (right) propeller angular velocity.

### Simulation

Figure 4-2 shows the highest level of the simulation, a design similar to many control loops can be seen. An input is fed to a control block which acts on the plant. Sensors are then fed back into the control block. The error signal between the sensor readings and the desired input determine how the quadcopter will adjust in order to minimize the error signal. Plots are attached to the sensors of the simulated quadcopter in order to view the changes in acceleration, gyroscope, and height values over time.

Figure 4-2: Highest level view of the Simulink model
Figure 4-3 shows the results of running the Simulink simulation via the mechanics explorer. This example was using an open-loop controller (i.e. no sensor values were used to create a feedback loop). A constant throttle input was given to each of the motors, causing instability. The pane of images represents the quadcopter simulation taking off of the green platform. In the first few panes, not much movement is seen as the quadcopter’s velocity starts from rest. As the panes progress, it is clear that the drone is acceleration against the force of gravity and picks up speed. By the last pane, the quadcopter has tilted forward and is still acceleration.

Many simulations like the one seen in Figure 4-3 were performed during the development of the physical constraints of the quadcopter. The open-loop controls variant was implemented first and more example control schemes were later created. One of either open-loop, negative feedback, or PID control schemes can be selected using Simulink’s build in variant control features. The open-loop variant is unstable due to the inherent dynamics of the quadcopter mentioned earlier in this work. The negative feedback controls quickly adapt to the error between the desired stability and the true sensor values but is still unstable. Finally, using discrete PIDs are tuned such that the quadcopter was stabilized.

This workflow of slowly improving the control scheme using more advanced techniques is a powerful concept emerging controls engineers should master in order to develop new systems. While the development of the current model was slow, the final result accurately models how the quadcopter would behave in a real-world scenario.
Figure 4-3: Simulation environment with unstable quadcopter takeoff.
CHAPTER 5. CONCLUSIONS & FUTURE WORK

Simulation Improvements

There are still many improvements that can be made to the Simulink model’s level of detail in order to make it as functional as possible. For example, the work already done has a simple CAD model which very little detail was added. By improving the details on this model, the simulation will become more accurate. Similarly, the experimental data collected can be collected at a higher frequency to provide an even closer representation of the true relationship between input and output. Collecting more data such as the input to thrust relationship as a function of battery percentage would also prove helpful in accounting for the non-constant battery voltage that occurs.

The Simulink model’s layout also has room for improvement. In the future, it would be ideal to have variable sources of inputs as well as variable control schemes and variable plant dynamics models. This is possible with the variants system available in Simulink and was utilized a small amount in this work but still needs to be expanded for optimal effectiveness of the model.

The control block requires more work in order to create a stable control system. While the majority of the work focused on setting up the physical properties of the quadcopter in the simulation, the controls used to stabilize the quadcopter have yet to be completed. This also made it impossible to try and hover the quadcopter at this time as the system was still unstable. Given more time, stabilizing the quadcopter should be simple due to the quick iterative process the simulations provide.
Educational Potential

A large motivating force for this work was for its application in educational environments. Many universities have clubs/teams working on quadcopter development. For example, Iowa State University’s electrical and computer engineering department has had a continuous team of students in senior design work on a project called MicroCART. By understanding the workflow used with the Rolling Spider minidrone, the development of quadcopter control systems will become much easier.

Given the growing number of practical applications of drones, using the Rolling Spider minidrone also has a valuable place in education. With the predeveloped workflow, designing quadcopter controls have never been easier. Adopting this workflow and incorporating it into the class room similarly to how MIT initially did would be a significant contribution to teaching controls concepts that Iowa State University and other universities would benefit from.
REFERENCES


