Design, Control and Estimation with the MultiMo-MHR: a Multi-Modal Monopedal High Performance Hopping Robot

A dissertation submitted by

Samuel J. Burns IV

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Mechanical Engineering

Tufts University

February 2026

Samuel J. Burns IV

Advisor: Dr. Matthew A. Woodward

Abstract

With the advancement of robot platforms for complex tasks, a deeper understanding of how the system can traverse an environment is required. Two common strategies concentrate on terrestrial motion or aerial flight. This thesis explores an alternative locomotion strategy, hopping, requiring both a sturdy design to survive continuous ground impacts and to be highly agile for aerial maneuvering. This thesis explores high-performance, high-energy, and highly dynamic hopping up to 4 m, a task no other hopping robot has achieved; to date, excluding this work, the highest hopping robot attains a 1.63 m hop [1]. Where low energy hopping generates easier control stabilization and limits the full range of potential for hopping.

This dissertation culminates the exploration into this higher energy and dynamic region of hopping locomotion. A hopping robot, the MultiMo-MHR, was developed, capable of hopping up to 4 m while weighing 684 g and standing 369 mm tall. Designed with a singular monopedal hopping leg and a quadrotor-based motor configuration for control and energy input. The thesis is comprised of four main parts, starting with a detailed robot overview that explores both the background of hopping robots, the design, control, and experimental elements. The first study explores design principles and control design to optimize efficiency. Presenting both the high-performance robot as well as design and control principles for efficient hopping. To expand the capabilities of hopping systems, the subsequent study explored full vertical state estimation for hopping robots. A region of study that needs to be studied as hopping systems reach new speeds and heights, and ballistic assumptions (i.e., no thrust, no drag) fail. Presenting an optimized Kalman filter estimation technique, requiring only accelerometer input while minimizing computational and weight load. Ultimately, enabling direct hopping height control and achieving a mean absolute percent error on the hop height of 12.5%. A final study begins entering higher dynamic horizontal motion with an improved controller design. Prior studies used PiD control and centered on vertical hopping, now

expanding to a model-based Lyapunov control. The work shows that the robot can handle disturbances and complex trajectories without sacrificing vertical height, thanks to the guaranteed stability and model-based control offered with the new controller. All works improve the hopping robots' capabilities and push the bounds of what hopping robots can do and explore.

Acknowledgments

I want to thank my advisor and committee chair, Dr. Matthew Woodward, for his years of support, guidance, and mentorship throughout my time at Tufts. Additionally, thank you to the various members over the years in the Robot Locomotion & Biomechanics Laboratory (RLB Lab), for listening to presentations and providing feedback.

Thank you to my dissertation committee, Dr. Felix Huang, Dr. Jivko Sinapov, and Dr. Yan Gu, for the support and feedback during this whole process.

Thank you to the mechanical engineering department at Tufts for both the support and the various questions I had over the years. Especially to Dr. Jason Rife, Courtney Russo, and Addie Cotter. With a special shoutout to everyone at the weekly Frisbee Friday put on by Dr. Chris Rogers.

Thank you to the Stern Family for the support with the Stern Endowed Graduate Research Fellowship towards the end of my time at Tufts. Their assistance enabled me to continue my research and reach this point of having a dissertation.

I would also like to thank those who led me to arrive at Tufts and pursue graduate studies. I got my start in research at the University of Massachusetts Lowell, under Dr. John Mack and the Energy and Combustion Research Lab (ECRL). Thank you to him and the lab members for all the support and guidance in my early career. Additionally, getting robotics research experience with Dr. Yan Gu and the TRACE lab at UMASS Lowell, further pushing my desire for graduate studies in robotics.

I also want to thank my friends and family who have supported and encouraged me through my entire academic journey. Specifically, my parents, Samuel and Annette Burns, and sisters Morgan Burns and Courtney (Alex) Fluet. They were always there when I needed them, encouraging me and pushing me to be the best version of myself.

Contents

Abstract . . . . . **i**

Acknowledgments . . . . . **iii**

Contents . . . . . **iv**

List of Tables . . . . . **viii**

List of Figures . . . . . **ix**

1 Introduction **1**

1.1 Motivation 1

1.2 Research Goals 3

1.3 Contributions 6

1.3.1 Hopping robot development 6

1.3.2 Design and control for high-performance hopping 6

1.3.3 Vertical state estimation via optimized Kalman filter and minimal sensor suite 7

1.3.4 Hopping robot non-linear controller implementation for higher system dynamics and disturbances 8

1.4 Dissertation Organization 8

2 MultiMo-MHR: robot & setup overview **11**

2.1 Introduction 11

2.2 Background 12

2.3 Design Goals and Inspiration 14

2.4 Design Overview 16

2.4.1 Robot General Specifications 16
2.4.2 Technical Specifications 20
2.4.3 Data Compression and Saving 23
2.4.4 Motor force output 25
2.4.5 Control Fundamentals 27
2.4.6 Hopping Methodology 32
2.5 Experimentation Setup 33
2.5.1 Room overview 33
2.5.2 Room safety 34
2.5.3 System axes and Rotation matrices 34
2.5.4 System Calibration 38
2.5.5 Experimental Data transmission Pipeline 40
2.6 Summary 41

3 Design and Control of a High-Performance Hopping Robot 43
3.1 Abstract 43
3.2 Introduction 43
3.3 Dynamic Model 46
3.3.1 Limited Rebound Control Input TWR, $\alpha_r \leq 99\%$ 53
3.3.2 Prescribed Rebound Control Input TWR, α_p 56
3.4 Robot Design 56
3.4.1 Energy Storage 57
3.4.2 Leg Design 59
3.4.3 Control 62
3.5 Experimental Results and Discussion 65
3.5.1 Energy Accumulation and Dissipation 67
3.5.2 Hopping Experimental Efficiency 68
3.6 Summary 69

4 Optimized Kalman Filter based State Estimation and Height Control in Hopping Robots . . . 70

4.1 Abstract . . . 70

4.2 Introduction . . . 71

4.3 Robot Design . . . 74

4.4 Hopping Phase Estimator (HPE) . . . 75

4.5 State Estimation Methodology . . . 77

4.5.1 Hopping Vertical State Estimator (HVSE) . . . 80

4.5.2 HPE and HVSE Training . . . 81

4.5.3 Training Results . . . 84

4.6 Sensing Frequency . . . 87

4.7 Hopping Height Control Experiments . . . 90

4.7.1 Results . . . 90

4.7.2 Agility Metric . . . 93

4.8 Summary . . . 97

4.9 Appendix . . . 98

4.9.1 Hopping Agility . . . 98

4.9.2 Robot Parameters . . . 101

4.9.3 Error Metrics . . . 102

4.9.4 Optical Flow Error . . . 103

5 Non-linear Controller Implementation for Enhanced Hop Stability . . . 105

5.1 Introduction . . . 105

5.2 Lyapunov controller selection and design . . . 107

5.3 Robot design and controller Implementation . . . 111

5.3.1 General control modifications . . . 112

5.3.2 Drop phase orientation modification . . . 113

5.3.3 Rebound phase vertical velocity control . . . 116

5.4 Experiments: disturbance recovery 116
5.4.1 PiD controller setup and disturbance performance 118
5.4.2 Lyapunov controller setup and disturbance performance 121
5.5 Experiments: variable positioning hopping 124
5.6 Summary 128

6 Summary and Future Work 130
6.1 Thesis Summary 130
6.2 Future Work 130
6.2.1 Hopping controller variation and analysis 130
6.2.2 System localization and mapping, outdoor testing 132
6.2.3 Active and passive foot variations & surface modifications 133
6.3 Publications 135

Bibliography 136

List of Tables

2.1 MultiMo-MHR Mechanical Components 21

2.2 MultiMo-MHR Electrical Components 22

3.1 Hopping Robot Systems 45

3.2 Robot Parameters 58

3.3 Experimental Trial Sets 67

3.4 Experimental results at constant hop heights of 1 and 2 m 68

4.1 Kalman Filters 78

4.2 Optimization Parameters 84

4.3 HVSE Performance Metrics 95

4.4 Hopping Platforms Vertical Agility 97

4.5 MultiMo-MHR Vertical Agility 101

4.6 Robot Parameters 102

4.7 HVSE Performance Metrics 103

5.1 PiD Controller Generated Disturbance Conditions 118

5.2 PiD Controller outcome after generated disturbance 119

5.3 Lyapunov Controller Generated Disturbance Conditions 122

5.4 Lyapunov Controller outcome after generated disturbance 124

5.5 MultiMo-MHR Variable Horizontal Positioning Accuracy 125

List of Figures

1.1 Benefits & applications of hopping robotics. Being usable in rough, uneven, or gap-filled terrain, where standard terrestrial systems may struggle. With increased energy efficiency, storing energy from one hop to the next which lowers the required input energy when compared to quadrotor systems. They can be highly agile, traversing a space fast and able to change motion directions by varying how they land. And using far fewer components in terrestrial systems, lowering cost and failure modes. This enables a wide use case, from search & rescue to mapping and exploration, given higher efficiency, lower cost, and high agility. To more extreme uses like non-earth bodies where they could traverse rapidly, assuming they can operate & control in that environment. Lastly, they can serve to entertain by mimicking what we see in film or cartoons. 4

2.1 Compilation of Monopedal Hopping Robots. Image sources: a (Raibert 3D Hopper [2]), b (ARL Hopper I [3]), c (ARL Hopper II [4]), d (3D Bow Leg Hopper [5]), e (Salto [6]), f (Salto-1P [7]), g (LEAP [8]), h (PogoDrone [9]), i (PogoX [10]), j (Hopcopter [1]), & k (insect scale hopper [11]). 15

2.2 MultiMo-MHR Overview: a. R1 version, b. CAD internal cutaway view, c: R2 version, c.1: R2 front profile, c.2: R2 side profile 17

2.3 Robot electronic components. a: ESP32, b: RF transceiver(nRF24L01), c: voltage regulator, d: battery(2200 mAh, 35C, 2S, 7.4 V), e: electronic speed controller(20 Amp), f: 9-DOF IMU (MPU6050), g: data logger Spark Fun Open log, h: 3-DOF accelerometer (H3LIS331DL), i: 9-DOF IMU (WT901) 18

2.4 Motor load cell test setup, utilizing the peedyBee 1404 motor, 4030 propeller, and 5 kg load cell. Used to evaluate the relation between commanded duty cycle and outputted thrust, test results can be seen in Figure 2.5. 26
2.5 Motor force fit equation for the SpeedyBee 1404 motor and 4030 propeller. Demonstrating a linear relationship for duty cycles over 145 microseconds. . 28
2.6 Quadrotor motor configuration strategies. a: cross configuration (same as MultiMo-MHR). b: plus configuration. 29
2.7 Hopping cycle overview. The cycle had four main phases: drop, stance down, stance up, and rebound. Where each phase is marked with a specific transition point: touchdown (TD), max squat (MS), lift off (LO), and hop apex (HA). Where each transition can be detected to then initiate a control strategy for the matching phase. 32
2.8 Early MultiMo-MHR testing setups. a: vertically constrained testing on a guide wire. b: orientation control testing. 35
2.9 Experimental setup with safety rope for three robot positions. Red: center of the room, Blue: far left edge with safety rope, Green: far right edge of safety rope, Orange: central rope operator can pull to shorten overall safety rope length. 36
2.10 Coordinate axes for, a: the motion capture studio, b: the MultiMo-MHR . . 37
2.11 Components for the Vicon Motion Capture studio. a. retro reflective tracker markers, b. calibration wand, c. Vicon vantage V5 camera 39
2.12 Data communication schematic for experimental testing of the MultiMo-MHR. The motion capture cameras capture the object's placement in the room and send it to the base PC over Ethernet. The base PC transfers the data over USB to an external ESP32 and transmits data over RF to the robot. An additional Bluetooth controller also connects to the ESP32 for extra commands and e-stopping. 42

3.1 a) Hopping cycle: Phases (drop, stance, and rebound) and transitions (touchdown, liftoff); with locomotion energy (LE) modifications: Control inputs (E_d, Es, E_r), and losses (E_{FDd}, E_{TD}, E_{mech}, E_{LO}, E_{FDr}). b) Symmetry plane of the MultiMo-MHR. c) CAD model of the hopping cycle with hop height measured as the center-of-mass (CM) displacement between LO and hop apex (excluding stance). c.1) Section view showing internal connections, 4 degrees-of-freedom, and dynamic parameters. c.2) Showing the two bodies (Body, Leg) at max spring extension during the stance phase. 46

3.2 Limited rebound control input TWR, $\alpha_r \leq 99\%$. a) Energy accumulation in the MultiMo-MHR with varying α_r, and corresponding cycle averaged TWR, α_{cyc}, for constant hopping height. b) Critical α_r (α_{cr}, blue lines, minimum TWR) and corresponding critical thrust, $F_{cr} = \alpha_{cr} m_T g$ (red lines) for continuous hopping at 0.25 m (solid lines) and 3 m (dashed lines). The MultiMo-MHR (black dot) with 200 g added to either the body (green dot) or leg (yellow dot) is also shown. 54

3.3 Hopping at various prescribed rebound control input TWRs ($\alpha_p = [\alpha_r$, 80, 90, 99, 125, 150, 200, 400]%); where each is maintained for the period, shown in the lower bar, necessary to reach the desired hop heights [1, 2, 4, 8] m. The resulting α_r (a) and α_{cyc} (b) are presented. 55

3.4 Rubber band stress-strain limit test for three bands. The results show a maximum stress and strain of 11.2 MPa, 6.2, with the total specific energy of the bands found to be 15800.3 J/kg. 59

3.5 Robot Design: a) MultiMo-MHR with monopedal hopping leg and quadrotor for control. a.1) Robot 1 (R1). a.2) Robot 2 (R2). b) Control scheme. 60

3.6 Poincaré map of the evolution of the TD velocity angle β_{TD}, using the presented dynamic model (1 m hops), as a function of $\beta_{TD(k)}$ and $\theta_{TD(k)}$; where the stable region, $[-1 < \beta_{TD(k+1)}/\beta_{TD(k)} < 1]$, indicates the range over which β_{TD} reduces. The inset shows the relationship between the angles and positive x-direction. 63

3.7 Snapshots of high-speed video (R1 Criteria-2): a) Constant height ($h_d = 1.5$ m) $\alpha_p = 74.2\%$, b) Increasing height ($h_d = 1.5$ m) $\alpha_p = 92.2\%$, c) Decreasing height ($h_d = 1.5$ m) $\alpha_p = 0\%$. 64

3.8 Experimental trials (R1 n = 843 hops, R2 n = 190 hops): a) Hopping height ratio (δ_{rd}) as a function of the rebound control input TWR (α_r) and drop height (h_d); where energy can be accumulated (gray region), dissipated (white region), and remain constant ($\delta_{rd} = 1$). b) Cycle average TWR (α_{cyc}) as a function of α_r 66

4.1 Hopping robot. a) Operational phases (drop, stance down, stance up, rebound) and transitions (touchdown (TD), max squat (MS), liftoff (LO), hop apex (HA)) a.1) Photo of the MultiMo-MHR platform with components labeled. a.2) CAD model with overall dimensions labeled. a.3) CAD model section view with spring extension (stance phases) showing the combined body and leg structures. b) Dynamic simulation showing the locally, highly nonlinear, behavior of the body at TD and LO for a 3 m hop. 73

4.2 Hopping robot control block diagram. 75

4.3 State machine diagram of the Hopping Phase Estimator (HPE), Hopping Vertical State Estimator (HVSE), and controllers showing both the prediction updates and Inferred Measurement Updates (IMUPTs) of the HVSE. Shown are phases (green), phase transition criteria (red), and transitions (blue) . . . 76

4.4 Sensitivity to changes in the optimized parameters; $\mathcal{L}_c = 100\gamma_1 + 10\gamma_2 + 10\gamma_3$ from equations 4.9, 4.10. 86

4.5 Change in the estimated ground height over multiple hops; where STD is the standard deviation, and the true ground height is zero. 86

4.6 Simulated State Estimation Error. a) Unbound and large bound position error, given state estimation at 10 to 400 Hz. b) Unbound and large bound velocity error, given state estimation at 10 to 400 Hz. c) Bound position error, given state estimation at 600 to 3360 Hz. b) Bound velocity error, given state estimation at 600 to 3360 Hz. 88

4.7 Example hopping trials using the hopping vertical state estimator (HVSE) based control a-b) Displays the vertical position and velocity error of the first 3 and last 2 hops for hopping trials at heights of 1,2,3&4 m. Each has a zoomed in region (a.1, b.1) showing the stance phase behavior. c) Snapshots from a high-speed video of the MultiMo-MHR hopping with the HVSE at a desired height of 3.25 m. 91

4.8 Mean and standard deviation of the position (a-d) and velocity (e-h) error for the experimental trials (n=70 hops) utilizing the true state (motion capture data) for height control. a&e) *Stance down* phase, TD to MS. a&e.1) Zoomed in region of (a&e) to show the behavior after TD). b&f) *Stance up* phase, MS to LO. c&g) *Rebound* phase, LO to HA. c&g.1) Zoomed in region from 0-15% of *Rebound* phase c&g.2) Zoomed in region from 15-100% *Rebound* phase. d&h) *Drop* phase, HA to next TD. Note) The phases are segmented by the true state data, while the dots indicate the points at which the robot identified the specified transition. 94

5.1 Control diagram for the PiD controller 114

5.2 Control diagram for the Lyapunov controller 115

5.3 PiD controller, position and orientation data for disturbance recovery. Showing both the low energy (n = 13 trials) and high energy (n = 6 trials) disturbances. 120

5.4 PiD controller, duty cycle average during and after the generated disturbance. Showing both the low energy (n = 13 trials) and high energy (n = 6 trials) disturbances. 120
5.5 Lyapunov controller (n = 7 trials), position and orientation data for disturbance recovery. 123
5.6 Lyapunov controller (n = 7 trials), duty cycle average during and after the generated disturbance. 123
5.7 Quadrant hopping locomotion path, Red dot: start/end position, travels in order along paths 1-8 landing in target positions a-d. 126
5.8 Quadrant hopping locomotion experiment trajectory path. a: 3D view, b: 2D XY view. 127
5.9 Quadrant hopping locomotion experiment positioning. a: Vertical or Z height, b: Horizontal or X and Y positioning. 127

1. Introduction

1.1. Motivation

Two common traversal strategies for robot locomotion involve terrestrial or aerial locomotion. Terrestrial locomotion typically utilizes legged or wheeled-based strategies, generating reliable and durable machines that usually travel at slower speeds. Aerial traversal, alternatively, is used by drones or fixed-wing aircraft, boasting higher speeds and agility with lower durability and survivability upon crashes/impacts. But what if a robot required the high speed/agility of aerial systems and the robustness of terrestrial systems? Hopping locomotion robots are one such platform requiring this; requiring survivability from impacts after extreme hop heights (up to 4m, as this work will show) and high maneuverability to direct/stabilize their trajectory while in the air. Hopping platforms also improve on terrestrial systems by being able to enter rougher terrain while also possessing fewer actuators and components, which reduces cost and prolongs life. At the same time, they can be more efficient than flying with the ability to store energy between jumps [1]. Researching hopping systems not only bridges the gap between terrestrial and aerial platforms by interacting with both, but also advances insight into the unexplored regime of high-energy impact dynamics, rapid state transition, and stabilization, all of which are beneficial for any robot.

Hopping or saltatorial locomotion is seen only in a handful of creatures around the world, like the common kangaroo to the small desert kangaroo rat [12], and in even fewer robot platforms. Hopping implies a continuous motion, wherein the system has regions of an aerial phase followed by a short stance phase in which kinetic energy is stored and used in the subsequent hop. This locomotion strategy induces large impacts and results in rapid system dynamics, which increase with larger system energy, and requires fast stabilization upon transitioning into the air for subsequent hops. Hopping systems have the added benefit of being able to store and use energy for subsequent hops, increasing the locomotion strategy's

overall efficiency. A close relative of hopping locomotion is jumping, being a singular hop with an extended stance time for reorientation, energy storage, and/or an alternative locomotion method, before the subsequent jump. With inspiration to be gleamed from jumping systems, especially energy storage with one capable of a 30 m jump [13]. Hopping locomotion faces several challenges, requiring both a sturdy yet lightweight design to maximize survivability and efficiency and a rapid and robust stabilization strategy to enable continuous hopping. Achievable through studying methods for design optimization, testing new control strategies, and designs with a high-performance hopping platform to ensure functionality. The study of hopping locomotion generates innovation in not only hopping but also the aforementioned terrestrial and aerial systems with new design ideas, knowledge on energy storage and control strategies for platforms facing rapid accelerations, and large impacts.

Amongst hopping platforms, until recently, a large majority were tethered systems with minimal ($\leq$ 1 m) hopping heights. Early hoppers focused on stance-based control and a ballistic aerial phase. Aerial-based control strategies have been explored, with one style being a flywheel-based method for body reorientation to ensure a proper landing state. Even recently, hopping robots have remained at lower energy states, meaning lower system dynamics and higher overall stability. As such, the goal of this work is to advance the understanding of hopping locomotion through studying high-performance hopping, where high-energy level systems will be tested both for design and control. This work will use an alternative quadrotor-based control method, which enables both orientation control and thrust input, enabling new control methods and higher energies to be studied. Specific investigations will include studying efficient control, energy input strategies, and key design components for high-performance hoppes, improving the system's efficiency and operational time. With a more reliable platform, investigations of state estimation techniques will be developed to expand hopping viability outside of experimental setups. Lastly, an additional controller type will be investigated to enhance the overall stability and accuracy of desired hopping trajectories, specifically for high-energy disturbances and rapid positional changes,

which will enable continued studies on high-performance hopping locomotion.

Overall, hopping robots bring about a means of locomotion, taking inspiration from pre-existing terrestrial and aerial locomotion strategies while facing benefits and challenges over both. Figure 1.1 presents an overview of why hopping robots can be beneficial, and applications for such systems. With Benefits including: an ability to traverse rough, challenging, gap filled terrain unlike terrestrial systems, being more efficient when compared to typical flight thanks to periods of energy storage, they boast agile design and control to survive harsh impacts, maneuver freely in air and be able to rapidly change direction, and finally they can have fewer components to typical systems meaning lower cost and fewer failure modes. These systems can then be applied in a wide range of applications, from search and rescue operations in unknown terrain to mapping and exploration of new spaces. These systems can also be utilized on non-Earth locations, enabling rapid traversal of an unknown planet with minimal energy required. Lastly, hopping robots can spark excitement and fun in entertainment forms, bounding great heights, bringing fantasy into reality, much like work by Disney and their Stuntronics robots bringing superheroes into reality [14].

1.2. Research Goals

Hopping locomotion in robotics is a research topic that has a solid foundation of existing work, with a substantial number of areas for further investigation. The goal in this work is to further expand upon the current understanding and capabilities of hopping systems, concentrating on high-performance hoppers which have had minimal study. Regions with limited/no presented research that were studied include: evaluating system and control efficiency for hopping, especially high-performance hopping, developing a state estimation methodology with only inertial measurement inputs enabling height control, and incorporating an alternative non-linear controller to improve system performance.

To evaluate efficient design and control for a hopping platform, a robot capable of operating in a wide energy envelope is necessary. Higher energy levels force the system to

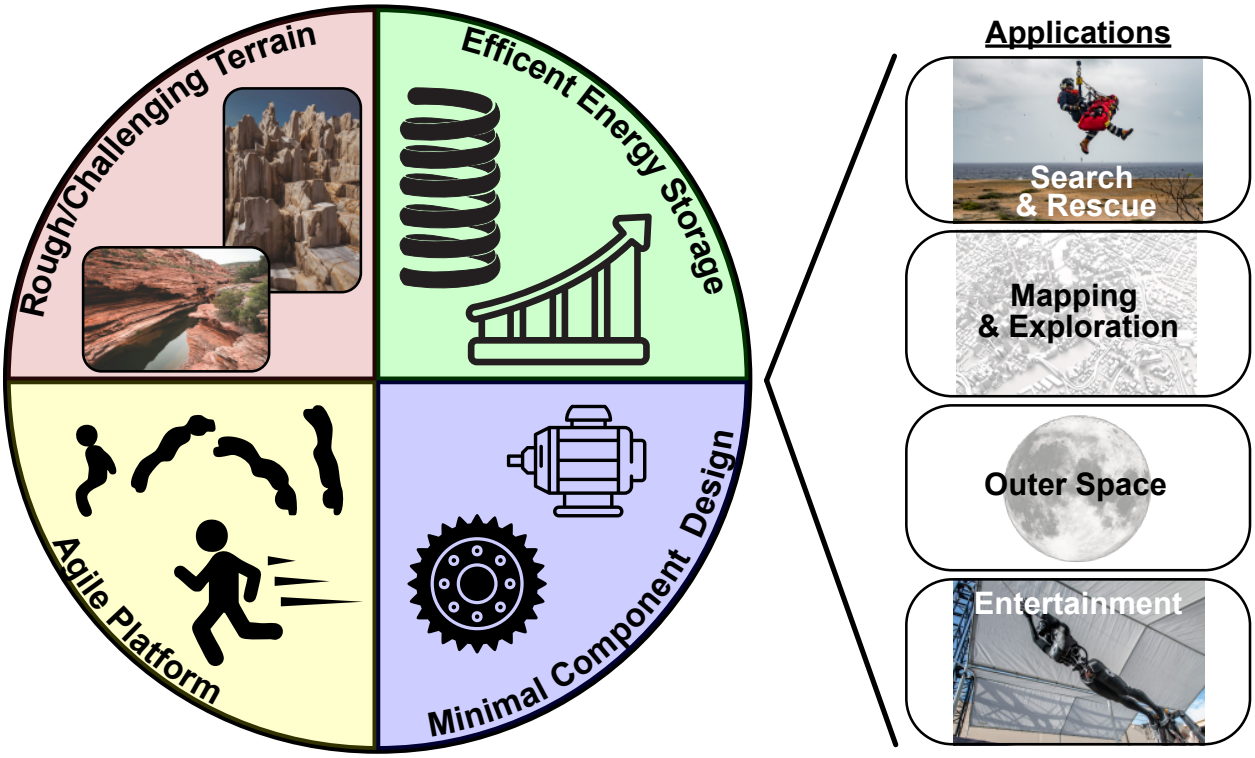

Figure 1.1: Benefits & applications of hopping robotics. Being usable in rough, uneven, or gap-filled terrain, where standard terrestrial systems may struggle. With increased energy efficiency, storing energy from one hop to the next which lowers the required input energy when compared to quadrotor systems. They can be highly agile, traversing a space fast and able to change motion directions by varying how they land. And using far fewer components in terrestrial systems, lowering cost and failure modes. This enables a wide use case, from search & rescue to mapping and exploration, given higher efficiency, lower cost, and high agility. To more extreme uses like non-earth bodies where they could traverse rapidly, assuming they can operate & control in that environment. Lastly, they can serve to entertain by mimicking what we see in film or cartoons.

experience higher system dynamics (both in terms of speed and force), meaning both the design and control method must be robust. Design aspects such as energy storage, leg design, and mass distribution will be studied both in simulation and experimentally to establish new design groundwork for higher-performance systems. The methods of control will be varied to optimize system efficiency and understand the impact it has on hopping performance. Research on a system is only as valuable as how generalizable it is; as such, having a high-performance system capable of up to 4m hopping will push design and control aspects to their limits and enable other platforms within our energy span to utilize our methods. Overall, presenting careful consideration of survivability versus efficient design, alongside generalized methodologies any hopping platform can use to improve efficiency.

Due to the relative infancy of hopping platforms, a majority of robots utilize external motion capture systems or human input for control. On-board state estimation would enable autonomous operation and allow for rigorous testing in non-lab environments. Currently, state estimation has been displayed for ballistic systems; however, these methods fail either as drag effects increase with higher energy hops or as the aerial phase sees additional thrust of forces. Establishing an estimation strategy that works regardless of the system's acceleration profile can greatly aid in robot development, as the strategy will work for any hopping style. This work develops a means for onboard estimation that is not only reliable enough for system control but can also be easily applied to any system, requiring only accelerometers.

Lastly, further research into the means of controlling the hopping systems is necessary to boost system optimization. Most hopping robots rely on PID-based control techniques, operating well for lower dynamic systems with minimal angular deviation. However, as hopping expands into more nonlinear and larger horizontal, vertical, and orientation-based trajectories, the linear PiD stability and controllability degrade. Applying alternative control schemes, like the Lyapunov controller, which can guarantee system stability and is model-based, would allow a larger dynamic region to be tested. Such a control scheme will allow more advanced trajectories to be tested, and in turn, advance control strategies to improve

efficiency and accuracy. This work will walk through the development of going from a PiD to a Lyapunov controller for a hopping system, and show functioning operation to higher dynamic disturbances and maneuvers. With future work on the controller being used to test advanced trajectories.

1.3. Contributions

1.3.1. Hopping robot development

A novel hopping robot platform was developed, which utilizes a monopedal leg and a central body that houses all control components. Control is achieved with a quadrotor-based motor configuration, allowing both orientation control and energy input. It is capable of hopping up to a tested 4 m vertical height and speeds over 7 m/s, with simulations showing capabilities up to 30 m assuming adequate energy storage. The system had two iterations to refine design aspects, make more permanent electronic/control components, and improve capabilities with data logging and minor sensor modifications.

Outside of just the robot itself, various methodologies for working with the robot were generated. An analysis of the robot motor force output was conducted, which serves to display methods to perform the analysis and how it can improve robot functionality. A data compression system was developed to minimize data packet size, enabling real-time data saving for each control loop iteration on board, without impacting overall system performance. Lastly, methods for experimentation are presented, including proper safety mechanisms for the robot and personnel. Overall, presenting the methodologies for repeatable and consistent experimentation with a high-performance hopping robot.

1.3.2. Design and control for high-performance hopping

Within the scope of hopping robot systems, there have been only a handful of in-depth studies on these platforms. To this end, the topic area of system control efficiency for high-performance hopping platforms was chosen, given the lack of research. Amongst other

hopping platforms, the current height record is around 1.63 m, with more systems operating in the 1 m height range [1]. A consequence of these lower heights is that the systems have less energy at touchdown, allowing for a larger misalignment tolerance, which results in smaller and slower perturbations at liftoff, making stabilization and control easier. To truly explore hopping systems, greater heights and energies are required, with our study producing a stable system capable of up to 4 m heights and 7 m/s speeds. This study saw the development of a 2D model that closely resembles the actual system performance. In addition to concentrating on understanding the impact that leg design, energy storage, mass distribution, and control have on the overall platform. The results identified that for hopping systems with mid-air thrust, a larger thrust over a short time upon lifting off generates higher efficiency when compared to less thrust for more time. Overall, this work shows a novel, efficient, and high-performance hopping robot and control methodology capable of operating at energy states seen by no other hopping platform to date.

1.3.3. Vertical state estimation via optimized Kalman filter and minimal sensor suite

One topic that has shown minimal development for hopping platforms is in the scope of state estimation. Unlike other locomotion modes, hopping experiences instances of a harsh impact and large accelerations during the stance phase. Due to the high-speed nature of the stance phase (especially at points of impacts for touchdown and liftoff), sensor aliasing is nearly guaranteed, meaning lost data and, in turn, an inability to use the data for direct state estimation. Here, a methodology of taking a Kalman filter, whose parameters have been optimized via collected training data and a genetic algorithm optimization method, was produced. This methodology eliminates the concern of lost data due to aliasing thanks to the training that adjusts for lost data. Through this work, the estimation technique is tested first in simulation to find an optimal methodology. Simulation was also used for both performance evaluations and for determining what parameters and sensor acquisition rates are required

for optimal performance. The net result shows a robust estimation methodology, sufficient for data capture and hopping height control for a high-performance hopping system, utilizing only accelerometers as data sources. Making a platform that can be easily implemented on any robot system and provide robust control.

1.3.4. Hopping robot non-linear controller implementation for higher system dynamics and disturbances

As previously discussed, to date, hopping robots have mainly remained at lower energy states and jump heights around 1 m. This enables simpler control schemes as the touchdown misalignments do not generate significant angle changes during stance and, in turn, less liftoff disturbances that need correcting. With the increased energy states explored in prior sections for this work, the focus has been on vertical hopping, which lessens the overall dynamics and, in turn, requires less controller effort and stability. This study enters dynamic regions where the base controller, a linear PiD, is less effective due to no stability guarantee, and uncertain operability at larger orientation angles and horizontal speeds. An alternative Lyapunov controller will be implemented, being model-based, non-linear, having guaranteed stability and energy convergence. Experiments have shown the ability to recover from larger disturbances and horizontal position variation. This sets up a controller that can be used for even more rigorous experimentation and trajectory analysis testing for high-performance hopping robots.

1.4. Dissertation Organization

This dissertation follows a structure of opening with the motivation, goals, and broad outcomes of the research in Chapter 1. Followed by Chapter 2, detailing the developed robot, the multi-modal monopedal hopping robot or MultiMo-MHR. Including technical specifications and platform revisions. The chapter contains a background on hopping robots and hopping locomotion from the early concepts to the most recent works. Possessing a

breakdown of the robot as a whole, along with general specifications for system parameters and the experimental setup.

The central studies conducted are presented in Chapters 3, 4 & 5. Chapter 3 presents a design and control methodology to understand system efficiency. This was achieved by studying high-performance hopping up to a maximum hop height of just over 4 m and speeds up to 7 m/s. The chapter begins by introducing current hopping robots and their capabilities, highlighting a gap in knowledge on topics like system energy and efficiency, which this chapter serves to fill. A full breakdown of a 2D, 4DOF model and accompanying simulations/discussions on efficient system control of thrust input is displayed. This leads to an overview of the robot platform and methodologies for the system's energy storage, leg design, and controller design. Culminating in an experimental analysis exploring system control for efficient hopping and understanding how energy can be accumulated, dissipated, and maintained in a hopping cycle.

Chapter 4 presents an optimized Kalman filter estimation methodology which can be used for hopping systems. The chapter opens with introducing current state estimation in hopping robots, which has been minimally explored, and the challenges associated with doing so. Specific discussion on the robot controller utilized in the estimation and the associated sensors/electronics used to achieve the results. The estimation technique is composed of two main sections: a hopping phase estimator, used to detect which phase in the hopping cycle the robot is in, and then the vertical state estimator itself. The vertical state estimator saw several implementation methodologies, tested in simulation, and used the best on the robot. An in-depth discussion on the optimization strategy employed for the estimator, as well as a simulation on how the sensing rate impacted the estimator's performance, was conducted. The chapter wraps up with the experimental procedure and quantifying the estimator's performance and the impact it has on hopping height control.

Chapter 5 showcases a new control method on the developed hopping robot. Opening with introducing current control methods that hopping robots utilize, with a discussion on

the benefits and drawbacks of the methods. Then the chapter examines the new controller implemented, a Lyapunov controller, to explain its selection and how the controller equations were derived for the robot. The new controller is then implemented onto the robot with any modifications to the control strategy discussed. Experiments showed testing with the original PiD controller and new Lyapunov controller for trials with higher disturbances, and detailed stability and performance capabilities for both controllers. With a subsequent experiment to showcase the controller's ability to hop between multiple horizontal positions.

Chapter 6 concludes this thesis. Summarizing the overall thesis, then presenting avenues for future work, and showing current publications with the hopping robot.

2. MultiMo-MHR: robot & setup overview

2.1. Introduction

In this chapter, the developed hopping robot, named the multi-modal monopedal hopper or MultiMo-MHR for short, is detailed. The platform possesses a monopedal leg with a quadrotor-based control scheme; this combination sets up a design that minimizes losses while necessitating reliable/stable control to ensure stability during hopping locomotion. The name originates from the robot platform developed by Dr. Matthew Woodward, the MultiMo-Bat, a multi-modal (jumping-gliding) robot which took inspiration from vampire bats [15]. For this work, the robot is multi-modal in that it can locomote via hopping and/or by flying thanks to four quadrotor motors, enabling orientation control and thrust input. The MHR tag stands for the robot design style, possessing a single monopedal leg to hop. The system can begin hopping with either a short burst of motor thrust or a limited flight time, either of which provides an initial level of energy to start the hopping locomotion mode. The hop cycle is comprised of three phases: drop, stance, and rebound. Where the drop phase uses motors for reorientation, stance sees no motor activity. At the same time, energy is stored from the drop, and the rebound phase sees motors engaged to provide thrust and reorientation/stabilization to maintain or vary the hopping trajectory.

This chapter serves as an overall robot walkthrough. Beginning with the background of hopping robots and design goals/inspiration for the MultiMo-MHR. Then, leading into robot specifications with discussions on the data saving and motor capabilities. Introductory discussion on the robot control is provided, but will see further development in later chapters as certain controller aspects were varied over the platform's development. The chapter then concludes with details on the testing environment for the robot. This is all done to provide

a clear picture of the robot and its capabilities, with subsequent chapters exploring specific goals in more detail.

2.2. Background

Hopping locomotion-based systems can be traced back to as early as 1967 by NASA with a set of ideas and principles for a lunar pogo stick, a theoretical lunar vehicle which utilizes hopping to traverse the lunar surface [16]. A subsequent theoretical design continued these ideas in 1969, proposing alternative lunar vehicles which could conserve fuel and achieve higher travel speeds than conventional lunar vehicles through the use of hopping [17]. While these ideas never made it to production for NASA missions, the idea of a hopping-based locomotion system was established, and by the 1980s, research and development in hopping robots began.

In 1984, Marc Raibert developed a 2D robot, then shortly after, an improved 3D hopping robot (Figure 2.1a). These works jump-started interest in hopping locomotion research, especially in the field of robotics [18–20]. Raibert's hopper centered around controlling horizontal velocity with leg placement at touchdown, controlling body orientation via motor torques in stance, and controlling the hopping height via a linear actuator in the leg. While this hopper did not experience high vertical heights, it displayed high stability and controllability for in-place and horizontal motion [19].

As time progressed, so did hopping robots. As a note, this portion will focus on monopedal (i.e., a single rigid leg) based robots; however, for some further insight on other jumping and hopping platforms, the following literature reviews can add some information [21–23]. Focusing on rigid-legged monopod hoppers, we advance to 1993 with Lebaudy et al., with a vertically constrained hopper used to focus on testing hopping height control [24]. Around the same time ranging from 1993-1999 with Gregorio et al. and Ahmadi et al. came the ARL Monopod I and II being a planar constrained hopper concentrated on horizontal hopping/running control based on Raibert's work (Figure 2.1b&c) [3, 4, 25–27]. All of

the aforementioned works have one major detraction for any robot system, which is that they require some external connection, either via boom arm, rigid frame, or a tether for power/control (a tether for general testing safety is not counted). The 3D bow legged hopper developed by Zeglin et al. and later updated by Jean-Christophe Zufferey brought a fully self-contained hopping platform into reality; this marking a shift in hopping robots as now all design/technical choices carry additional literal weight and thoughts of balancing a systems efficiency with its capabilities (Figure 2.1d) [5, 28].

Advancing once again, getting closer to the present day, with hopping robot capabilities improving, higher hops, and overall better controllability and stability. Two robot platforms originating around 2017, which highlight this are the LEAP and Salto/Salto-1P platforms. LEAP or the linear elastic actuator in parallel combines a voice coil actuator and gimbal setup to achieve a continuous in-place hopping locomotion for ~ 7 seconds (Figure 2.1g) [8]. The work with the robot platform Salto (Figure 2.1e) and its improved second version, Salto-1P (Figure 2.1f), boosted hopping robot capabilities and analysis immensely. While not a monopod, using a parallel-elastic leg mechanism, Salto presented a small (100 g) robot capable of fully untethered single leaps up to 1 m and wall jumps [6]. Salto-1P then brought Salto into a continuous hopping condition up to 1.25 m while retaining its lightweight (98 g) form [7]. Salto-1P additionally showed extreme precision in control horizontally and vertically, as well as orientation estimation, enabling control outside of a lab space [29–31].

One constant of all aforementioned systems is the means by which energy is input into the system. That is being done within the stance phase between hops. While the benefit of this is that it can allow for relatively simple energy input means, it comes at the cost of occurring over a very short time, meaning it cannot have much of an impact from a control perspective, being mainly used for energy addition. Taking Salto/Salto-1P as an example, it stores energy while in the air via a spring and releases during stance to maintain hopping; however, it requires an additional flywheel and sideways thruster pair to control body orientation [6, 7]. This limitation led to the idea of changing energy addition to occur in the aerial phase,

beginning the development of monopedal hoppers with quadrotor-based attachments for aerial-based energy addition and control. The earliest showings of such systems began in 2022-2023 with robots PogoDrone (Figure 2.1h) and this robot, the MultiMo-MHR; both showing untethered continuous hopping systems with PogoDrone seeing best stability at hops around 0.5 m and the MultiMo-MHR reaching just over 4m hops, a unseen height for any other hopper to date [9, 32]. Similar design based hoppers then were seen through the untethered systems PogoX (Figure 2.1i) and Hopcopter (Figure 2.1j), as well as a tethered micro insect-scale continuous hopping robot by Hsiao et al. (Figure 2.1k) [1, 10, 11, 33]. Hopcopter displayed a high level of controllability and the second highest hop height of 1.63 m, with the ability to hop stably outside [1]. And PogoX presents a far heavier system, with an expanded computation and sensing suite ultimately showing solid vertical and horizontal velocity estimation and orientation state estimation with an extended Kalman filter at the cost of hopping only $\sim 0.6m$ [10, 33]. These platforms enable a new control paradigm, no longer being limited to a purely ballistic trajectory.

2.3. Design Goals and Inspiration

The design of the MultiMo-MHR is that of a single monopedal leg and main body with a single prismatic joint. Joining the two components are a series of rubber bands, which enable energy storage when the system is in the stance phase of a hopping cycle.

Design inspiration was based on the 2/3D monopedal hopper of Raibert et al. as well as the work of Salto/Salto 1-P; utilizing a single monopedal leg, with a fully self contained control/power system to enable hopping locomotion [6, 7, 18–20, 29–31, 34]. A single leg was desired to minimize the energy lost at the largely inelastic collision between the leg and the surface during ground contact. For this same reason, the system design was focused on minimizing mass addition of any kind (especially on the leg) to reduce any possible additional energy loss. The monopedal nature reduces system complexity greatly, enabling minimal losses while facing the added challenge of less natural stability, necessitating reliable

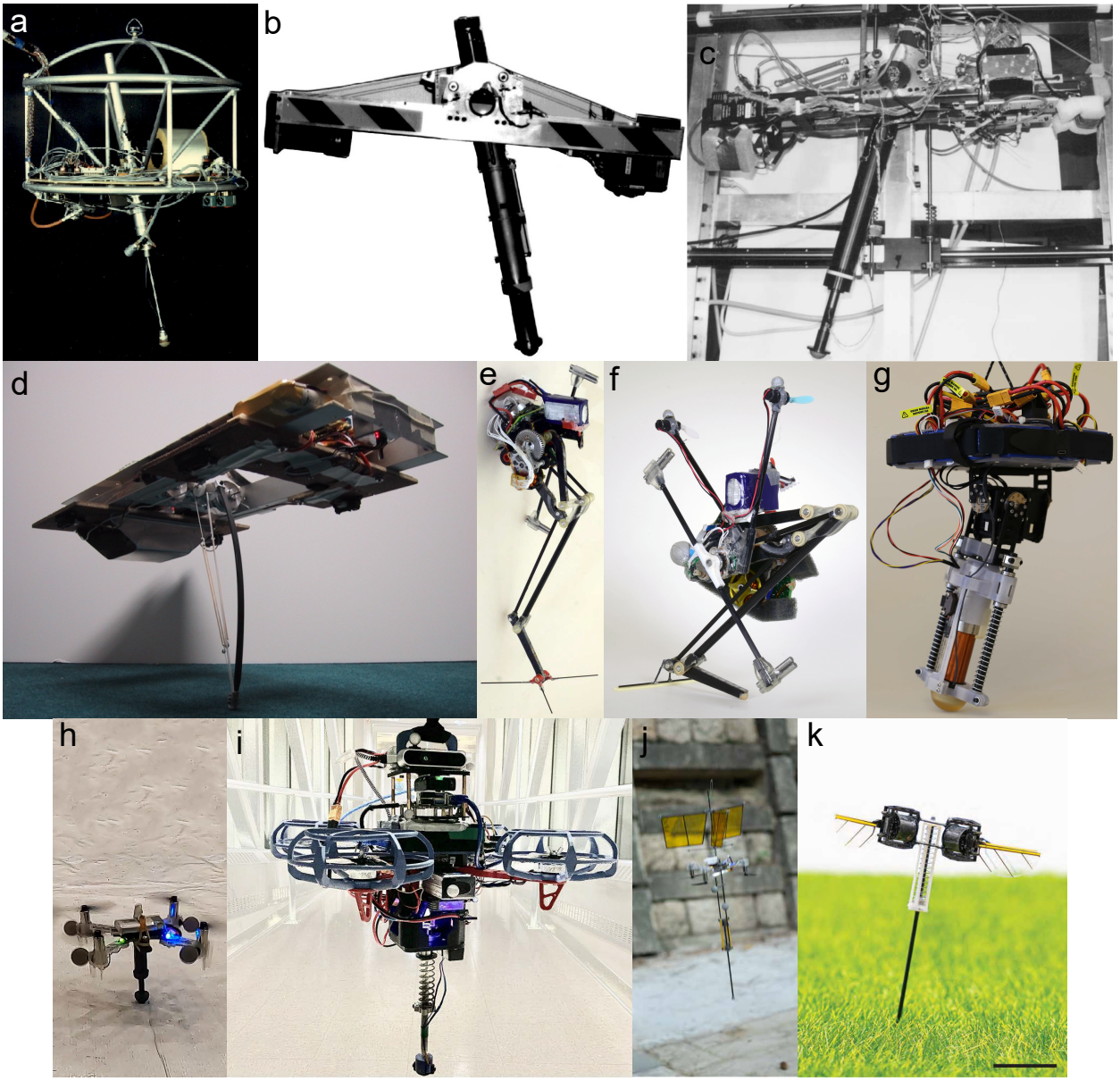

Figure 2.1: Compilation of Monopedal Hopping Robots. Image sources: a (Raibert 3D Hopper [2]), b (ARL Hopper I [3]), c (ARL Hopper II [4]), d (3D Bow Leg Hopper [5]), e (Salto [6]), f (Salto-1P [7]), g (LEAP [8]), h (PogoDrone [9]), i (PogoX [10]), j (Hopcopter [1]), & k (insect scale hopper [11]).

and robust control.

System design/control also took inspiration from quadrotor/drone systems. These platforms are capable of high levels of control authority and precision while remaining relatively lightweight platforms [35, 36]. The addition of quadrotor motors enables two main benefits: stability control and energy addition. Past hopping systems tend towards flywheel-based mid-air stabilization (Salto/Salto 1-p), or leg reorientation and stance-based applied forces/torques (Raibert Hopper, ARL Hopper). These paths ultimately limit the advancement of hopping systems as flywheels solely enable reorientation at the cost of added mass, and stance-based methods become less achievable with higher hop heights, and in turn higher ground forces requiring larger motors to resist. Ideally, a control means would both enable control and the potential to add energy while not adding significant mass to the system. Thus, a quadrotor motor approach was devised, enabling control authority while in the air (and stance if desired) and allowing for energy addition as thrust if desired.

Additional design constraints included limiting the computational power of the system with low-cost micro-controllers and sensors, this was done to ensure controllability and stability is possible on a fundamental level with the future ability to add more to boost robustness or capabilities.

2.4. Design Overview

The MultiMo-MHR is a two-body system: a leg and main body, as shown in Figure 2.2. For specifics on the design methodology of the robot, see chapter 3.4. This section will provide insight into a complete breakdown of the system components and specifications for motor capabilities, on-board data saving methodology, as well as the fundamentals for controlling the robot.

2.4.1. Robot General Specifications

The MultiMo-MHR platform is a small, light-weight platform with a high specific energy, enabling hopping to heights of at least 4 m. Two revisions (named R1 and R2 for revision

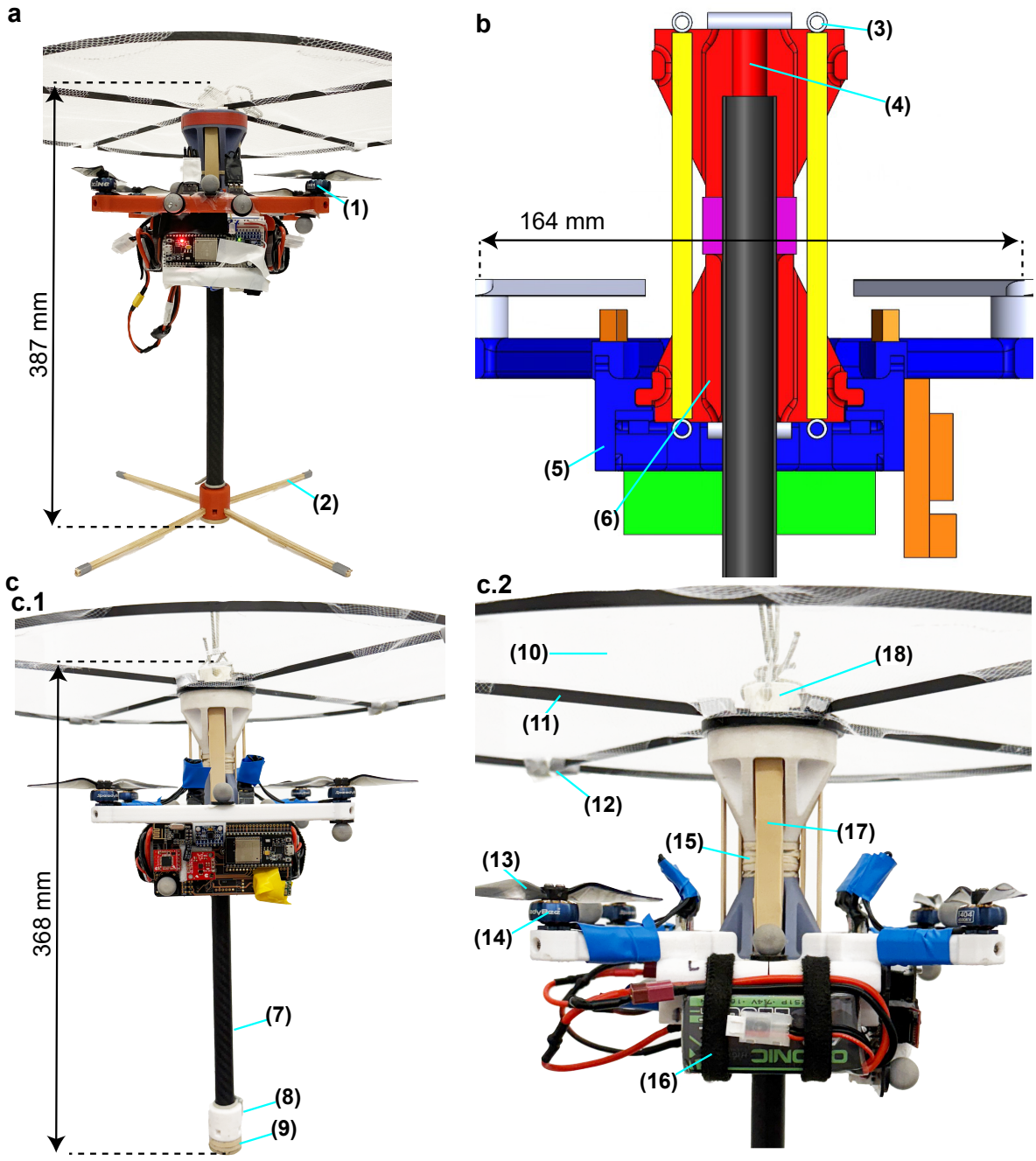

Figure 2.2: MultiMo-MHR Overview: a. R1 version, b. CAD internal cutaway view, c: R2 version, c.1: R2 front profile, c.2: R2 side profile

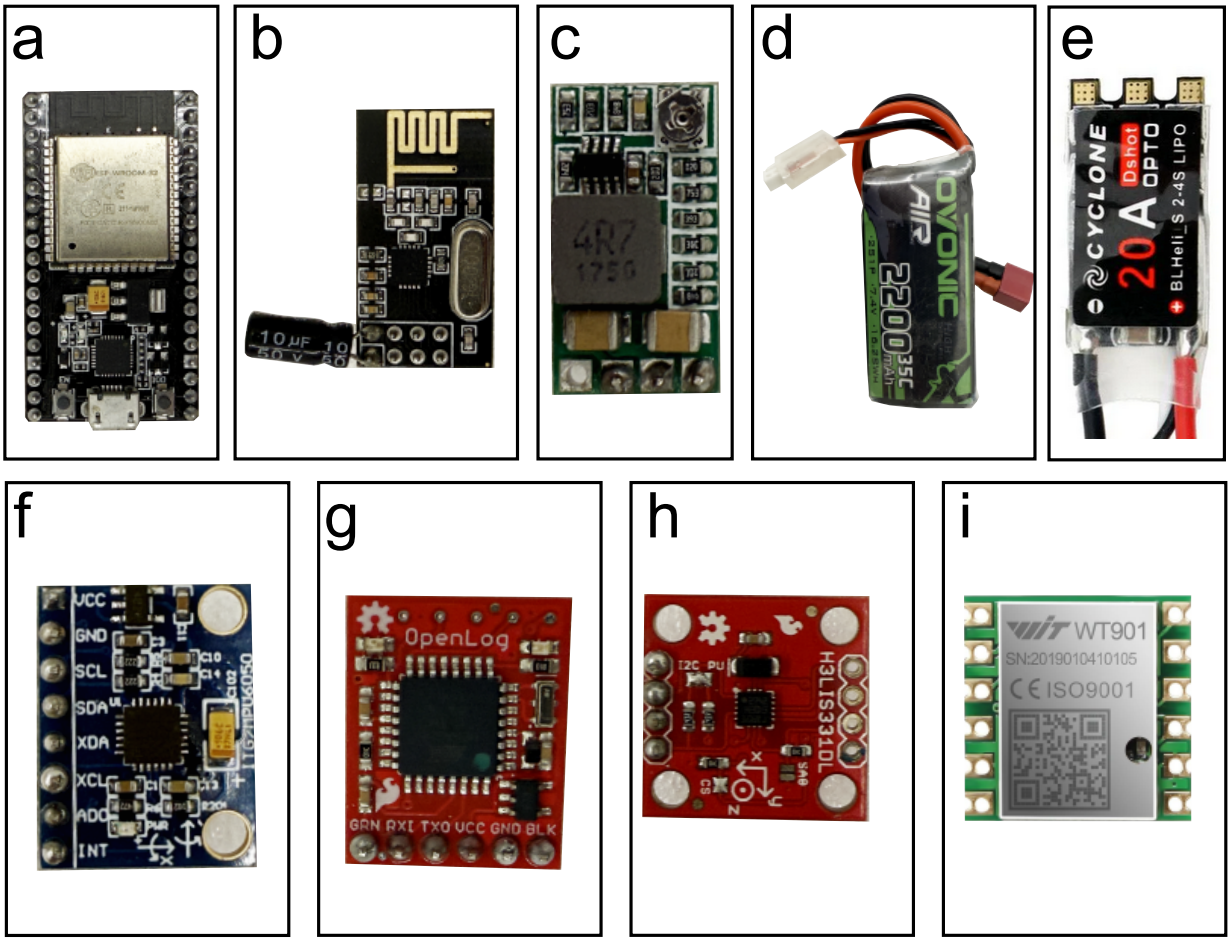

Figure 2.3: Robot electronic components. a: ESP32, b: RF transceiver(nRF24L01), c: voltage regulator, d: battery(2200 mAh, 35C, 2S, 7.4 V), e: electronic speed controller(20 Amp), f: 9-DOF IMU (MPU6050), g: data logger Spark Fun Open log, h: 3-DOF accelerometer (H3LIS331DL), i: 9-DOF IMU (WT901)

1 and revision 2, Figure 2.2 a&c, respectively) were produced through the research, with most modifications occurring on the electronic components to clean up and improve the functionality.

The robot is comprised of a mix of 3D printed (ABS plastic) and off-the-shelf components. As a whole, the robot stands 387 mm & 368 mm tall for R1 and R2, respectively, with a width of 164 mm at the widest portion. Additionally, the system weighs 697 g & 684 g for R1 and R2, respectively.

The robot has two main parts, the leg and the body, connected by rubber bands. A complete overview of the robot and versions is presented in Figure 2.2, with all components listed in Table 2.1 with identifiers to match between the figures. The leg design saw little modification between R1 and R2. The majority of the leg is a carbon fiber tube, with 3D printed ends on either side. The upper end top is used to connect the leg to the body, using a set of four aluminum rods in channels that the bands wrap around, before using a similar system on the body to bridge the two parts. As well, the upper portion contains a netted bumper, comprised of thin carbon fiber strips joined by 3D printed caps; the bumper prevents a safety rope from colliding with the propellers while in operation. At the end of the leg is a 3D printed foot with a flat foot design. This foot has two versions between R1 and R2. Between both versions, the base foot uses the same 3D printed design. Added to the feet are rubber pads at the end of the foot, reducing the leg impact force by extending the leg compression distance. R1 has a single rubber pad on the foot, and taking inspiration from Salto, it has four wooden stabilizers enabling a resistive torque in stance to prevent robot rotation. The stabilizers come at the cost of only allowing vertical straight hopping to be tested, necessary in the early stages of the robot but later removed for R2. The foot in R2 uses the same base 3D print, but removes the foot stabilizers (improved control and a desire to test horizontal motion) and adds two extra rubber pads at the end (totaling 3) to lessen the overall impact at touchdown.

The second robot part is the body. The body has three 3D printed parts; an inner

print rides along the leg tube and has slots for the aforementioned aluminum rods and rubber bands. Next are two identical halves, which hold the power, control, and sensor systems. Between R1 and R2, the body saw little variation aside from electronic component simplifications, which will be detailed in the next section.

2.4.2. Technical Specifications

The MultiMo-MHR is comprised of a number of electronic components, shown in Figure 2.3. It is controlled from the ESP32 (Figure 2.3a) with a dual-core 240 MHz microcontroller. The chip was compiled and programmed with the Arduino IDE. This microcontroller does all data processing and controller calculations/commands. The sole external processing done is the acquisition of position and orientation data from a Vicon motion capture system, however only position and angles are obtained from the system, with any additional calculations or rotations being done on the robot; this ensures a seamless transition with addition of state estimation or translation/orientation sensors o make the system not dependent on the motion capture system.

The system's sensor suite is comprised purely of IMU's. R1 utilized a MPU6050 (6-axis, 1000Hz, ± 16g, ±2000 deg/s, Figure 2.3f) and a WT901 (9-axis + tilt sensor & DMP, 200 Hz dead reckoned to 1000Hz, Figure 2.3i). While R2 utilized the same MPU6050 and a Sparkfun H3LIS331DL(using ± 100g, but can do up to ± 400g, Figure 2.3h). In the sensor suite for R1, the MPU6050 accelerometer is used for state detection (i.e., touchdown or lift off), while the gyroscope is used for control and to dead reckon between angle readings from the WT901, as the WT901 operates at 200 Hz and the MPU6050 at 1000Hz. Alternatively, R2 uses a combination of the MPU6050 and H3LIS331DL accelerometers to capture the acceleration profile better; the combined accelerometer and MPU6050 gyroscope data are then used for state detection and state estimation in the controller.

An external motion capture system (Vicon, Vantage v5) is utilized for both R1 and R2. The system serves two purposes: to provide ground truth measurements (captured at 600

Table 2.1: MultiMo-MHR Mechanical Components

Item Name	**Material**	**Dimensions**	**Robot ID**	**Figure Match**
Main Leg Tube	Carbon Fiber	ID = 16 mm OD = 14 length = 330 mm	R1 & R2	Fig 2.2 (7)
Foot	3D Print - ABS		R1 & R2	Fig 2.2 (8)
Foot Impact Cushion	Natural Rubber 40A	Thick (each) = 3.175 mm Num Pads: R1 = 1, R2 = 3	R1 & R2	Fig 2.2 (9)
Upper leg tube attachment	3D Print - ABS		R1 & R2	Fig 2.2 (4)
Netted topper attachment hub	3D Print - ABS		R1 & R2	Fig 2.2 (18)
Netted topper netting	Fine mesh netting		R1 & R2	Fig 2.2 (10)
Netted topper frame	carbon fiber	Cross Beam: L,W,H = 200, 5, 1 mm Circular: L,W,H = 343, 5, 0.5 mm	R1 & R2	Fig 2.2 (11)
Netted topper frame connectors	3D Print - ABS		R1 & R2	Fig 2.2 (12)
Rubber band connectors	Aluminum rod	ID = 3.18 mm OD = 6.35 mm length = 25 mm num = 8	R1 & R2	Fig 2.2 (3)
Rubber bands	Natural crepe rubber	width = 13mm length = 89 mm	R1 & R2	Fig 2.2 (17)
Inner body tube component	3D Print - ABS		R1 & R2	Fig 2.2 (6)
Body main structure	3D Print - ABS		R1 & R2	Fig 2.2 (5)
Leg-body impact cushion	Natural Rubber 40A	Thick (each) = 3.175 mm Num Pads: R1 = 1, R2 = 5	R1 & R2	Fig 2.2 (15)
Wooden foot stabilizers	wood	L,W,H = 127, 4.11, 4.73 mm	R1	Fig 2.2 (2)

Table 2.2: MultiMo-MHR Electrical Components

General Name	**Specific name**	**Capabilities**	**Robot ID**	**Figure Match**
Main Controller	ESP32	dual-core 240 MHz	R1 & R2	Fig 2.3 a
Low-g IMU	MPU6050	9-DOF, ±16g, ±2000 rad/s,±1000 HZ	R1 & R2	Fig 2.3 f
High-g Accelerometer	H3LIS331DL	3-DOF, ±400g,±1000 HZ	R2	Fig 2.3 h
Orientation sensor	WT901	9-DOF, ±16g, ±2000 rad/s,±200 HZ	R1	Fig 2.3 i
RF Transceiver	nRF24L01	2.4GHz	R1 & R2	Fig 2.3 b
Voltage regulator	N/A	5V regulator, input 5-24V	R1 & R2	Fig 2.3 c
Data logger	SparkFun OpenLog	uses 64Gb micro SD at baude = 500000	R2	Fig 2.3 g
R1 motor	Iflight XING2 1404	4600 kv	R1	Fig 2.2 (1)
R2 motor	SpeedyBee 1404	4500 kv	R2	Fig 2.2 (14)
Motor Propellor	N/A	4030 propellor	R1 & R2	Fig 2.2 (13)
Electronic Speed Controller	N/A	20 Amp, DSHOT	R1 & R2	Fig 2.3 e
Battery	N/A	2200 mAh, 35C, 2S, 7.4 V	R1 & R2	Fig 2.3 d, Fig 2.2 (16)

Hz), and to provide data to the robot for control (at 100 Hz). R1 purely used translation data from the system, while R2 used both translation and orientation to boost robot precision and accuracy while testing. Data from the motion capture system is sent to the robot via an RF transceiver (nRF24L01, Figure 2.3b).

To aid in system initialization as well as for emergency stopping the system, a dual-shock 4 Bluetooth controller is utilized. The controller can be used to both control the system's movements, and to shut down system motors to end a test or stop the system in the event of an emergency. The controller does not directly talk to the robot, but instead an external ESP32, which both receives Vicon information and the Bluetooth controller data and transmits it using the aforementioned RF transceiver.

The original robot (R1) had all components attached via a breadboard for easy prototyping and configuration during initial testing. R2 swapped that with a fabricated PCB, which aided in reducing the overall system weight, and ensured components were more rigidly attached. Added to R2 was a method of data logging. A Sparkfun OpenLog (Figure 2.3g) enables desired data to be saved to a micro SD card for later processing; this entire process is detailed in the next section.

Lastly, to control and provide thrust, four quadrotor motors (iFlight XING2 1404 for R1 and SpeedyBee 1404 for R2 with 4030 props) are utilized. They are powered by a 14.7V battery pack (two 7.4V, 2s, 2200mAh, 35C batteries in series, Figure 2.3d). The motors are controlled from four ESCs (one per motor, Figure 2.3e), each rated at 20 Amps, with the commanded drive signal being sent from the ESP32.

2.4.3. Data Compression and Saving

To monitor processes and data from the robot, a method of internal data collection is required. The MultiMo-MHR, version R2, possesses a data logger (SparkFun OpenLog in Figure 2.3g), and here the process required to save the data is detailed.

The MultiMo-MHR utilized a data logger for storage of content including: current data

frame from the motion capture, state estimated position/velocity, accelerometer & gyroscopic readings, and time, to name a few. Data, like the current motion capture frame, enables a way to align the robot's measured data to the ground truth motion capture data for comparison and data analysis. Due to the limitations of the ESP32, data could not be stored in RAM (for any sufficient length of time), and even if packages of data could be buffered, then sent, the sending would slow the control code. The MultiMo-MHR operates at a control loop rate of 840 Hz with the ultimate specific data logging methodology, which will be detailed here. However, an important note is that the details to follow are needed due to the large amount of data that is desired to be logged; if minimal saved data is required, the process can be omitted.

The specific form of data compression or encoding used is base-N (or radix) encoding. This pipeline serves only to shrink the size of saved data into the smallest form to speed up the data process of writing data to an onboard micro SD card (micro SD XC card, 64 GB, C10 U3 V30), not for data security or any form of data saving redundancy. The used means of connecting an SD card to the MultiMo-MHR was with a device called the SparkFun OpenLog, through testing it was found a base 91 set of characters could be used to convert from the standard base 10 digits. The specific characters used for the base 91 conversion include the 26 base letters A-Z in lower and upper case, digits 0-9, as well as a variety of mathematical and punctuation symbols such as but not limited to $!\#\%\&()*+,-./\ :;<=>?@[]$. The OpenLog could not support other character-based languages or letter accents, which can aid in further data compression. To optimize the compression further, the desired data was concatenated into longer integer numbers; this was done as the larger the integer number converted to base 91, the more efficient the conversion. For example, a 4-digit long integer sees a 25% compression from length four digits to length of 3 digits in base 91, while a 19-digit long integer sees a 47% compression to 10 digits in base 91.

Ultimately, the robot saw 23 unique data values ranging from sizes of one integer to six integers to ensure adequate significant figures; with the most desired data for a single

robot loop being 82 total characters of data. With the aforementioned base 91 compression methodology, the 82 characters were reduced to 44 characters. Testing of the robot found that the maximum number of characters that could be saved per loop before degrading the control loop time was around 50 characters.

The decoding of the saved data was done off the robot on a separate PC for data processing once a trial ended. As no error correction means were applied, like a checksum, the data saving was not perfect, and occasional erroneous data was saved. As erroneous data could not be recovered, it was removed based on two checks. One check was on the length of the saved character string for any loop, as each was of known length (like the 44 saved characters previously mentioned), any line of different length was removed. Additionally, certain measured values had known ranges, like the low g accelerometer values being between +- 16g. Saved data was also cross-checked against the measured motion capture data to ensure consistency in data such as acceleration and gyroscopic rates. These combined checks ensured an accurate capture of data for use in data analysis.

2.4.4. Motor force output

In order to ensure proper control of the system, an understanding of the force output of the motor-propeller combination is required. The electronic speed controllers (ESCs) receive a commanded duty cycle ranging from 0% to 100% (indicated as pulse widths modulations of 48 to 2048 ms, respectively). The robot itself cannot go past the 80% duty mark as the current draw will exceed the limit (20 Amp) of the electronic speed controller.

A test rig was created to hold one motor-propellor pair. The system was powered by the same battery configuration as the robot, a 14.7V battery pack (two 7.4V, 2s, 2200mAh, 35C batteries in series) to ensure consistency. Battery voltage was measured with a multimeter (AstroAI DM6000AR), and an ACS712 current sensor was used to measure current draw. Motor commands were sent via an ESP32 to a 20-AMP ESC, then the motor and any sensors were recorded with the same ESP32.

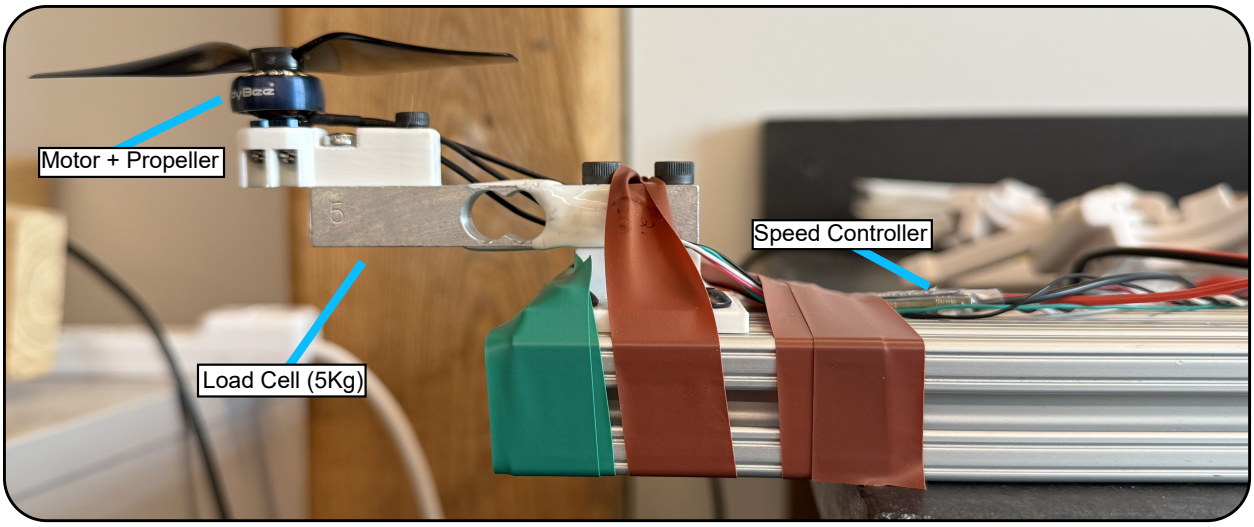

Figure 2.4: Motor load cell test setup, utilizing the peedyBee 1404 motor, 4030 propeller, and 5 kg load cell. Used to evaluate the relation between commanded duty cycle and outputted thrust, test results can be seen in Figure 2.5.

The testing rig, seen in Figure 2.4, consists of a 5 Kg load cell (with an HX711 load cell amplifier) and 3D printed components that mimic the motor mounting method used by the MultiMo-MHR and hold the test rig to a table. The motor-propeller setup was placed overhanging the edge of the table to avoid any surface effects of the thrust output.

Testing consisted of running the motor from 0% to 65% duty, ideally testing would go up to 80% duty however the measuring tools could not exceed 10 Amps (approximately achieved at 65% duty). The duty would increase in 5% increments every 2 seconds, then once at the apex, drop back to zero percent; this pattern was done four times for statistical variance. While running the load cell measurement, the multimeter voltage and current sensor data were all recorded.

The relation between the duty cycle and output force was the primary focus of these tests to aid in understanding the motor force profile and to assist in system control ultimately. Standard convention for the relation between the motor RPM and resulting force is that of $F \propto \Omega^2$; however, this is not guaranteed, and the relation can be varied based on the used ESC, motor, and propeller combination. For this reason, to ensure a known relation between a specified duty cycle and resulting thrust force, the setup should be experimentally

evaluated.

Testing enabled the generation of Figure 2.5, showing the specified duty cycle percentage as a function of the measured force. This indicates that above a duty cycle of 5%, a linear relation between the duty cycle and output thrust as the motors themselves do not take straight duty percentages but instead a specified pulse width specified in microseconds, where 0% duty is 48 microseconds and 100% duty is 2048 microseconds. As such, a fit equation was created to get the motor force output (F_{motor} in units of Newtons) based on the duty cycle (T_{duty} in units of microseconds); the fit equation is as follows:

$$F_{motor} = 0.00217 * T_{duty} - 0.26991 \tag{2.1}$$

$$\text{where } T_{duty} \geq 148 \text{ microseconds} \tag{2.2}$$

Given this finding, the robot was limited to have a minimum idling motor duty cycle of 5%. Additionally, with this knowledge, it is possible to convert a target force into the necessary duty cycle for the specific robot setup. This knowledge will later be implemented on the robot, see Chapter 5, to convert a controller output force into a motor-specific duty cycle to improve control.

2.4.5. Control Fundamentals

All robot control is done via a set of four quadrotor motors oriented around the body of the robot. Heavy inspiration from standard drones/quadrotors was taken for both the general design and control methodology [37]. Quadrotors have two typical motor configurations, cross(x) or plus(+) (visible in Figure 2.6 a & b respectively), which alter how the motors are situated about the front of the system. Between the two the cross configuration is more common as it moves the motors away from the front of the system; this slightly reduces drag when moving in the forward direction, and aids in lowering chances of arm breakage as falling will typically not land on a singular arm, however this comes at the cost of coupling

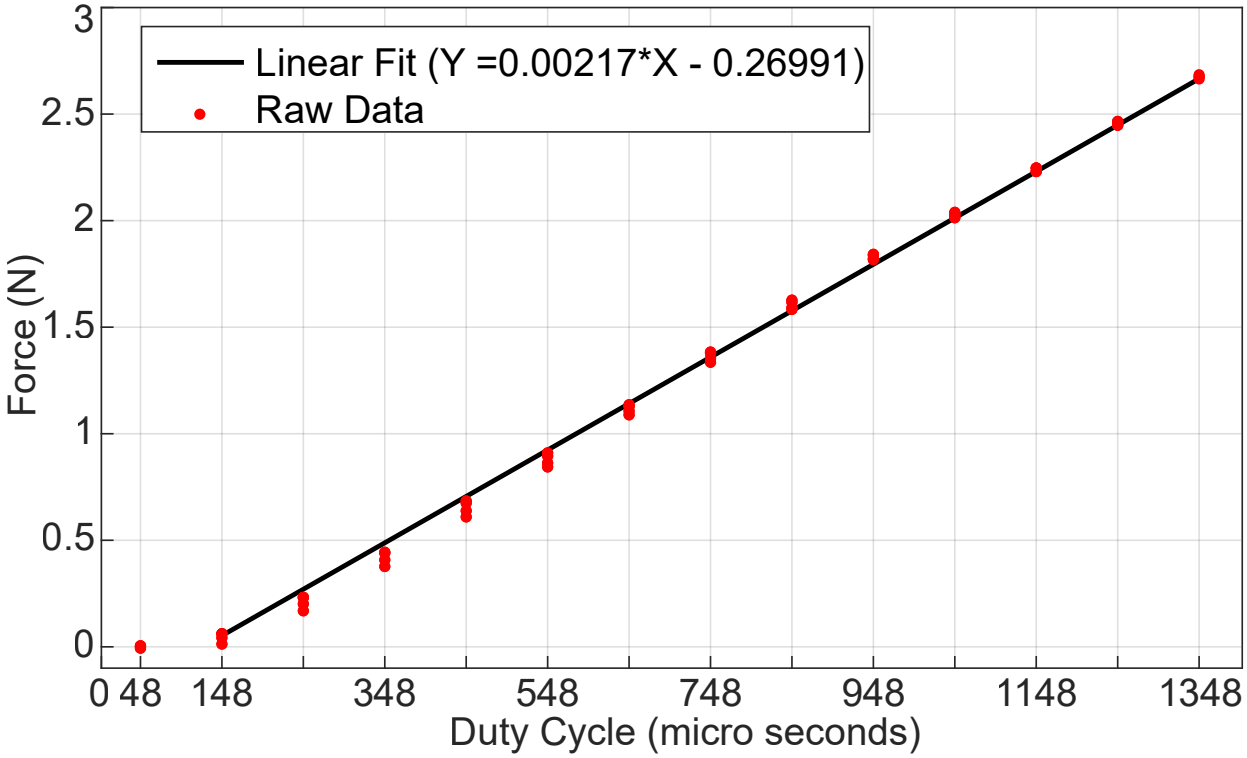

Figure 2.5: Motor force fit equation for the SpeedyBee 1404 motor and 4030 propeller. Demonstrating a linear relationship for duty cycles over 145 microseconds.

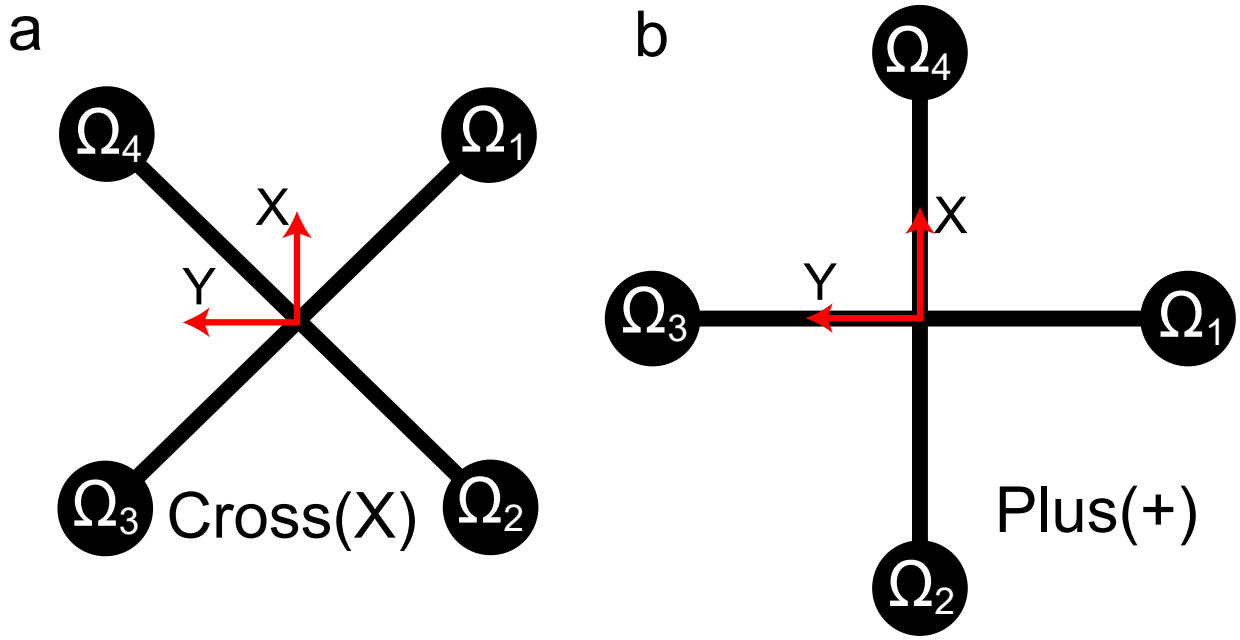

Figure 2.6: Quadrotor motor configuration strategies. a: cross configuration (same as MultiMo-MHR). b: plus configuration.

motors for forward/side-to-side motion requiring proper control and motor timing [38]. As such, the cross configuration was chosen for the MultiMo-MHR.

Standard notation for control over quadrotor systems uses U1, U2, U3 & U4 for the thrust(z), pitch, roll, and yaw control, respectively, where U1, U2, and U3 are control forces and U4 is a control torque. These control forces are based on the rotation rate of the motors; a general setup for a plus configuration can be found in [37]. From which our cross configuration setup was derived as:

$$U1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \tag{2.3}$$

$$U2 = b(-\Omega_1^2 - \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \tag{2.4}$$

$$U3 = b(-\Omega_1^2 + \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \tag{2.5}$$

$$U4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \tag{2.6}$$

where Ω1-4 is the respective rotation rate of the motor (units of rad/sec), the thrust (b) and

torque (d) factors, respectively (units of $Kg * m$ & $Kg * m^2$ respectively), used to convert U1,2,3 to a force and U4 to a torque.

The control forces (U's) are calculated from a controller, such as a PID. This calculation will not be explored in depth in this section. It will be detailed in subsequent chapters as various controllers were utilized on the robot, which calculated U1-4 in specific ways.

Reworking equations 2.3-2.6 and solving for the rotation rates enables a target motor action as a function of the control forces to be calculated. They can be solved as:

$$\Omega_1 = \sqrt{\frac{1}{4}(\frac{U1}{b} - \frac{U2}{b} - \frac{U3}{b} - \frac{U4}{d})} \tag{2.7}$$

$$\Omega_2 = \sqrt{\frac{1}{4}(\frac{U1}{b} - \frac{U2}{b} + \frac{U3}{b} + \frac{U4}{d})} \tag{2.8}$$

$$\Omega_3 = \sqrt{\frac{1}{4}(\frac{U1}{b} + \frac{U2}{b} + \frac{U3}{b} - \frac{U4}{d})} \tag{2.9}$$

$$\Omega_4 = \sqrt{\frac{1}{4}(\frac{U1}{b} + \frac{U2}{b} - \frac{U3}{b} + \frac{U4}{d})} \tag{2.10}$$

A slight issue arises though with only having the rotation rate for each respective motor, in that when controlling a motor a specified rate is not necessary known but instead some target duty cycle or thrust/force output. Based on the knowledge that U1-3 are forces equivalent to:

$$F = b * \Omega^2 \tag{2.11}$$

Equations 2.7-2.10 can be reworked to solve for the desired force each motor should output as a function of the U's:

$$F_1 = b\Omega_1^2 = \frac{1}{4}(U1 - U2 - U3 - U4\frac{b}{d}) \tag{2.12}$$

$$F_2 = b\Omega_2^2 = \frac{1}{4}(U1 - U2 + U3 + U4\frac{b}{d}) \tag{2.13}$$

$$F_3 = b\Omega_3^2 = \frac{1}{4}(U1 + U2 + U3 - U4\frac{b}{d}) \tag{2.14}$$

$$F_4 = b\Omega_4^2 = \frac{1}{4}(U1 + U2 - U3 + U4\frac{b}{d}) \tag{2.15}$$

where F_{1-4} are control forces, which, due to the linear relation found in the Chapter 2.4.4, can be converted to a known specified motor duty cycle to command the motors. As it stands, however, the current equations to determine the force still depend on the thrust and torque factors. As the U's are calculated by a controller, each U will be multiplied by some control gain ($K_{u1,u2,u3,u4}$). Each control force and torque can be calculated as:

$$U1^* = K_{U1} * U1 \tag{2.16}$$

$$U2^* = K_{U2} * U2 \tag{2.17}$$

$$U3^* = K_{U3} * U3 \tag{2.18}$$

$$U4^* = K_{U4} * \frac{d}{b} * U4 = K_{U4}^* * U4 \tag{2.19}$$

where each new control force (equations 2.16-2.18) and control torque (equation 2.19) can be simplified to a gain (K) multiplied by the control unit (U). This enables a final simplification of:

$$F_1 = \frac{1}{4}(U1^* - U2^* - U3^* - U4^*) \tag{2.20}$$

$$F_2 = \frac{1}{4}(U1^* - U2^* + U3^* + U4^*) \tag{2.21}$$

$$F_3 = \frac{1}{4}(U1^* + U2^* + U3^* - U4^*) \tag{2.22}$$

$$F_4 = \frac{1}{4}(U1^* + U2^* - U3^* + U4^*) \tag{2.23}$$

The ultimate equations 2.20 - 2.23 allow one to calculate a desired control force for each motor using only a controller output, assuming the controller utilizes tunable gains. The thrust and torque factors can still be computed, but this process simplifies the testing process

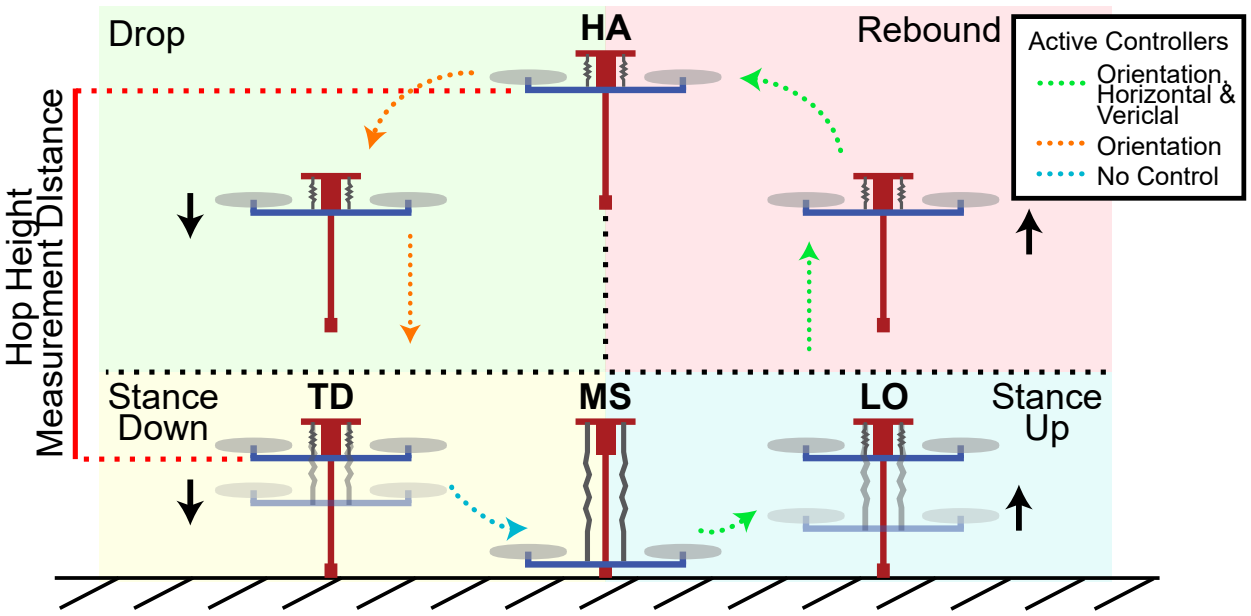

Figure 2.7: Hopping cycle overview. The cycle had four main phases: drop, stance down, stance up, and rebound. Where each phase is marked with a specific transition point: touchdown (TD), max squat (MS), lift off (LO), and hop apex (HA). Where each transition can be detected to then initiate a control strategy for the matching phase.

when a gain-based controller is already in use.

2.4.6. Hopping Methodology

The method by which the robot hops involves four main phases, with four transition points. The phases include: drop, stance down, stance up, and rebound. While the transitions occur between the phases as: touchdown (TD), max squat (MS), liftoff (LO), and hop apex (HA). The phases and respective transitions can be seen in Figure 2.7.

A general overview of the hop cycle is provided here, but see Chapters 3 and 4 for finer detail on exact procedures and processes in the cycle. The robot has three main forms of control, as detailed in the prior section: thrust (U1), pitch (U2), roll (U3), and yaw (U4) control. Thrust, U1, operates in the vertical body axis of the robot and can provide added energy to set a hopping height. The remaining three allow for orientation control with U2, U3, and U4. And can employ horizontal and vertical motion by combining thrust with orientation control (i.e., pitch oneself over and then apply thrust to move horizontally).

The general framework of the robots' hopping cycle is as follows. Starting with the drop phase, in which the robot falls using only orientation control to set a desired body angle for touchdown. Once in contact with the ground, control is disabled until the robot body reaches its maximum stretch (MS). At this point, based on the desired trajectory, all controllers (vertical, horizontal, and orientation) activate to command a desired trajectory and achieve a target hopping position in space. Once at the apex of the hop, the cycle renews with the drop phase until the cycle is terminated.

2.5. Experimentation Setup

2.5.1. Room overview

Experimental evaluation of the MultiMo-MHR was conducted in a motion capture studio, utilizing a Vicon Vantage V5 system. The system is comprised of 12 cameras (for full camera specifications see [39]) which are attached to a rigid metal frame on the edges of the space. The usable volume is a 5x5x5 m region shown in Figure 2.9, in which objects with tracking markers (Figure 2.11a) affixed can be tracked.

The space itself has a hard concrete floor with a sealed epoxy coating on top. The floor is mainly flat, with some inconsistencies in certain regions due to seam lines and/or settling. The studio is the sole use of the space, helping prevent possible measurement artifacts from external sources picked up by the camera.

Additionally, the space has supplied AC and heating via a series of ducts, which can generate disturbances due to the blowers. As such, all experiments were done with any fans and AC/heating disabled to prevent external disturbances. This ensures all control effort seen is a function of the robot system (or any system) and not as a result of external stimuli, ideal for experimental evaluation.

2.5.2. Room safety

Due to the volatile nature of robots, especially one that can hop up to 4 m, safety precautions must be taken to ensure personal and system care. Prior to any testing in the motion capture studio, experiments were conducted to test feasibility and early controller planning. One setup saw a metal wire attached from the ceiling to a cement bucket on the floor, visible in Figure 2.8a. The robot would ride along the guide wire, enabling both hopping and yaw orientation control to be established. A subsequent rig saw the robot mounted sideways by strings to test the roll/pitch control, visible in Figure 2.8b. Both test setups enabled the early robot to transition more easily into the less restricted motion capture studio.

With the initial validation completed, the robot was brought to the motion capture studio, where all subsequent robot testing was conducted. To ensure safety, two guide ropes (Paracord) were installed. The rope went through a set of rungs from the operator location to the center of the space. The rope was given enough slack such that the robot could hop within a desired circular region (up to the entire width of the space) and with a pull of the rope the robot would be unable to impact the ground or walls therein ending a trial, the rope is visible with color lines in Figure 2.9. The combined pair of the safety rope and an external Bluetooth controller produces a reliable means of terminating a hopping trial. In the event of a system error (i.e., part break or unexpected behavior) or at the end of an experimental trial, the safety rope can be pulled in conjunction with a button press on the Bluetooth controller, which disables the motors, drops the system, and prevents it from impacting anything.

2.5.3. System axes and Rotation matrices

To ensure proper control, a known coordinate axis and set of Euler angles is required. The MultiMo-MHR has coordinates as: X-axis positive body forward, Y-axis positive body left & Z-axis positive is body axis up. The environment/motion capture studio has the same

Figure 2.8: Early MultiMo-MHR testing setups. a: vertically constrained testing on a guide wire. b: orientation control testing.

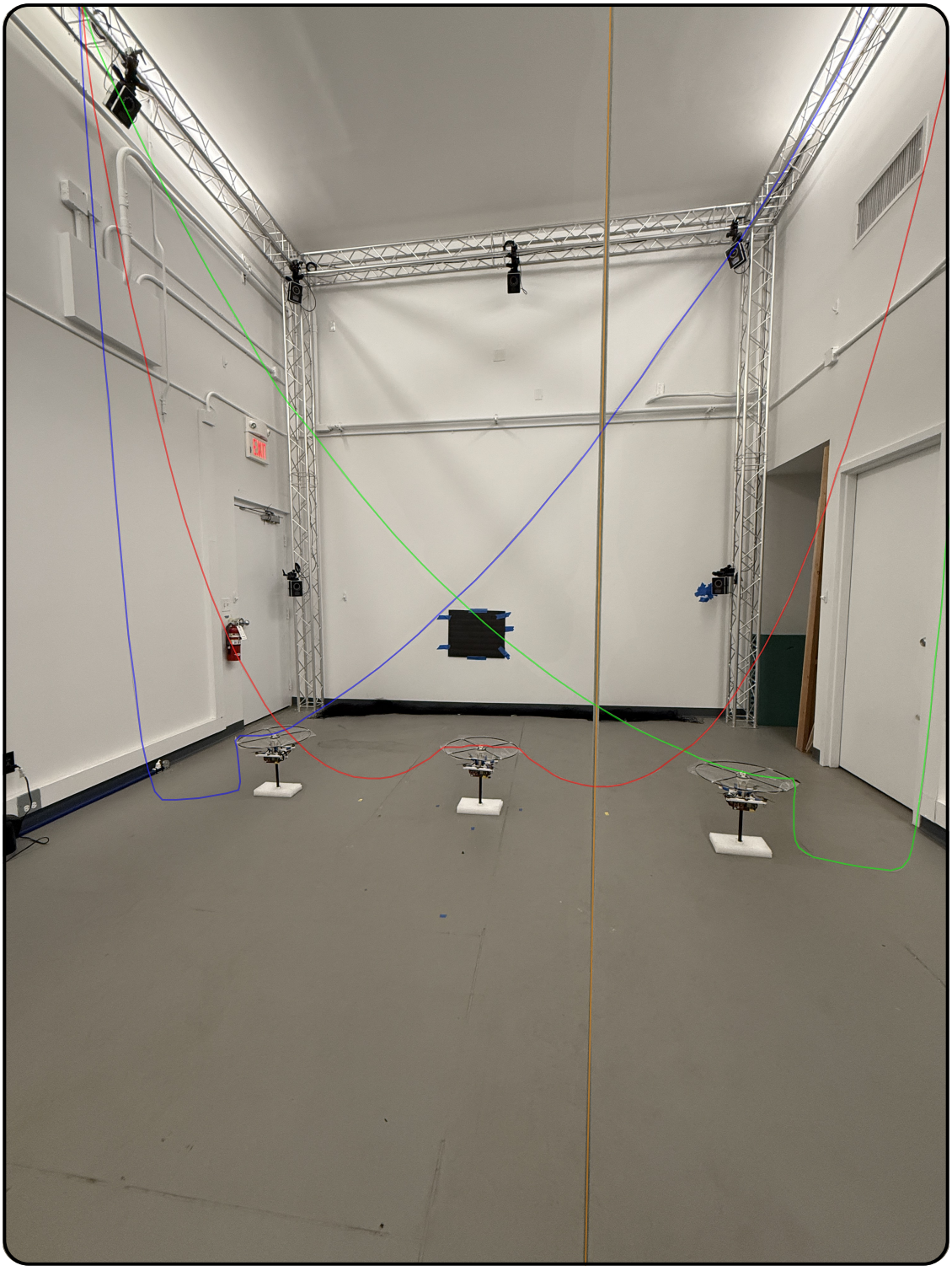

Figure 2.9: Experimental setup with safety rope for three robot positions. Red: center of the room, Blue: far left edge with safety rope, Green: far right edge of safety rope, Orange: central rope operator can pull to shorten overall safety rope length.

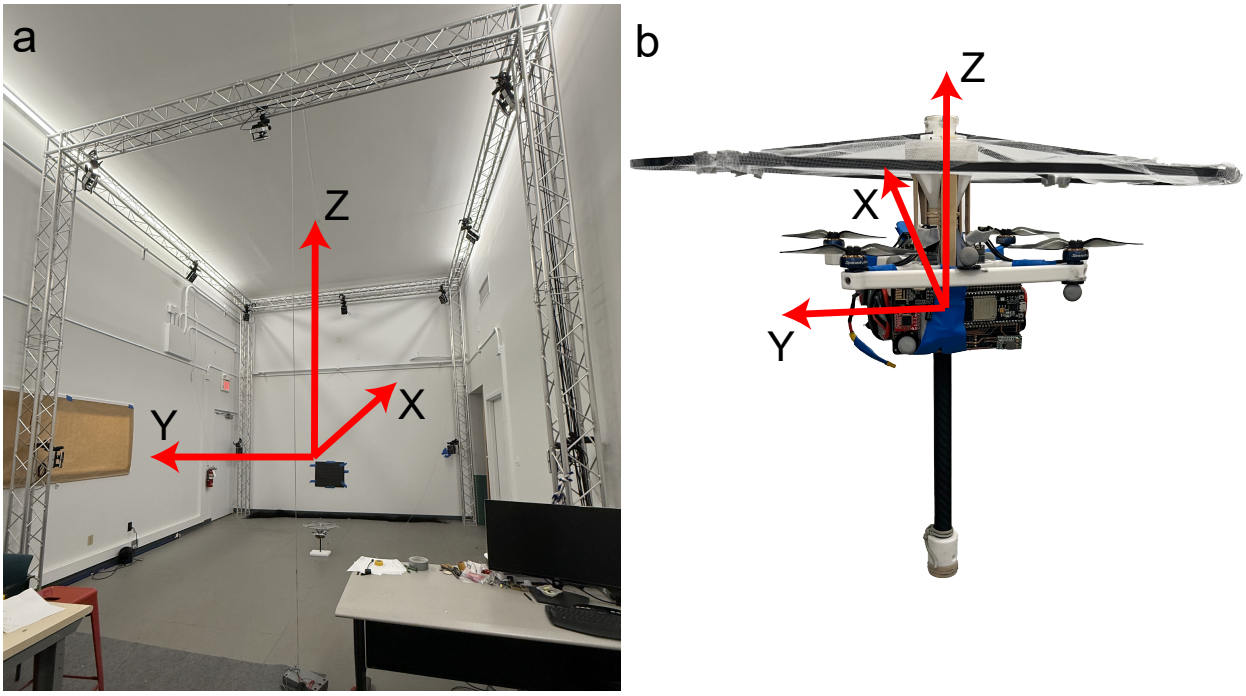

Figure 2.10: Coordinate axes for, a: the motion capture studio, b: the MultiMo-MHR

establishment as: X-axis positive room forward, Y-axis positive room left & Z-axis positive is room axis up. Both associated axes in the respective regions can be visualized in Figure 2.10.

Orientation measurements used for control were primarily obtained from the motion capture system; for R1, the WT901 sensor was used to provide body frame angles, but was later removed while control was perfected. The motion capture system provides XYZ Euler angles of the tracked object, which are directly used for orientation control. These measurements also enable transformation between body and global frame coordinate systems. As an example, a desired and measured global location (x,y,z) can be rotated into the robot body frame and used as inputs for a position-based PID controller.

The rotations to achieve a proper rotation are as follows. A counterclockwise XYZ Euler rotation is utilized with rotation matrices seen in equations 2.24-2.26 for rotating about the x-axis (roll or ϕ), y-axis (pitch or θ), and z-axis (yaw or ψ) respectively. And combine for a world to body rotation (R_{wb}) per equation 2.27, or the body to world rotation (R_{bw}) per

equation 2.28.

$$R_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & sin(\phi) \\ 0 & -sin(\phi) & cos(\phi) \end{bmatrix} \tag{2.24}$$

$$R_\theta = \begin{bmatrix} cos(\theta) & 0 & -sin(\theta) \\ 0 & 1 & 0 \\ sin(\theta) & 0 & cos(\theta) \end{bmatrix} \tag{2.25}$$

$$R_\psi = \begin{bmatrix} cos(\psi) & sin(\psi) & 0 \\ -sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.26}$$

$$R_{wb} = R_\phi * R_\theta * R_\psi \tag{2.27}$$

$$R_{bw} = (R_{wb})^T \tag{2.28}$$

2.5.4. System Calibration

Two forms of calibration are conducted for experimental tests. One is to calibrate the motion capture system, and the other is for the on-board sensors.

Starting with the calibration of the motion capture space, specifically for the Vicon motion capture Vantage V5 setup. With the cameras orientated in the desired manner and no retro reflective materials (i.e. reflecting IR light) or tracker markers (see Figure 2.11a) are visible to the cameras (i.e. remove the thing which is to be tracked from view of the cameras), an initial mask is generated which hides the camera IR lights and any other reflections from tracking. Then, using a calibration wand (see Figure 2.11b), the target space is swept with the wand, especially in the locations where tracked motion is expected. Once completed, the wand is placed at the set room origin, oriented such that the long handle portion of the wand

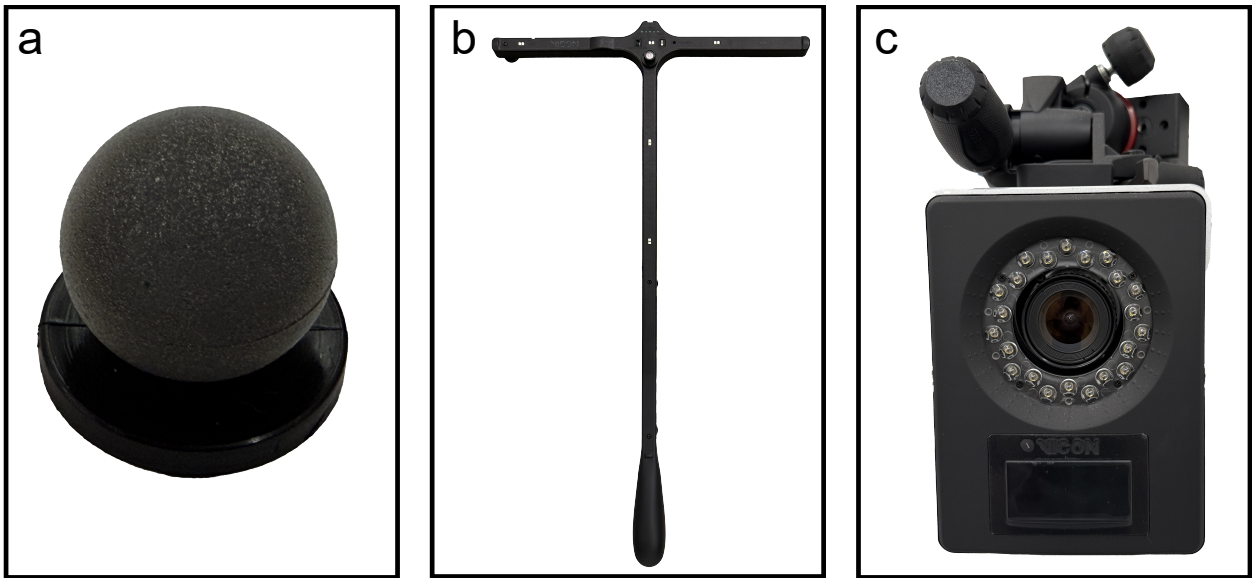

Figure 2.11: Components for the Vicon Motion Capture studio. a. retro reflective tracker markers, b. calibration wand, c. Vicon vantage V5 camera

aligns with the desired positive room Y-axis to set the room origin (i.e., room (0,0,0)). The last step is to ensure the desired markers on the robot are being tracked, and the tracked object's COM aligns with the actual robot COM. The robot COM is approximated with a SolidWorks model, and the robot COM is measured based on the distance away from a known tracker marker location. This enables the Vicon tracked COM to be shifted from the known maker placement to the approximate true robot COM. With that, the calibration of the motion capture space is complete and ready for testing; this calibration occurs periodically, not before every trial.

The calibration on the robot itself comes in the form of sensor calibration. Before every trial the robot is placed vertically upright on the ground, on the robot startup a calibration process as follows goes on for the following sensor. The low-g IMU (MPU6050) has a constant set offset for the accelerometer and gyroscopic readings. The offsets were found experimentally by averaging 10 trials in which the sensor was read with the robot stationary for 10 seconds; this was repeated periodically and updated if needed. The high-g accelerometer (H3LIS331DL) was found to have a far higher noise floor; as such, before each experimental trial, the robot would be placed upright and have 10 seconds of data read and averaged to

collect offsets. Lastly, the robot had a measured starting height captured by reading Vicon position data during the aforementioned 10-second vertical upright phase, which gives the robot's height when the body is unstretched and sets a zero height for the system.

2.5.5. Experimental Data transmission Pipeline

The processes and technologies involved in data transmission and acquisition for experimental trials with the MultiMo-MHR are comprised of several strategies. This section will detail those methodologies, but will exclude any data measured explicitly on board the robot itself, as that content is detailed in Chapter 3.4.3. There are five components for data transmission and acquisition shown in Figure 2.12: the MultiMo-MHR, a separate desktop computer, an ESP32, a PlayStation 4 dual shock 4 (DS4) controller, and the Vicon motion capture systems (comprised of 12 cameras and a data acquisition system).

The entire data pipeline is displayed in Figure 2.12. The bulk of the process starts at the main desktop computer. The computer receives data from the Vicon motion capture cameras via Ethernet cables. The motion capture data itself is captured from the 12 cameras (Vicon vantage V5 as seen in Figure 2.11c), which utilize infrared light to detect retro reflective markers placed on the tracked object. With this, the desktop can acquire position and orientation (and their derivatives) data for the object within view of the cameras. For the desired experiment, certain data is transmitted to the robot, whose transmission process starts by sending data packets over USB to a separate ESP32.

The ESP32 takes in two sources of information: USB and Bluetooth. Motion capture data packets are sent from the desktop computer to the ESP32 over USB. Bluetooth commands from a DS4 controller are transmitted to the ESP32 as well. The DS4 controller connects to the ESP32 over Bluetooth low energy, with the ESP32 reading any button or analog stick input. Both the motion capture data packet and DS4 inputs are packaged as a string of values/commands and are transmitted to the robot using an RF transceiver (nRF24L01) connected to the ESP32. The data packet is received on the robot with a matching RF

transceiver at a rate of 100Hz. Where the motion capture data packet is extracted and used for control, and the DS4 commands are read, and certain actions are taken based on the robot mode.

The DS4 controller does not have any direct input on the robot's hopping locomotion aside from being able to stop the robot and for general initialization commands like engaging motors and to allow a trial to begin. It was primarily used in early stages to test system flight and allow for remote gain adjustment before the controller was finalized.

2.6. Summary

This chapter presented the MultiMo-MHR, delving into the origin of hopping robots from the past to modern days. The components that make up the robot are displayed, along with sections focusing on specific elements, including data saving, motor force, and control fundamentals, to explain certain processes and methodologies. The chapter then wraps up with details concerning the testing environment and setup, both from a physical and software perspective. This chapter serves to both introduce the robot and explain aspects of the operation of the robot.

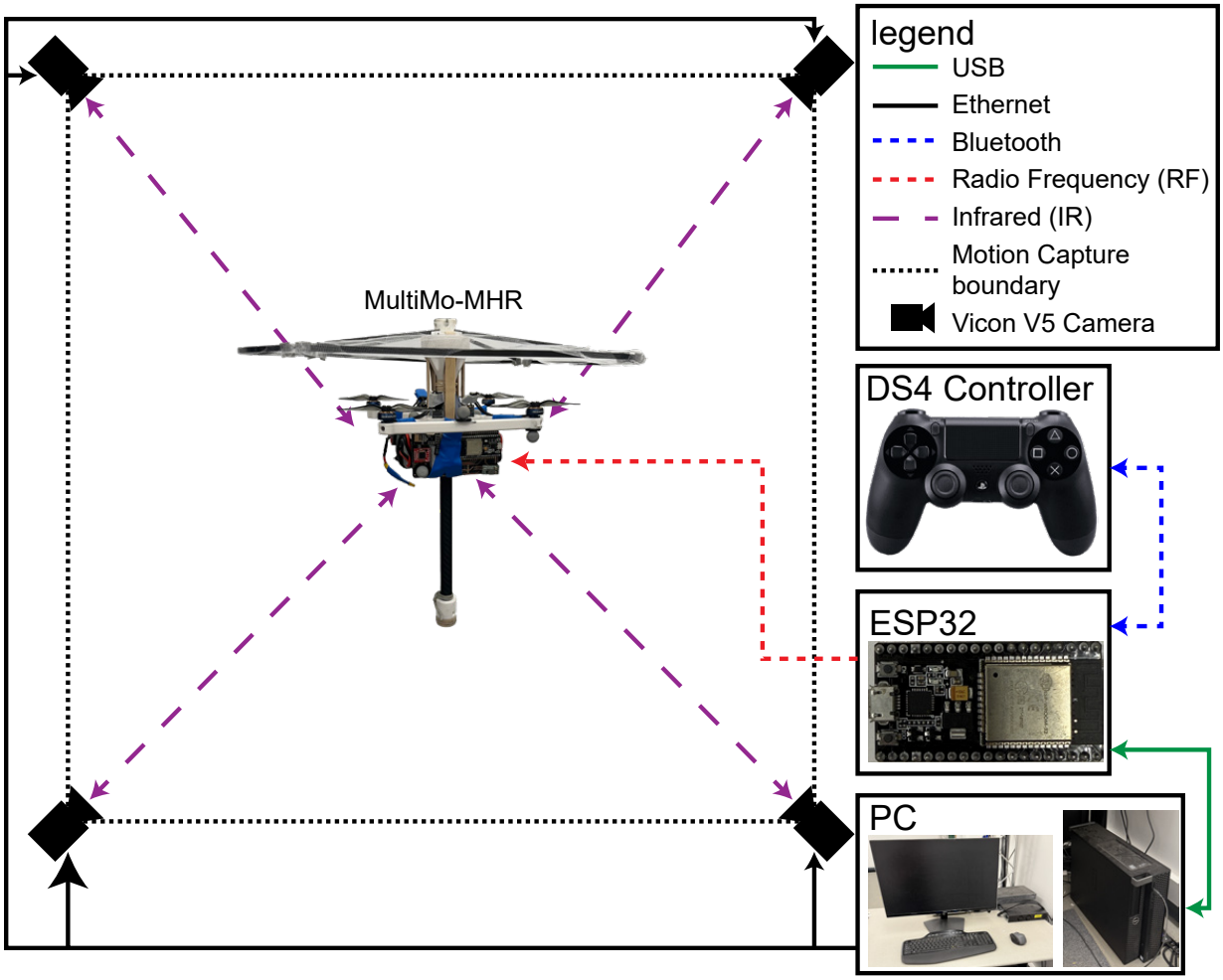

Figure 2.12: Data communication schematic for experimental testing of the MultiMo-MHR. The motion capture cameras capture the object's placement in the room and send it to the base PC over Ethernet. The base PC transfers the data over USB to an external ESP32 and transmits data over RF to the robot. An additional Bluetooth controller also connects to the ESP32 for extra commands and e-stopping.

3. Design and Control of a High-Performance Hopping Robot

This chapter is comprised of a journal article under the same name; and was published in *IEEE Robotics and Automation Letters (Volume: 10, Issue: 6, June 2025)*

3.1. Abstract

Jumping and hopping locomotion are efficient means of traversing unstructured rugged terrain with the former being the focus of roboticists; a focus that has recently been changing. This focus has led to significant performance and understanding in jumping robots but with limited practical applications as they require significant time between jumps to store energy, thus relegating jumping to a secondary role in locomotion. Hopping locomotion, however, can preserve and transfer energy to subsequent hops without long energy storage periods. However, incorporating the performance observed in jumping systems into their hopping counterparts is an ongoing challenge. To date, hopping robots typically operate around 1 meter with a maximum of 1.63 m whereas jumping robots have reached heights of 30 m. This is due to the added design and control complexity inherent in developing a system able to input and store the necessary energy while withstanding the forces involved and managing the system's state. Here we report hopping robot design principles for efficient, robust, high-specific energy, and high-energy input actuation through analytical, simulation, and experimental results. The resulting robot (MultiMo-MHR) can hop over 4 meters or ∼2.4x the current state-of-the-art.

3.2. Introduction

Jumping robots have been under significant development for decades [23,40], demonstrating a diversity of designs, energy storage and release mechanisms, and additional behaviors,

resulting in the specific energy (energy/mass) and performance of jumping robots far surpassing their hopping counterparts. The challenge then becomes how to incorporate that level of performance into hopping robots.

High-performance jumping platforms have achieved jump heights as high as 30 meters [13, 41]. Additionally, jumping offers the advantage of extended ground contact time, compared to the rapid stance time of hopping platforms, which facilitates self-righting [42–44] and traversal [45, 46] (via legs [47–50] or wheels [51–53]). However, the principles used to achieve high specific energy in jumping platforms can be applied to hopping systems. The design of the hopping leg and its connection to the energy storage mechanism is crucial for efficient energy transfer. Jumping systems have demonstrated various leg mechanisms, such as linkage [34,46,51,54], linear [55–58], buckling [59,60], and spring-based legs [13,61]. Of these, linkage-based legs may present challenges for hopping systems due to their added complexity and potential failure points from repeated impacts. Jumping platforms have also employed a number of aerial locomotion strategies, from gliding [15,62–67] to flying [9] to further enhance their locomotion capabilities and overcome a wider range of obstacles.

Hopping platforms, however, have seen much less development due to their design and control complexity necessitated by the requirement to maintain orientation and input energy over the extremely short time periods and complex ground interactions. They do have the advantage though, as energy can be added in the stance and/or aerial phases to alter or maintain a hopping height (h_r); measured as the center-of-mass (CM) displacement between liftoff and hop apex (excluding stance) for the ascent (rebound) phase. This is achieved by carrying forward energy from previous hops; whereas, jumping platforms must add some energy in stance. As seen in Table 3.1 Salto/Salto 1-p , LEAP, moving-mass robot, and ARCHER are platforms which hop with stance based energy methods [6–8, 29–31, 34, 68, 69]. Alternatively, energy added in the aerial phase has been demonstrated by rotor-based platforms including PogoDrone, PogoX and Hopcopter [1, 9, 10, 33]. This style of hopping platform is also more efficient than flight, as hopping is possible with a thrust-to-weight ratio

Table 3.1: Hopping Robot Systems

Robot	**Energy Input**	$h_r{}^{\dagger}(m)$	$L_T(m)$	$m_T(kg)$
Salto [6, 34]	stance	1.0	∼0.29	0.100
Salto-1P [7, 29–31]	stance	1.3	∼0.31	0.0981
Moving mass [68]	stance	∼0.15	0.52	6.025
Archer [69]	stance	NA	NA	NA
Leap [8]	stance	NA	NA	2.470
PogoX [10, 33]	aerial-rotor	∼0.6	0.45	2.500
PogoDrone [9]	aerial-rotor	∼0.7	∼0.084	0.031
Hopcopter [1]	aerial-rotor	∼1.63	∼0.21	0.0348
MultiMo-MHR	aerial-rotor	4.49	0.387	0.697

max rebound height ($h_r{}^{\dagger}$), robot height (L_T) & system mass (m_T)

(TWR) < 1 while flight requires a TWR≥ 1 [1].

Hopping platforms typically maintain stable heights with consistent h_r, though there are exceptions such as hopping onto higher surfaces [10, 31], leaping off walls or platforms [6], or modifying energy between hops [70, 71]. Of these, the mechanisms behind energy modification have not been thoroughly investigated. Specifically, we are interested in energy modification at lower constant actuator power output and short periods of high output. Lower power output results in lighter actuators and higher performance, as seen in a key performance metric for vertical jumping and hopping, $height_{jump} = energy_{jump}/(mass * gravity)$; which is closely tied to mass. This is particularly relevant as hopping robots vary greatly in weight, from Salto's 100 g to the Cargo robot's 150 kg [72]. Additionally, through energy accumulation, this concept can allow the robot to achieve greater hopping heights over multiple cycles. This is inspired by the desert kangaroo rat (Dipodomys deserti), which increase their locomotion energy (LE, defined as the total kinetic, gravitational, and elastic energies) over successive hops by accumulating LE in both kinetic energy and elastic tissues,

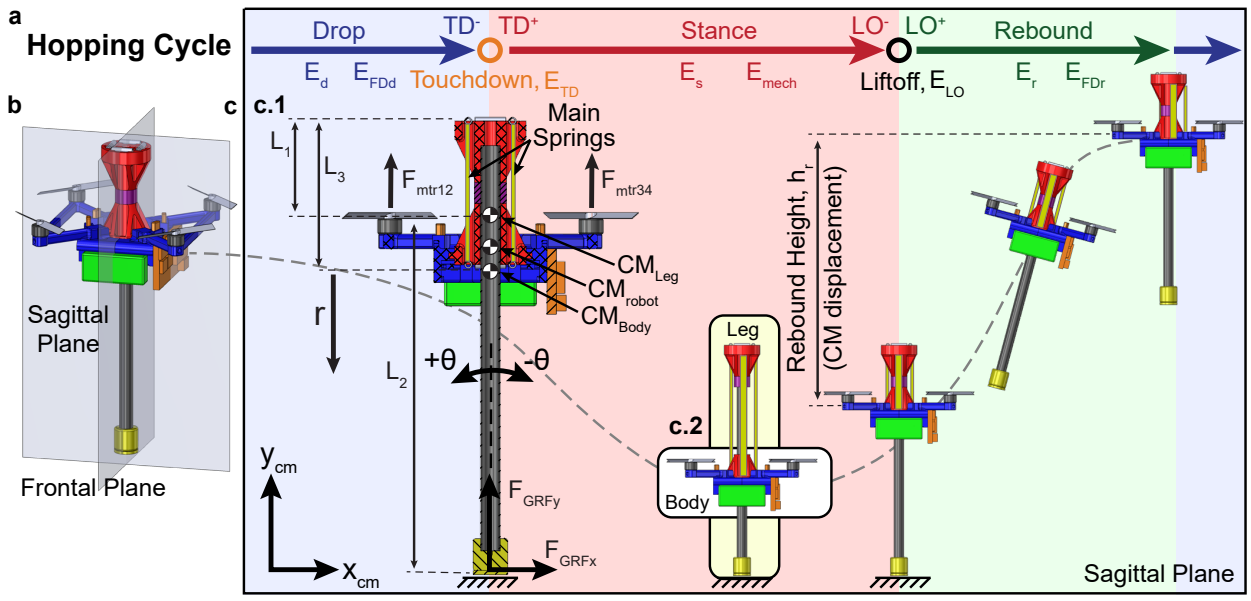

Figure 3.1: a) Hopping cycle: Phases (drop, stance, and rebound) and transitions (touchdown, liftoff); with locomotion energy (LE) modifications: Control inputs (E_d, Es, E_r), and losses (E_{FDd}, E_{TD}, E_{mech}, E_{LO}, E_{FDr}). b) Symmetry plane of the MultiMo-MHR. c) CAD model of the hopping cycle with hop height measured as the center-of-mass (CM) displacement between LO and hop apex (excluding stance). c.1) Section view showing internal connections, 4 degrees-of-freedom, and dynamic parameters. c.2) Showing the two bodies (Body, Leg) at max spring extension during the stance phase.

all while maintaining a constant metabolic rate [73]. This approach of accumulating and releasing energy without increasing actuator power provides a model for efficient hopping. This work develops a hopping robot model to explore energy input and control, fundamentals for high-performance hopping robot designs, and a control system, resulting in a robotic prototype (MultiMo-MHR) able to reach hopping heights up to $\sim 4m$ ($\sim$2.4x the current state-of-the-art [1]).

3.3. Dynamic Model

Locomotion tends to require cyclic motions in which the LE is modified over the course of the cycle. Figure 3.1a shows the hop cycle divided into three phases, including drop, stance, and rebound, and two transitions, including touchdown (TD) and liftoff (LO). Each phase has the potential for a control input to either add or remove LE, whereas both phases and

transitions have characteristic losses of LE. This results in eight distinct energy modifications with the total change in LE (apex to apex) as, $m_T g h_d(\delta_{rd} - 1) = E_d + E_s + E_r - E_{FDd} - E_{TD} - E_{mech} - E_{LO} - E_{FDr}$; where the total mass, m_T, gravity, g, drop height, h_d, rebound height, h_r, height ratio, $\delta_{rd} = h_r/h_d$, represent the total change in energy. The control input energies E_d (drop), E_s (stance), and E_r (rebound), and the energy losses from drag, E_{FDd} (drop) and E_{FDr} (rebound), touchdown, E_{TD}, mechanical, E_{mech}, and liftoff, E_{LO}, are represented accordingly. Given the new class of rotor-based hopping robots, the potential to add energy in rebound (E_r) with a thrust input has been facilitated.

To explore rotor-based hopping a dynamic model of the MultiMo-MHR (Fig. 3.1) is developed. The model consists of two masses (body, m_B, and leg, m_L) connected by a prismatic joint along the long axis of the leg. Due to the robot's symmetry about the frontal and sagittal planes (Fig. 3.1b), it can be modeled in the sagittal plane (Fig. 3.1c) resulting in four degrees-of-freedom (DoF) including: the center-of-mass (CM) position [x_{cm}, y_{cm}], orientation, θ, and spring stretch, r; state vector, $q = [x_{cm}, y_{cm}, r, \theta]$. The system dynamics will be solved using the Lagrange method, the process of deriving the equations is as follows.

As the system dynamics will utilize Lagrange's method the kinetic and potential energies of the system are required. Utilizing the robot diagram seen in Figure 3.1c.1 the equation development can be performed. Starting with kinetic energy which appears in the form of the body and legs translational and rotational energies. As the total systems center of mass can change as a function of the location of the body COM relative to the leg COM, these energies will be found for the separate bodies and combined.

Looking at the position of the leg COM ($\overrightarrow{X_l}$) and body COM ($\overrightarrow{X_b}$) a few relations are needed. Namely the relation between the separate COM's and center of mass for whole system ($\overrightarrow{X_{cm}}$) as:

$$m_l * \overrightarrow{X_l} + m_b * \overrightarrow{X_b} = m_t * \overrightarrow{X_{cm}} \tag{3.1}$$

Additionally the relative location between the body and leg COM's ($\overrightarrow{d}$) which can be

written in two methods:

$$\overrightarrow{d} = \overrightarrow{X_b} - \overrightarrow{X_l} \tag{3.2}$$

or as:

$$\overrightarrow{d} = R_\theta \begin{bmatrix} 0 \\ (r + L_3 - L_1) \end{bmatrix} \tag{3.3}$$

where both equations 3.2 & 3.3 give the location of the body COM relative to the leg COM. For equation 3.3 it uses direct measurements in the robots body frame and requires a rotation (R_θ) which has the form:

$$R_\theta = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \tag{3.4}$$

The aforementioned equations can then be used to solve for the body and leg COM positions given the location of the entire systems COM and relative location of the body to the leg:

$$\overrightarrow{X_b} = \overrightarrow{X_{cm}} + \frac{m_l}{m_t}\overrightarrow{d} = \overrightarrow{X_{cm}} + \frac{m_l}{m_t}(R_\theta \begin{bmatrix} 0 \\ (r + L_3 - L_1) \end{bmatrix}) \tag{3.5}$$

$$\overrightarrow{X_l} = \overrightarrow{X_{cm}} - \frac{m_b}{m_t}\overrightarrow{d} = \overrightarrow{X_{cm}} - \frac{m_b}{m_t}(R_\theta \begin{bmatrix} 0 \\ (r + L_3 - L_1) \end{bmatrix})) \tag{3.6}$$

Additional positions for components of the robot are necessary to calculate the force components on the robot. The robot has four main forces: motor thrust, ground reaction forces, drag and damping between the body-leg contact. As the model is simplified to be a 4 DOF model, the thrust simplifies the four motor thrusts into two forces called F_{mtr12} and F_{mtr34}, which are applied along the body z-axis of the robot. Next the ground reaction forces, existing when the robot contacts the ground, are applied at the end of the leg, or

foot, in a global reference frame dubbed F_{GRFx} and F_{GRFy} for the X and Y components of the force. Lastly, a drag force (F_{drag}) is applied about the system COM and a damping term (F_{sd}) applied to account for body-leg oscillations about the system COM. For these forces the location and direction of application is necessary, as such positions for the motor forces ($\overrightarrow{X_{mtr12}}$ and $\overrightarrow{X_{mtr34}}$) and foot location ($\overrightarrow{X_{foot}}$) are required. These come out as:

$$\overrightarrow{X_{foot}} = \overrightarrow{X_l} + R_\theta \begin{bmatrix} 0 \\ -L_2 \end{bmatrix} \tag{3.7}$$

$$\overrightarrow{X_{mtr12}} = \overrightarrow{X_b} + R_\theta \begin{bmatrix} -d_{mtx} \\ d_{mty} \end{bmatrix} \tag{3.8}$$

$$\overrightarrow{X_{mtr34}} = \overrightarrow{X_b} + R_\theta \begin{bmatrix} d_{mtx} \\ d_{mty} \end{bmatrix} \tag{3.9}$$

Moving on now to the actual Lagrangian method which will have four degrees of freedom. The DOF being: the system COM position ($\overrightarrow{X_{cm}} = [x_{cm}, y_{cm}]^T$), the system orientation (θ) and the spring stretch (r). As such the overall state is:

$$q = \begin{bmatrix} x_{cm} \\ y_{cm} \\ r \\ \theta \end{bmatrix} \tag{3.10}$$

With the Lagrangian method using the following form:

$$M\ddot{q} + Cg(\dot{q}, q) = \tau \tag{3.11}$$

With components being the mass matrix(M), a combined Coriolis, centrifugal, and gravitational forces (Cg) and the input torques and forces (τ). The next step in the model is

derivation of these matrices which can be achieved with a handful of steps. Starting with the kinetic (T) and potential (V) energies of the system. For kinetic energy there are translational and rotational energies, best broken up based on the system components being the body and leg. The kinetic energy can be found as:

$$T = \frac{1}{2}m_B V_B^2 + \frac{1}{2}m_L V_L^2 + \frac{1}{2}I_{CML}\dot{\theta}^2 + \frac{1}{2}I_{CMB}\dot{\theta}^2 \tag{3.12}$$

where V_B&V_L are the body and leg velocities which can be obtained with the time derivatives of equations 3.5 & 3.6 respectively. For rotational energies the moment of inertia of the body and leg (I_{CMB} &I_{CML}), alongside the rotational rate for the robot ($\dot{\theta}$) are required.

Moving on to the potential energies, the robot has three sources of potential energy. Coming in the from of height potential and spring potential (two sets of springs with the main rubber bands and leg-body spring). The potential energy is as follows:

$$V = m_L\, g\, h_L + m_B\, g\, h_B + \frac{1}{2}K_s(r + r_{sp})^2 + \frac{1}{2}K_{bl}\, r^2 \tag{3.13}$$

Which uses gravity(g) with the y-component of the body and leg positions (h_B and h_L) from equations 3.5 & 3.6 for the height potential terms. And for spring potential the spring constants for the main spring(K_s) and body-leg spring (K_{bl}) alongside the main body stretch (r) and an initial main spring pre-stretch (r_{sp}) for the equation.

With this all the necessary components to derive the M, Cg and τ matrices are achieved. Each matrix requires the following process to compute. Starting with the mass matrix (M) which has the base form of:

$$T = \frac{1}{2}\dot{q}^T M \dot{q} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} M\dot{q}_i\dot{q}_j \tag{3.14}$$

Equation 3.14 can then be solved for M with partial derivatives with respect to $\dot{q}_i$&$\dot{q}_j$ to have (based on [74]):

$$M = \frac{\partial^2 T}{\partial \dot{q}_i \partial \dot{q}_j} \tag{3.15}$$

Next the combined Coriolis, centrifugal, and gravitational forces (Cg) which can be achieved as follows (based on [75]):

$$Cg = \frac{d}{dt}\frac{\partial T}{\partial \dot{q}} - \frac{\partial T}{\partial q} + \frac{\partial V}{\partial q} \tag{3.16}$$

Lastly is the input torques and forces (τ) which can be found using the applied forces (f) which are mapped to the generalized coordinates using the CM to force application location(x). The simple expression of this is as (based on [76]):

$$\tau_x = J_x(q)^T f \tag{3.17}$$

For this model there are three applied forces which need to be shifted as they are not applied at the CM: F_{mtr12}, F_{mtr34} and F_{GRF} (where $F_{GRF} = [F_{GRFx};F_{GRFy}]$). With the Jacobians taken about the relative force application positions $\overrightarrow{X_{mtr12}}$ and $\overrightarrow{X_{mtr34}}$ and $\overrightarrow{X_{foot}}$. The remaining applied forces, F_{drag} & F_{sd} are applied at the CM so can be used as is. The ultimate equation to derive τ is as follows:

$$\tau = J_{\overrightarrow{X_{mtr12}}}(q)^T \begin{bmatrix} 0 \\ F_{mtr12} \end{bmatrix} + J_{\overrightarrow{X_{mtr34}}}(q)^T \begin{bmatrix} 0 \\ F_{mtr34} \end{bmatrix} + J_{\overrightarrow{X_{Foot}}}(q)^T \begin{bmatrix} F_{GRFx} \\ F_{GRFy} \end{bmatrix} + F_{drag} + F_{sd} \tag{3.18}$$

With everything one can solve equations 3.15, 3.16 and 3.18 to achieve the system dynamics for the Lagrange method as:

$$M\ddot{q} + Cg(\dot{q}, q) = \tau \tag{3.19}$$

$$M = diag[m_T,\, m_T,\, (m_B m_L)/m_T,$$
$$c_1/m_T + I_{cmB} + I_{cmL}]^T$$

$$Cg = \left[0,\, g * m_T,\, c_2/m_T,\, m_B m_L \dot{\theta}\dot{r}(2L_3 - 2L_1 + 2r)/m_T\right]^T$$

$$\tau = \begin{bmatrix} -U_1 sin(\theta) \\ U_1 cos(\theta) \\ -(U_1 m_L)/m_T \\ -U_2 d_{mtrx} \end{bmatrix} + \begin{bmatrix} F_{GRFx} \\ F_{GRFy} \\ c_3 \\ c_4 \end{bmatrix} + F_{drag} + F_{sd}$$

$$c_1 = m_B m_L (L_1^2 - 2L_1 L_3 - 2L_1 r + L_3^2 + 2L_3 r + r^2)$$

$$c_2 = m_T(K_s(r_{sp} + r) + K_{bl} r) + m_B m_L \dot{\theta}^2 (L_1 - L_3 - r)$$

$$c_3 = -m_B(F_{GRFx} sin(\theta) - F_{GRFy} cos(\theta))/m_T$$

$$c_4 = c_5(L_2 m_T + m_B(L_1 - L_3 - r))/m_T$$

$$c_5 = F_{GRFy} sin(\theta) + F_{GRFx} cos(\theta)$$

where, all of the aforementioned the parameters and values are shown in Table 3.2 and Fig. 3.1c.1 and c.2. The forces and torques exerted on the robot throughout hopping locomotion including, motor thrust (F_{mtr12}, F_{mtr34}), ground reaction forces (F_{GRFx}, F_{GRFy}), drag (F_{drag}), and body-leg contact damping force (F_{sd}). First, the motor thrusts in the body frame can be converted into the control inputs $U_1 = F_{mtr12} + F_{mtr34}, U_2 = F_{mtr12} - F_{mtr34}$. The world frame ground reaction forces, modeled as springs and dampers, arise as the robot's foot makes contact with the ground. The drag force accounts for an atmosphere and is assumed to act at the CM of the system. Finally, the body-leg contact damping force, reduces the oscillation of the body and leg after collision at LO. The experientially measured losses are scaled accordingly: drag force scaling factor, $\gamma_{drag} = (1 + \delta m/m_T)^{2/3}$ (change in mass, δm, from

the MultiMo-MHR, m_T), touchdown efficiency $\eta_{TD} = m_B/m_T$, and mechanical $\eta_{mech} = 0.75$ and liftoff $\eta_{LO} = 0.8328\, h_d^{-0.058}$ efficiencies, (fitted parameters); where the losses not included in the model are applied to the robot's energy state as efficiencies at the following phase transition. To understand the relationship between thrust and robot mass the control input, $U1$, in rebound is normalized; creating an average rebound control input thrust-to-weight ratio (TWR) from LO (t_{LO}) to hop apex (t_{HA}), $\alpha_r = 1/(t_{HA} - t_{LO}) \int_{t_{LO}}^{t_{HA}} U_1/(m_T g) dt$.

3.3.1. Limited Rebound Control Input TWR, $\alpha_r \leq 99\%$

Figure 3.2a, shows the maximum performance of the MultiMo-MHR at varying α_r from 50% to 99% with the associated cycle average TWR,
$\alpha_{cyc} = 1/(t_{HA(k+1)} - t_{HA(k)}) \int_{t_{HA(k)}}^{t_{HA(k+1)}} U_1/(m_T g) dt$, representing the equivalent continuous TWR. The MultiMo-MHR can sustain hopping at an $\alpha_r(\alpha_{cyc}) = 50\%(26.8\%)$ and reach a maximum height of 30.7 m at $\alpha_r(\alpha_{cyc}) = 99\%(83.7\%)$. In all cases, $\alpha_{cyc} \geq 100\%$ (flight) and $\alpha_{cyc} \geq 84.4\%$ (flight with leg removed) suggests that the hopping robot would always be more efficient than continuous flight, resulting in longer operation times; albeit with a reduced operating environment. While differing losses will result in differing performance for other robots, the trend will remain as approaching $\alpha_r = 99\%$ will accentuate the rebound phase, resulting in increased α_{cyc}.

Figure 3.2b, presents the critical α_r (α_{cr}, blue lines), defined as the minimum TWR a hopping robot must have to sustain hopping at 0.25 m (solid lines) and 3 m (dashed lines), and the resulting critical thrust (red lines), $F_{cr} = \alpha_{cr} m_T g$. The figure covers three orders-of-magnitude in robot mass, [0.05,5] kg, and body mass percentages from 40% to 95%, thus encompassing the range of current hopping robots. It demonstrates that α_{cr} is more sensitive to the body mass ratio, m_B/m_T, and less sensitive to the total robot mass; primarily due to the preservation of body energy, and loss of leg energy, at TD. The critical thrust, F_{cr} Fig. 3.2b (red lines), also suggests that adding mass to the robot body has a significantly lower impact on the required thrust magnitude as compared to adding mass

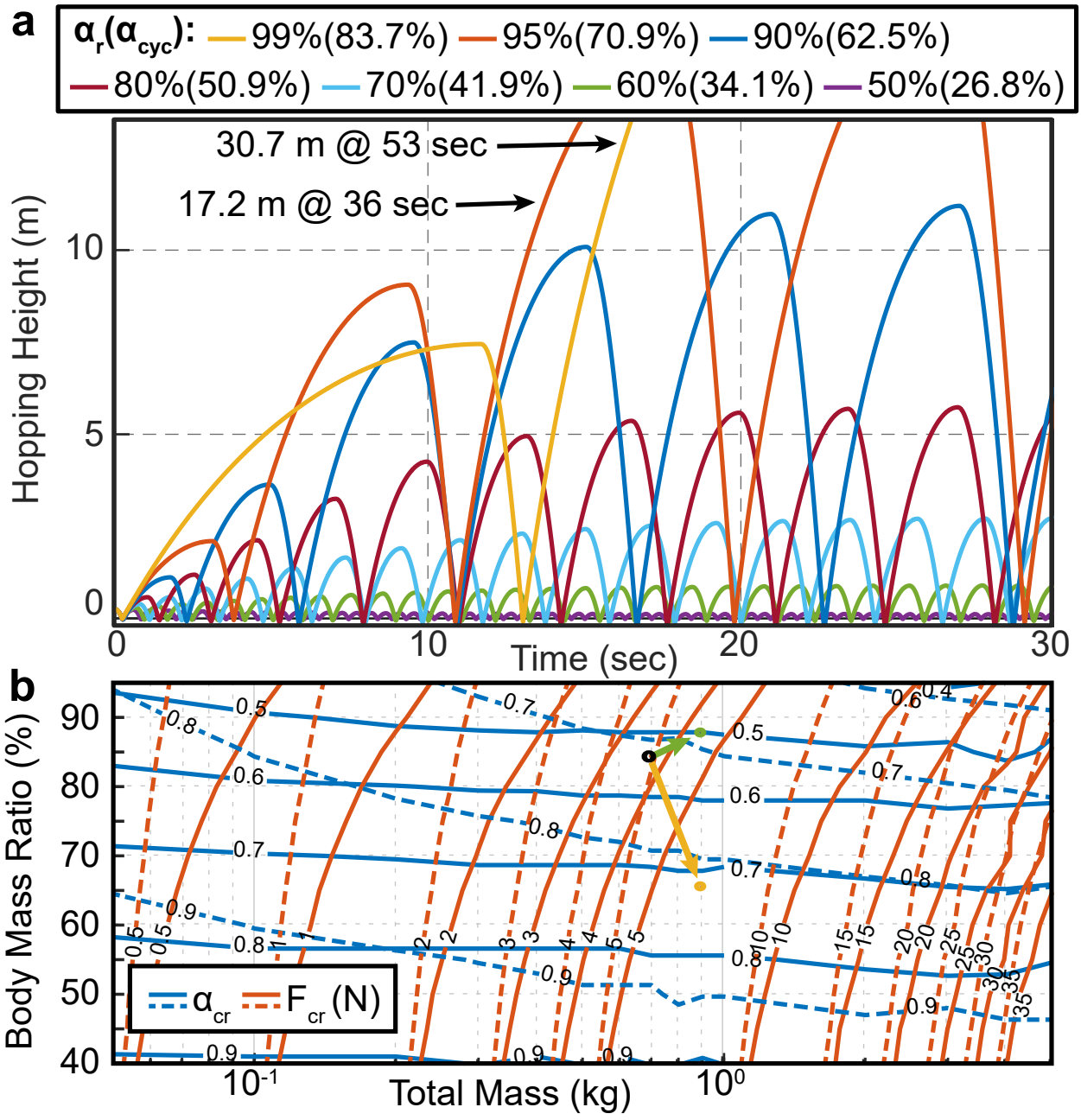

Figure 3.2: Limited rebound control input TWR, $\alpha_r \leq 99\%$. a) Energy accumulation in the MultiMo-MHR with varying α_r, and corresponding cycle averaged TWR, α_{cyc}, for constant hopping height. b) Critical α_r (α_{cr}, blue lines, minimum TWR) and corresponding critical thrust, $F_{cr} = \alpha_{cr} m_T g$ (red lines) for continuous hopping at 0.25 m (solid lines) and 3 m (dashed lines). The MultiMo-MHR (black dot) with 200 g added to either the body (green dot) or leg (yellow dot) is also shown.

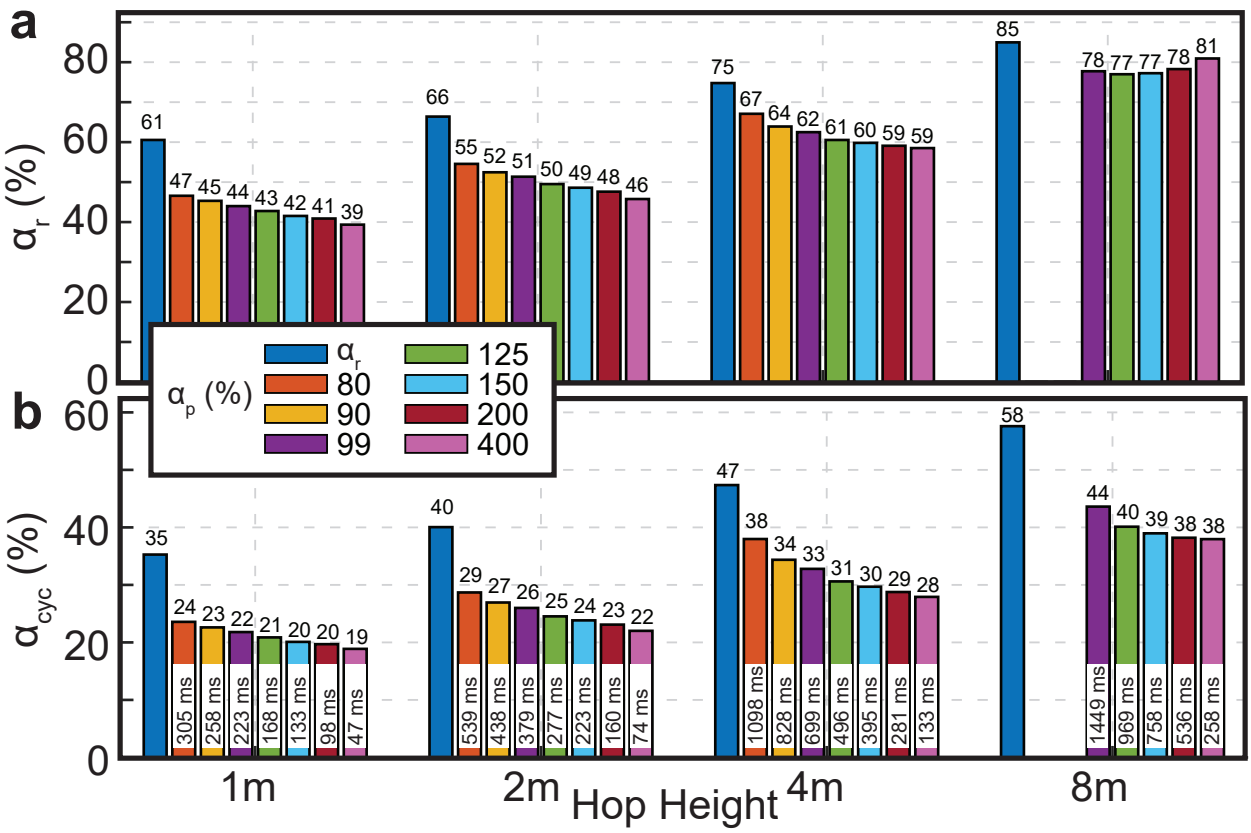

Figure 3.3: Hopping at various prescribed rebound control input TWRs (α_p = [α_r, 80, 90, 99, 125, 150, 200, 400]%); where each is maintained for the period, shown in the lower bar, necessary to reach the desired hop heights [1, 2, 4, 8] m. The resulting α_r (a) and α_{cyc} (b) are presented.

to the leg. As an example in Fig. 3.2b, the MultiMo-MHR ($\alpha_{cr} = 53.9\%$) is represented by the black dot; where, 200 g (28.7% increase) has been added to either the body (green dot) or leg (yellow dot). The addition of body mass at [0.25, 3, 6] m hops shows a modest increase in required F_{cr} (17.5%, 23.2%, 25.0%) compared to the addition of leg mass (73.5%, 48.3%, 40.0%) respectively; with diminishing return at higher heights due to the increasing dominance of drag. This therefore suggests, larger body masses require proportionally less F_{cr}, as a function of the mass, to sustain hopping; freeing up mass for additional components and complexity.

3.3.2. Prescribed Rebound Control Input TWR, α_p

Assuming a hopping robot has a TWR greater than that required to reach a desired height, instead of an applied α_r over the full rebound phase, the robot can specify a prescribed rebound control input TWR, $\alpha_p \geq \alpha_r$, for a set period of time during rebound. Figure 3.3 shows the resulting α_r (a) and α_{cyc} (b) over $\alpha_p = [\alpha_r$,80,90,99,125,150,200,400]% for hopping heights of [1,2,4,8] m. This demonstrates that between 1 and 4 meter hops, operating at higher α_p will result in overall lower α_r and α_{cyc}, thus increasing efficiency and operational time. Moreover, comparing a 1 m to 4 m hop, shows that rotor-based hopping robots with greater TWR can operate at higher hopping height with equal efficiency [$\alpha_{cyc}(1\text{ m}, \alpha_p = \alpha_r) = 35\%$ and $\alpha_{cyc}(4\text{ m}, \alpha_p = 90\%) = 34\%$]. At 8 m the α_r is not monotonically decreasing with increasing α_p, indicating that drag is becoming a dominate force. This suggests that as the height increases, the minimum α_{cyc} may not be at the maximum α_p available. Therefore, rotor-based hopping robots should, where possible, be designed with a TWR greater than that required for the desired hopping height to improve efficiency; where the majority of the improvement is achieved at modest increases in the TWR.

3.4. Robot Design

The MultiMo-MHR (Table 3.2), is designed with a monopedal hopping leg for strong and efficient hopping locomotion, and a quadrotor configuration for energy input and control

(Fig. 3.5a); similar to existing robots [1, 9, 10, 33]. However, unlike other systems, the body surrounds the leg rather than being above it; minimizing the structural elements required to connect the body and leg. This also maximizes the available leg motion, d_{spr}, for a given robot height, d_{robot}; whereas, all other hopping robots are limited to a size of $d_{spr} \leq d_{robot}/2$, the MultiMo-MHR ($d_{robot} = 1.37 d_{spr}$) is limited to a size of $d_{robot} \geq d_{spr}$. Given a specified d_{robot}, this design difference allows the required energy ($E_{spr} = 1/2 K_{spr} d_{spr}^2$) to be stored with minimum ground reaction forces ($F_{GRF} = K_s d_{spr}$) and therefore minimum loading on the leg structure.

3.4.1. Energy Storage

A major loss of energy in jumping and hopping robots is in the connection between the energy input device and the main elastic energy storage (i.e., spring) device. These mechanisms tend to be complex and experience high forces, which produces significant resistance to the energy release. To remove this impediment to hopping locomotion and maintain high efficiency, the MultiMo-MHR is designed with no connection between the main elastic energy storage device and the energy input device, and therefore these losses are entirely removed. While the MultiMo-MHR utilizes unconnected aerodynamic energy inputs, unconnected inertial energy inputs could be used as well.

Developing a high-specific energy (SE = energy/mass) hopping robot requires consideration of the hopping energy storage medium's SE. Low speed experimental testing (equipment limitations) of elastomer bands were conducted. Testing consisted of having the bands cycled 10 times at 150 mm/min then stretched to breakage at the same rate. The initial cycle pre-stretch simulated a used band then the stretch to break enabled a full understanding to its limit. The results from the break test are shown in in Figure 3.4, correlating the measured stress strain plot for three tested bands. The testing shows a maximum stress (average cross-sectional), strain, and SE of 11.2 MPa, 6.2, and 15800.3 J/kg respectively; where the high speed nature of hopping locomotion and viscoelastic effects of elastomers are expected

Table 3.2: Robot Parameters

Parm.	Description	R1	R2	Units
m_B	Body Mass	0.588	0.562	kg
m_L	Leg Mass	0.109	0.122	kg
m_T	$m_B + m_L$			
I_{cmB}	Body CM Rotational Inertia	0.0012	0.0012	kg m^2
I_{cmL}	Leg CM Rotational Inertia	0.0019	0.0021	kg m^2
K_s	Main Power Spring Const.*	622	828	N/m
K_{lb}	Leg-Body Spring Const.†	$400K_s$	$400K_s$	N/m
b_{lb}	Leg-Body Damping Coef.†	100	100	N s/m
L_1	Leg Top to CM_L	0.1053	0.0795	m
L_2	Leg Bottom to CM_L	0.2821	0.2899	m
L_3	Body Offset from Top Leg	0.1191	0.1175	m
d_{mtrx}	Motor Offset from Body CM	0.0820	0.0820	m

*: Calculated as the number of bands times ∼103.5 N/m per band (experimental),†: Estimated from simulation data.

to increase the stress at lower strains. To increase robustness and longevity of the elastomers, the MultiMo-MHR operates with a maximum strain of 2.1; verified with no significant difference observed in elastomer stretch between two groups of 10 hops at 1.5 m separated by $\sim$ 100 hops (t(18) = 1.49, p = 0.15). Comparing elastomer bands to optimal helical metallic springs (SE $\sim$ 575 J/kg), the elastomers have the potential for $\sim 27.5x$ greater SE [15]. The MultiMo-MHR possesses 6 (Robot 1, R1) or 8 (Robot 2, R2) bands resulting in a SE of $\sim$ 24 J/kg ($\sim$ 16.6 J) and $\sim$ 32 J/kg ($\sim$ 22.2 J) for R1 and R2 respectively, reaching a height of 4 m as compared to the MultiMo-Bat (SE = 78.1 J/kg) and 7g Jumper (SE = 22.4 J/kg) reaching heights of just 3.6 and 1.4 m respectively [67, 77]. Hopping robots, with rebound phase LE input, can achieve equal hopping heights to their jumping counterparts but with significantly lower SE, due to the inherent lower liftoff energy which is supplemented with LE input in rebound.

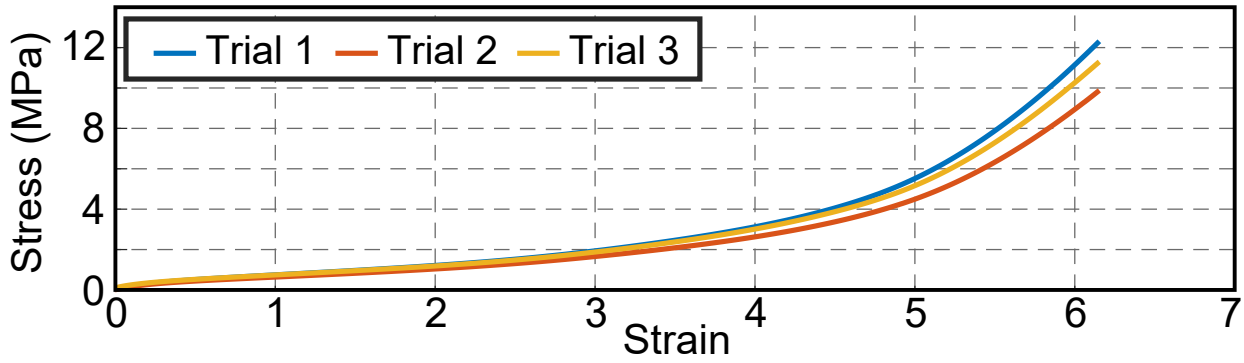

Figure 3.4: Rubber band stress-strain limit test for three bands. The results show a maximum stress and strain of 11.2 MPa, 6.2, with the total specific energy of the bands found to be 15800.3 J/kg.

3.4.2. Leg Design

The monopedal design produces a hopping leg with a single prismatic joint aligned with the locomotion forces. This significantly decouples the friction from the locomotion forces; as the forces are predominantly parallel to the friction direction. Additionally, with the inherent simplicity of the design and primarily compressive forces on the structure, the design produces an extremely strong and robust leg.

Mass is usually assumed to scale isometrically; i.e., as body mass increases so too would the leg mass to withstand the increased locomotion forces. However, due to the touchdown in hopping robots, the maximum force experienced by the leg is typically not applied by the body during the stance phase and is instead applied by the foot at touchdown; and is observed in the dynamic model. Additionally, the natural frequency of the foot-ground interface tends to be significantly greater than that of the body-leg. Assuming high damping in the foot-ground interface, the high impact forces will be dissipated before the body can apply any significant forces. Therefore, the determining factor in required leg strength is the leg mass, foot-ground interface stiffness, and touchdown velocity.

Reducing leg mass is generally challenging, and drag couples the total mass to the touchdown velocity at a given height. However, the force during touchdown is highly sensitive to the stiffness of the foot-ground interface, which can be adjusted by modifying the foot con-

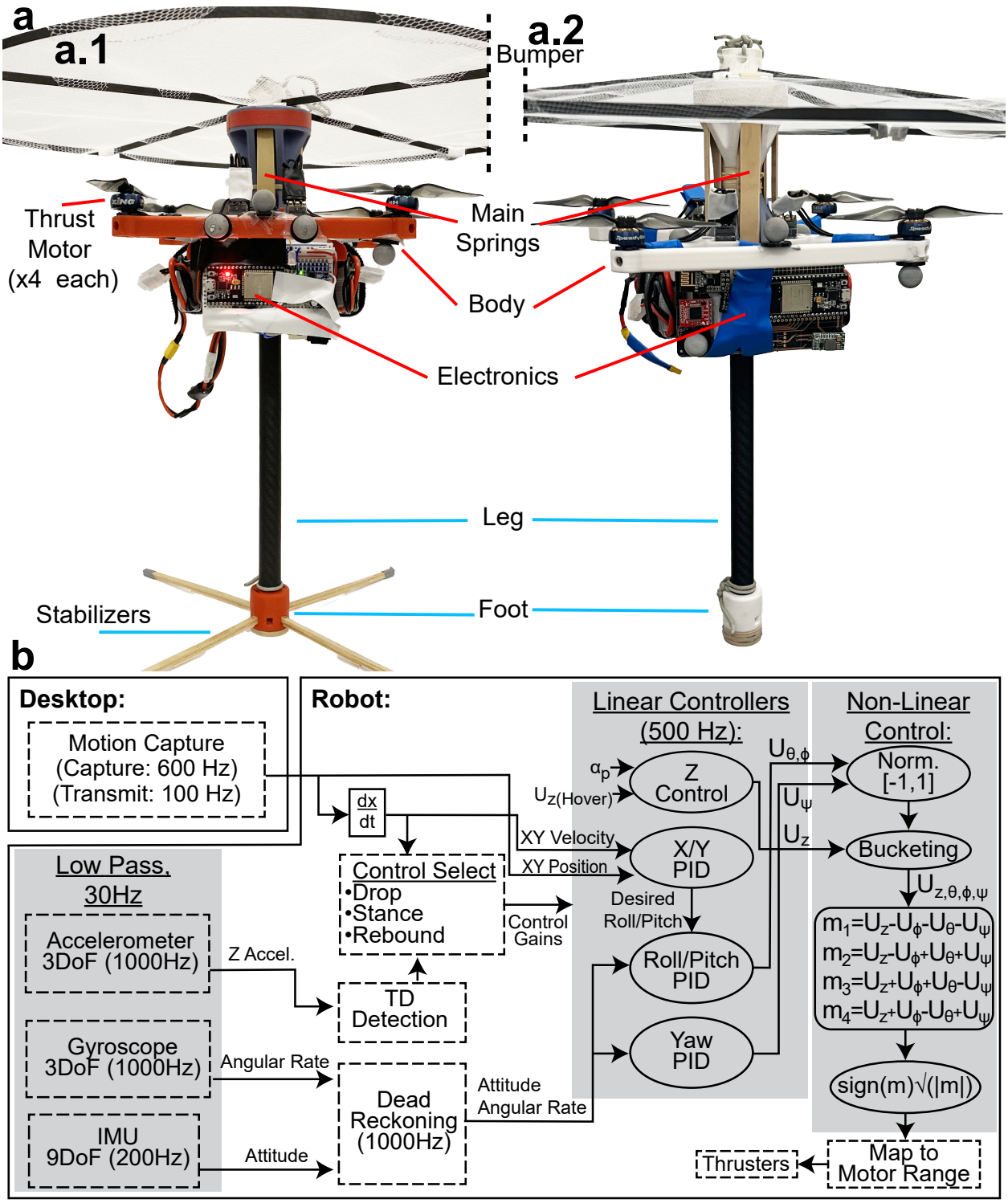

Figure 3.5: Robot Design: a) MultiMo-MHR with monopedal hopping leg and quadrotor for control. a.1) Robot 1 (R1). a.2) Robot 2 (R2). b) Control scheme.

tact material and thickness; where the MultiMo-MHR uses one to several layers of 3.175 mm thick natural rubber (40A) to vary foot compression distance. For example, if we assume a touchdown velocity of $\dot{y}_{TD} = 7.5$ m/s (corresponding to a 4-meter hop) and a maximum foot compression of $d_f = 5$ mm, energy balance predicts a peak leg acceleration ($\ddot{y}_{max} = \dot{y}_{TD}^2/d_f$) of 1147 g's (1222 N) at touchdown. In comparison, the maximum body acceleration during stance is 25 g's (146 N); thus, the touchdown forces are $\sim 8x$ greater. The MultiMo-MHR's theoretical maximum hop height of 30.7 m ($\alpha_r = 99\%$, touchdown velocity = 12.8 m/s) approaches terminal velocity (12.9 m/s) which results in a peak leg g-force and load of 3393 g's (3615 N); where losses in the dynamic model show these to be over estimates. This therefore decouples the leg mass from the body mass allowing for independent scaling and design.

Given the high compressive strength of carbon fiber tubes, the likely failure would be in buckling. Using the Euler bucking equation, the critical buckling force (CBF) is calculated as, $P_{cr} = \pi^2 E_L I_L/(K\gamma_L L_L)^2$; where, the young's modulus, E_L, area moment of inertia for a tube, $I_L = \pi((\gamma_{OD}d_{OD})^4 - (\gamma_{OD}d_{OD} - 2\gamma_T d_T)^4)/64$, outside tube diameter, d_{OD}, tube thickness, d_T, outside diameter scaling factor, γ_{OD}, thickness scaling factor, γ_T, length factor, K, leg length scaling factor, γ_L, and leg length, L_L, determine the value. The ratio of the CBF divided by the CBF with scaling factors set to one, $R_{CBF} = P_{cr}/P_{cr}([\gamma_{OD}, \gamma_T, \gamma_L] = 1)$, presents the overall percent difference in CBF, P_{cr}, as the outside diameter, leg length, and tube thickness are varied. Using the MultiMo-MHR, the CBF scales, given constant tube thickness ($\gamma_T = 1$) and length ($\gamma_L = 1$), as follows [$R_{CBF}(\gamma_{OD} = 1.25) = 2$, $R_{CBF}(\gamma_{OD} = 1.5) = 3.6$, $R_{CBF}(\gamma_{OD} = 2) = 8.8$]. The corresponding mass of a tube, $m_{tube} = \pi((\gamma_{OD}d_{OD}/2)^2 - ((\gamma_{OD}d_{OD} - 2\gamma_T d_T)/2)^2)\gamma_L L_L \rho_{cf}$, with material density, ρ_{cf}, can also be calculated. As with the CBF, the mass scaling is calculated as, $R_m = m_{tube}/m_{tube}([\gamma_{OD}, \gamma_T, \gamma_L] = 1)$; where the MultiMo-MHR's tube mass scales as follows [$R_m(\gamma_{OD} = 1.25) = 1.27$, $R_m(\gamma_{OD} = 1.5) = 1.53$, $R_m(\gamma_{OD} = 2) = 2.07$]. Therefore, the CBF can be increased significantly without significantly increasing the mass.

Additionally, as the tube mass represent only $\sim 3.2\%$ of the total mass, doubling it

would result in minor changes to the system's performance with $\sim$ 8.8x increase in the CBF. Finally, CBF can be increased with no change in mass by reducing the tube thickness as the outside diameter increases; this can be seen by solving the mass scaling equation, R_m, for the thickness scaling factor, γ_T, and substituting it into the CBF scaling equation, R_{CBF} with $R_m = 1$. The CBF of the MultiMo-MHR $P_{cr}(E_L = 150$ GPa, $d_{OD} = 16$ mm, $d_T = 1$ mm, $K = 2$, $L_L = 368$ mm) = 3638 N, demonstrating that the current system can fully withstand ground contact at terminal velocity; where the real CBF will be greater as the mass is distributed over the leg and therefore the loading is not actually concentrated at end of the rod.

3.4.3. Control

Precise control of the TD angle is of paramount importance in high-performance hopping robots as small angular deviations from vertical at TD can lead to significant changes in the liftoff angle and control effort (Fig. 3.6, red lines). The aerial dynamics of the MulitMo-MHR follow a standard quadrotor control structure; with a body z-axis thrust (U_z), and body torques about the roll, pitch & yaw axes (U_ϕ, U_θ, U_ψ) [37]. Modifications were applied to aid control with TWR < 1. To enable control, seen in Figure 3.5b, two onboard IMUs are utilized: a MPU6050 (6-axis, 1000Hz) for ground detection and angular rates and a WT901 (9-axis + tilt sensor & DMP, 200 Hz dead reckoned to 1000Hz) for orientation. A motion capture system (Vicon, Vantage v5, 100Hz) provides the translational information sent to the robot via an RF transceiver (nRF24L01). Finally, the thrust is provided by four quadrotor motors (iFlight XING2 1404 4600 KV with 4030 props), powered all by a 14.7V battery pack (two 7.4v, 2s, 2200 mAh, 35C batteries in series). Due to the electronic speed controllers (ESC), the system can only output thrust in a single direction (i.e., cannot reverse motors); this enables energy input during rebound, but only energy removal in drop.

The PID controller, hand tuned, with a nonlinear transformation is seen in Fig. 3.5b. The controller is divided into three components: horizontal [X,Y], orientation [roll, pitch, yaw]

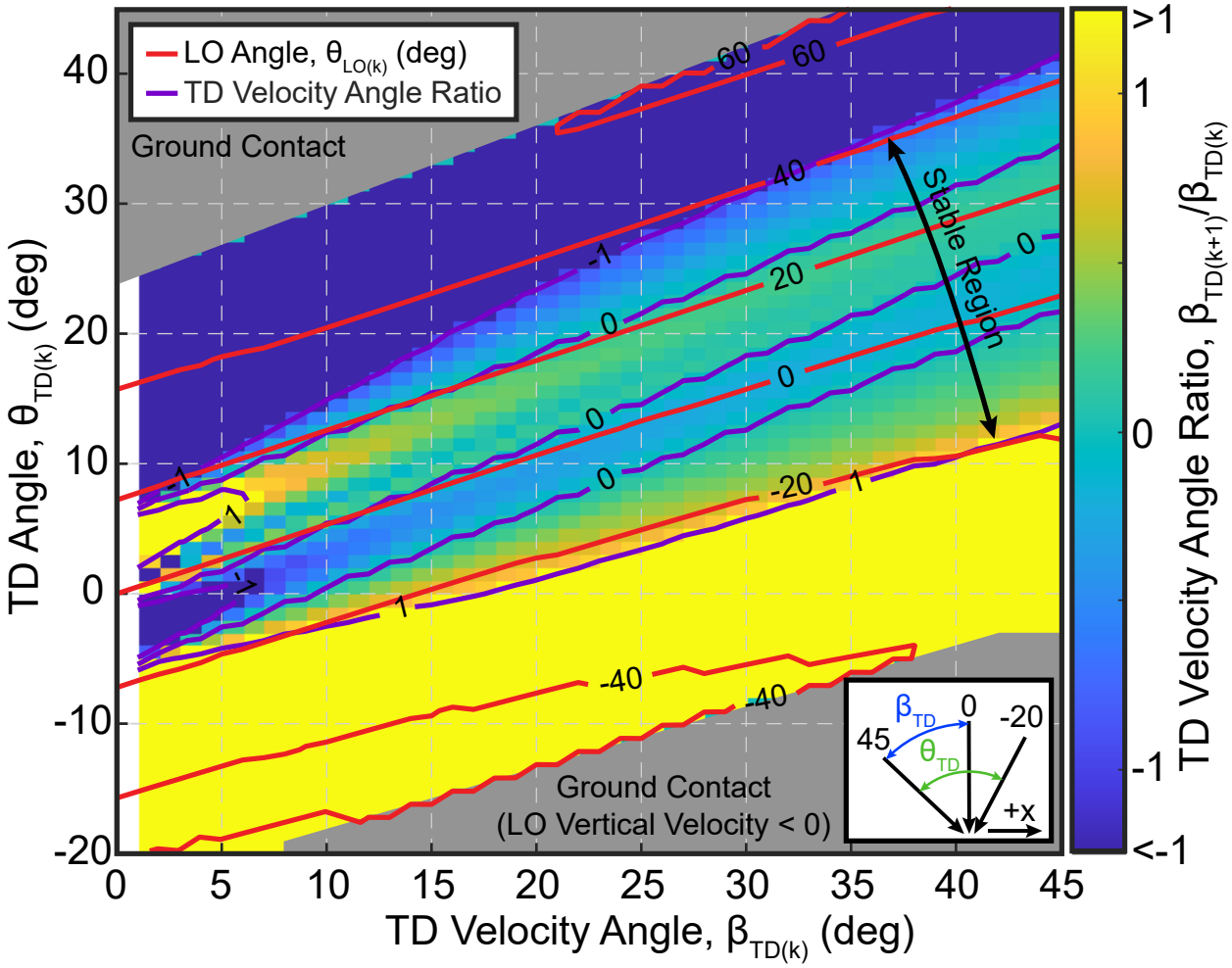

Figure 3.6: Poincaré map of the evolution of the TD velocity angle β_{TD}, using the presented dynamic model (1 m hops), as a function of $\beta_{TD(k)}$ and $\theta_{TD(k)}$; where the stable region, $[-1 < \beta_{TD(k+1)}/\beta_{TD(k)} < 1]$, indicates the range over which β_{TD} reduces. The inset shows the relationship between the angles and positive x-direction.

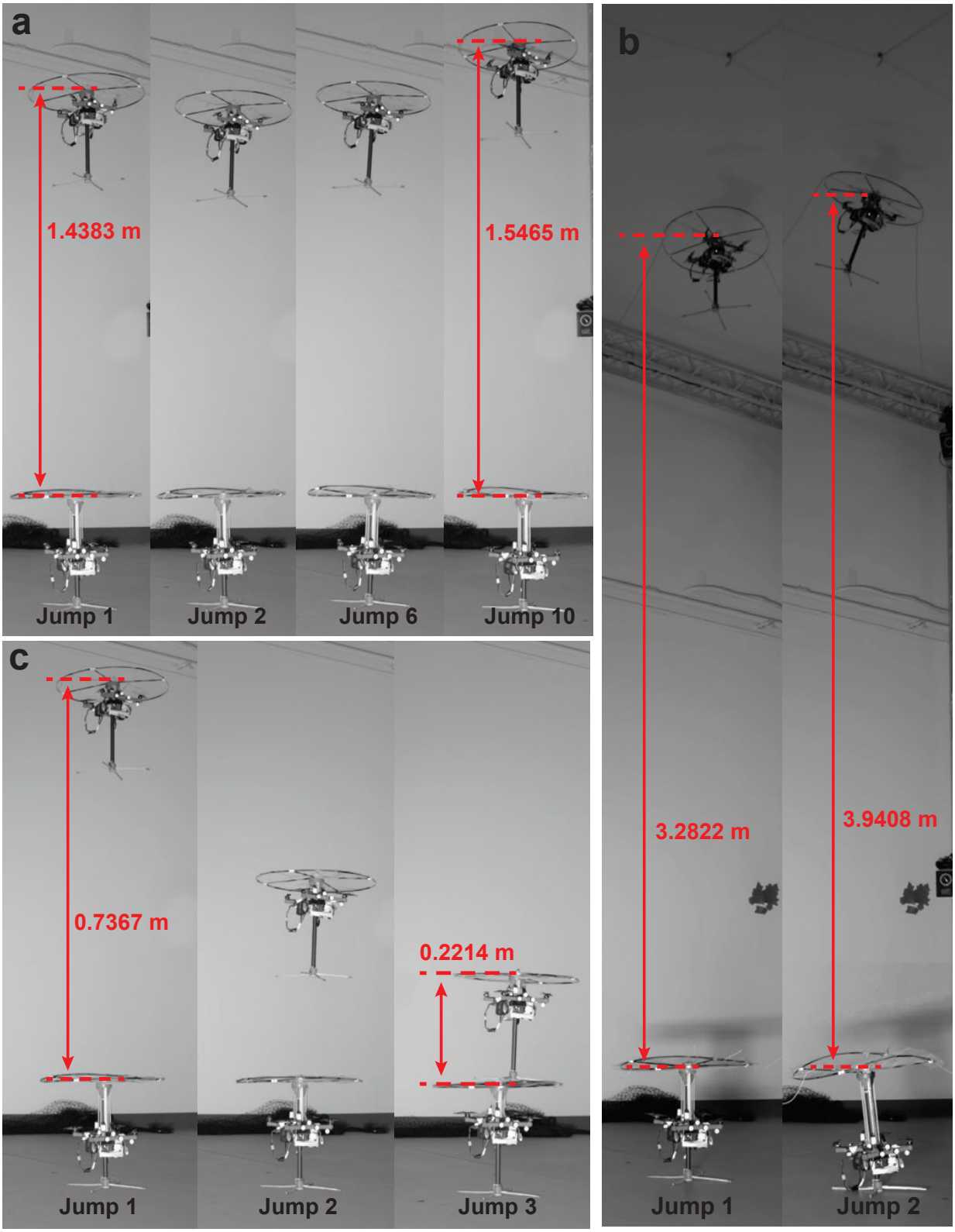

Figure 3.7: Snapshots of high-speed video (R1 Criteria-2): a) Constant height ($h_d = 1.5$ m) $\alpha_p = 74.2\%$, b) Increasing height ($h_d = 1.5$ m) $\alpha_p = 92.2\%$, c) Decreasing height ($h_d = 1.5$ m) $\alpha_p = 0\%$.

and LE input [Z/vertical]. The horizontal PD controllers, active only during rebound, are setup as cascades, generating desired angles for the the roll and pitch controllers. The orientation PD controllers, active during rebound and drop, drive the platform to the desired angles. Lastly, the LE input applies a constant feed-forward thrust in rebound ($U_z = \alpha_p * U_{z(Hover)}$). These combined linear controllers are then sent through a nonlinear transformation. The U_ϕ, U_θ, and U_ψ are first normalized by $[\frac{\pi}{2}, \frac{\pi}{2}, \pi]$, respectively, and saturated to a range of [-1,1]. A bucketing scheme then allocates power: U_ϕ, U_θ up to 60%, U_z 20% plus prior remaining, and U_ψ 20% plus prior remaining (e.g., if U_ϕ, U_θ use 10%, U_z can have 70%). The signals are combined relative to the motor configuration and square rooted, accentuating near zero control inputs to maintain the precise TD angle; in this case [roll, pitch] = 0. Using the control stability metric presented by [1] (Fig. 3.6), the Poincaré map of the evolution of the TD velocity angle as a function of the TD angle ($\theta_{TD(k)}$) and TD velocity angle ($\beta_{TD(k)}$) shows the stable region of the controller as $[-1 < \beta_{TD(k+1)}/\beta_{TD(k)} < 1]$; where the TD velocity angle reduces at the subsequent hop $(k + 1)$. Experiential results showed stable operation for [roll, pitch] angles and [X, Y] velocities at TD of 0.78 ± 4.21, -1.74 ± 1.66 deg and -0.03 ± 1.17, 0.001 ± 0.99 m/s; and for LO -1.14 ± 5.39, -2.39 ± 3.08 deg and -0.12 ± 1.47, 0.16 ± 0.62 m/s.

3.5. Experimental Results and Discussion

To validate the model, experimental trials were conducted. Two iterations of the MultiMo-MHR, Fig 3.5a, were utilized, robot 1 (R1, n = 843 hops total) and a slightly updated version robot 2 (R2, n = 190 hops total). R1 uses onboard orientation measurements from the IMU as compared to R2 that uses motion capture data for orientation control; this enhances system stability and enabled the removal of the foot stabilizers.

The experimental procedure begins by calibrating the sensors with the robot in an upright and stationary position. The robot is then flown up to the desired drop height, ranging from 0.5 to 4 m, and begins hovering. Once stabilized, the hover motor value ($U_{z(Hover)}$) is

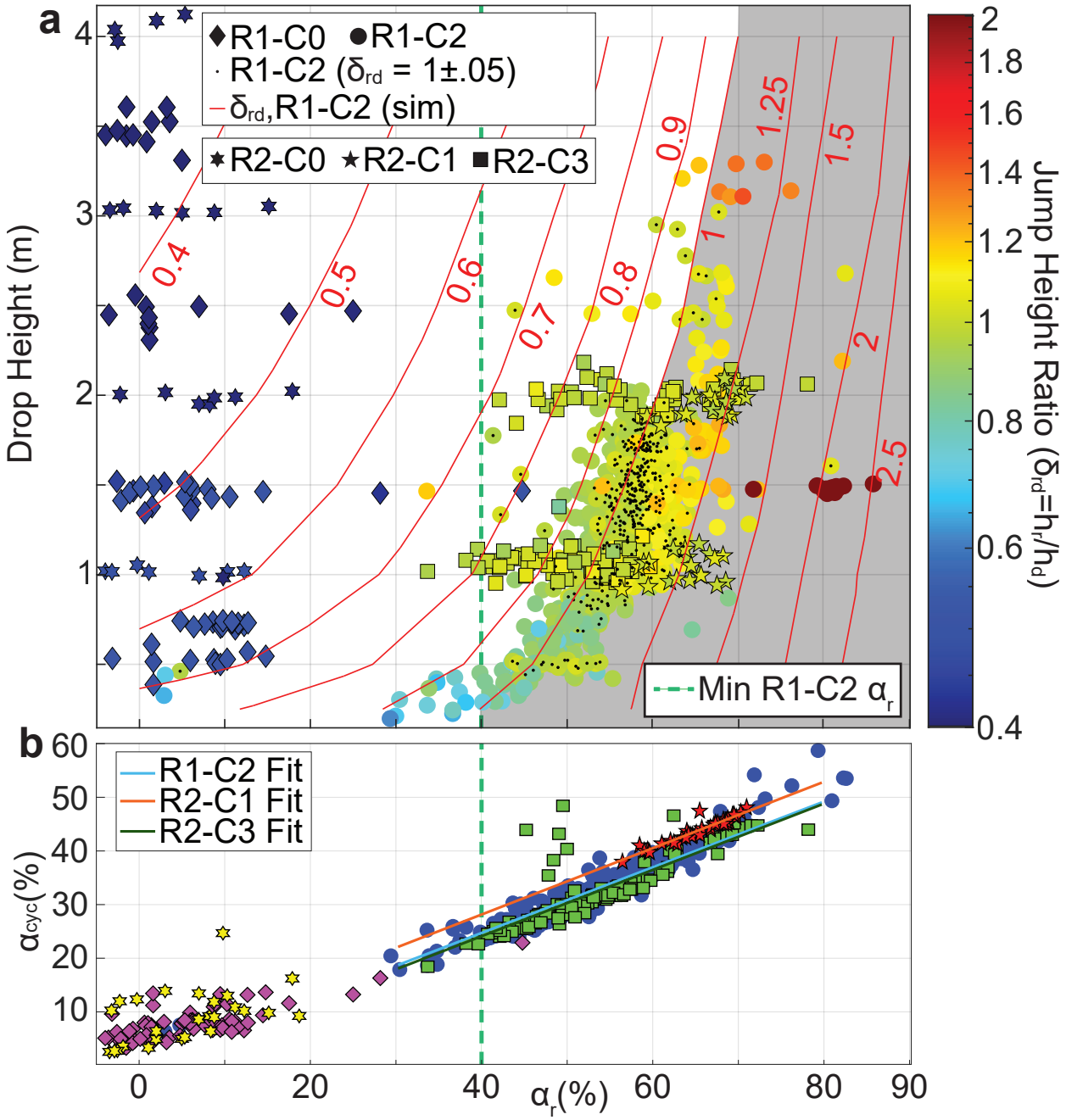

Figure 3.8: Experimental trials (R1 n = 843 hops, R2 n = 190 hops): a) Hopping height ratio (δ_{rd}) as a function of the rebound control input TWR (α_r) and drop height (h_d); where energy can be accumulated (gray region), dissipated (white region), and remain constant ($\delta_{rd} = 1$). b) Cycle average TWR (α_{cyc}) as a function of α_r

.

Table 3.3: Experimental Trial Sets

Robot#- **Criteria#**	h_d (m)	α_p **(TWR)**
R1-C0	0.5,1.5,2.5,3.5	[0], [0], [0], [0]
R2-C0	1,2,3,4	[0], [0], [0], [0]
R2-C1	1,2	[73,77], [73,77]
R1-C2	0.5,1,1.5,2	[77.5], [70.7,74.2,77.5], [74.2,80.6,92.2], [74.2]
R1-C3	1,2	[80, 99, 125], [80, 99, 125]

Each height corresponds to the matching α_p range in the respective [ ].

recorded, enabling repeatable actuator performance per trial with changing battery voltage. Then drop is initiated with only the orientation controller active. All controllers are then deactivated at TD. At LO the orientation controller is activated, with the horizontal and LE input controllers activating during rebound under specific criteria; drop then follows restarting the cycle.

Rebound controller activation requirements include: criteria-0 (C0) not activated; criteria-1 (C1) vertical velocity $\geq$ 0 m/s; criteria-2 (C2) vertical velocity $\geq$ 1 m/s (see Fig. 3.7); and criteria-3 (C3) elapsed time $<$ specified value. C0 is designed to determine the cycle efficiency, η_{cyc}. C1 and C2 explore the maximum hop height at limited TWRs, and C3 explores efficiency at higher TWRs. To isolate and explore the LE input control from that required for stabilization, the robot is commanded to hop vertically; minimizing stabilization inputs. All trials conducted per robot and criteria are presented in Table 3.3; at least 2-3 hops for higher accumulation/dissipation tests and up to 17 hops for constant hop height was achieved.

3.5.1. Energy Accumulation and Dissipation

Figure 3.8a shows that the experiential results for R1-C0/C2 are well aligned with the simulation data (δ_{rd}, R1-C2 (sim)). With the exception of the low drop height ($< 1m$) R1-C0 cases that show a lower ratio as compared to the simulation; likely a result of increased

Table 3.4: Experimental results at constant hop heights of 1 and 2 m

Robot#-Criteria#	**Hop Num**	α_p **(TWR)**	$h_d(m)$	δ_{rd}	α_r **(TWR)**	α_{cyc} **(TWR)**
R2-C1	20	73	1.04±0.08	1.01±0.03	63.6±3.5	42.8±2.4
R1-C2	48	74.2	1.03±0.11	0.98±0.04	55.2±2.4	34.1±1.4
R1-C2	51	77.5	1.07±0.08	1.06±0.04	59.6±2.8	37.1±1.9
R2-C3	21	80	1.06±0.04	1.00±0.03	49.5±4.4	30.7±4.0
R2-C3	20	99	1.07±0.04	1.01±0.03	48.6±5.6	28.1±3.0
R2-C3	33	125	1.07±0.09	1.00±0.07	**49.6±6.3**	**29.2±4.6**
R2-C1	20	77	1.96±0.08	1.00±0.03	67.0±3.1	44.9±2.0
R1-C2	20	74.2	1.92±0.06	0.95±0.02	57.9±2.0	35.1±1.0
R2-C3	15	80	2.02±0.07	1.00±0.03	66.7±4.0	42.2±2.3
R2-C3	16	99	1.89±0.24	1.00±0.02	53.1±6.4	33.5±5.9
R2-C3	15	125	2.03±0.08	1.00±0.05	**56.2±8.3**	**33.7±5.6**

losses in the elastomer due to their non-linear behavior.

The MultiMo-MHR (R1-C2) is able to sustain hopping at $\alpha_r \geq 40\%$ (Fig. 3.8a) as compared to $\alpha_{cr} = 53.9\%$ for R1-C1 (Fig. 3.2b) further supporting the efficiency gains by operating at increase α_p (Fig. 3.3). Assuming the robot begins in the gray region of the plot, it will accumulate energy (increasing the rebound or subsequent drop height) until it reaches the $\delta_{rd} = 1$ line. If it begins in the white region, it will dissipate energy until it reaches the $\delta_{rd} = 1$ line or a drop height of zero which indicates cessation of hopping. Furthermore, the maximum reduction in LE per cycle is the hopping cycle efficiency, η_{cyc}, which can be extracted from the C0 trials (Fig. 3.8a) and shows an as expected height dependent δ_{rd} between 0.35 (3.5-4 m drop height) and 0.50 (0.5 m drop height).

3.5.2. Hopping Experimental Efficiency

The MultiMo-MHRs (R1,R2) are shown to operate at $\alpha_{cyc} \leq 100\%$ (Fig. 3.8b), indicating higher efficiency and therefore longer operational time as compared to conventional flight; with α_{cyc} between 28% to 45% for 1 to 2 m continuous height hops (Table 3.4), and a total range of 23% to 59% for $\alpha_r \geq 40\%$ (Fig. 3.8b). This further validates our simulation

results and matches the findings from Hopcopter which operates with an $\alpha_{cyc} = 28\%$ at $h_r \simeq 0.7m$ [1].

Comparing the three control criteria C1, C2, and C3 (Fig. 3.8b, fit lines), we see similar slopes with differing vertical offsets. Confirming the simulation data (Fig. 3.3), when the motors are on for a majority of rebound (C1) we see the largest α_{cyc} for a given α_r. Figure 3.8b (fit lines) shows the cycle average reduces with C2 and is marginally lower with C3. This is further observed in Table 3.4 showing average values for 1 and 2 m hops where $\delta_{rd} \simeq 1$. Comparing R2-C1 ($\alpha_p = 73\%$) to R2-C3 ($\alpha_p = 125\%$) at an $h_d \simeq 1$ & 2 m, we see a reduction in the α_{cyc} of $\sim 13\%$ and $\sim 11\%$ respectively. This follows in Fig. 3.8a where R2-C3 trials can operate at the same height with lower α_r to R1-C2; even operating lower then the minimum R1-C2 $\alpha_r(40\%)$ value. This validates the simulation findings in Figure 3.3, indicating a larger α_p used for less of the rebound phase leads to a lower α_r and α_{cyc}; indicating increased efficiency.

3.6. Summary

Our results demonstrate a novel high-performance hopping robot (MultiMo-MHR) able to hop nearly 4 m ($\sim 2.4x$ the current state-of-the-art) with the theoretical strength to survive impact at terminal velocity; allowing for aerial deployment. Higher control input TWRs, over shorter periods of the rebound phase, were shown to improved overall efficiency. More broadly this work creates a platform with sufficient payload (sensing, communication, computation) and performance to study artificial intelligence, machine learning, and edge computing under highly dynamic locomotion conditions in a miniature robot. Additionally, as high-speed terrestrial locomotion dynamics converge to hopping type behaviors, the concepts developed here, and those that could be developed with the platform, have the potential to be broadly applicable to advancing terrestrial robotics in general [18].

4. Optimized Kalman Filter based State Estimation and Height Control in Hopping Robots

This chapter is comprised of a tentative journal article by the same name, which is currently under review. It was submitted to *IEEE Transactions in Mechatronics* on June 4^{th}, 2025.

4.1. Abstract

Rotor-based hopping locomotion significantly improves efficiency and operation time as compared to purely flying systems; where most hopping robots use the liftoff states and an assumed ballistic trajectory to determine the hopping height. However, significant aerial phase force (e.g., thrust and drag) can invalidate this assumption and lead to poor estimation performance. To combat this issue, a group has implemented multiple sensors (active and passive optical, inertial, and contact) and significant computational power to achieve full state estimation. This, however, poses a significant challenge to the development of light-weight, high-performance, low observable, jamming and electronic interference resistant hopping systems; especially in perceptually degraded environments (e.g., dust, smoke). Here we show a training procedure for a coupled hopping phase and Kalman filter-based vertical state estimator, requiring only inertial measurements, which is able to learn the characteristics of the target system, sensors, locomotion behaviors, environment, and acceleration measurement aliasing conditions. The resulting estimator, given hop heights up to 4 m and velocities up to ± 7 m/s, achieves a mean absolute percent error in the hop apex height of 12.5% with an aerial trajectory average normalized mean absolute error in position and velocity of 19% and 16.5%, respectively; while operating at 840 Hz, on a dual-core 240 MHz processor, with a total robot mass of 672 g. Due to the low mass and computational

power, the presented estimator could also be used as a degraded operational mode in cases of sensor damage, malfunction, or occlusion in more complex robots.

4.2. Introduction

Hopping robots combine the agility of aerial systems with the efficiency of terrestrial systems [1, 10, 32]; where the focus has been placed on predicting the vertical states for control, from the liftoff characteristics [19, 20]. However, this method is not applicable to the recent generation of rotor-based hopping robots that may not follow strictly ballistic trajectories in the aerial phase [1, 9, 10, 32, 33]. Therefore, a new method for continuous state estimation is necessary. This is particularly challenging in light-weight hopping robots, without the payload for significant increases in sensors, processing, and power; furthermore high-performance hopping robots, where aliasing of sensors measurements can be significant, especially during the impacts associated with hopping. The challenge is further compounded in perceptually degraded environments (e.g., dust, smoke, darkness), in contested environments in which active jamming may be present, and where sensing modalities that radiate energy into the environment may lead to detection and destruction. Currently, robot rotational state estimation can be achieved through inertial measurement units (IMUs), and translational state estimation can be achieved through GPS and combined optical flow and range finder senors; however, range finder sensors radiate energy into the environment for sensing, and can have difficultly operating in perceptually degraded environments. As IMUs are ubiquitous in robots and small in size, this work seeks to develop an IMU-based vertical state estimation methodology for hopping height control that could be extended to horizontal states in the future; fully replacing the optical flow sensor for operation in highly perceptually degraded environments.

To date numerous non-continuous jumping and hopping robots have been developed including miniature jumping robots [61, 77–82], high specific energy robots [13, 41], soft jumping robots [83, 84], multimodal robots [85], with wings for gliding [15, 62–67, 86, 87],

and inertial tails or reaction wheels for orientation control [7, 29–31, 88]. These robots have also used a wide variety of hopping leg designs including: linkage [34, 46, 51, 54], and linear [55, 57, 58, 89] legs. These robots represent a highly diverse range of characteristics and constraints. However, given the minimal impact of an IMU integration, the addition of the proposed method is highly feasible for any system.

Currently, five untethered continuous hopping robots exist: MultiMo-MHR (our system) [32], PogoDrone [9], Hopcopter [1], Salto/Salto-1P [6, 7, 29–31, 34, 90, 91], and PogoX [10, 33] and one tethered continuous insect-scale hopping robot [11]. While PogoDrone and the insect-scale hopping robot shows no form of state estimation, both the Hopcopter and Salto robots use the estimated or measured liftoff states and an assumed ballistic aerial trajectory to estimate the hopping height. However, the ballistic trajectory assumption invalidates this technique for robots with applied aerial phase forces. PogoX (robot mass = 3.65 kg) displays the most thorough state estimation to date, using an extended Kalman filter, moving horizon estimation, and visual inertial odometry, requiring multiple sensors, advanced computational hardware, large weight budget, and direct measurement of both vertical position and acceleration to estimate vertical velocity; thus posing a challenge to broad applicability. This work seeks to estimate both position and velocity using only direct acceleration measurements allowing for a broader integration across platforms while also being used in a degraded operational mode when primary sensors are damaged or malfunction in more complex systems; such as PogoX.

This paper is organized as follows. Section 1 introduces the topic, and Section 2 briefly covers the robot design with focus on the changes from previous work [32]. Section 3 discusses the hopping phase estimator, and Section 4 presents the hopping state estimator with associated training procedure and results. Section 5 explores the effects of the sensing frequency on measurement aliasing and ultimately state estimation. Section 6 presents the experiential procedure and results of the deployed state estimation technique on the MultiMo-MHR, and Section 7 summarizes the work.

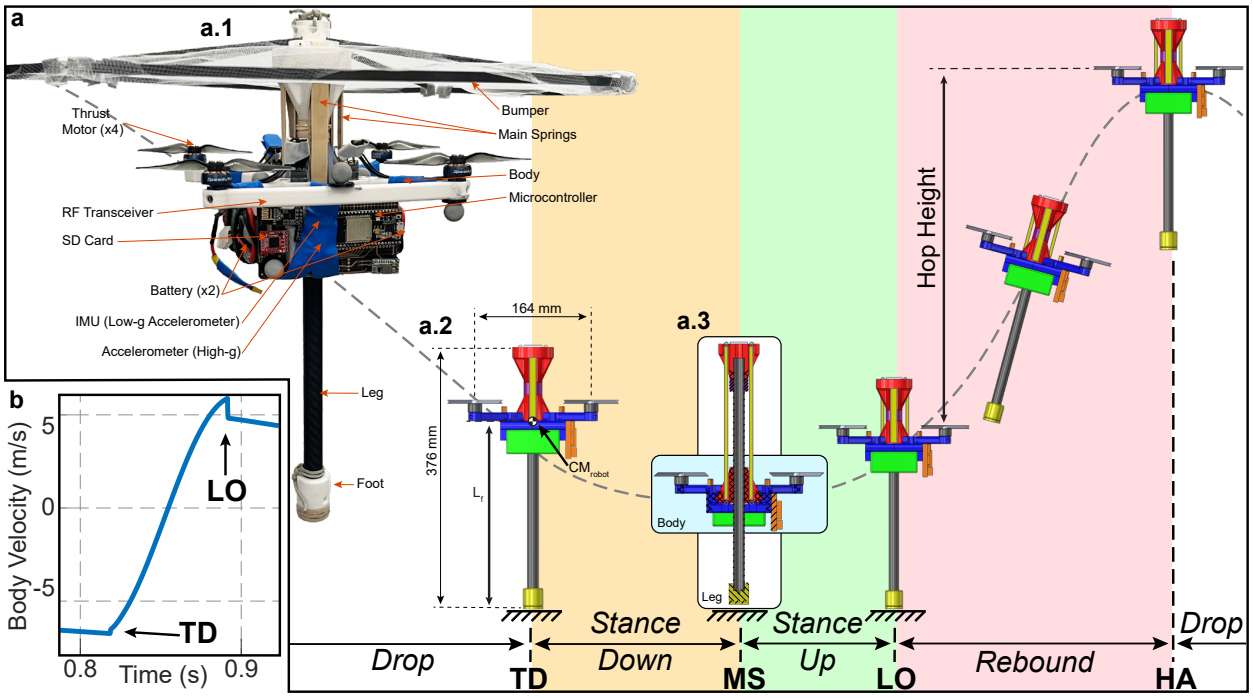

Figure 4.1: Hopping robot. a) Operational phases (drop, stance down, stance up, rebound) and transitions (touchdown (TD), max squat (MS), liftoff (LO), hop apex (HA)) a.1) Photo of the MultiMo-MHR platform with components labeled. a.2) CAD model with overall dimensions labeled. a.3) CAD model section view with spring extension (stance phases) showing the combined body and leg structures. b) Dynamic simulation showing the locally, highly non-linear, behavior of the body at TD and LO for a 3 m hop.

4.3. Robot Design

The MultiMo-MHR, developed in previous work [32], is a monopedal hopping robot that utilizes rotors for orientation control and energy input (Fig. 4.1a, Appendix Robot Parameters). The robot consists of two bodies where a single unactuated prismatic joint, with the motion-axis aligned to the leg-axis, connects the body to the leg. At touchdown the body's momentum causes it to move down the leg, extending the main power springs. The springs consist of multiple rubber bands that are able to store a maximum usable locomotion energy of approximately 34 J resulting in a maximum hopping height of over 5 meters; where additional bands can increase the height.

The MultiMo-MHR uses an on-board microcontroller (ESP32-WROOM-32, two Tensilica Xtensa LX6 cores at 240MHz) for real-time control (Fig. 4.1a). Given the significant difference in the accelerations experienced during ground contact and the aerial phases, the platform has two inertial measurement units (IMUs), including a low-g (MPU-6050: ±16g, ±2000 deg/s, 1000 Hz) and high-g (H3LIS331DL: ±100g, 1000Hz), ensuring accurate measurements tailored to the varying magnitudes of forces encountered; where the static measurement noises are $\pm 0.013g$ (MPU-6050) and $\pm 0.32g$ (H3LIS331DL). A data logger (Sparkfun OpenLog) is used to store both the sensor data and calculated values during operation. The robot is powered by a 14.7v battery pack (two 7.4v, 2s, 2200 mAh, 35C batteries in series) to provide sufficient power to the four electronic speed controllers (ESC, 20 Amp) and motors (SpeedyBee 1404 4500 kv). Finally, a radio frequency transceiver (nRF24L01) receives data from an external PC which is used to simulate additional sensors, via a motion capture system (Vicon vantage V5), while focusing on hopping height control (e.g. PMW3901 Optical Flow Sensor [horizontal positions] and WT901 [orientation]).

The previously developed controller, composed of the horizontal [X, Y], orientation [roll, pitch, yaw], and locomotion energy (LE) input [Z or height] control, is augmented with a vertical state estimator and a new LE input controller for hopping height control (Fig. 4.2).

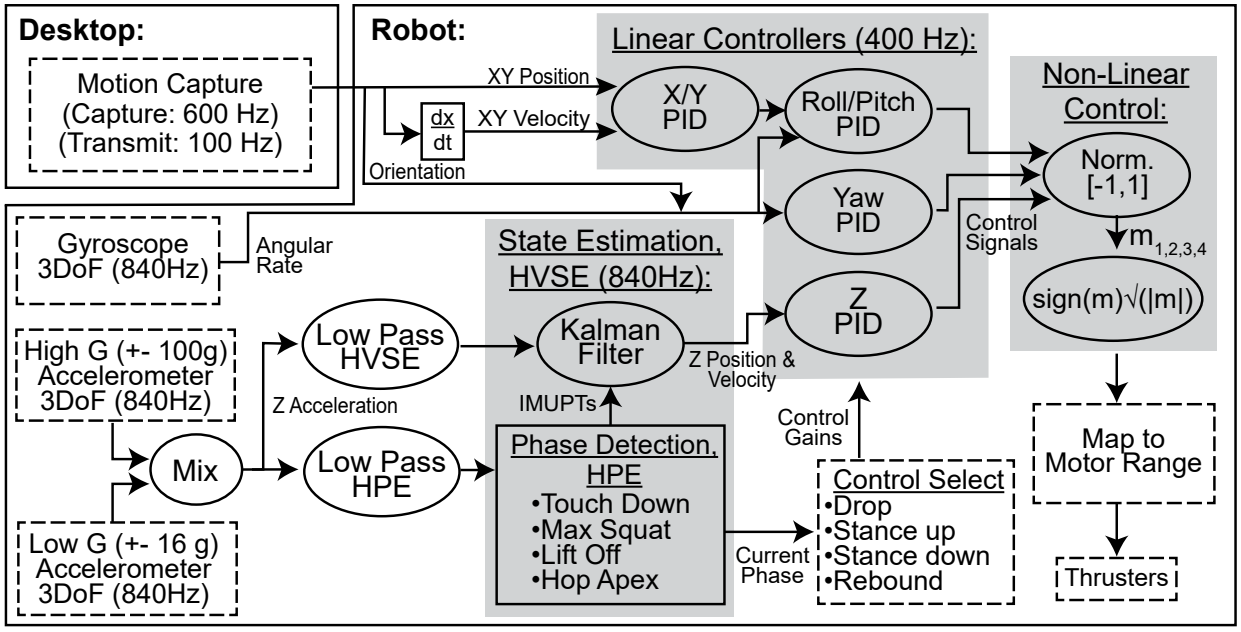

Figure 4.2: Hopping robot control block diagram.

Due to the highly damped nature of the vertical states, a proportional controller is used for the LE input control. The P-controller output is mapped to a thrust-to-weight ratio (TWR) of between 0% and 83.7% (i.e., 100% equates to the same thrust required for hover); where the minimum motor command is set to 5% duty-cycle to increase the motor's responsiveness while producing almost no thrust. The controller operates as a sliding mode control at ranges greater than 0.33 m from the desired height.

4.4. Hopping Phase Estimator (HPE)

A hopping phase estimator (HPE) is developed (Fig. 4.3) where the operational phases (green), phase transition criteria (red), and transitions (blue) are shown. The HPE determines the *drop*, *stance down*, *stance up*, and *rebound* phases through identification of the four transitions; touchdown (TD), maximum squat (MS), liftoff (LO), and hop apex (HA). Previous work in hopping robots has indicated the TD and LO transitions can be determined from the states [20], with two specific methods including leg motion through the use of a leg encoder [6,7,29–31] and ground contact through a force sensor on the foot [10,33]. However,

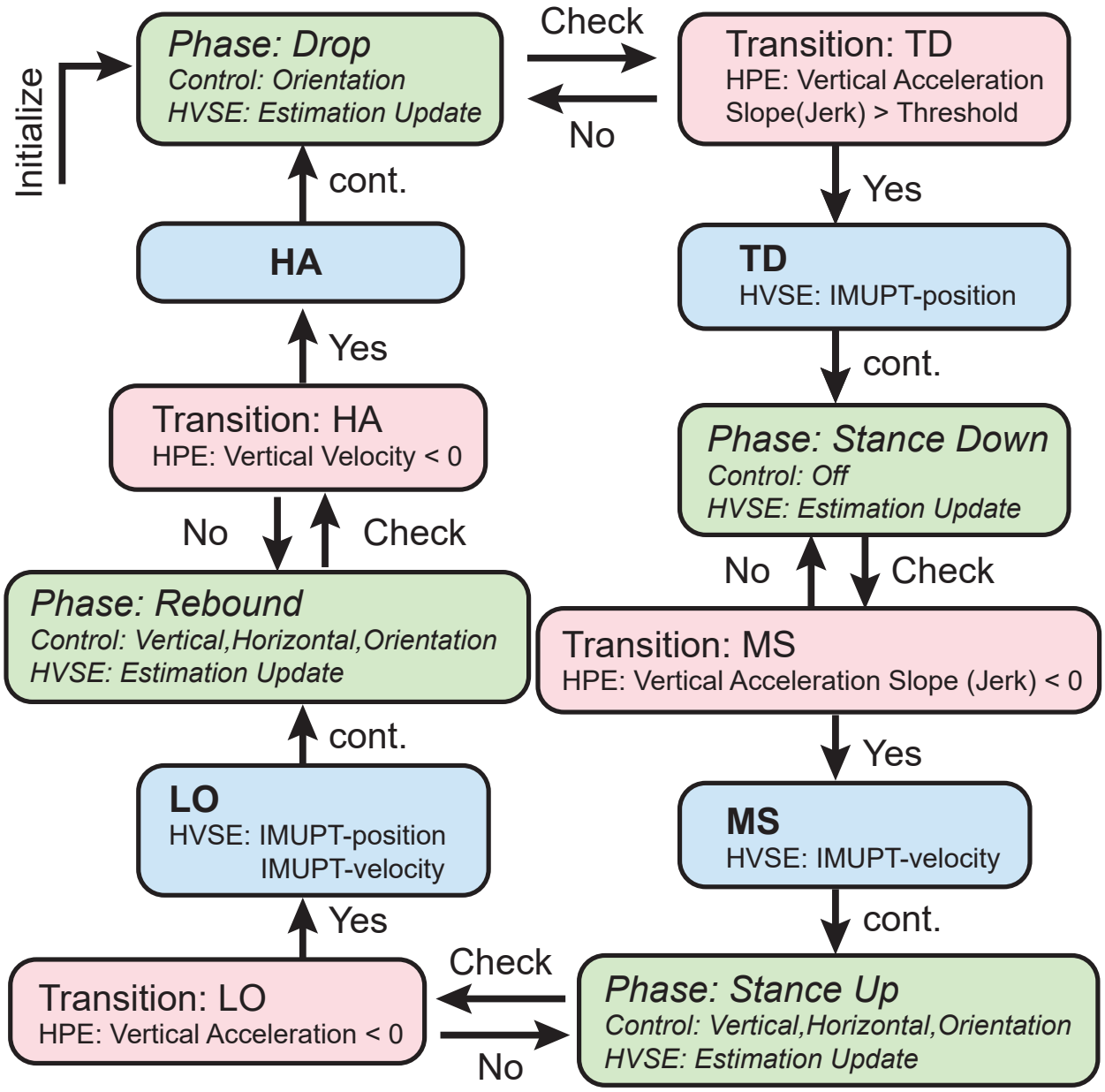

Figure 4.3: State machine diagram of the Hopping Phase Estimator (HPE), Hopping Vertical State Estimator (HVSE), and controllers showing both the prediction updates and Inferred Measurement Updates (IMUPTs) of the HVSE. Shown are phases (green), phase transition criteria (red), and transitions (blue)

both methods require an additional sensor with complex mounting requirements beyond the typical 9-axis inertial measurement unit (IMU) required for aerial phase control. Instead, inspired by work in inertial pedestrian navigation or pedestrian dead-reckoning [92–95], multiple acceleration measurements (IMU) are used to calculate the slope (jerk) and a set positive threshold determines TD. This same measurement is then used to determine MS as the point at which the jerk becomes negative. LO is determined by a single filtered acceleration measurement that is less than zero; indicating gravity is the dominant acceleration. Finally, previous works have predicted the HA from measured liftoff states and an assumed ballistic trajectory. However, as hopping height increases so too will drag, resulting in increased deviation from gravity-based ballistic motion. Furthermore, rotor-based hopping robots can significantly alter their aerial trajectory from that of ballistic motion; due to thrust produced in the aerial phase. While range finding sensors can directly measure the height and thus the HA, they significantly increase the observability of the system, provide and opening for external interference, and increased height can degrade the measurement. As proprioceptive sensors are unable to directly or indirectly detect it, the transition criteria for the HA requires an accurate estimation of the vertical velocity.

4.5. State Estimation Methodology

Two common approaches involve a Kalman filter (KF) with states ($x_k = [p_k, v_k, \theta_k, b_{ak}, b_{gk}]$) or an error state Kalman filter (ESKF) with error states ($\delta x_k = [\delta p_k, \delta v_k, \delta\theta_k, \delta b_{ak}, \delta b_{gk}]$) and nominal states ($x_k$); where, the subscript k is the time step, p_k is the position, v_k is the velocity, θ_k is attitude, b_{ak} is the accelerometer bias, b_{gk} is the gyroscope bias, and δ indicates the error between the true and estimated states. The inertial measurement unit (IMU) and accelerometer provide both the linear acceleration and rotational velocity as,

$$a^{IMU} = \hat{a} + b_a + n_a + \alpha_a \tag{4.1}$$

$$\omega^{IMU} = \hat{\omega} + b_g + n_g \tag{4.2}$$

Table 4.1: Kalman Filters

Filter Type	Filter Equations	Filter Parameters	IMUPTs (Phase/Transition)	Errors (998 hops)
KF1	KF - Prediction $x_k^- = Fx_{k-1} + Gu_k$ $P_k^- = FP_{k-1}F^T + Q$	$z_k = H\hat{x}_k + n_a,\ u_k = [0,0,1]\,a_k^w$ $F = \begin{bmatrix}1 & \Delta t\\ 0 & 1\end{bmatrix},\ G = \begin{bmatrix}0.5\Delta t^2\\ \Delta t\end{bmatrix},\ x_k = \begin{bmatrix}z\\ \dot{z}\end{bmatrix}$	Position (TD, LO) Velocity (MS, LO)	$\gamma_1 = 0.108$ $\gamma_2 = 0.329$ $\gamma_3 = 0.594$
KF2	KF - Meas. Update $K = P_k^- H^T(HP_k^- H^T + R)^{-1}$ $x_k = x_k^- + K(z_k - Hx_k^-)$ $P_k = (I - KH)P_k^-$	$z_k = H\hat{x}_k + n_a,\ u_k = [0,0,1]\,a_k^w$ $F = \begin{bmatrix}1 & \Delta t & -0.5\Delta t^2\\ 0 & 1 & -\Delta\\ 0 & 0 & 1\end{bmatrix},\ G = \begin{bmatrix}0.5\Delta t^2\\ \Delta t\\ 0\end{bmatrix}$ $x_k = [z\ \dot{z}\ \ddot{z}_{bias}]^T$	Position (TD, LO) Velocity (MS, LO) Accel. Bias (Aerial Phases)	$\gamma_1 = 0.111$ $\gamma_2 = 0.497$ $\gamma_3 = 0.903$
ESKF1	ESKF - Nominal State Prediction $x_k^- = f(x_{k-1}) + g(x_{k-1})u_k$ ESKF - Prediction $P_k^- = FP_{k-1}F^T + Q$ ESKF - Meas. Update $K = P_k^- H^T(HP_k^- H^T + R)^{-1}$ $\delta x_k = \delta x_k^- + K(z_k - H\delta x_k^-)$ $P_k = (I - KH)P_k^-$ $x_k = x_k^- + \delta x_k$	$z_k = H\hat{x}_k + n_a - Hx_k^-,\ u_k = [0,0,1]\,a_k^w$ $F = \begin{bmatrix}1 & \Delta t & -0.5\Delta t^2\\ 0 & 1 & -\Delta t\\ 0 & 0 & 1\end{bmatrix},\ G = \begin{bmatrix}0.5\Delta t^2\\ \Delta t\\ 0\end{bmatrix}$ $x_k = [z\ \dot{z}\ \ddot{z}_{bias}]^T$ $\delta x_k = [\delta z\ \delta\dot{z}\ \delta\ddot{z}_{bias}]^T$	Position (TD, LO) Velocity (MS)	$\gamma_1 = 0.127$ $\gamma_2 = 0.417$ $\gamma_3 = 0.724$
ESKF2			Position (TD, LO) Velocity (MS) Accel. Bias (Aerial Phases)	$\gamma_1 = 0.134$ $\gamma_2 = 0.423$ $\gamma_3 = 0.725$

Parameters: state transition matrix, F, control matrix, G, estimated covariance, P, process noise covariance, Q, observation matrix, H, measurement noise covariance, R, Kalman gain, K, input, u_k, and measurements, z_k. Abbreviations and descriptions: Measurement (Meas.), Acceleration (Accel.), Kalman Filter (KF), Error State Kalman Filter (ESKF), true state ($\hat{x}_k$).

where, $\hat{a}$ is the true acceleration, n_a is the accelerometer noise, α_a is the accelerometer aliasing, $\hat{\omega}$ is the true rotational rate, and n_g is the gyroscope noise. As with [96], neglecting the Coriolis forces and curvature of the earth, the state transition equations are as follows,

$$p_{k+1}^w = p_k^w + v_k^w \Delta t + \frac{1}{2}((R_k^{bw}(a_k^{IMU} - b_{ak}) + g^w)\Delta t^2 \tag{4.3}$$

$$v_{k+1} = v_k + (R_k^{bw}(a_k^{IMU} - b_{ak}) + g^w)\Delta t \tag{4.4}$$

$$R_{k+1}^{bw} = R_k^{bw}(I + [\omega_k^{IMU} - b_{gk}]_\times \Delta t) \tag{4.5}$$

$$b_{ak+1} = b_{ak} + n_{ba} \tag{4.6}$$

$$b_{gk+1} = b_{gk} + n_{bg} \tag{4.7}$$

where, the superscript w indicates the world frame, R^{bw} is the rotation matrix from the body to the world frame, $I + [\omega_k^{IMU} - b_{gk}]_\times \Delta t$ is the first-order Taylor expansion of Rodrigues' rotation formula, a^{IMU} is the measured acceleration in the IMU frame, ω^{IMU} is the measured rotation rate in the IMU frame, the subscript $\times$ indicates the cross-product operator matrix

for the bias compensated angular rate measurement, and g^w is the gravity vector.

Developing a KF from the state transition equations 4.3-4.7, shows that measurement updates impact only the measured state and its time integrals and derivatives; thereby allowing for extraction and independent development of the KF for the vertical states. Extending this to an ESKF, it does show a small amount of coupling between non-integral and non-derivative states, due to the attitude error impacting the velocity error [93]. However, assuming small attitude errors and therefore limited coupling, this isolation and development can be applied to both KFs and ESKFs. This allows for a focused study on managing significant and variable aliasing in acceleration measurements (MultiMo-MHR: Nyquist frequency $\sim$420 Hz). The challenge is further compounded by the fact that the most significant aliasing occurs at the regions of maximum energy transfer during TD and LO; as seen in the previously developed model (Fig. 4.1b) and experimental testing [32].

The used Kalman Filter concentrates on the purely vertical axis in the global reference frame, with all inputs (i.e., accelerometer control inputs) being rotated into the same frame. Naturally, the filter operates under a series of base assumptions. One is an assumed non-rotating vertical acceleration, as the measured acceleration is pre-rotated into the world frame using orientation data from the motion capture system and deadreckoned to the control rate with the on-board gyroscope. We assume there is no error in our measured orientation; however, any error (noise or other obstructions) in the orientation data will lead to an estimation error. Additionally, the filter is naturally linear by nature, which can degrade our estimator's performance under increased drag or with inaccuracies/high variability in acceleration. System non-linearities are marginally accounted for through system optimization to compensate for the drift/error they cause, although higher/larger system dynamics may require advancing to an extended Kalman filter or other nonlinear approach. The controller is also only observable at the points of the position and velocity IMUPTs, given that those are points where a measured position or velocity occurs. Increasing the observability of the system will require additional sensors, like a downward-facing Lidar, to provide more con-

tinuous position readings. This reason is why the optimization strategy enables improved performance, as it accounts for the estimation drift when the estimator is unobservable.

4.5.1. Hopping Vertical State Estimator (HVSE)

Differing from [1, 9], with thrust-to-weight ratios (TWRs) ≥ 1, the MultiMo-MHR operates with a maximum TWR limited to 83.7%. Operating at TWR < 1 poses an additional challenge as the true *drop* phase transition is determined by the dynamic behavior and not the controller; where an early estimated *drop* phase transition reduces the maximum performance while a late *drop* phase transition reduces hopping efficiency. Vertical state estimation at a TWR of $\sim$90% has been shown, however, the work required both position and acceleration to estimate vertical velocity [10, 33].

Previous studies in pedestrian tracking employing Kalman Filters (KF) to estimate both position and velocity, have shown that assuming zero velocity of the foot during ground contact can allow for inferred velocity measurements, termed zero-velocity updates (ZUPTs), which can significantly reduce drift in the state estimation [92, 95]. Further studies have added zero-altitude updates [93] to reduce the drift in the vertical axis as well. Expanding on the concept for vertical state estimation in high-performance hopping robots, three of the four phase transitions including touchdown (TD), max squat (MS), and liftoff (LO) as well as the motor drive commands allow for inferred-measurements updates (IMUPTs) including:

IMUPT-position (TD,LO) $= [0, 0, 1]\, R^{bw}[0, 0, -L_f]^T$ at TD and LO where the vertical position is inferred from the known distance between the robot's center-of-mass and the foot ($L_f = (m_B L_3 + m_L L_1)/(m_B + m_L) - (L_1 + L_2)$) without spring extension.

IMUPT-velocity (MS) $= 0$ at MS where the vertical velocity is inferred from the body's change in direction.

IMUPT-velocity (LO) $= v_{LO}\delta_{vLO}$ at LO where the vertical velocity is inferred from an understanding of the system losses, momentum transfer, and inherent acceleration aliasing at liftoff. The scaling factor $\delta_{vLO} = (c_{vel2} v_{LO}^2 + c_{vel1} v_{LO} + c_{vel0})(c_{ch1} h_{ch} + c_{ch0})$; where,

liftoff velocity estimate, v_{LO}, and preset commanded height for the subsequent hop, h_{ch}, determine the velocity adjustment (Fig. 4.7b.1 ∼2.06% mark). The $c_{vel2,vel1,vel0}$, and $c_{ch1,ch0}$ are the v_{LO} and h_{ch} fit coefficients respectively, and account for estimation errors during the stance and subsequent rebound phases.

IMUPT-acceleration (Aerial Phases) $= [0, 0, 1](R^{bw}F_{U1}/m - g^w)$ during the aerial phase where the acceleration is inferred from the total commanded motor thrust, $F_{U1} = c_{m1}(\sum_{i=1}^{4} d_{mi}) + c_{m0}$; where $c_{m1,m0}$ are the fit coefficients to convert from duty-cycle to thrust, d_{mi} are the commanded motor duty-cycles, m is the overall robot mass.

where, Appendix Robot Parameters shows the variables.

Based on the identified IMUPTs, two Kalman filters (KF1, KF2) and two error state Kalman filters (ESKF1, ESKF2) are developed (Table 4.1). The vertical acceleration input $u_k = [0, 0, 1]\, a_k^w$ is composed of the world frame acceleration measurement $a_k^w = R_k^{bw}(a_k^{IMU} - b_{ak}) + g^w$. These filters required both a process and measure noise covariance matrix containing up to four additional parameters including the standard deviation of the acceleration (σ_{az}), position (σ_{pz}), velocity (σ_{vz}), and bias (σ_{bz}) measurements. Additional adaptability is included in the state estimation, by first low pass filtering the accelerometer measurements resulting in two cutoff frequency parameters, f_{HVSE} and f_{HPE}, for the HVSE and HPE, respectively. Finally, as two accelerometers are used, the switching g-force parameter, g_s, is included to determine at which point to switch from the low-g to high-g accelerometer measurements. The total Kalman filter parameters are presented in Table 4.2; highlighting the challenge to hand tuning these filters.

4.5.2. HPE and HVSE Training

Initial experimental trials (n=998 hops using the true state for control) were conducted at hopping heights between 1 and 4 meters; where the acceleration readings from both the low-g and high-g accelerometers were captured (robot approx. 840 Hz) along with the true state data from a motion capture system (Vicon Vantage, 600 Hz). Implementing in simulation

(MATLAB) a particular HPE and HVSE on the captured experimental data, allows for an exact recreation of what the robot would experience and the associated state estimation, without the risks and time associated with testing on the real robot. Therefore, assuming the initial experiments span the complete range of desired robot operational characteristics, this allows for a complete understanding of the HVSE error including all environmental (e.g. wind, substrate properties), locomotion (e.g., hopping heights, traveling speeds), control (e.g., controller driven behaviors), and sensor (e.g., noise, positioning) characteristics.

The experimental trials used to optimize the estimator concentrated on vertical hopping. Some angular deviations were present due to impact disturbances; however, the system remained within small, near-zero orientation angles. We are solely concentrated on interpolating within the space of experimental trials, not extrapolating performance under other conditions or surface types. For this reason, it is uncertain exactly how the vertical estimation will perform under larger dynamics or on different surfaces, and would require the acquisition of trials encompassing the range of desired hopping regions and optimization across that region. Likewise, we assume a solid sensor calibration that does not vary significantly; however, should the system encounter variability or changes from its original calibration, a recalibration would be required, with the potential for the added accelerometer bias term to adjust for calibration errors.

The ability to capture large amounts of experimental data and exactly simulate the HVSE errors, suggests that the Kalman filter parameters can be learned from the data. Therefore, due to the challenging parameter space with significant local minima, a genetic algorithm was developed to optimize the HPE and HVSE parameters listed in Table 4.2. To control the hopping height of the robot, the most important parameter is the hop apex height, z_{HA}. Therefore, the cost function ($\mathcal{L}_c$) is set as the mean-absolute-percentage-error (MAPE, γ_1)

in the apex height as,

$$\min \mathcal{L}_c = \begin{cases} w_1\gamma_1 & \text{if } n_{HA} = \hat{n}_{HA} \\ w_2\gamma_2 + w_3\gamma_3 & \text{otherwise} \end{cases} \tag{4.8}$$

$$\gamma_1 = \frac{1}{n_{HA}} \sum_{i}^{n_{HA}} \left| \frac{z_{HA_i} - \hat{z}_{HA_i}}{\hat{z}_{HA_i}} \right| \tag{4.9}$$

$$\gamma_2 = \sqrt{\frac{1}{n_p} \sum_{i}^{n_p} (z_i - \hat{z}_i)^2}, \; \gamma_3 = \sqrt{\frac{1}{n_v} \sum_{i}^{n_v} (\dot{z}_i - \hat{\dot{z}}_i)^2} \tag{4.10}$$

where, the ground true values $[\hat{z}_{HA}, \hat{z}, \hat{\dot{z}}]$ of the hop apex height, z_{HA}, vertical position, z, and vertical velocity, $\dot{z}$, are as shown respectively. The RMSE in both vertical position, γ_2, and velocity, γ_3, are used to maintain a gradient in the event that the estimation does not capture the correct number of hops. The number of estimated hop apexes, n_{HA}, the true hop apexes, $\hat{n}_{HA}$, vertical heights, n_p, and vertical velocities, n_v, are represented accordingly. Finally, the weights are as follows, $[w_1 = 100, \; w_2 = w_3 = 10]$.

The genetic algorithm is initialized as follows: individuals = 14 parameters (Table 4.2), initial population = 1000 randomly generated individuals, maximum generations $n_{ga} = 20$, elites carried over to the subsequent generation = 5%, crossover fraction = 80%, and the mutation fraction = 15%. Stochastic universal sampling is used to select two parents and the children are created through a uniform scattering crossover; where each optimization parameter is given a random one (parent-1) or zero (parent-2), determining which parent provides the parameter for the child. Finally, a constraint-aware adaptive mutation algorithm is used that creates the child as follows, $x'_{ga} = x_{ga} + \alpha_0 \exp(n_{ga_i}/n_{ga})\, d/||d||$; where the direction vector, $d_{(15\times1)}$, is a random vector in which each component is drawn from a normal distribution $\sim \mathcal{N}(0, 1)$. The n_{ga_i} is the current generation number, and α_0 is the initial step size that can be reduced if the constrains are violated. Finally, as stated previously, learning the parameters requires that the training data covers all desired robot operational characteristics. However, initial testing has shown that a much more limited set of data

Table 4.2: Optimization Parameters

Parm.	**Description**	**Opt. Value**	**Bounds**	**Units**
f_{HVSE}	Cutoff Freq. (HVSE)	7	[400,5]	Hz
f_{HPE}	Cutoff Freq. (HPE)	8	[400,5]	Hz
g_s	Accel. Switching g-force	14.24	[14.5,12]	g-force
σ_{az}	Accel. Meas. STD	9.9857	[10,0.0001]	m/s^2
σ_{bz}	Accel. Bias Meas. STD	NA	[10,0.0001]	m/s^2
σ_{vz}	Velocity Meas. STD	9.5722	[10,0.0001]	m/s
σ_{pz}	Position Meas. STD	0.0091	[10,0.0001]	m
c_{vel2}	Fit Constant Coef. vel^2	-1.1246	[10,-10]	
c_{vel1}	Fit Constant Coef. vel^1	5.9203	[10,-10]	
c_{vel0}	Fit Constant Coef. vel^0	6.7054	[10,-10]	
c_{ch1}	Fit Constant Coef. ch^1	6.2138	[10,-10]	
c_{ch0}	Fit Constant Coef. ch^0	8.4355	[10,-10]	
c_{m0}	Fit Constant Coef. d^1_{mi}	NA	[10,-10]	
c_{m1}	Fit Constant Coef. d^0_{mi}	NA	[10,-10]	

Abbreviations: Measurement (Meas.), Acceleration (Accel.), standard deviation (STD), coefficient (Coef.).

than that collected sufficiently well cover the operating range allowing for more efficient optimization on a reduced set containing 111 hops spread across the [1,2,3,4] m hop trials. This results in a reduction from multiple days per optimization cycle, to approximately 5 hours of parallel processing across eight cores; where code optimization could improve this further.

4.5.3. Training Results

Optimizing across the four filters (KF1, KF2, ESKF1, ESKF2) for the MAPE in the apex height (γ_1) results in KF1 with the lowest error parameters (Table 4.1 Error column). The addition of the accelerometer bias measurement (IMUPT-Accel. Bias (Aerial Phase)) to KF2 results in increased errors. This suggests that the measurement adds more uncertainty than useful information; likely due to the delay between motor command and thrust production. Shifting to ESKF1 and ESKF2, IMUPT-Velocity(LO) was removed because of an observed

instability suggesting that the short-term significant change in the error, cause by the aliasing at TD and LO, is not well managed by the ESKF.

Selecting KF1, the optimized parameter values can be seen in Table 4.2, which result in a MAPE in the apex height of 10.77% (n=998 hops) over the complete set of training data. Analyzing the optimized values in more detail, we can further understand the system's characteristics. First, the cutoff frequencies for both the HVSE and HPE show a trend towards low values, causing significant phase lag and smoothing of the acceleration measurements. This suggests a preference towards removing the inherently arbitrary aliasing of the acceleration measurements over phase accuracy for both the HVSE and HPE; where the colored dots in Fig. 4.8 show the HPE's transitions during the subsequent experimental trials. Second, the liftoff velocity and commanded height show a correlation with the HVSE error as the exponential coefficients (c_{vel2}, c_{vel1}, c_{ch1}) are not equal to zero. Third, as expected, the smoothing does show an increase in the average error as evident by the non-unity constant coefficient values (c_{vel0}, c_{ch0}). Finally, arbitrary initialization of the estimated covariance matrix, P, yielded poor initial state estimations at the first TD. Therefore, to improve the initial state estimation and accelerate convergence of the KF, the P is initialized to an average P after convergence; yielding a much closer estimation of the real value for the first TD (MultiMo-MHR: initial $P = [0.0582, 0.0774; 0.0774, 0.1441]. * 10^{-4}$).

Figure 4.4 shows the sensitivity of the optimized parameters to variation, where the cutoff frequencies (f_{HVSE}, f_{HPE}) and the switching g-force parameter (g_s) are the most sensitive; resulting in larger percent changes to the cost function than the percent change to the parameter. Furthermore, the significant change caused by increasing the g_s, suggests that the low-g accelerometer's noise characteristics change abruptly as its limits are approached, and that the optimization methodology can learn the sensor characteristic from the data.

As can be seen in [93], zeroing altitude (IMUPT-position) removes drift but also the ability to track absolute height in environments where the ground is not flat; resulting in an error relative to the previous contact surface of approximately 0.05 m per step. Absolute

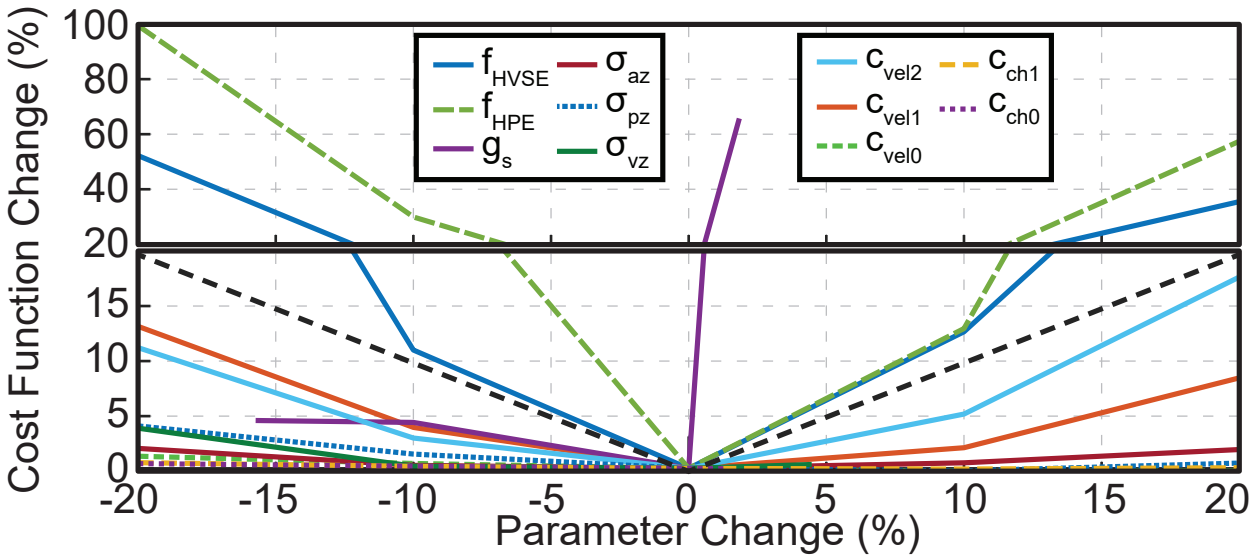

Figure 4.4: Sensitivity to changes in the optimized parameters; $\mathcal{L}_c = 100\gamma_1 + 10\gamma_2 + 10\gamma_3$ from equations 4.9, 4.10.

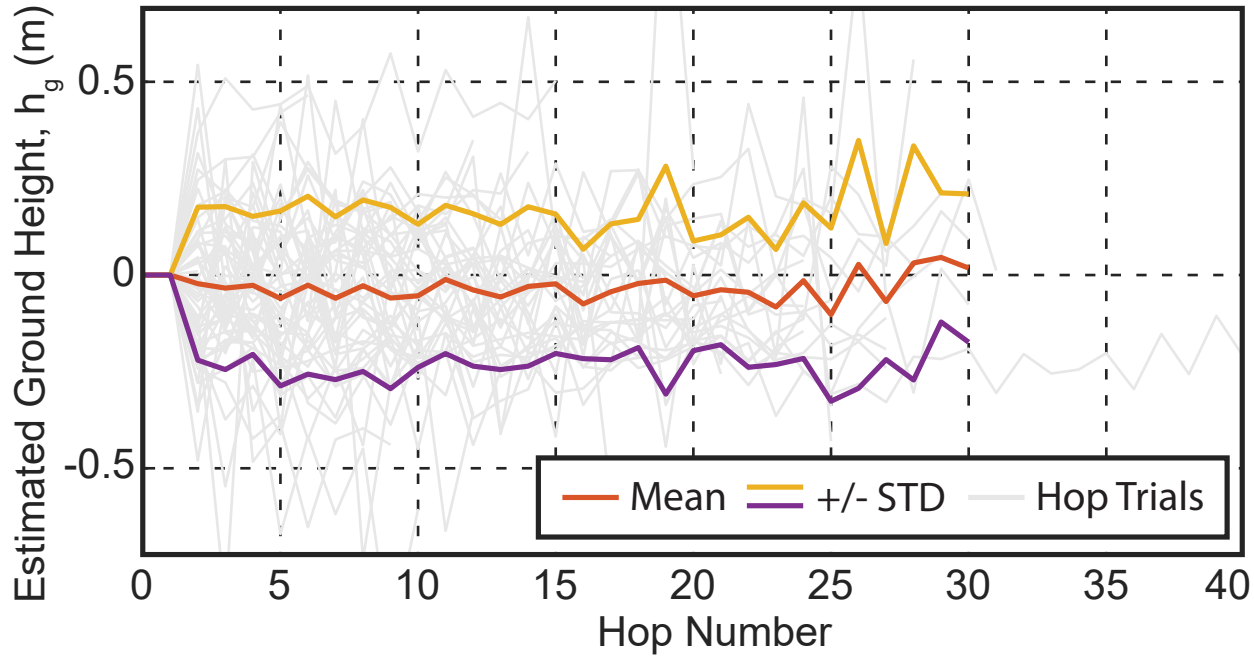

Figure 4.5: Change in the estimated ground height over multiple hops; where STD is the standard deviation, and the true ground height is zero.

height can be recovered through summing the change in height of each stepping or hopping cycle prior to the zeroing. The change in the ground height between cycles can be calculated in two ways including,

$$\Delta h_{1n} = (h_{HAn} - h_{TDn}) - (h_{HAn-1} - h_{TDn-1}) \tag{4.11}$$

$$\Delta h_{2n} = h_{TDn} \tag{4.12}$$

where the estimated height at the hop apex (h_{HA}) and TD (h_{TD}) are represented accordingly, and n indicates the hop number. The difference between them ($h_{HA} - h_{TD}$) gives a measure of the drop height, and the difference between subsequent drop heights, provides a measure of the change in the ground height (Δh_1). Second, the estimated TD height (h_{TD}) provides a direct measure of the change in the ground height (Δh_2). Taking the average of these two measures,

$$h_{gn} = h_{gn-1} + (\Delta h_{1n} + \Delta h_{2n})/2 \tag{4.13}$$

yields an estimate of the change in the ground height for the subsequent hop or step (h_g). Comparing the estimated ground height error of KF1 (Figure 4.5) to that of an untrained KF [93], shows that not only can drift be significantly reduced but that the absolute ground height can be accurately tracked. With an overall average ground height error of 0.018 m after 30 hops of up to 4 meters, the trained KF1 can achieve a 64% improvement on the single step error and significantly better than the sum over the same 30 steps. This is due to the optimization tending towards zero mean error with minimum standard deviation; resulting in a minimum error with an equal chance to be positive or negative for each hop.

4.6. Sensing Frequency

The previously developed dynamic model in [32] can be used to understand the impact of the sensing frequency on the HVSE; which fundamentally controls the aliasing of the ac-

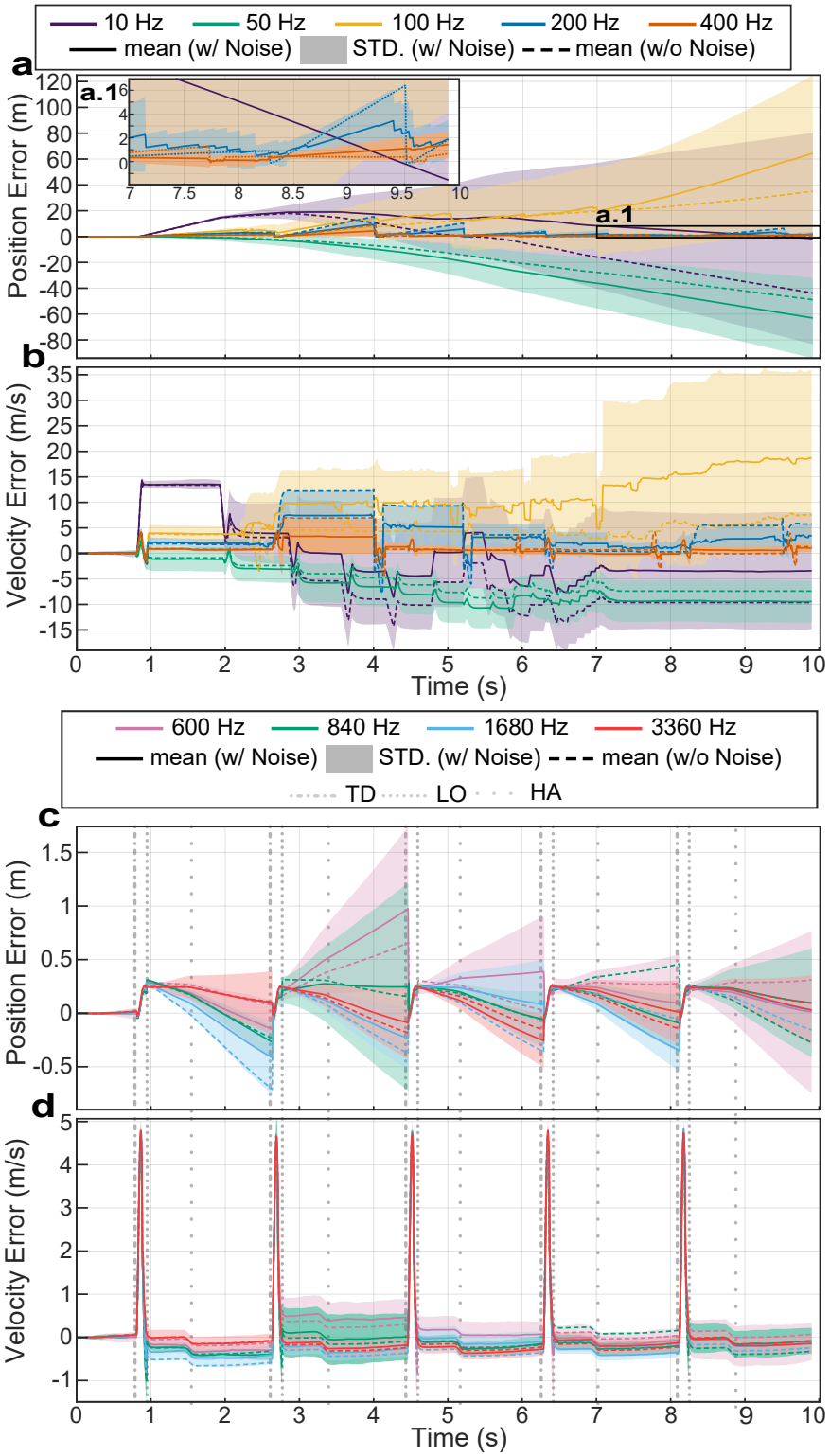

Figure 4.6: Simulated State Estimation Error. a) Unbound and large bound position error, given state estimation at 10 to 400 Hz. b) Unbound and large bound velocity error, given state estimation at 10 to 400 Hz. c) Bound position error, given state estimation at 600 to 3360 Hz. b) Bound velocity error, given state estimation at 600 to 3360 Hz.

celerometer measurements. Specifically, the state estimation error boundedness and growth as a function of the limiting frequency; that being sensor read speeds or state estimation frequency. To explore the estimation error behavior, the measurement and estimation frequencies are varied over 12 values including [3360, 1680, 840, 700, 600, 500, 400, 300, 200, 100, 50, 10] Hz; where the MultiMo-MHR's HVSE is optimized to, and operates at, 840 Hz. The simulation (MATLAB) results are shown in Fig. 4.6 with a control frequency of 400Hz, commanded height, h_{ch}, of 3 m, and the robot parameters in Appendix Robot Parameters. The results are shown for simulations both with and without sensor noise, and the controller is using the true state data, at the measurement frequency, as reference for control. The simulation without sensor noise (mean (w/o Noise)) can highlight effects of aliasing alone while the simulation with noise (mean and STD. (w/ Noise)) can provide an understanding of the true error growth characteristics.

As can be seen in Fig. 4.6a,b, frequencies from 10 to 100 Hz show an unbounded state estimation error with significant growth rate in the standard deviation as indicated by the large monotonic increase over time (Fig. 4.6a,b STD. (w/ Noise)); at $\sim$7 seconds both the 10 and 50 Hz trials are unable to sustain hopping and simply remain stationary on the ground. This unboundedness is similar to the drift associated with direct accelerometer integration (DAI) and therefore, provides no improvement in state estimation over DAI.

Frequencies from 200 to 400 Hz show a bound error characteristic in which the standard deviation in the error is not increasing monotonically (Fig. 4.6a,b STD. (w/ Noise)). However, both mean and standard deviation remain sufficiently large so as to cause variation in the subsequent hops (Fig. 4.6a,b mean and STD. (w/ Noise)).

Frequencies from 600 to 3360 Hz show both a bound mean error and a significant reduction in the standard deviation of the error, in both position and velocity, within each hop cycle (Fig. 4.6c,d mean and STD. (w/ Noise)); creating a deterministic starting point for the subsequent hop. Additionally, from Fig. 4.6c,d, we see both a low sensitivity to minor changes in frequency and diminishing returns as the frequency is increased. This suggests

that the impact of accelerometer aliasing is saturating, and that the HVSE, trained at 840 Hz, is robust to frequency changes in the range of 600 to 3360 Hz. Interestingly, as seen in Fig. 4.6d, it is not required that the mean error in velocity be reduced; only that the standard deviation be reduced as the velocity scaling factor (δ_{vLO}) accounts for mean errors in velocity.

4.7. Hopping Height Control Experiments

Hopping experiments were conducted with the MultiMo-MHR (Fig. 4.1a) at target heights of 1, 2, 3, and 4 m. Trials had two possible measurement inputs for vertical control; true state (GT, motion capture system, n = 70 hops) or the HVSE position and velocity (n = 121 hops). The HVSE is recorded in all scenarios to identify if the HVSE and vertical controller are decoupled and to evaluate the estimator performance.

The experimental procedure begins with zeroing of the accelerometers and motion capture system with the robot positioned vertically on the ground. The robot is then flown to the set initial height with a separate controller (only used in the startup procedure) before entering the first drop state, beginning the hop cycle and activating the hopping height controller. For a complete breakdown of the hop cycle and corresponding HPE and HVSE see Fig. 4.3; where, Fig. 4.1 and 4.7 provide a visual reference of the hop cycle and HVSE.

4.7.1. Results

Figure 4.8 shows the experimental results of the vertical position and velocity estimation using the HVSE. Table 4.3 compares the HVSE to other vertical position and velocity estimation techniques used in hopping robots across five error metrics (Appendix: Error Metrics) including: the mean of the normalized MAE in the vertical position (M1), the mean of the normalized MAE in the vertical velocity (M2), mean absolute percent error (MAPE) in the apex height (M3), mean absolute error (MAE) in the apex time (M4), and MAE in the desired apex height (M5). The table compares three vertical state estimation techniques to the HVSE. These include a Newtonian approach that assumes a ballistic aerial trajectory where

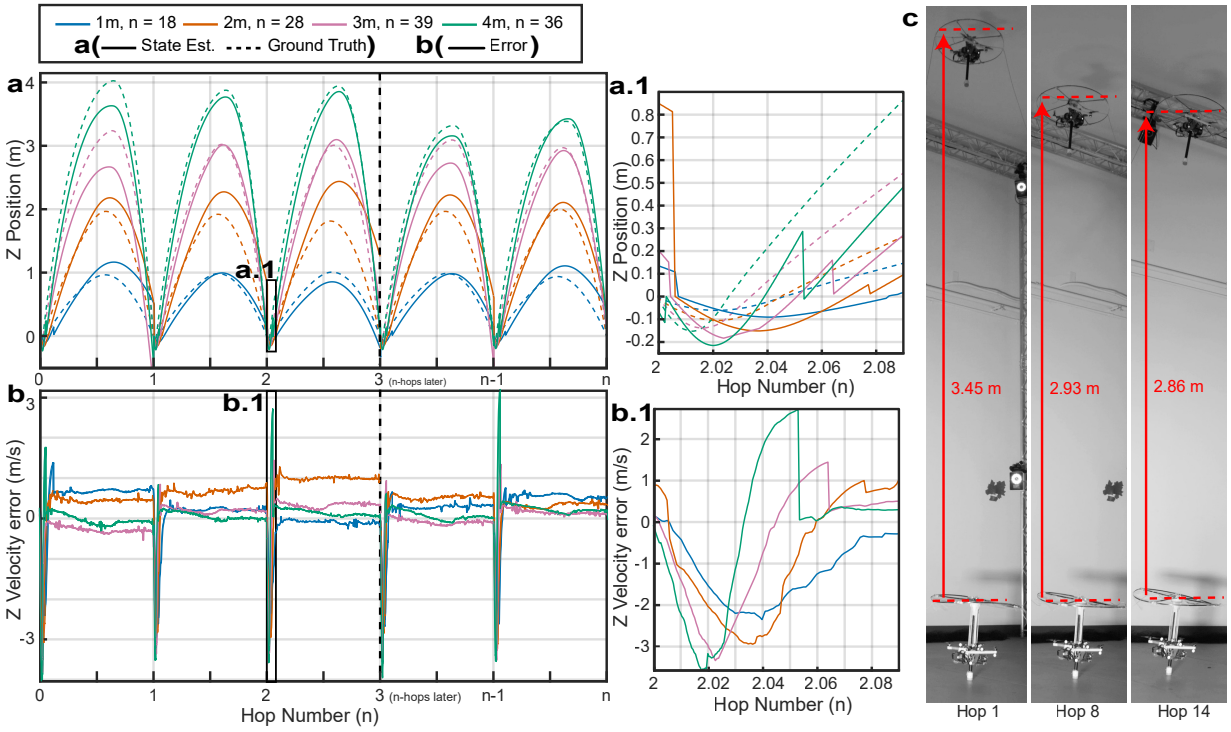

Figure 4.7: Example hopping trials using the hopping vertical state estimator (HVSE) based control a-b) Displays the vertical position and velocity error of the first 3 and last 2 hops for hopping trials at heights of 1,2,3&4 m. Each has a zoomed in region (a.1, b.1) showing the stance phase behavior. c) Snapshots from a high-speed video of the MultiMo-MHR hopping with the HVSE at a desired height of 3.25 m.

the LO state is determined by a leg encoder (BA1) [1, 20, 29]. However, as rotor based hopping robots may not follow a ballistic trajectory, a natural extension is to dead-reckoning the accelerometer measurements (LO to TD) to improve the overall estimates (DR1). Finally, as shown in [92, 93, 95], a Kalman filter with a zero altitude update at TD can be employed to further improve the state estimation (KF3 = KF1 with only IMUPT-position(TD)). Four variations of the HVSE are shown including HVSE1, using the true state (GT) for control, and HVSE2, using the estimated state (SE) for control; where HVSE3 and HVSE4 only include the aerial phase errors. The controller for each technique can use either the true state (GT), focusing on the estimation characteristic, or the estimated state (SE), focusing on the overall hop height control performance.

Validating the simulation (Sim.) data, the experimental (Exp.) results show a similar reduction in both the error mean and standard deviation in position estimation at TD; as a result of the IMUPT-Position(TD) ((Sim.) Fig. 4.6c, (Exp.) Fig. 4.8a, a.1). The IMUPT-Velocity(MS) in both cases reduces the velocity mean error growth rate but with limited impact on the standard deviation ((Sim.) Fig. 4.6d, (Exp.) Fig. 4.8f). Finally, in both cases at LO (IMUPT-Position(LO), Velocity(LO)), a significant reduction in the velocity error's mean and standard deviation is observed ((Sim.) Fig. 4.6d, (Exp.) Fig. 4.8g.1). The significant reductions in both the mean and standard deviations in the position and velocity errors creates a deterministic starting point for the subsequent hop cycle estimation; minimizing the error cause by the significant aliasing of the accelerometer measurements.

Table 4.3 highlights a number of key concepts. First, comparing the common hopping robot vertical state estimation technique (BA1) to its natural extension for rotor based hopping (DR1), suggests that the ballistic trajectory assumption may not be valid. Second, comparing DR1 to the Kalman filter based pedestrian navigation technique (KF3), suggests that a leg encoder can achieve similar results to that of a zeroing altitude condition at TD in a Kalman filter based approach. Third, comparing KF3 to the HVSE, suggests that a trained Kalman filter with the specified IMUPTs can improve the state estimation under significant

aliasing conditions. Fourth, comparing HVSE1 to HVSE2 only shows a significant different in M5, suggesting that the controller and HVSE are independent; however, as expected M5 does show a significant difference, suggesting that the HVSE does introduce error beyond that caused by the controller itself. Fifth, comparing HVSE1 and HVSE2 to HVSE3 and HVSE4, suggests that the estimation of the stance phase states is more challenging; however, as the vast majority of the motion of the robot occurs in the aerial phase this will have a minimal overall impact. Sixth, from basic optics, the M1 metric is shown to be good approximation of the error in the horizontal state estimation, given that an optical flow sensor's range finder (height measurement) is replaced with the HVSE (Appendix: Optical Flow Error). Seventh, given the MultiMo-MHR travels up to $\sim$ 8 meters at velocities that range between $\sim \pm 7$ m/s in $\sim$2.6 seconds coupled with the significant and stochastic nature of aliasing, a certain amount of error is inherent without further direct measurements of the states. Finally, using experimental data, a KF or ESKF can be trained to extract as much information as possible from a highly aliased signal, and be incorporated back into a full state KF or ESKF.

4.7.2. Agility Metric

The commonly used hopping performance metric, vertical jumping agility (ν_{VJA}), defined as an average climb speed in a gravitational environment, allows for quantifying and comparing jumping and hopping robot's approximate average locomotion speed [6]. The metric is defined as follows, $\nu_{VJA} = \frac{h_1}{t_{apogee}} = \frac{h_1}{t_s + t_r}$ where, the apex height, h_1, stance through apex time, t_{apogee}, stance time, t_s, and *rebound* time, t_r, define the metric. However, being focused on climb rate, the ν_{VJA} is unable to account for the *drop* phase time of continuous hopping robots. A subsequent group amended this metric to the hopping agility metric (ν_{HA}) adding the *drop* phase time, t_d, to quantify the overall average locomotion speed of continuous hoppers. The metric is defined as follows [1], $\nu_{HA} = \frac{2h_1}{t_{cycle}} = \frac{2h_1}{t_s + t_r + t_d}$ where, the total cycle time is $t_{cycle} = t_s + t_r + t_d$. However, the base metric, ν_{VJA}, was developed to quantify existing systems' performance from videos, and not as a robot design tool. There-

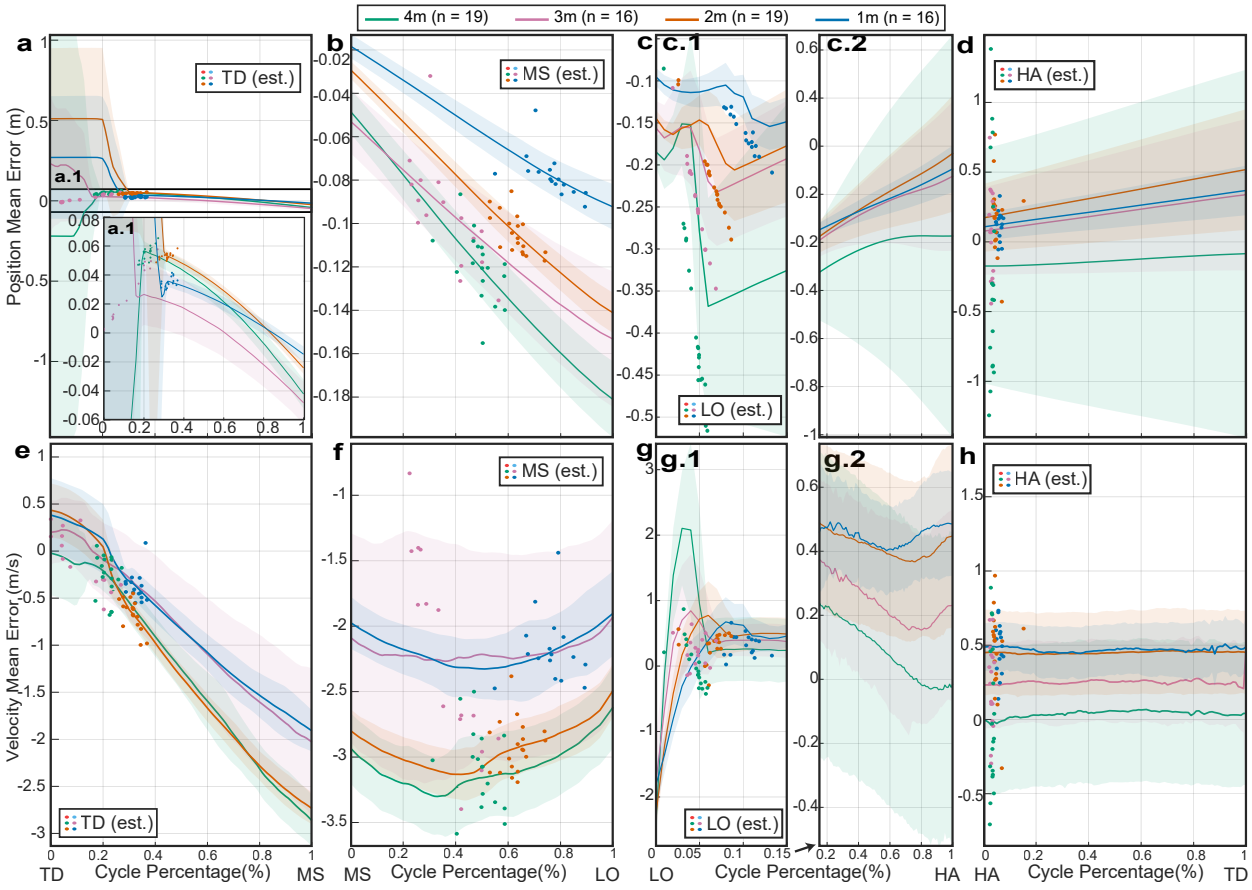

Figure 4.8: Mean and standard deviation of the position (a-d) and velocity (e-h) error for the experimental trials (n=70 hops) utilizing the true state (motion capture data) for height control. a&e) *Stance down* phase, TD to MS. a&e.1) Zoomed in region of (a&e) to show the behavior after TD). b&f) *Stance up* phase, MS to LO. c&g) *Rebound* phase, LO to HA. c&g.1) Zoomed in region from 0-15% of *Rebound* phase c&g.2) Zoomed in region from 15-100% *Rebound* phase. d&h) *Drop* phase, HA to next TD. Note) The phases are segmented by the true state data, while the dots indicate the points at which the robot identified the specified transition.

Table 4.3: HVSE Performance Metrics

Metric	**Est.**	**Meas.**	**1-4m**	**Units**	**Statistics** ‡
(M1) Mean	BA1	GT	196.8	%	t(138)=30.97, p=0.00
nMAE Traj	DR1	GT	44.05	%	t(138)=5.62, p=0.00
Pos.	KF3	GT	60.45	%	t(138)=7.49, p=0.00
	HVSE1	GT	21.24	%	t(189)=0.99, p=0.32
	HVSE2	SE	19.46	%	Comparison
	HVSE3†	GT	20.21	%	
	HVSE4†	SE	**18.97**	%	
(M2) Mean	BA1	GT	179.9	%	t(138)=24.15, p=0.00
nMAE Traj	DR1	GT	28.15	%	t(138)=3.17, p=0.00
Velo.	KF3	GT	37.84	%	t(138)=5.24, p=0.00
	HVSE1	GT	19.58	%	t(189)=0.31, p=0.76
	HVSE2	SE	19.11	%	Comparison
	HVSE3†	GT	17.08	%	
	HVSE4†	SE	**16.50**	%	
(M3)	BA1	GT	53.85	%	t(138)=32.01, p=0.00
MAPE	DR1	GT	43.36	%	t(138)=5.90, p=0.00
Apex	KF3	GT	50.11	%	t(138)=7.38, p=0.00
Height	HVSE1	GT	13.77	%	t(189)=0.83, p=0.41
	HVSE2	SE	**12.49**	%	Comparison
(M4)	BA1	GT	0.7140	s	t(138)=16.83, p=0.00
MAE	DR1	GT	0.1323	s	t(138)=2.66, p=0.00
Apex Time	KF3	GT	0.1388	s	t(138)=4.18, p=0.00
	HVSE1	GT	0.0641	s	t(189)=0.96, p=0.34
	HVSE2	SE	**0.0588**	s	Comparison
(M5) MAE	HVSE1	GT	0.0912	m	t(170)=5.87, p=0.00
Apex Height	HVSE2	SE	**0.2597**	m	Comparison
(desired)					

Error metrics M1-M5 (n=191 hops, Appendix Error Metrics); where *Meas.* column indicates the measurement used for control (true state (GT), estimated state (SE)). * Values indicate the mean of the RMSE value for a complete hop cycle; TD to subsequent TD. † Data uses only the aerial phase as detected by the robot for analysis. ‡ Comparing the 1-4 m data to the HVSE2 using a two-tailed t-test given the null hypothesis with a 95% confidence interval. Related works, BA1 [1, 20, 29], KF3 [92, 93, 95].

fore, while increasing height and decreasing cycle time can increase both metrics, little can be said about control schemes, system losses, and non-constant hopping heights.

The similarity between the metrics allows for their unification, and a mathematical re-derivation allows for a deeper understanding of the effect of system design and operation parameters (Appendix: Hopping Agility). The unified hopping agility (UHA) metric is as follows:

$$\nu_{UHA} = \frac{h_1 + \beta h_0}{t_s + t_r + \beta t_d} \tag{4.14}$$

$$= \frac{h_1 + \beta h_0}{t_s + \sqrt{\frac{2h_1}{g(1-\gamma_r+\gamma_{lr})}} + \beta\sqrt{\frac{2h_0}{g(1-\gamma_d-\gamma_{ld})}}} \tag{4.15}$$

$$h_0 = \zeta_s^2 \frac{h_1(1-\gamma_r+\gamma_{lr})}{(1-\gamma_d-\gamma_{ld})} \tag{4.16}$$

where, the previous hop height, h_0, and gravitational acceleration, g, are represented accordingly. The average *rebound* phase acceleration, $a_{rebound} = -g(1-\gamma_r+\gamma_{lr})$, and the average *drop* phase acceleration, $a_{drop} = -g(1-\gamma_d-\gamma_{ld})$, are composed of both energy input, $[\gamma_r, \gamma_d] < 1$, representing average thrust forces in *rebound* and *drop*, and energy loss, $[\gamma_{lr}, \gamma_{ld}] \geq 0$, representing average drag forces in *rebound* and *drop*; where positive $[\gamma_r, \gamma_d]$ indicates upward thrust and $[\gamma_{lr}, \gamma_{ld}]$ are always opposite the velocity vector. The change in stance phase velocity between TD and LO, $\zeta_s > 0$, can capture energy lost, $0 < \zeta_s < 1$, no energy change, $\zeta_s = 1$, and energy gained, $\zeta_s > 1$ in the stance phase. The previous vertical jumping agility metric (ν_{VJA}) can be derived from $\beta = 0$ whereas, the hopping agility metric (ν_{HA}) can be derived from $\beta = 1$, thus unifying the two metrics within a mathematical framework to inform future hopping robot design. Comparing the MultiMo-MHR to existing hopping robots (Table 4.4) shows that it is able to outperform not only all existing robots but the Galago animal as well.

Table 4.4: Hopping Platforms Vertical Agility

Platform	**Hop Height(m)**	t_{apogee} **(s)**	t_{cycle} **(s)**	ν_{VJA} **(m/s) [6]**	ν_{HA} **(m/s) [1]**
Salto [6]	1.008	0.58*	1.03*	1.75	1.96
Salto-1P [7]	1.252	0.68*	1.15*	1.83	2.18
Galago [6]	1.742	0.78*	1.55*	2.24	2.24
PogoDrone [9]	0.7	4.18†	8.25†	0.17	0.17
PogoX [10]	0.6	0.44†	0.88†	1.37	1.37
HopCopter [1]	1.63	0.71‡	1.37‡	2.30	2.37
MultiMo-MHR§	**3.92**	1.58	2.60	**2.47**	**3.01**
MultiMo-MHR¶	**4.02**	1.85	2.86	**2.18**	**2.81**

Note: Vertical Jumping Agility (ν_{VJA}) [6]. Hopping Agility (ν_{HA}) [1]. *: assumes drop time equals rise time given ballistic motion,†: estimated by paper data,‡: assumes rise and drop time are half the total air time,§: MultiMo-MHR max agility,¶: MultiMo-MHR max achieved height.

4.8. Summary

Here we have shown a light-weight, low computational load, IMU only, Kalman filter-based hopping vertical state estimator (position, velocity) coupled to a hopping phase estimator (drop, stance down, stance up, rebound) and the training procedure to learn the specific characteristics of the target system, including significant aliasing of the acceleration measurements. As discussed in the paper, the resulting filter can be reincorporated into a full state Kalman filter and an error state Kalman filter. While training was conducted under specific conditions, the methodology can be applied to alternative dynamics or surfaces by acquiring hopping data within the targeted regime. Furthermore, with surface identification and state knowledge, one can switch between optimal parameters given the current conditions, which could be an avenue for future work. Finally, unlike previous works that trade absolute for relative height estimation to remove drift, the characteristics of the learned filter result in both an absolute height estimation and a significantly reduced drift (average ground height drift of 0.018 m after 30 hops of up to 4 m). The filter was experimentally validated on

the MultiMo-MHR with hops up to 4.02 m, 2.47x greater than the next highest continuous hopping robot, at velocities approaching ±7 m/s. This results in a mean absolute percent error in hop apex height (M3) of 12.5% with an aerial trajectory average normalized mean absolute error in position (M1) and velocity (M2) of 19% and 16.5%, respectively.

4.9. Appendix

4.9.1. Hopping Agility

Expanding upon the Vertical Jumping Agility (ν_{VJA}) [6] and Hopping Agility (ν_{HA}) [1] metrics, we developed a unified agility metric (UHA: Unified Hopping Agility). Using plots from [6], the hopping cycle for the agility metric proceeds from the prior HA to current HA. While this is irrelevant for constant hopping height cycles, it can create a critical difference in agility for non-constant hopping height cycles.

Using fundamental Newtonian equations and starting with the sequence of prior HA to TD, the fundamental equations are,

$$a_{drop} = (-g + a_d + a_{ld}) = -g(1 - \gamma_d - \gamma_{ld}) \tag{4.17}$$

$$v_{TD} = v_0 + a_{drop} t_d \tag{4.18}$$

$$h_{TD} = h_0 + v_{TD} t_d + \frac{1}{2} a_{drop} t_d^2 \tag{4.19}$$

where we have a control input in the *drop* phase of $a_d = g\gamma_d$ and accompanying loss shown as $a_{ld} = g\gamma_{ld}$. The system starts at an initial height and velocity (h_0, v_{h0}) proceeds to a touchdown position and velocity of (h_{TD}, v_{TD}) over a *drop* phase time window (t_d). In the stance phase, between TD and the subsequent LO, the total energy may change do to loss or addition,

$$\zeta_s = \frac{v_{LO}}{-v_{TD}} \tag{4.20}$$

over a time period (t_s). The last segment is the rebound phase, between LO and the current HA, in which the equations are as follows,

$$a_{rebound} = (-g + a_r - a_{lr}) = -g(1 - \gamma_r + \gamma_{lr}) \tag{4.21}$$

$$v_1 = v_{LO} + a_{rebound} t_r \tag{4.22}$$

$$h_1 = h_{LO} + v_{LO} t_r + \frac{1}{2} a_{rebound} t_r^2 \tag{4.23}$$

where we have a control input in the *rebound* phase of $a_r = g\gamma_r$ with accompanying loss shown as $a_{lr} = g\gamma_{lr}$. The system starts at LO with an initial height and velocity (h_{LO}, v_{LO}) and continues to an ultimate height and velocity of (h_1, v_1) over a *rebound* time window (t_r).

Solving for the *drop* phase time, t_d, from Eq. 4.19 and 4.18 and the *rebound* phase time, t_r, from Eq. 4.22 and 4.23, assuming $[v_0, v_1, h_{TD}, h_{LO}] = 0$ results in,

$$t_d = \sqrt{\frac{2h_0}{g(1 - \gamma_d - \gamma_{ld})}} \tag{4.24}$$

$$t_r = \sqrt{\frac{2h_1}{g(1 - \gamma_r + \gamma_{lr})}} \tag{4.25}$$

Using a switching parameters, β, the two agility metrics, ν_{VJA} and ν_{HA}, can then be combined as,

$$\begin{aligned} \nu &= \frac{h_1 + \beta h_0}{t_s + t_r + \beta t_d} \\ &= \frac{h_1 + \beta h_0}{t_s + \sqrt{\frac{2h_1}{g(1-\gamma_r+\gamma_{lr})}} + \beta\sqrt{\frac{2h_0}{g(1-\gamma_d-\gamma_{ld})}}} \end{aligned} \tag{4.26}$$

where, $\beta = 0$ gives the vertical jumping agility metric and $\beta = 1$ gives the hopping agility metric, thus unifying the two metrics within a mathematical framework to inform future hopping robot design. To include changes in energy during the stance phase (ζ_s), first solve eq. 4.18 and 4.22 for their time and substitute them into eq. 4.19 and 4.23, respectively. Solve each of these for their respective velocity at TD (v_{TD}) or LO (v_{LO}), substitute those

into eq. 4.20, and solve for h_0 resulting in,

$$h_0 = \zeta_s^2 \frac{h_1(1 - \gamma_r + \gamma_{lr})}{(1 - \gamma_d - \gamma_{ld})} \tag{4.27}$$

where, the previous hop height, h_0, and gravitational acceleration, g, are represented accordingly. The average *rebound* phase acceleration, $a_{rebound} = -g(1 - \gamma_r + \gamma_{lr})$, and the average *drop* phase acceleration, $a_{drop} = -g(1 - \gamma_d - \gamma_{ld})$, are composed of both energy input, $[\gamma_r, \gamma_d] < 1$, representing average thrust forces in *rebound* and *drop*, and energy loss, $[\gamma_{lr}, \gamma_{ld}] \geq 0$, representing average drag forces in *rebound* and *drop*; where positive $[\gamma_r, \gamma_d]$ indicates upward thrust and $[\gamma_{lr}, \gamma_{ld}]$ are always opposite the velocity vector. The change in stance phase velocity between TD and LO, $\zeta_s > 0$, can capture energy lost, $0 < \zeta_s < 1$, no energy change, $\zeta_s = 1$, and energy gained, $\zeta_s > 1$ in the stance phase. Comparing the MultiMo-MHR at differing hopping heights highlights how the hopping height affects the metrics (Table 4.5); where an increase is observed at lower heights and a saturation appears to occur at higher heights.

Analysis of the metric can highlight key characteristics for enhancing robot agility. First, given Eq. 4.14, assuming ballistic hopping at a constant height with no inputs or losses, and zero stance time ($[\gamma_{lr}, \gamma_{ld}] = 0, \zeta_s = 1, [\gamma_r, \gamma_d] = 0, h_1 = h_0, t_s = 0$), the resulting equation, $\nu = \sqrt{2gh_1}/2$, shows the metric will increase with increasing gravity or hopping height. Moreover, as the metric switching parameter, β, cancels out, the two metrics are equivalent under these conditions.

Second, given Eq. 4.14, it is clear that increasing the metric requires decreasing the denominator time parameters for the specified heights. The t_s is inversely related to the robot's natural frequency ($\sim \sqrt{m_B/K_s}$); where, the spring constant, K_s, and body mass, m_B are represented accordingly. Therefore, the *stance* time (t_s) can be minimized by increasing K_s and decrease m_B. The *drop* and *rebound* phase times (t_d, t_r) can be minimized by maximizing the acceleration magnitudes ($a_{drop}, a_{rebound}$); thus requiring maximum aerial forces. As the

Table 4.5: MultiMo-MHR Vertical Agility

Meas. (n=hop num.)	**Hop Height(m)**	t_s **(s)**	t_{apogee} **(s)**	t_{cycle} **(s)**	ν_{VJA} **(m/s) [6]**	ν_{HA} **(m/s) [1]**
GT (n=16)	1.03 ± 0.05	0.059 ± 0.001	0.68 ± 0.04	1.18 ± 0.04	1.52	1.74
HVSE (n=18)	1.00 ± 0.00	0.058 ± 0.002	0.69 ± 0.06	1.19 ± 0.08	1.45	1.68
GT (n=19)	2.02 ± 0.02	0.067 ± 0.001	0.98 ± 0.02	1.68 ± 0.01	2.05	2.41
HVSE (n=28)	1.97 ± 0.11	0.07 ± 0.001	0.96 ± 0.11	1.65 ± 0.13	2.04	2.39
GT (n=16)	3.03 ± 0.06	0.069 ± 0.001	1.26 ± 0.03	2.14 ± 0.04	2.40	2.84
HVSE (n=39)	2.85 ± 0.15	0.07 ± 0.001	1.29 ± 0.11	2.13 ± 0.13	2.20	2.68
GT (n=19)	3.78 ± 0.13	0.071 ± 0.001	1.71 ± 0.07	2.72 ± 0.08	2.21	2.78
HVSE (n=36)	3.41 ± 0.29	0.071 ± 0.001	1.64 ± 0.19	2.59 ± 0.24	2.08	2.63
GT (max)	3.92	0.071	1.58	2.60	2.47	3.01
HVSE (max)	4.02	0.072	1.85	2.86	2.18	2.81

Note: Vertical Jumping Agility (ν_{VJA}) [6]. Hopping Agility (ν_{HA}) [1].

MultiMo-MHR can only generate thrust in the positive vertical direction, $0 \leq [\gamma_r, \gamma_d] < 1$, the metric can be maximized by minimizing $[\gamma_r, \gamma_d, \gamma_{ld}] = 0$, and maximizing the *rebound* loss γ_{lr}. This will inherently result in non-constant hopping height unless energy is added in stance; the amount of which can be determined by solving Eq. 16 for the velocity ratio between TD and LO, ζ_s. This does highlight an additional important consideration of the metric. The metric can be increased with a higher previous hop, h_0, followed by a lower hop, h_1. However, this would reduce the hopping height with each cycle, and therefore could be used to increase agility for individual hops at the expense of long-term agility.

4.9.2. Robot Parameters

Table 4.6 shows the robot parameters used in the dynamic simulation from previous work [32].

Table 4.6: Robot Parameters

Parm.	Description	Value	Units
m_B	Body Mass	0.5619	kg
m_L	Leg Mass	0.0981	kg
I_{cmB}	Body Rotational Inertia	0.0012	kg m^2
I_{cmL}	Leg Rotational Inertia	0.0019	kg m^2
K_s	Main Power Spring Const.	704	N/m
K_{lb}	Leg-Body Spring Const.	$400K_s$	N/m
b_{lb}	Leg-Body Damping Coef.	100	N s/m
L_1	Leg Top to CM$_L$	0.1053	m
L_2	Leg Bottom to CM$_L$	0.2821	m
L_3	Body CM$_B$ from Top Leg	0.1191	m
d_{mtrx}	Motor Offset from Body CM	0.0820	m
d_{mtry}	Motor Offset from Body CM	0	m

Full schematic diagram shown in [32].

4.9.3. Error Metrics

Five error metrics are shown in Table 4.7 including:

$$M1 = 1/n_d \sum_{j}^{n_d} \left(\frac{\frac{1}{n_p} \sum_{i}^{n_p} \left| z_i - \hat{z}_i \right|}{\frac{1}{n_p} \sum_{i}^{n_p} \hat{z}_i} \right)_j \tag{4.28}$$

$$M2 = 1/n_d \sum_{j}^{n_d} \left(\frac{\frac{1}{n_p} \sum_{i}^{n_p} \left| \dot{z}_i - \hat{\dot{z}}_i \right|}{\frac{1}{n_p} \sum_{i}^{n_p} |\hat{\dot{z}}_i|} \right)_j \tag{4.29}$$

$$M3 = \frac{1}{n_{HA}} \sum_{i}^{n_{HA}} \left| \frac{z_{HA_i} - \hat{z}_{HA_i}}{\hat{z}_{HA_i}} \right| \tag{4.30}$$

$$M4 = \frac{1}{n_{HA}} \sum_{i}^{n_{HA}} \left| t_{HA_i} - \hat{t}_{HA_i} \right| \tag{4.31}$$

$$M5 = \frac{1}{n_{HA}} \sum_{i}^{n_{HA}} \left| z_{HA_i} - \bar{z}_{HA_i} \right| \tag{4.32}$$

where, n_d is the number of hops, n_p is the number of points per hop (TD to subsequent TD), n_{HA} is the number of hop apexes, and $[z, \hat{z}], [\dot{z}, \hat{\dot{z}}], [z_{HA}, \hat{z}_{HA}], [t_{HA}, \hat{t}_{HA}]$ represent the estimated and true vertical position, velocity, hop apex height, and hop apex time, accordingly.

Table 4.7: HVSE Performance Metrics

Metric Description	Est. Type	Meas.	Set Height (m) 1	2	3	4	1-4m	Units	Comparison‡ to HVSE
(M1) Mean	BA1	GT	171.9	157.7	193.5	248.8	196.8	%	t(138) = 30.97, p = 0.00
nMAE Traj Pos.	DR1	GT	39.10	30.06	29.10	74.01	44.05	%	t(138) = 5.62, p = 0.00
	KF3	GT	50.02	44.12	51.64	89.64	60.45	%	t(138) = 7.49, p = 0.00
	HVSE1	GT	24.64	20.29	13.81	25.60	21.24	%	t(189) = 0.99, p = 0.32
	HVSE2	SE	25.18	22.48	13.88	19.79	19.46	%	Statistical Comparison
	HVSE3†	GT	21.36	19.71	13.51	25.38	20.21	%	
	HVSE4†	SE	25.17	21.91	13.48	19.52	**18.97**	%	
(M2) Mean	BA1	GT	143.1	136.5	174.7	245.0	179.9	%	t(138) = 24.15, p = 0.00
nMAE Traj Velo.	DR1	GT	24.24	19.14	17.97	48.46	28.15	%	t(138) = 3.17, p = 0.00
	KF3	GT	30.91	27.19	31.70	57.35	37.84	%	t(138) = 5.24, p = 0.00
	HVSE1	GT	28.52	21.52	12.44	16.13	19.58	%	t(189) = 0.31, p = 0.76
	HVSE2	SE	25.53	21.95	15.73	17.36	19.11	%	Statistical Comparison
	HVSE3†	GT	24.88	18.78	11.37	13.63	17.08	%	
	HVSE4†	SE	21.98	19.59	13.35	14.78	**16.50**	%	
(M3)	BA1	GT	57.34	51.72	54.53	53.23	53.85	%	t(138) = 32.01, p = 0.00
MAPE	DR1	GT	29.04	22.13	22.47	73.30	43.36	%	t(138) = 5.90, p = 0.00
Apex Height	KF3	GT	37.28	33.24	40.75	81.82	50.11	%	t(138) = 7.38, p = 0.00
	HVSE1	GT	13.50	12.25	8.81	19.68	13.77	%	t(189) = 0.83, p = 0.41
	HVSE2	SE	14.79	14.36	8.11	14.62	**12.49**	%	Statistical Comparison
(M4)	BA1	GT	0.3305	0.4653	0.6987	0.9950	0.7140	s	t(138) = 16.83, p = 0.00
MAE	DR1	GT	0.0554	0.0583	0.0659	0.2430	0.1323	s	t(138) = 2.66, p = 0.00
Apex Time	KF3	GT	0.0726	0.0830	0.1178	0.2483	0.1388	s	t(138) = 4.17, p = 0.00
	HVSE1	GT	0.0611	0.0581	0.0369	0.0956	0.0641	s	t(189) = 0.96, p = 0.34
	HVSE2	SE	0.0524	0.0611	0.0505	0.0695	**0.0588**	s	Statistical Comparison
(M5) MAE Apex	HVSE1	GT	0.0477	0.0294	0.0562	0.2190	0.0912	m	t(170) = 5.87, p = 0.00
Height (desired)	HVSE2	SE	0.0353	0.0959	0.1773	0.5885	**0.2597**	m	Statistical Comparison

Error metrics M1-M5 (n=191 hops, Appendix Error Metrics); where *Meas.* column indicates the measurement used for control (true state (GT), estimated state (SE)). * Values indicate the mean of the RMSE value for a complete hop cycle; TD to subsequent TD. † Data uses only the aerial phase as detected by the robot for analysis. ‡ Comparing the 1-4 m data to the HVSE2 using a two-tailed t-test given the null hypothesis with a 95% confidence interval. Related works, BA1 [1, 20, 29], KF3 [92, 93, 95].

Finally, $\bar{z}_{HA}$ represents the desired hop apex height.

4.9.4. Optical Flow Error

To determine the potential error in the horizontal distance traveled, measured by optical flow. From basic optics, assuming the sensor is traveling parallel to a surface in a single direction, we know the the ratio between the distance an object moves on a camera's image plane, $\hat{d}_c$, divided by the focal distance, $\hat{h}_c$, is equal to the distance the object moves in

the world, d_w, divided by the distance to the surface, h_w, as follows $d_w/h_w = \hat{d}_c/\hat{h}_c$. If the measurement of the distance to the object includes error, $h_w = \hat{h}_w + \delta h_w$, then, so too will the distance traveled, $d_w = \hat{d}_w + \delta d_w$. Substituting this into the previous equation results in,

$$\hat{d}_w + \delta d_w = (\hat{d}_c/\hat{h}_c)\hat{h}_w(1 + \alpha_h) \tag{4.33}$$

where, the $\hat{}$ represents the true value, δ represents the error, and the height error is converted to a percent of the true height, $\delta h_w = \alpha_h \hat{h}_w$. The α_h can be approximated as the mean of the normalized mean-absolute-error (nMAE) in the height measurement (Table 4.7, M1); defined as the mean of the mean-absolute-error in the height divided by the mean of the true height per hop cycle. The error as a percent of the distance traveled can therefore be determined from the ratio of the horizontal distance both with and without the error as,

$$\frac{\hat{d}_w + \delta d_w}{\hat{d}_w} = 1 + \alpha_h \rightarrow \alpha_h = \frac{\delta d}{d_w} \tag{4.34}$$

where, it can be seen that α_h also represents the horizontal error, as a percent of the distance traveled, that would be incurred by replacing an optical flow sensor's range finder with the HVSE.

5. Non-linear Controller Implementation for Enhanced Hop Stability

5.1. Introduction

One of the most critical components to prescribe motion or actions for a robot platform is the implemented controller type. One well-known and understood controller is the PiD controller, a linear controller that utilizes gains to set the response of an output based on an input. The PiD controller is both simple yet reliable, requiring only a means to measure the target and time to adjust and refine gains for optimal performance. While a very solid controller, there are limitations, including the lack of knowledge about the system itself and the linear assumption that more complex systems tend to deviate from, as instabilities and nonlinearities take over. For this reason, understanding and utilizing alternative controller types can generate better controller behavior, in addition to being better suited for extraneous situations.

Hopping locomotion introduces instances of large impacts and, in turn, highly dynamic motion into the robot as it continues along its trajectory. The utilized controller must be able to handle the large resulting disturbances and maintain controllability after lifting off. Within our past work with the MultiMo-MHR, the robot used a PiD controller, but the locomotion was concentrated on vertical hopping while maintaining a horizontal positioning [32,97]. The PiD controller was found to maintain stability within the tested locomotion, but performance for trajectories with increased horizontal motion and more extreme orientations was untested and may require additional system tuning or gain scheduling to achieve stable hopping.

Looking outward to other continuous hopping robots, the control systems vary, with

a number utilizing a similar PiD controller. The work with Salto/Salto-1P saw a variety of controller designs with the early Salto employing basic open-loop control, delving into PiD-based control, and later a dead-beat controller with Salto-1P [6, 7, 31]. The controllers Salto/Salto-1P employed enabled precise, reliable locomotion. However, the locomotion did not see extreme maneuvers, concentrating on back-and-forth or path-style motion with some hopping on/over obstacles. Moving to systems that are more similar in design to the MultiMo-MHR with quadrotor-based control methods, the platforms PogoDrone, PogoX, and Hopcopter can be evaluated. PogoDrone utilized a model-based geometric controller, a step above PiD being model-based and guaranteed stability; the work, however, does not explore the controller capabilities as the robot operates under low energy states (stated ideal operation at 0.3 m and functioning at 0.7 m), indicating less aggressive control is required [9]. PogoX utilizes separate controllers for vertical and horizontal motion, with a energy controller which uses a control Lyapunov function-based quadratic program to stabilize vertical energy and a step-to-step (S2S) dynamic controller for horizontal motion; while these controllers are model based and have improved stability over PiD, it is unknown how the robot operates in high disturbance/energy situations as testing occurred under lower energy conditions ($\sim$ 0.6m vertical displacement) with some mid-air pushes [10]. Lastly, there is Hopcopter, which uses a model-based controller for trajectory prediction and planning for landing into the stance phase, then a PiD controller is used in the air for orientation. Hopcopter sees higher agility motions while in a jump-flying mode, not pure hopping, leading to uncertainty in agility capabilities for pure hopping [1]. All aforementioned works as well remain at lower energy states overall with either low hop heights and/or system masses (see Table 3.1 from Chapter 3 for additional parameters on these systems), which lessens the overall amount of stabilization and energy needed to control the system.

With the variety of controllers implemented on hopping platforms, a gap is seen in implementing and testing a controller of increased reliability and stability for highly dynamic robots. For this reason, this work will see a new controller applied to the MultiMo-MHR,

in the form of a Lyapunov controller. This model-based controller will utilize quadrotor dynamics to establish a controller with guaranteed stability. This ensures the system will see fewer instabilities and should eventually converge to the desired state, making it more ideal for hopping overall for both purely vertical and higher dynamic hopping. This controller will be tested to see its capabilities under system disturbances causing high energy disturbances, as well as for locomotion between multiple positions to test high dynamic motion.

5.2. Lyapunov controller selection and design

The Lyapunov controller is a well-established control methodology, utilized for its model-based design and ability to guarantee system stability and convergence. While in the air, the MultiMo-MHR operates like a quadrotor, matching the overall system dynamics and force inputs. Looking within the work of quadrotors, the Lyapunov controller is well established in research, both from early design and control of quadrotors to more recent applications of the controller for high-speed operations with flips [36, 37]. Given the similarity between the MultiMo-MHR and that of typical quadrotors whilst in the air, utilizing this controller was deemed viable for implementation.

The development of a Lyapunov controller requires knowledge of the system dynamics of what is being controlled. For the MultiMo-MHR, control is only performed while in the air, during which time the system dynamics match those of a quadrotor. For this reason, a quadrotor dynamic model can be utilized as the basis of the controller, of which the system dynamics in the 6-DOF space are obtained as (adapted from [37]):

$$\ddot{x} = \frac{U_1}{m} \left(\sin\left(\phi\right) \sin\left(\psi\right) + \cos\left(\phi\right) \cos\left(\psi\right) \sin\left(\theta\right) \right) \tag{5.1}$$

$$\ddot{y} = -\frac{U_1}{m} \left(\cos\left(\psi\right) \sin\left(\phi\right) - \cos\left(\phi\right) \sin\left(\psi\right) \sin\left(\theta\right) \right) \tag{5.2}$$

$$\ddot{z} = \frac{U_1}{m}\left(\cos\left(\phi\right)\,\cos\left(\theta\right)\right) - g \tag{5.3}$$

$$\ddot{\phi} = \frac{L_t\,U_2}{I_x} + \frac{\dot{\psi}\,\dot{\theta}\,(I_y - I_z)}{I_x} \tag{5.4}$$

$$\ddot{\theta} = \frac{L_t\,U_3}{I_y} - \frac{\dot{\phi}\,\dot{\psi}\,(I_x - I_z)}{I_y} \tag{5.5}$$

$$\ddot{\psi} = \frac{U_4}{I_z} + \frac{\dot{\phi}\,\dot{\theta}\,(I_x - I_y)}{I_z} \tag{5.6}$$

Where the translational directions (X, Y, Z) are shown in equations 5.1 - 5.3, and rotational (roll/ϕ, pitch/θ, yaw/ψ) in equations 5.4 - 5.6, relevant parameters include total system mass (m), gravity (g) and the body inertia (I_x, I_y, I_z). As this is a forced system, the force inputs are prescribed as U1, U2, U3, & U4 for the thrust, roll, pitch, and yaw input forces, respectively; additionally, a lever arm between the force application point and system COM is required (L_t).

Subsequent steps include defining target goals. For the hopping platform, control over the 6-DOF and their respective derivatives is required. Netting 18 total error metrics defined generally as:

$$e_{param} = X_{desired} - X_{measured} \tag{5.7}$$

$$\dot{e}_{param} = \dot{X}_{desired} - \dot{X}_{measured} \tag{5.8}$$

Where e_{param} (and the derivative $\dot{e}_{param}$) is the error between the desired or goal state and the measured state. The error metric is calculated on the states including: position (X, Y, Z), velocity ($\dot{X}$, $\dot{Y}$, $\dot{Z}$), orientation (ϕ, θ, ψ) and rotational velocity ($\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$). Where each

state has some desired value and a measured value, linear and rotational acceleration will follow similarly, however replacing the measured value will be the state dynamic equation, being represented not as a singular error metric but two variables per state for the desired state ($\ddot{X}_d$, $\ddot{Y}_d$, $\ddot{Z}_d$, $\ddot{\phi}_d$, $\ddot{\theta}_d$, $\ddot{\psi}_d$) and the state based on the dynamic equation (equations 5.1 - 5.6). A key point component to mention is in computing the error metrics for the roll and pitch, U2 and U3, controllers the system dynamics for translational motion, equations 5.1 - 5.3, are in the world world frame and must be converted to the body frame using the rotation matrix from equation 2.27. This shifts the dynamics into the robot body frame, which can then be used to compute the ultimate U2 and U3 controller equations.

With error metrics defined, a Lyapunov function (candidate function, V) is required, needing to be positive definite. To meet this criterion, the function that will be used has the form:

$$V = \frac{1}{2} r_{err}^2 \tag{5.9}$$

Where the function is positive definite. The parameter r_{err} is an error term that will vary depending on the controller. There are four total controllers, which will be developed from the base equations, one for each control force U1, U2, U3, and U4. For each controller, the error term (r_{err}) was developed based on the axis direction upon which that controller operated within. As an example, the U2 or roll controller commands the roll orientation and, as a result, can change the y-axis positioning; thus, the error term will incorporate both of those terms. The complete set of error sets are as follows:

$$U1 : r_{err} = Kp_z * e_z + Kd_z * \dot{e}_z \tag{5.10}$$

$$U2 : r_{err} = Kp_y * e_y + Kd_y * \dot{e}_y + Kp_\phi * e_\phi + Kd_\phi * \dot{e}_\phi \tag{5.11}$$

$$U3 : r_{err} = Kp_x * e_x + Kd_x * \dot{e}_x + Kp_\theta * e_\theta + Kd_\theta * \dot{e}_\theta \tag{5.12}$$

$$U4 : r_{err} = Kp_{\psi} * e_{\psi} + Kd_{\psi} * \dot{e}_{\psi} \tag{5.13}$$

where control gains Kp_{param} and Kd_{param} are multiplied by the aforementioned state error metrics from equations 5.7-5.8. With this basis, one last requirement is that the time derivative of the Lyapunov function (equation 5.9) needs to be negative definite. This can be achieved by setting the time derivative as (again, for each respective controller):

$$U1 : \dot{V} = -K_{U1}V \tag{5.14}$$

$$U2 : \dot{V} = -K_{U2}V \tag{5.15}$$

$$U3 : \dot{V} = -K_{U3}V \tag{5.16}$$

$$U4 : \dot{V} = -K_{U4}V \tag{5.17}$$

This adds an additional controller-based gain (K_{U1}, K_{U2}, K_{U3}, K_{U4}), which can be tuned in addition to the prior gains. Through plugging the respective r_{err} term into equation 5.9, as well as calculating the time derivative of equation 5.9. The ultimate equations can be substituted into equations 5.14-5.17 and solved for the control inputs (U1, U2, U3, U4). Leading to the final equations for the Lyapunov controller, with guaranteed stability thanks to defining the time derivative of the Lyapunov function (equation 5.9) as negative definite, as:

$$U1 = \frac{m\left(2\,\dot{e}_{\mathrm{zcm}}\,\mathrm{kp}_z + 2\,g\,\mathrm{kd}_z + 2\,\mathrm{kd}_z\,\ddot{z}_d + \dot{e}_{\mathrm{zcm}}\,k_{\mathrm{U1}}\,\mathrm{kd}_z + e_{\mathrm{zcm}}\,k_{\mathrm{U1}}\,\mathrm{kp}_z\right)}{2\,\mathrm{kd}_z\,\cos\left(\phi\right)\,\cos\left(\theta\right)} \tag{5.18}$$

$$\begin{aligned} U2 = {} & \frac{I_x\,\ddot{\phi}_d + \frac{I_x\,\dot{e}_{\mathrm{phicm}}\,k_{\mathrm{U2}}}{2} - I_y\,\dot{\psi}\,\dot{\theta} + I_z\,\dot{\psi}\,\dot{\theta}}{L_t} \\ & + \frac{I_x\,\dot{e}_{\mathrm{ycm}}\,\mathrm{kp}_y + I_x\,\dot{e}_{\mathrm{phicm}}\,\mathrm{kp}_\phi + I_x\,\mathrm{kd}_y\,\ddot{y}_d + \frac{I_x\,\dot{e}_{\mathrm{ycm}}\,k_{\mathrm{U2}}\,\mathrm{kd}_y}{2} +}{L_t\,\mathrm{kd}_\phi} \\ & + \frac{\frac{I_x\,e_{\mathrm{phicm}}\,k_{\mathrm{U2}}\,\mathrm{kp}_\phi}{2} + \frac{I_x\,e_{\mathrm{ycm}}\,k_{\mathrm{U2}}\,\mathrm{kp}_y}{2} + I_x\,g\,\mathrm{kd}_y\,\cos\left(\theta\right)\,\sin\left(\phi\right)}{L_t\,\mathrm{kd}_\phi} \end{aligned} \tag{5.19}$$

$$U3 = \frac{I_y\,\ddot{\theta}_d + \frac{I_y\,\dot{e}_{\text{thetacm}}\,k_{\text{U3}}}{2} + I_x\,\dot{\phi}\,\dot{\psi} - I_z\,\dot{\phi}\,\dot{\psi}}{L_t} + \frac{I_y\,\dot{e}_{\text{xcm}}\,\text{kp}_x + I_y\,\dot{e}_{\text{thetacm}}\,\text{kp}_\theta + I_y\,\text{kd}_x\,\ddot{x}_d - I_y\,g\,\text{kd}_x\,\sin(\theta)}{L_t\,\text{kd}_\theta} + \frac{+\frac{I_y\,\dot{e}_{\text{xcm}}\,k_{\text{U3}}\,\text{kd}_x}{2} + \frac{I_y\,e_{\text{thetacm}}\,k_{\text{U3}}\,\text{kp}_\theta}{2} + \frac{I_y\,e_{\text{xcm}}\,k_{\text{U3}}\,\text{kp}_x}{2}}{L_t\,\text{kd}_\theta} \quad (5.20)$$

$$U4 = \frac{I_z\,\dot{e}_{\text{psicm}}\,k_{\text{U4}}}{2} + \frac{I_z\left(\dot{e}_{\text{psicm}}\,\text{kp}_\psi - \frac{\text{kd}_\psi\,\dot{\phi}\,\dot{\theta}\,(I_x - I_y)}{I_z}\right)}{\text{kd}_\psi} \quad (5.21)$$

The above equations display the control force calculations for the Lyapunov controller, utilizing the system dynamics to establish control. Each controller has three associated gains, which (Kp, Kd, and $K_{U\#}$, all positive) require tuning (for the system, the Kd was set to 1.0 while the Kp and $K_{U\#}$ were adjusted). All remaining terms are either constant system parameters (system inertia from the solid works model and system mass was directly measured), state variables, or error terms, which are directly calculated from either sensor inputs or motion capture data. And desired states are specific to the hopping mode.

5.3. Robot design and controller Implementation

This section will detail modifications to the control methodology compared to previous chapters. The robot design for the MultiMo-MHR remained the same version, R2, as was utilized in both chapters 3 and 4 [32, 97]. All design, energy storage, and electronics remain unchanged for this chapter and can be found in Chapter 3.4 (for R2 specifically) and in Chapter 4.3. All modifications from this work occurred at the software/programming perspective level on the ESP32 microcontroller, which will be detailed in the following subsections. The robot has the same phases of drop, stance down, stance up, and rebound, with transitions between states of touchdown (TD), max squat (MS), liftoff (LO), and hop apex (HA). Within the phases the overall control method matches prior chapter in which the drop phase only uses orientation control with no active thrust, the stance down phase

has all control disabled, and at the start of the stance up phase (at MS) all control both orientation and thrust activates.

5.3.1. General control modifications

Both the PiD controller and Lyapunov control diagrams are presented in Figures 5.1-5.2, respectively. Both controllers utilize the same sensor inputs as those of Chapter 4.3. With the low g IMU (MPU6050 IMU) and high g IMU (H3LIS331DL), where the high g is used during acceleration values over 14.24 g. Additionally, the same phase detection low-pass filter of 8 Hz is used for the hopping phase estimator (HPE) from Chapter 4.3. The gyroscope provides the angular rate directly (MPU6050); the gyroscope data is additionally used to dead reckon orientation data between motion capture data transmissions (sent at 100 Hz) up to the gyroscope sampling rate of 840 Hz (same as the main control loop rate) to improve stability. The position and aforementioned orientation were obtained from a motion capture system (Vicon vantage V5) and transmitted to the robot over the RF transceiver (nRF24L01) at a rate of 100 Hz. The velocity was calculated on board the robot from the transmitted position data, done to minimize the data packet size of the transmitted data. The last main commonality between controllers is the control selection, which sets active controllers, gains, and desired trajectory values for a given robot phase; additional modifications for some actions in the robot phase follow in subsequent sections.

The PiD controller portion (see Figures 5.1) operates in the same overall strategy as in Chapters 3.4.3 and 4.3. The main modifications include using motion capture measurements as inputs for the vertical (Z) controller, not a constant value as in Chapter 3.4.3 or a state-estimated value as in Chapter 4.3. Once the controller outputs were calculated, the non-linear control portion saw simplifications from Chapters 3.4.3 and 4.3. Namely, the bucketing strategy (see Chapter 3.4.3) was removed as it was intended to prevent one controller from asking for too much but hindered control. Additionally, while the normalization strategies for all control outputs (U1, U2, U3, and U4) follow from Chapter 3.4.3, the non-linear

transformation was moved earlier, prior to calculating the direct motor commands, and was not applied to the thrust control force (U1). The control signals were then combined based on the motor configuration and sent to the motors.

The implementation of the Lyapunov controller (see Figures 5.2) on the MultiMo-MHR was done with equations 5.18 - 5.21 being directly implemented. Direct inputs for the Lyapunov controller included the following system parameters: system mass, moments of inertia, and motor offset, which are presented in Table 3.2, column R2 from chapter 3.4. Sensor inputs required for the Lyapunov controller are the measured translational position and velocity, along with the measured orientation and angular velocity. Additionally, as the Lyapunov controller outputs forces (U1, U2, U3, and U4) due to being derived from the system dynamics, the methods of commanding the motors were slightly altered. First, the output force values are normalized, where the U1 force is normalized by a maximum U1 set by the force of the system due to gravity times the maximum desired thrust-to-weight ratio. Additionally, the orientation controllers, U2-4, are normalized by a general max force (named max U), set to be 1.6 times larger than the force of the system due to gravity. This value prevented the controller from asking for power that draws too much current for the motors. Once normalized the relation between motor force and used duty cycle which was obtained in Chapter 2.4.4 equation 2.1 was utilized. This converts the controller output force into a motor-specific duty cycle, which was then used for control. This motor force addition applies only to the Lyapunov controller. The PiD controller does not use the motor-force scaling relation due to controller variation and currently established gains (all other modifications discussed in this section and subsequent sections are applied to both control strategies unless stated).

5.3.2. Drop phase orientation modification

To enable smoother operation under higher dynamic operation a addition to setting of a desired orientation during the drop phase of the hop cycle was applied. As discussed in

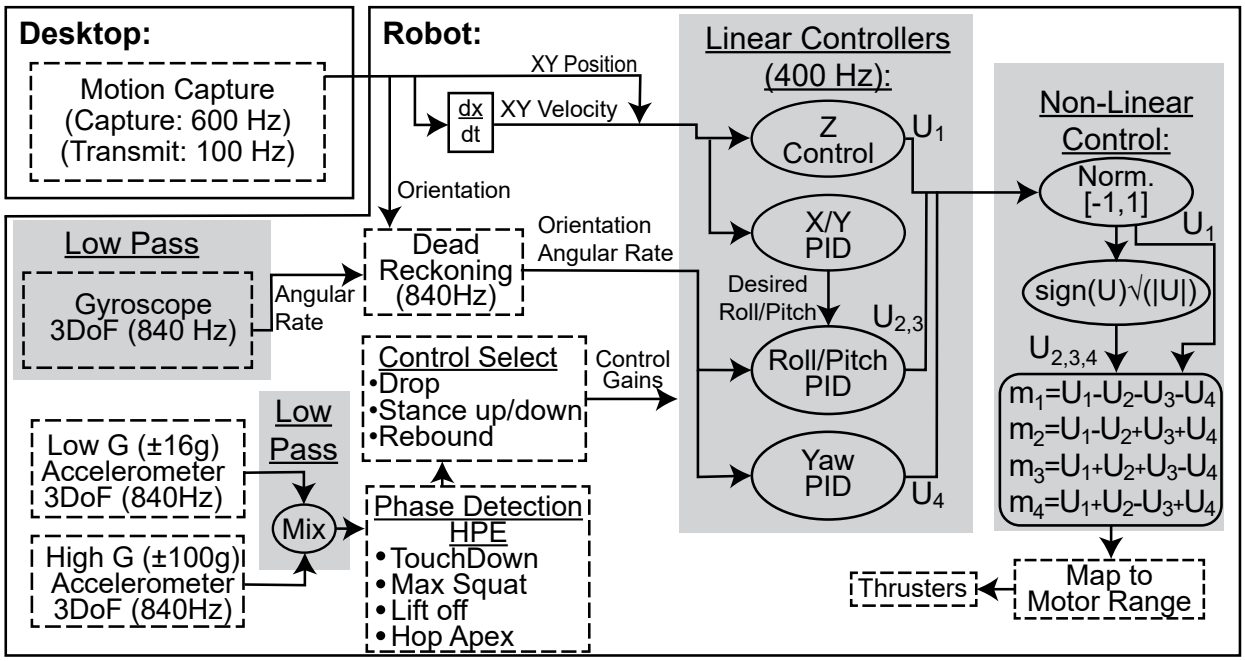

Figure 5.1: Control diagram for the PiD controller

Chapter 3 with Figure 3.6, which showed the impact horizontal velocity has as a function of touchdown angle and resulting liftoff angle, being the larger horizontal velocities lead to larger liftoff angles. A method of reducing the impact of non-zero horizontal velocities during the drop phase was developed, in which the touchdown angle is adjusted as a function of the vertical and horizontal velocities to achieve a zero-degree liftoff angle. This was done by first computing an estimated touchdown vertical velocity (V_{TD-EST}) given the current vertical velocity (V_z) and current system height(h) per equation 5.22. A new desired roll and pitch angle (ϕ_{des} & θ_{des}) are then computed to resist the horizontal velocity (V_x & V_y) done with equations 5.23-5.24 for the new roll and pitch angles respectively. Equations 5.23-5.24 include an adjustment parameter (α_ϕ & α_θ) which is calculated using equations 5.25-5.26. The adjustment parameter is modified with each hop cycle to converge on an optimal value using the sign of the roll and pitch angle at touchdown and liftoff (ϕ_{TD}, θ_{LO} & ϕ_{TD}, θ_{LO}), and an error between a desired liftoff angle (ϕ_{LO-des} & θ_{LO-des}) and measured lift off angle (ϕ & θ). For the robot, a desired liftoff angle of zero degrees for roll and pitch was used. By achieving a near-zero degree liftoff, it prevents additional horizontal velocities from being

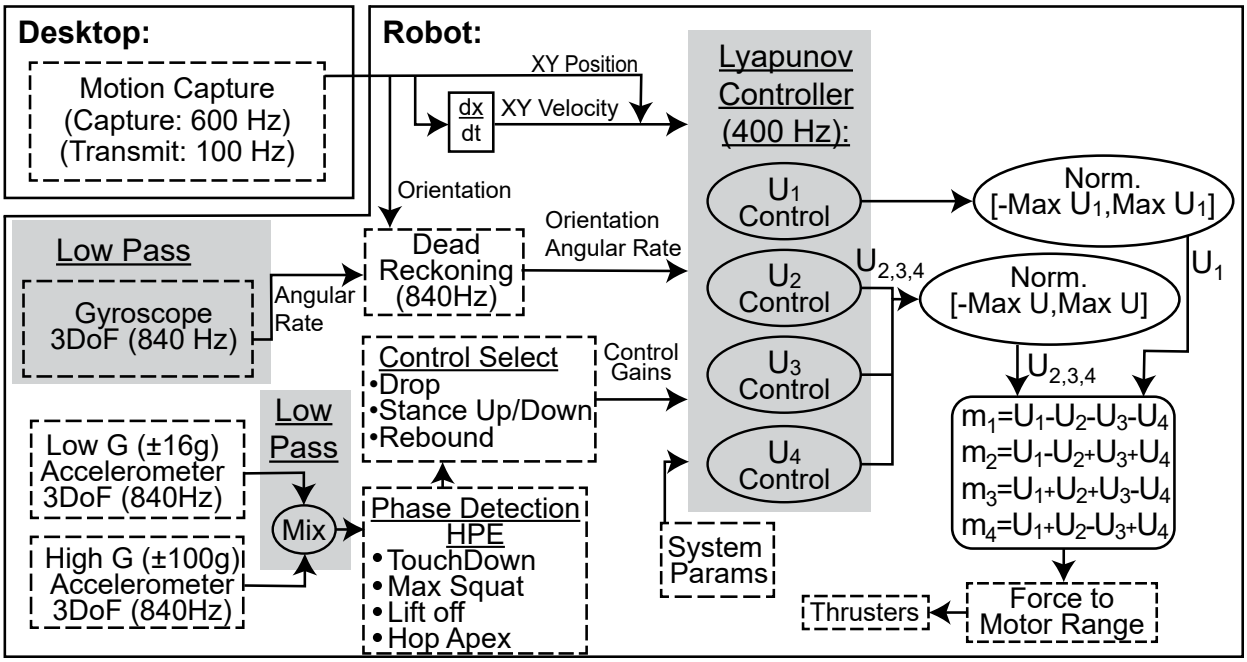

Figure 5.2: Control diagram for the Lyapunov controller

added by putting a lot of energy into the vertical axis, and in turn, making control more reliable for subsequent hops.

$$V_{TD-EST} = \begin{cases} \sqrt{V_z^2 + 2gh} & \text{if } V_z^2 + 2gh > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.22}$$

$$\phi_{des} = -tan^{-1}(\frac{V_y}{V_{TD-EST}}) * \alpha_\phi \tag{5.23}$$

$$\theta_{des} = tan^{-1}(\frac{V_x}{V_{TD-EST}}) * \alpha_\theta \tag{5.24}$$

$$\alpha_\phi = \alpha_\phi - (0.1 * sign(\phi_{TD}) * sign(\phi_{LO}) * |\phi_{LO-des} - \phi_{LO}|) \tag{5.25}$$

$$\alpha_\theta = \alpha_\theta - (0.1 * sign(\theta_{TD}) * sign(\theta_{LO}) * |\theta_{LO-des} - \theta_{LO}|) \tag{5.26}$$

5.3.3. Rebound phase vertical velocity control

The last addition is a trajectory alteration feature, used to set a desired vertical velocity while in the rebound phase of the hopping cycle. Prior work had a desired velocity of zero, and adjustments to the Kp and Kd gains for the PiD controller were made to achieve a desired hop height. The issue with setting the desired vertical velocity to zero for the entire rebound phase originates with the work in Chapter 3 on hopping efficiency. In Chapter 3, it was found that efficiency is optimized by maximizing one's velocity at the point of liftoff (utilizing a large thrust) before entering a ballistic mode (disabling thrusters); where setting a desired velocity to zero right at liftoff can only serve to degrade the overall system efficiency and performance. A desired rebound velocity was calculated using a ballistic Newtonian method and assuming no losses; this generates a known overestimated desired velocity given the absence of losses like drag, which will be accounted for separately. The desired vertical velocity (V_{Z-des}) is calculated from the current (h_z) and desired (h_{des}) vertical heights with equation 5.27. The estimated desired vertical velocity is then reduced by 30% and 20% for the PiD and Lyapunov controllers, respectively, accounting for the aforementioned losses and gain/controller variation. This addition generates a velocity profile that decreases as the robot approaches the target height, ensuring control effort is maximized at liftoff and reduces as the robot rebounds, aiding in boosting system efficiency.

$$V_{Z-des} = \begin{cases} \sqrt{2g(h_{des} - h_z)} & \text{if } 0 < h_z < h_{des} \\ 0 & \text{otherwise} \end{cases} \tag{5.27}$$

5.4. Experiments: disturbance recovery

To test the Lyapunov controller, a disturbance/recovery test with higher system dynamics was utilized. Both the original PiD controller and the new Lyapunov controller had

disturbance testing applied to see how each handled the higher dynamic trajectories. Tests consisted of starting the MultiMo-MHR at a predefined position of (-1, 0, 3)m in XYZ coordinates. Once the robot was in position, it would enter the drop phase, during which a non-zero pitch angle was prescribed with all remaining angles holding zero. Upon ground contact, the initial angle caused increases in the angle and rotation rate of the robot, leading to a large disturbance at liftoff in the form of a large pitch angle, rotation rate, and horizontal velocity. The generated disturbance also serves to replicate a running locomotion that seeks to stop; both this and a stationary angled drop would see similar stance phase reactions leading to a large disturbance at liftoff. Each test had both a prescribed initial disturbance landing angle ($\theta_{disturb}$), and a maximum thrust-to-weight ratio (TWR) which could be applied during the rebound phase (see chapter 3 for an extended talk on the thrust to weight ratio). Upon the point of max squat, both controllers would become active, attempting to both recover/slow down from the disturbance and hop at the originally defined global position. A trial would conclude anywhere from around 7 to 10 sequential hops, as the hopping remained consistent after that period. The following sections will display the results from each respective controller and how each handled the higher dynamic disturbance.

Each controller had a series of evaluations to assess its performance. Translational and orientation measurements were captured, being used to evaluate the applied disturbance and subsequent recovery. Additionally, the duty cycle thrust average over the hop cycles is presented. This presents the duty cycle the motors use as a percentage of the system weight (i.e., 100% = thrust required to hover). The percentages are given for all robot phases (stance, drop, and rebound) as an average of all phases. This allows instances of high or low control effort to be captured, and another method to see when the system stabilizes and where energy is applied.

5.4.1. PiD controller setup and disturbance performance

PiD controller testing served to test if the base PiD controller utilized in prior works (See chapters 3-4) could operate under higher horizontal velocities and orientations. To evaluate the controller, two disturbances were applied, a low and high energy case; the generated disturbance conditions can be seen in Table 5.1 for both the imparted touchdown angle and resulting liftoff disturbance. For both cases, a maximum rebound TWR of $TWR = 83.7\%$ was set, matching the value used in chapter 4. The target stable position is in XYZ coordinates of (-1, 0, 3)m, as previously stated. Testing was attempted at a higher TWR of $TWR = 110\%$, but the system failed to operate after an initial disturbance and will be discussed more later.

Table 5.1: PiD Controller Generated Disturbance Conditions

Variable Condition	**Low Energy Disturbance**	**High Energy Disturbance**
Touchdown Pitch Angle (deg)	3.5	9.0
Touchdown Roll Angle (deg)	0.0	0.0
Liftoff Pitch Angle (deg)	11.5 ± 1.0	31.5 ± 1.1
Liftoff Roll Angle (deg)	0.8 ± 1.8	-1.1 ± 3.8
Liftoff Pitch Rate (deg/s)	6.0 ± 17.8	146.7 ± 14.3
Liftoff Roll Rate (deg/s)	-68.4 ± 38.8	-37.9 ± 50.1
Liftoff X Velocity (m/s)	1.3 ± 0.1	2.9 ± 0.1
Liftoff Y Velocity (m/s)	-0.1 ± 0.1	0.1 ± 0.3
Liftoff Z Velocity (m/s)	4.4 ± 0.1	3.6 ± 0.03

Number of trials per disturbance. Low energy disturbance: n = 13 trials, High energy disturbance: n = 6 trials. Where 1 trial consists of at least 10 continuous hops.

General performance from the disturbances can be seen in Figure 5.3. With a breakdown of key measurement points presented in Table 5.2, which highlights maximum measured values caused by the disturbance and state parameters once the system stabilized. Overall, the PiD saw horizontal stabilization after 2 and 4 hop cycles, respectively, for the lower and higher disturbances. The hop height, however, took longer to stabilize, requiring 4 and 10 hop cycles for the low and high disturbances, respectively. The higher energy disturbance shows the limit of the PiD controller, as it traveled nearly 2 m in displacement as a result of the

initial disturbance before changing directions; it required more energy to slow the system down horizontally, resulting in a great reduction in hop height, which could slowly build back up. While the PiD operates well under minor disturbances, it sees a great reduction in vertical height for high-energy disturbances. This highlights a fault with the PiD as it underperforms as the system enters the higher angle regions, and would require a separate gain tuning or control style to attempt and manage, if at all.

Table 5.2: PiD Controller outcome after generated disturbance

Measured Condition	**Low Energy Disturbance**	**High Energy Disturbance**
Max X Velocity (m/s)	1.43 ± 0.13	3.31 ± 0.10
Max X position (m)	-0.28 ± 0.09	0.91 ± 0.14
Stable X Position (m)	$-0.92 \pm 0.05^{*}$	$-0.98 \pm 0.04^{\dagger}$
Stable X Velocity (m/s)	$-0.01 \pm 0.04^{*}$	$-0.01 \pm 0.04^{\dagger}$
Average Y Position (m)‡	0.10 ± 0.05	0.10 ± 0.06
Y Velocity Bounds (m/s)‡	± 0.2	± 0.2
Max Pitch Angle (deg)	-32.56 ± 1.65	-37.82 ± 3.65
Max Pitch Rate (deg/s)	319 ± 20	545 ± 32
Stable Pitch Angle (deg)	$\pm 10^{*}$	$\pm 10^{\dagger}$
Stable Pitch Rate (deg/s)	$\pm 200^{*}$	$\pm 200^{\dagger}$
Roll Angle Bound (deg)	$\pm 5^{\ddagger}$	$\pm 10^{\diamond}$, $\pm 5^{\dagger}$
Roll Rate Bound (deg/s)	$\pm 100^{\ddagger}$	$\pm 200^{\diamond}$, $\pm 100^{\dagger}$
Yaw Angle Bound (deg)‡	± 5	± 5
Yaw Rate Bound (deg/s)‡	± 50	± 50
Apex Z Height after disturbance (m)	2.49 ± 0.08	1.34 ± 0.08
Min Apex Height (m)	2.49 ± 0.08	0.99 ± 0.38
Z Apex at Hop 10 (m)	2.91 ± 0.11	2.58 ± 0.22
Touchdown Z Velocity at Hop 10 (m/s)	-6.79 ± 0.13	-6.25 ± 0.38
Liftoff Z Velocity at Hop 10 (m/s)	4.92 ± 0.11	4.63 ± 0.34

Notes: all parameters in this table are for conditions after the system lifts off with the disturbance. *: low energy condition after hop 2. †: high energy condition after hop 4.$^{\diamond}$: high energy condition hops 1-4. ‡: condition for all hops in trial.
Number of trials per disturbance. Low energy disturbance: n = 13 trials, High energy disturbance: n = 6 trials. Where 1 trial consists of at least 10 continuous hops.

Looking next to the duty cycle percentages, which can be seen in Figure 5.4. This data follows similar trends to the positioning data, where the first 2 and 4 hop cycles see increased levels of effort for the low and high energy disturbances, respectively. The data

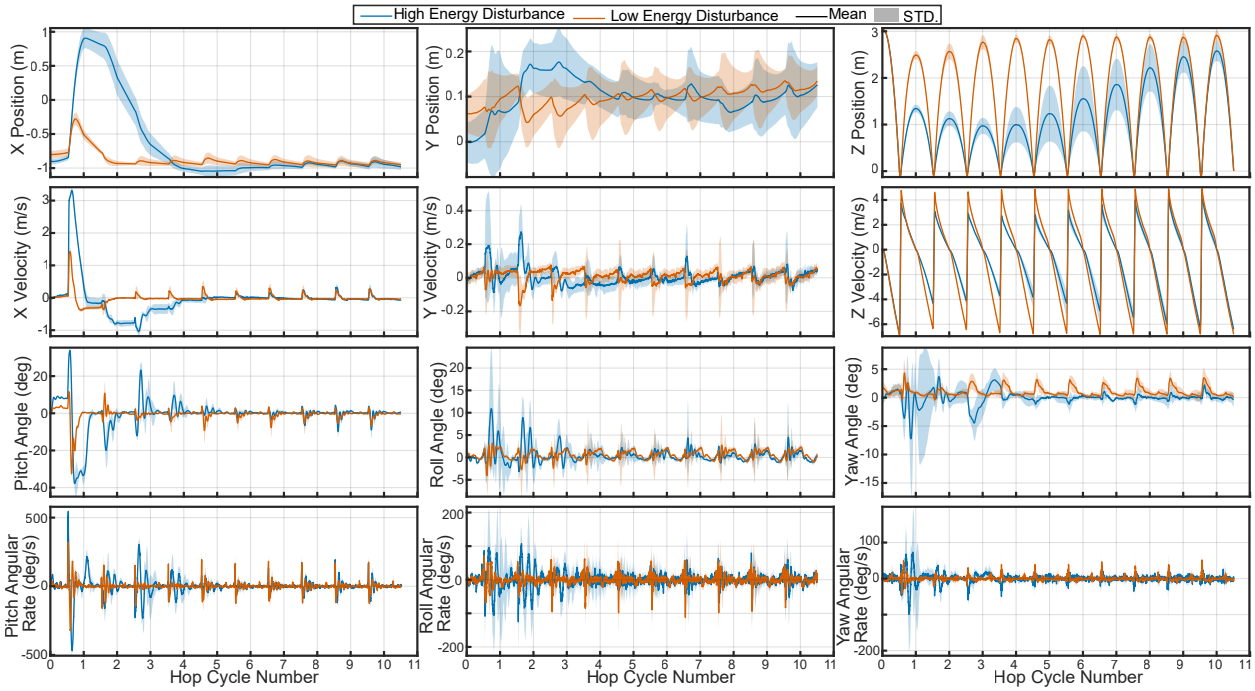

Figure 5.3: PiD controller, position and orientation data for disturbance recovery. Showing both the low energy (n = 13 trials) and high energy (n = 6 trials) disturbances.

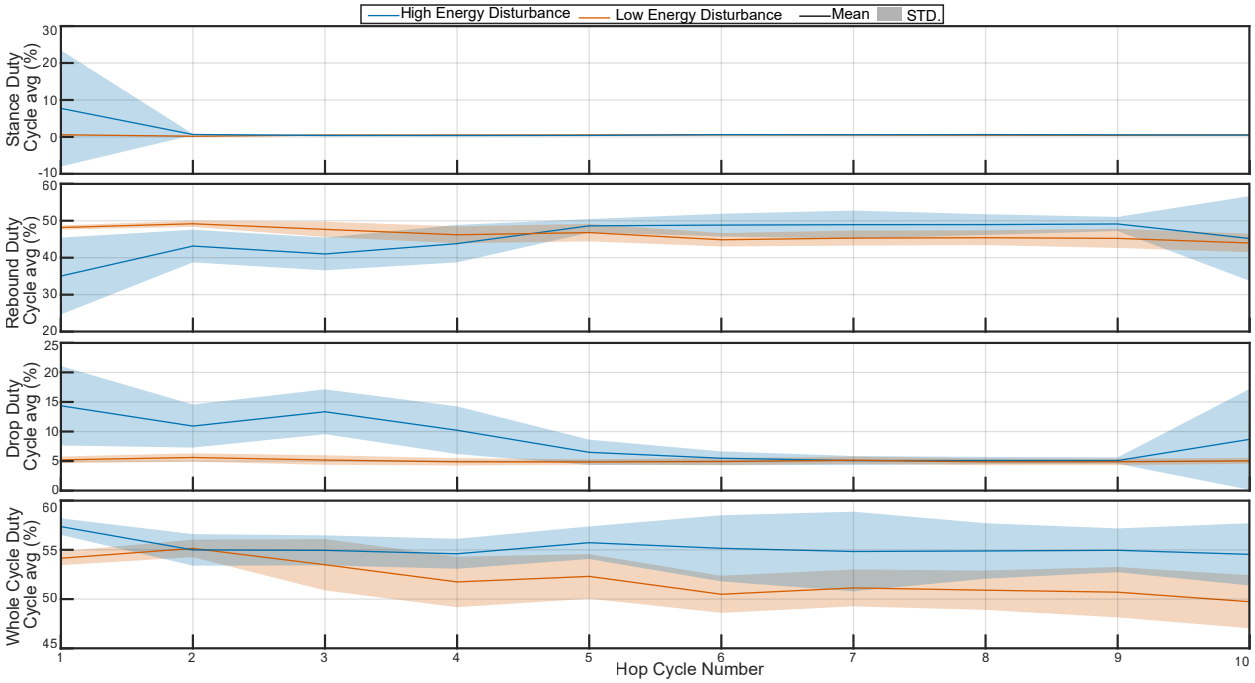

Figure 5.4: PiD controller, duty cycle average during and after the generated disturbance. Showing both the low energy (n = 13 trials) and high energy (n = 6 trials) disturbances.

for the whole cycle shows the first few hop cycles having slightly larger percentages as a result of the disturbance recovery, which then stabilizes. The low energy percentage for the first two hops sees 54.6 ± 0.7 % then stabilizes to 51.3 ± 2.3 %, and the high energy sees a high first hop of 57.4 ± 1.4 % then stabilizes to 54.9 ± 2.4 %. Overall, the low energy trials saw a very consistent duty cycle across all phases, with practically all work applied in the rebound phase. Looking at the high energy specifically, some interesting notes include the rebound percentage starting low, then building due to the lower hopping height caused by the disturbance. The drop phase also sees increased power for the first four hops around 15 % due to some of the added pitch stabilization required.

Another important note on the PiD is an inability to function with a higher TWR of $TWR = 110\%$ for the same disturbances. These tests led to failure as the system would liftoff from the initial disturbance and grow unstable rapidly. While additional gain tuning and adjustment may have enabled operation, it again highlights the limitation in the PiD methodology where stability is not guaranteed, and as it was not tuned for the higher power outputs, it could not operate.

Overall, the PiD controller could operate under an applied disturbance. However, greater disturbances generate large reductions in the vertical energy, which recovered slowly over 10 hops. Both disturbances showed stabilization, but it is worth noting that the PiD is better suited to the linear region with near-zero orientation angles it was initially designed for, and operating outside that region would require a method like a state-specific gain selection process and additional gain tuning.

5.4.2. Lyapunov controller setup and disturbance performance

Moving to the Lyapunov controller tests, focused on implementing the controller at higher power levels and a higher disturbance to evaluate the capabilities. The disturbance generated for this test is displayed in Table 5.3. With a maximum rebound TWR of $TWR = 110\%$, and a target position of XYZ coordinates of (-1, 0, 3)m.

Table 5.3: Lyapunov Controller Generated Disturbance Conditions

Variable Condition	**Disturbance**
Touchdown Pitch Angle (deg)	6.0
Touchdown Roll Angle (deg)	0.0
Liftoff Pitch Angle (deg)	20.9 ± 1.8
Liftoff Roll Angle (deg)	1.0 ± 3.0
Liftoff Pitch Rate (deg/s)	53.6 ± 58.6
Liftoff Roll Rate (deg/s)	-43.4 ± 81.0
Liftoff X Velocity (m/s)	2.0 ± 0.2
Liftoff Y Velocity (m/s)	-0.2 ± 0.3
Liftoff Z Velocity (m/s)	4.3 ± 0.1

Number of trials for the disturbance: n = 7 trials. Where 1 trial consists of at least 7 continuous hops.

The positional and orientation tracking for the test can be seen in Figure 5.5. Key data points such as the maximum value after the disturbance and post-disturbance stabilized value range are presented in Table 5.4. Like the PiD controller, the generated disturbance was along the pitch angle and x-axis. It can be seen that the system has a disturbance velocity of 2.35 ± 0.27 m/s, from which it needs to recover and slow down. It was capable of recovering horizontally in around two hop cycles and stabilizing the vertical height immediately. This shows the benefits of the Lyapunov control method, as it is model-based and aids in improving efficiency, better controller output, and guaranteed stability and convergence, ensuring it stabilizes.

Accompanying average duty cycle percentages are provided in Figure 5.6. General expected trends exist with slightly larger percentages at the first and second hop as the system stabilizes, after which, across the board, all percentages come to a stable, consistent value range. The first two hops see a total average percentage for all phases of $56.4 \pm 8.2\%$ before stabilizing to $47.0 \pm 2.9\%$ for the remaining hops. The same trend can be seen in all phases, with a slightly higher initial few hops which then stabilize; this is thanks to the rapid system stabilization both horizontally and vertically, aiding in a faster consistency in values.

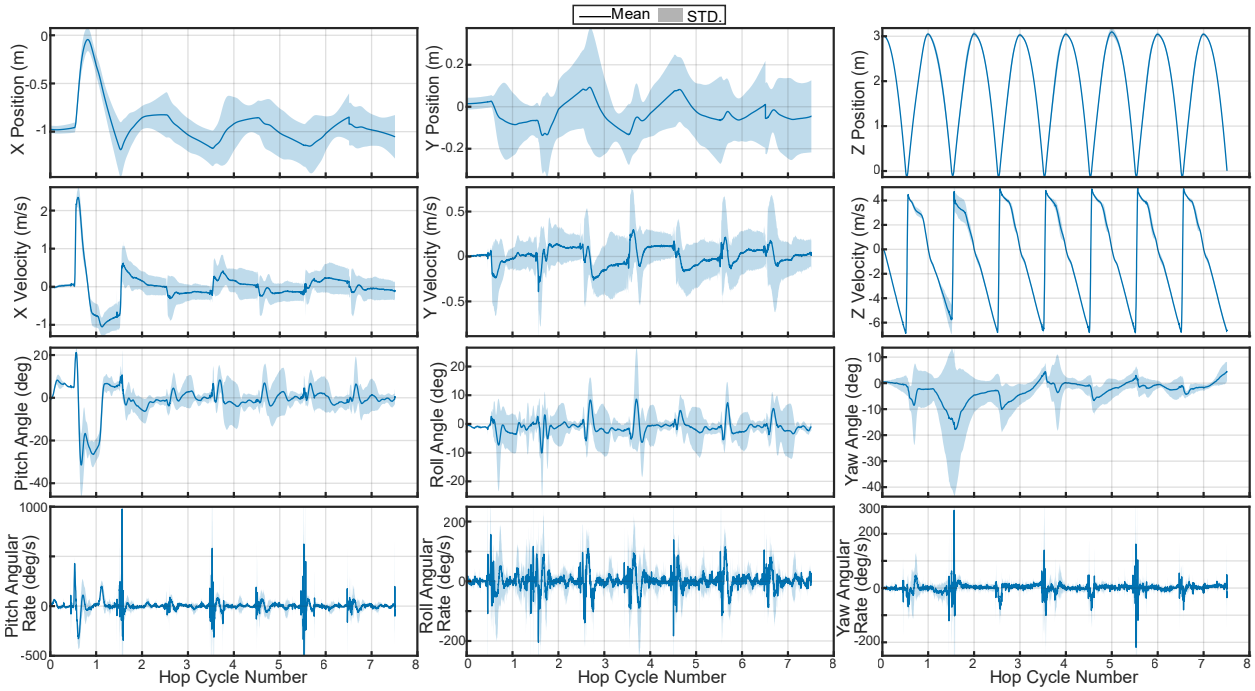

Figure 5.5: Lyapunov controller (n = 7 trials), position and orientation data for disturbance recovery.

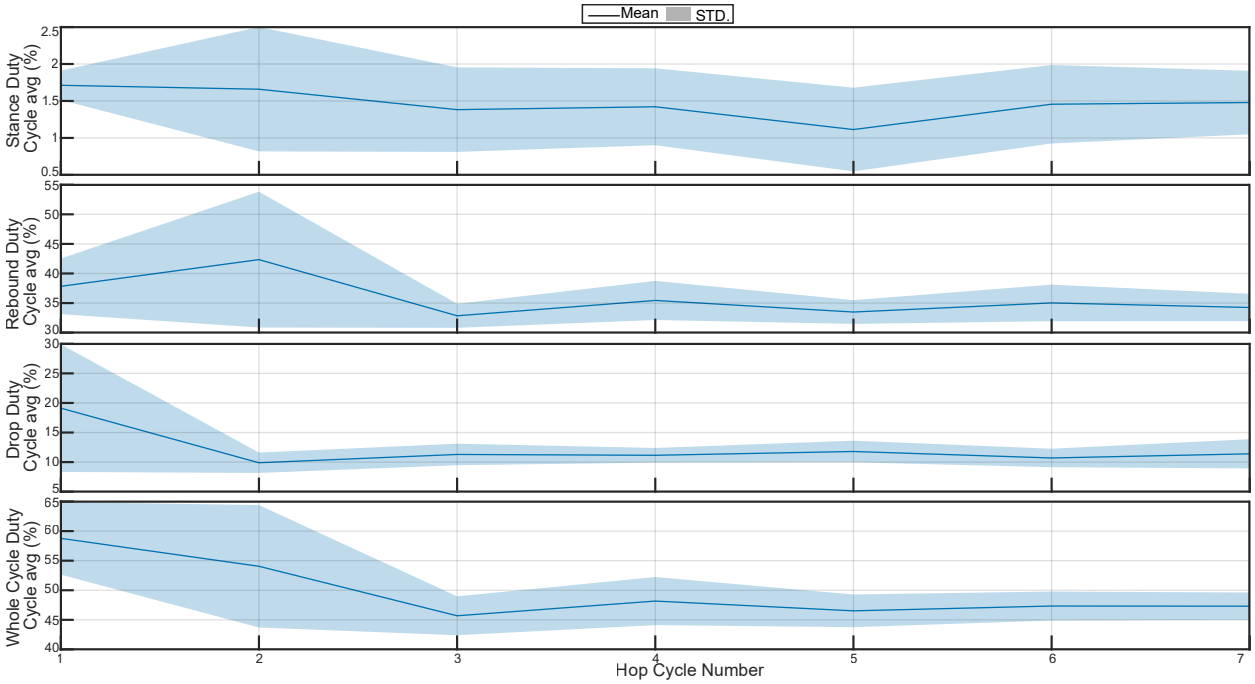

Figure 5.6: Lyapunov controller (n = 7 trials), duty cycle average during and after the generated disturbance.

Table 5.4: Lyapunov Controller outcome after generated disturbance

Measured Condition	**Measured Value**
Max X Velocity (m/s)	2.35 ± 0.27
Max X position (m)	-0.04 ± 0.12
Stable X Position (m)	$-0.98 \pm 0.20^{*}$
Stable X Velocity (m/s)	$-0.01 \pm 0.31^{*}$
Average Y Position (m)‡	-0.028 ± 0.14
Y Velocity Bounds (m/s)‡	± 0.25
Max Pitch Angle (deg)	-31.48 ± 14.83
Max Pitch Rate (deg/s)	428 ± 67
Stable Pitch Angle (deg)	$\pm 10^{*}$
Stable Pitch Rate (deg/s)	$\pm 500^{*}$
Roll Angle Bound (deg)‡	± 10
Roll Rate Bound (deg/s)‡	± 200
Yaw Angle Bound (deg)*	± 10
Yaw Rate Bound (deg/s)*	± 100
Avg Z height (m)‡	3.05 ± 0.05
Avg Touchdown Z Velocity (m/s)‡	-6.65 ± 0.27
Avg Liftoff Z Velocity(m/s)‡	4.87 ± 0.46

Notes: all parameters in this table are for conditions after the system lifts off with the disturbance. *: condition after hop 2. ‡: condition for all hops in trial.
Number of trials for the disturbance: n = 7 trials. Where 1 trial consists of at least 10 continuous hops.

5.5. Experiments: variable positioning hopping

Expanding upon the disturbance test, another highly dynamic trajectory was tested. Hopping robots have demonstrated the ability to hop between predefined locations, namely the Salto-1P platform, which could precisely move between target locations along one main horizontal motion axis [29].

To test the higher stability and ability to follow a trajectory, the Lyapunov controller was evaluated with a hopping trajectory between four positions. The trajectory pattern was designed to ensure the robot could move reliably in all axes. The pattern designed is visualized in Figure 5.7; the robot starts at the red dot, and traverses to the subsequent target points (set at a distance of ΔX and ΔY about the room origin). The generated path between

Table 5.5: MultiMo-MHR Variable Horizontal Positioning Accuracy

Landing Spot	**Desired X(m)**	**Desired Y(m)**	**Measured X(m)**	**Measured Y(m)**
a	-0.75	0	-0.891±0.112	0.203±0.145
b	0	0.75	-0.093±0.156	0.869±0.257
c	0	-0.75	-0.151±0.170	-0.640±0.187
d	0.75	0	0.847±0.216	0.085±0.198

Note: Landing Spot letters correspond to letters in Figure 5.7. Both measured and desired are with respect to the landing position, specifically the max squat position for the measured values.

hop positions ensured the robot hopped both on axis and on the 45-degree rotated axis in all directions. This path enables testing of motion that utilizes all motor configurations for control, as forward/back/side motion uses pairs of motors to direct motion, while the 45-degree axis uses singular motors that align with the respective direction. The pattern ensures reliability regardless of orientation and motion style (a yaw angle of zero degrees is required as well through the locomotion).

An experimental test of this hopping pattern was conducted with the MultiMo-MHR. The pattern was set to have a hop height of 1.5m and ΔX,ΔY of 0.75 m. Before shifting to a new position along the pattern, the robot was allowed one additional hop at the same position to aid in stabilization before transitioning to the next point. Once reaching the path's end, the cycle would continue until terminated. The MultiMo-MHR accuracy for holding the various positions in the pattern are presented in Table 5.5; additionally through the trajectory the hop apex maintained a height of 1.567±.129 m. The general 3D trajectory and associated individual axis path can be seen in Figures 5.8 and 5.9, respectively. The pattern was attempted with the PiD controller; however, horizontal instabilities grew rapidly, leading to an inability to hop.

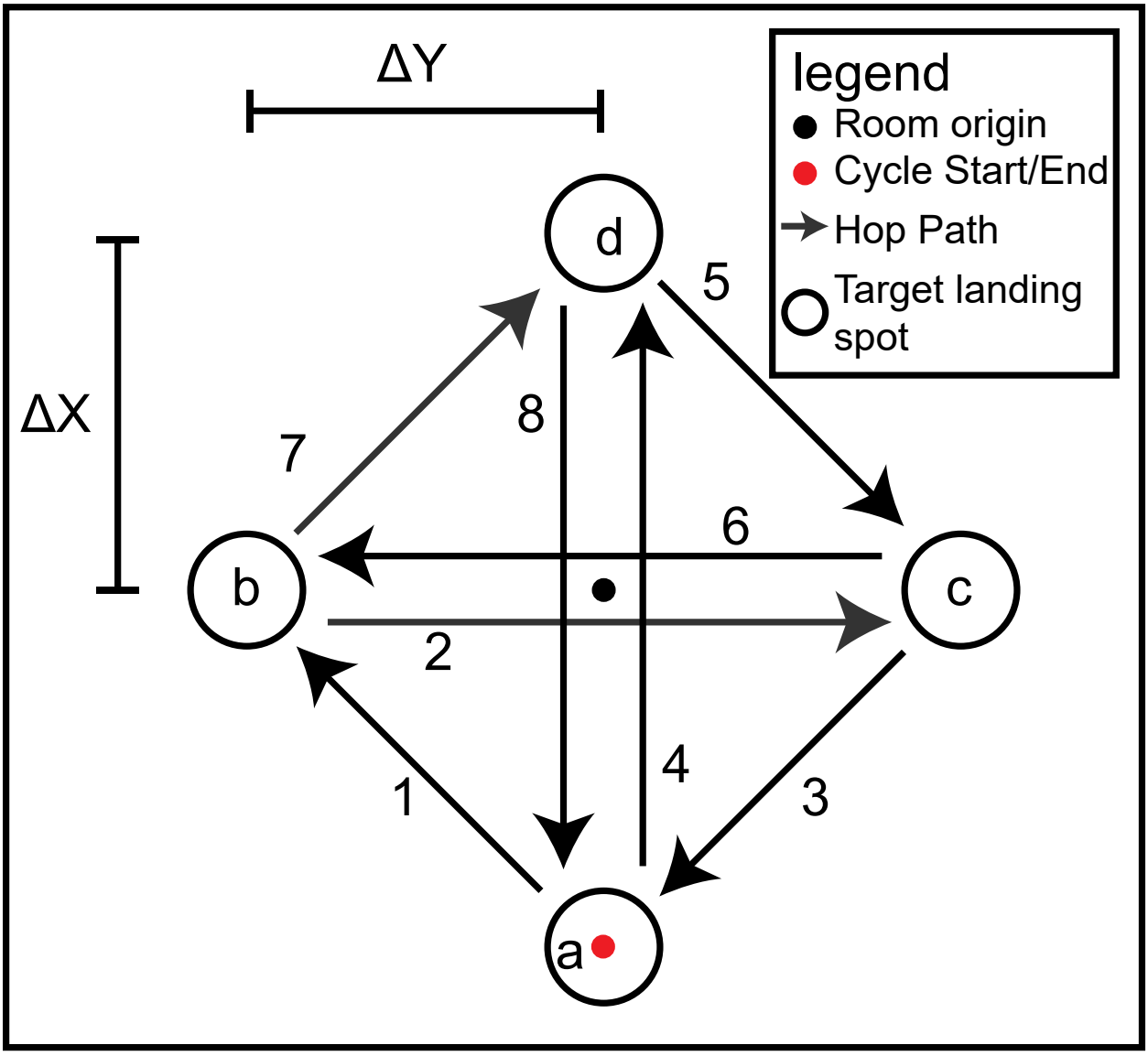

Figure 5.7: Quadrant hopping locomotion path, Red dot: start/end position, travels in order along paths 1-8 landing in target positions a-d.

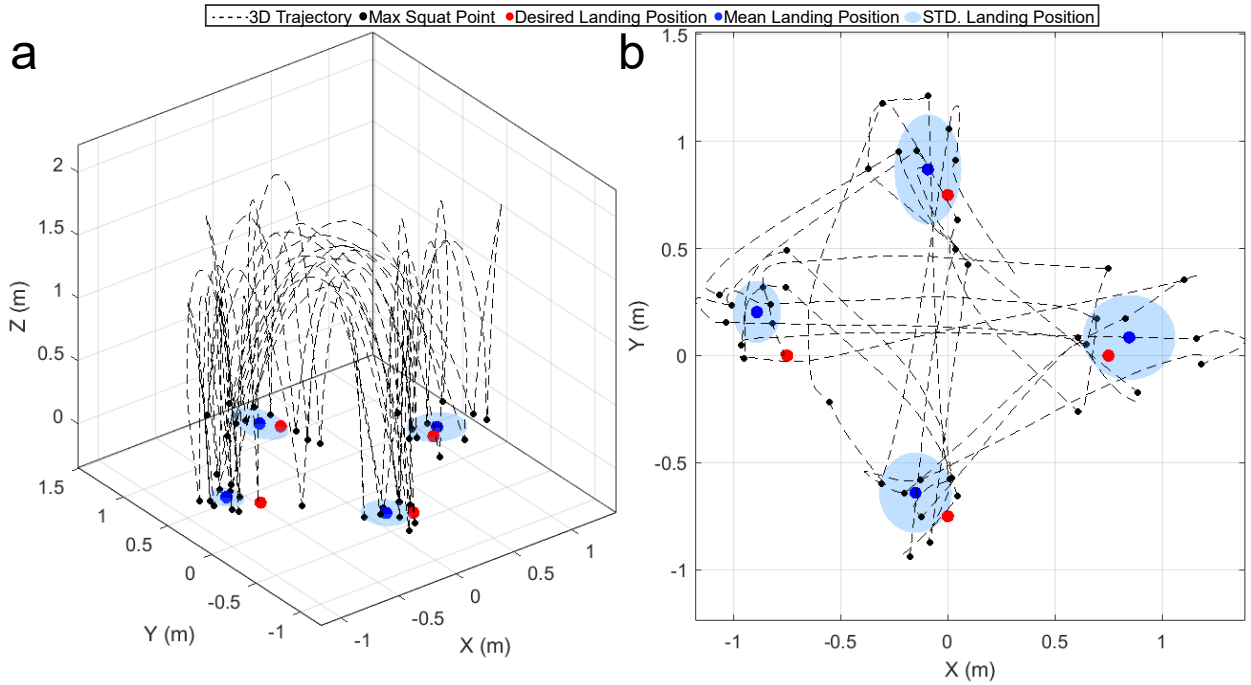

Figure 5.8: Quadrant hopping locomotion experiment trajectory path. a: 3D view, b: 2D XY view.

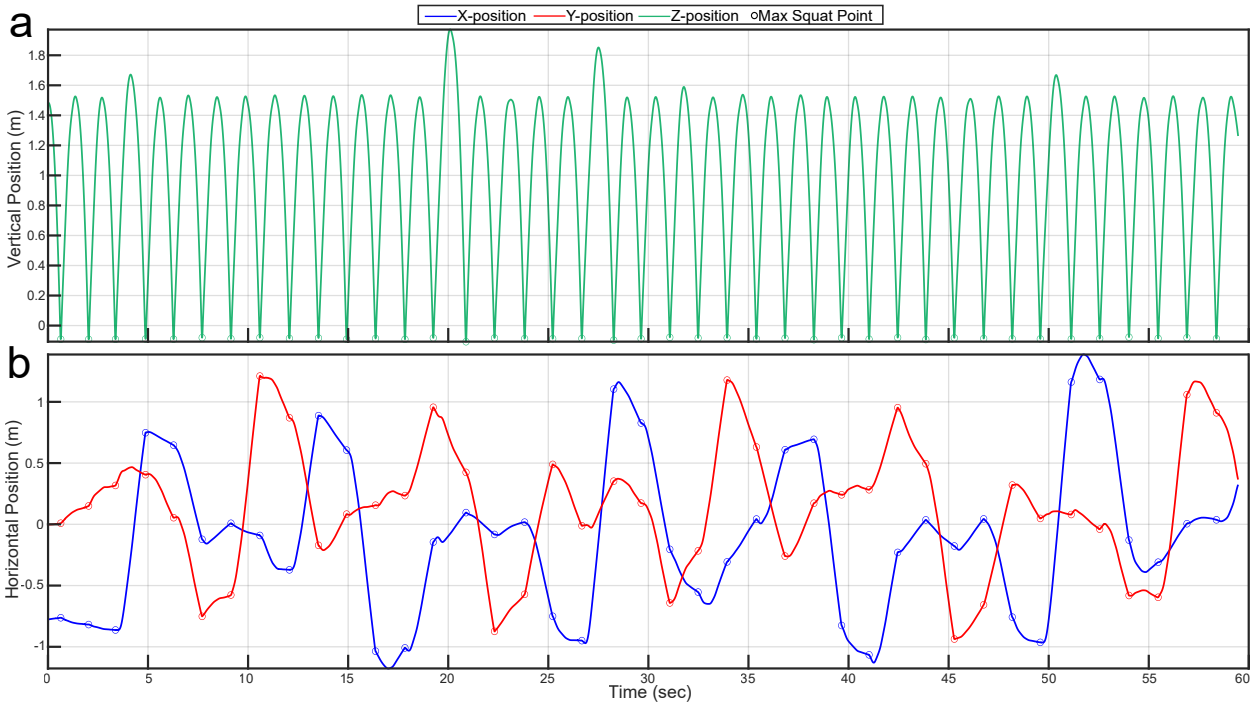

Figure 5.9: Quadrant hopping locomotion experiment positioning. a: Vertical or Z height, b: Horizontal or X and Y positioning.

5.6. Summary

Here, an alternative control strategy was applied to an existing hopping robot platform, the MultiMo-MHR. Prior work saw a PiD controller utilized, with hopping locomotion centered around vertical in-place hopping with no self-imposed or external disturbances. Due to the nature of PiD controllers being best suited for that for which the gains were tuned, and given their linear nature, stability is not guaranteed, and control for alternative or more intensive hopping may not function the same or at all. A Lyapunov controller was applied, known for being model-based, with guaranteed stability and eventual state convergence. This control strategy has been utilized for highly dynamic drone work, and given the close relation between the MultiMo-MHR and quadrotors while in the air, it was selected as a solid new control strategy that could handle more extreme system dynamics. The new controller was developed and implemented on the robot, alongside several additional controller trajectory setting improvements to better respond to higher levels of energy in the horizontal and orientation axes. The base Lyapunov controller was selected due to the aforementioned benefits and knowledge of system parameters. If this is not the case, alternative controllers, such as model reference adaptive control, could be used to improve overall adaptability and robustness, and could be a source of future work. As the controller focuses on purely aerial dynamics, the system is truly a hybrid system with varying dynamics between stance and aerial phases. This was accounted for by the adaptive parameter on the touchdown angle, which reduces error. Although it does not guarantee energy reduction for any single hop, it does reduce energy in subsequent continuous hops. Expanding system dynamics to incorporate the true hybrid nature can be a source of future research.

Moving to the experiments, a test was established in which the MultiMo-MHR would have a self-imposed disturbance, which it would then have to recover from and maintain a desired target hop positioning. The PiD controller was tested with a disturbance and was found to function well for minor disturbances, but while functional at larger disturbances,

it saw a great reduction in hopping height, which then had to build up over several hops. The PiD also failed to perform at higher thrust-to-weight ratios, due to too much variation between the originally tuned gains and the dynamics experienced with a disturbance and higher allowed power. The new Lyapunov controller was tested and found that stability was readily achieved with a disturbance, and saw no reduction in hopping height as a result of disturbance recovery. Additionally, the Lyapunov controller was able to follow a hopping variable horizontal position trajectory, showing reliable hopping height and solid foot placement between hops. This work sets the groundwork for further testing and higher dynamic trajectories, where the controller can be further refined to both better follow a set path and handle extreme energy disturbances.

6. Summary and Future Work

6.1. Thesis Summary

This work presents a novel hopping platform with a focus on design and control elements for efficient hopping, presenting a novel state estimation methodology and an implementation of a Lyapunov controller to boost system stability and operation under higher dynamic states. The work strives to expand the scope of hopping locomotion by entering higher energy conditions. Achieving an understanding of system design and controller operation to improve system efficiency for systems with aerial-based energy input methods. And showing a state estimation scheme which relies only on accelerometer input to achieve full vertical state estimation that is reliable and robust enough to enable accurate hopping height control. Concluding with advancing the controller of the MultiMo-MHR through a nonlinear Lyapunov control scheme, which allows the platform to achieve far more dynamic motions in all axes. The concepts presented, while conducted on the MultiMo-MHR, can be generalized for any hopping platform to improve the performance of hopping systems overall.

6.2. Future Work

6.2.1. Hopping controller variation and analysis

Expanding upon the work conducted in Chapter 5, future work is to further the investigation on the non-linear Lyapunov controller for hopping locomotion. By testing alternative control schemes and advanced trajectories, we can understand the overall impact they have on system parameters, like stability and energy usage. Further expanding upon more advanced trajectories, from increasing system angles and horizontal velocities to more challenging maneuvers like wall jumps or climbs.

Continuing on from the work in Chapter 5, the scope would be twofold. One is the

development of an analysis methodology for evaluating a controller's performance. To achieve the analysis pipeline, from an energy perspective, the addition of a new electronic speed controller with the telemetry feature for real-time RPM measurements would serve to greatly improve energy calculations. The current MultiMo-MHR can record the commanded motor order, not the actual experience, given motor ramp-up/down time. Even without the RPM measurement, the energy analysis methodology can still be employed. Outside of pure energy usage, controller stability and performance are another key metric that would be required for this future work. A critical question to ask is how to evaluate a hopping robot; one such metric can be how long the system takes to stabilize from a disturbance, including the number of hops, time to stability, hop height variation, horizontal displacement required to achieve stability, and general energy usage required. Establishing a dimensionless parameter for evaluating system stability could be one strategy, though careful consideration is required not to favor one style of hopping over another. The Unified Hopping Agility metric from Chapter 4.9.1 is an example of a metric for hopping which focuses on hop height and time of the various phases, though such a metric does have limitations as previously discussed.

The second aspect of this work would be the implementation of alternative control schemes and advanced trajectories on the robot, and ideally in simulation for initial testing for feasibility. Currently, a linear transformed PiD and a non-linear Lyapunov controller have been utilized on the robot, with functioning and reliable control being seen with the current testing regime. The PiD functions well for basic operation, but given its linear and non-model-based characteristics, it underperformed with more extreme locomotion strategies. Hence, the implementation of the Lyapunov controller, whose current testing shows solid functionality under disturbances. Testing on these controllers is currently ongoing, with fine-tuning of controller gains and establishment of a simulation to test the aforementioned and other control strategies. More intensive experimental evaluations are target points, both higher horizontal velocities to recover from, to more complex maneuvers such as wall jumps or wall climbs, and to more precise locomotion and trajectory following.

6.2.2. System localization and mapping, outdoor testing

A hopping-based locomotion strategy can enable a rapid environment traversal, both horizontally and vertically, while being more efficient than a typical quadrotor. Hopping locomotion can also explore very similar spaces to quadrotors, only requiring some surface to interact with occasionally, but also being able to reach extreme heights and having the potential for rapid horizontal motion. Further, when compared to terrestrial-centric systems, hopping can explore far more rugged, non-flat, or spaces with gaps, thereby expanding its viability in various locations. To this end, establishing a methodology for a deeper level of state estimation in addition to environment mapping is a target for future work. Reliable full state estimation with mapping capabilities would enable advanced testing within the real world (i.e., beyond a lab setting), allowing the pursuit of further controller testing. This would also help the system to be utilized for more tasks, such as environment exploration or search and rescue tasks; being able to explore environments like a drone with the ability to operate for longer with improved efficiency.

To achieve this improved feature set, the base MultiMo-MHR would require improvements to its sensor suite. Additionally, the algorithms for estimation and mapping will need new methodologies given the constant motion and hard impacts of hopping locomotion. As displayed in Chapter 4, reliable state estimation is achievable in part thanks to clear detectable measurement updates (IMUPTs) and estimator optimization. Similar methodologies are expected to work in other axes, which can be advanced with additional sensors.

Initial proof of concept work can see minimal sensor addition, including LiDar, barometer, and incorporating additional sensor data from IMU's, i.e., in addition to the base accelerometer, using gyroscopic and magnetometer readings. The aforementioned sensors come in small packages and can easily function indoors or outdoors. These would have the capacity to improve the already displayed vertical height and velocity estimation, with the goal of producing horizontal and orientation-based state estimation.

Subsequent sensor suites can add GPS and radar to boost the estimation capabilities.

The LiDAR and radar technologies can serve as initial means for localization and mapping for the robot. Having high sample rates on a majority of the sensor suite would be a key aspect due to the constant, rapid motion nature of hopping locomotion. The GPS can be used in outdoor spaces to provide reliable position data. Cameras can be another path; however, this is anticipated to require additional work, as a high frame rate will be required on top of the image processing and visual odometry. Additional computational hardware will be required to do these calculations, which can then send commands to the already established ESP32 microcontroller.

6.2.3. Active and passive foot variations & surface modifications

Future work that can greatly improve the MultiMo-MHR is that of foot design. The current foot is of a flat-foot design, good for vertical hopping, and enables horizontal motion. However, as more horizontal motion and non-flat surfaces or walls are used in testing, a more refined and compliant foot is needed to improve efficiency and control. Due to the current flat foot design, limitations exist on its capabilities, especially at high touchdown angles and high horizontal velocities at TD, where the combined creates minimal foot contact area, enabling slippage and failed hops. This can be amended with a redesigned foot, which can be developed.

For a new foot, there are a few requirements. One being the weight it adds, ideally, the foot redesign would be as minimal as possible, given that any added mass will lead to lost energy on surface contact, which will need to be added back during the hop. Next is the strength and durability of the foot, requiring being able to withstand high forces, ground reaction forces, and torques, and to do so for many continuous hops (ideally in the thousands life cycle range to enable repeated testing). Given that these two requirements lead to polar opposite design requirements, compromises will be required.

To improve on general foot design, a baseline start can be seen with the foot of the MultiMo-Bat and the multi-Surface Locust Inspired Passively-adaptable (SLIP) Foot created

for it [87,98]. This work shows a passive foot compliant with many surface types and reliable hopping, both starting from rest and for high-velocity impact-based jumps. Alternatively, passive soft feet can enable conformity and some rigidity. However, material selection and design will be critical to ensure survivability and a solid lifespan to ensure functionality for prolonged hopping.

Alternatively, the foot or ankle joint is another method to add passive/active compliance. Joints such as a universal or ball joint can enable multi-axis motion and surface conformation. A ball joint is an expected solid candidate, due to the minimal joints involved (especially compared to the universal joint) and fewer failure points. Design choices to prevent joint disconnection and to determine the right amount of friction in the joint will be required. An active foot can be possible, though it is more likely that any active system orientation will be done with the main system motors and not at the foot, as adding any motors or similar to the foot would greatly increase the weight and losses.

The topic of foot design then simultaneously leads to the topic of what surfaces the robot is compatible with. As it stands, only a solid surface has been tested with the MultiMo-MHR. Evaluating both the base flat foot design along with alternative feet on other natural surfaces such as grass, dirt, or sand with variability in surface compliance and conditions (i.e., wet or dry). Based on knowledge of the aforementioned other surfaces, sand will likely require the largest foot modification, as the amount of energy removed due to sand penetration is expected to greatly hinder hopping performance. This can be tested with an experimental testing rig. Such work has been shown in literature for hopping on granular media, albeit at low heights of $\sim 60mm$, and then later moved to the real system [58]. This expansion of both foot design for alternative surfaces and non-flat surfaces would greatly expand the capabilities of the hopping robot, and generate an understanding for better methodologies to traverse these environments.

6.3. Publications

- S. Burns and M. Woodward, "Design and Control of a High-Performance Hopping Robot," *IEEE Robotics and Automation Letters*, vol. 10, pp. 5641–5648, June 2025.

- S. Burns and M. Woodward, (under review) "Optimized Kalman Filter based State Estimation and Height Control in Hopping Robots," *IEEE Transactions on Mechatronics*.

Bibliography

[1] S. Bai, Q. Pan, R. Ding, H. Jia, Z. Yang, and P. Chirarattananon, "An agile monopedal hopping quadcopter with synergistic hybrid locomotion," *Science Robotics*, vol. 9, p. eadi8912, Apr. 2024.

[2] "3D One-Leg Hopper."

[3] P. Gregorio, M. Ahmadi, and M. Buehler, "Experiments with an electrically actuated planar hopping robot," in *Experimental Robotics III* (T. Yoshikawa and F. Miyazaki, eds.), vol. 200, pp. 267–281, London: Springer-Verlag, 1994. Series Title: Lecture Notes in Control and Information Sciences.

[4] M. Ahmadi and M. Buehler, "Preliminary experiments with an actively tuned passive dynamic running robot," in *Experimental Robotics V* (M. Thoma, A. Bensoussan, M. J. Grimble, P. Kokotovic, H. Kwakernaak, J. L. Massey, Y. Z. Tsypkin, A. Casals, and A. T. De Almeida, eds.), vol. 232, pp. 312–324, Berlin, Heidelberg: Springer Berlin Heidelberg, 1998. Series Title: Lecture Notes in Control and Information Sciences.

[5] J.-C. Zufferey, "First Jumps of the 3D Bow Leg Hopper," Feb. 2001.

[6] D. W. Haldane, M. M. Plecnik, J. K. Yim, and R. S. Fearing, "Robotic vertical jumping agility via Series-Elastic power modulation," *Science Robotics*, vol. 1, no. 1, 2016.

[7] D. W. Haldane, J. K. Yim, and R. S. Fearing, "Repetitive extreme-acceleration (14-g) spatial jumping with Salto-1P," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Vancouver, BC), pp. 3345–3351, IEEE, Sept. 2017.

[8] Z. Batts, J. Kim, and K. Yamane, "Untethered One-Legged Hopping in 3D Using Linear Elastic Actuator in Parallel (LEAP)," in *2016 International Symposium on Exper-*

imental Robotics (D. Kulić, Y. Nakamura, O. Khatib, and G. Venture, eds.), vol. 1, pp. 103–112, Cham: Springer International Publishing, 2017. Series Title: Springer Proceedings in Advanced Robotics.

[9] B. Zhu, J. Xu, A. Charway, and D. Saldaña, "PogoDrone: Design, Model, and Control of a Jumping Quadrotor," Apr. 2022. arXiv:2204.00207 [cs].

[10] Y. Wang, J. Kang, Z. Chen, and X. Xiong, "Terrestrial Locomotion of PogoX: From Hardware Design to Energy Shaping and Step-to-step Dynamics Based Control," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3419–3425, May 2024.

[11] Y.-H. Hsiao, S. Bai, Z. Guan, S. Kim, Z. Ren, P. Chirarattananon, and Y. Chen, "Hybrid locomotion at the insect scale: Combined flying and jumping for enhanced efficiency and versatility," *Science AdvAnceS*, 2025.

[12] C. P. McGowan and C. E. Collins, "Why do mammals hop? Understanding the ecology, biomechanics and evolution of bipedal hopping," *Journal of Experimental Biology*, vol. 221, June 2018. Publisher: The Company of Biologists.

[13] E. W. Hawkes, C. Xiao, R.-a. Peloquin, C. Keeley, M. R. Begley, M. T. Pope, and G. Niemeyer, "Engineered jumpers overcome biological limits via work multiplication," *Nature*, vol. 604, no. December 2020, 2022. Publisher: Springer US.

[14] Disney, "Stuntronics."

[15] M. A. Woodward and M. Sitti, "MultiMo-Bat: A biologically inspired integrated jumping-gliding robot," *International Journal of Robotics Research*, vol. 33, no. 12, pp. 1511–1529, 2014. ISBN: 0278364914541.

[16] H. S. Seifert, "The lunar pogo stick.," *Journal of Spacecraft and Rockets*, vol. 4, pp. 941–943, July 1967.

[17] M. H. Kaplan and H. S. Seifert, "Hopping transporters for lunar exploration.," *Journal of Spacecraft and Rockets*, vol. 6, pp. 917–922, Aug. 1969.

[18] M. H. Raibert, *Legged robots that balance*. The MIT Press series in artificial intelligence, Cambridge, Mass.: MIT Press, 1986.

[19] M. H. Raibert, H. B. Brown, and M. Chepponis, "Experiments in Balance with a 3D One-Legged Hopping Machine," *The International Journal of Robotics Research*, vol. 3, no. 2, pp. 75–92, 1984.

[20] M. H. Raibert, "Hopping in legged systems — Modeling and simulation for the two-dimensional one-legged case," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-14, pp. 451–463, May 1984.

[21] Z. Zhang, J. Zhao, H. Chen, and D. Chen, "A Survey of Bioinspired Jumping Robot: Takeoff, Air Posture Adjustment, and Landing Buffer," *Applied Bionics and Biomechanics*, vol. 2017, pp. 1–22, 2017.

[22] A. Sayyad, B. Seth, and P. Seshu, "Single-legged hopping robotics research - A review," *Robotica*, vol. 25, no. 5, pp. 587–613, 2007.

[23] C. Zhang, W. Zou, L. Ma, and Z. Wang, "Biologically inspired jumping robots: A comprehensive review," *Robotics and Autonomous Systems*, vol. 124, p. 103362, Feb. 2020.

[24] A. Lebaudy, J. Presser, and M. Kam, "Control algorithms for a vertically-constrained one-legged hopping machine," in *Proceedings of 32nd IEEE Conference on Decision and Control*, pp. 2688–2693 vol.3, Dec. 1993.

[25] P. Gregorio, "Design Control and Energy Minimization Strategies for the ARL Monopod,"

[26] P. Gregorio, M. Ahmadi, and M. Buehler, "Design, control, and energetics of an electrically actuated legged robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 27, pp. 626–634, Aug. 1997.

[27] M. Ahmadi and M. Buehler, "The ARL monopod II running robot: control and energetics," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 3, pp. 1689–1694 vol.3, May 1999. ISSN: 1050-4729.

[28] G. Zeglin and H. B. Brown, "First Hops of the 3D Bow Leg Hopper," *International Conference on Climbing and Walking Robots (CLAWAR)*, pp. 357–364, 2002.

[29] J. K. Yim, B. R. P. Singh, E. K. Wang, R. Featherstone, and R. S. Fearing, "Precision Robotic Leaping and Landing Using Stance-Phase Balance," *IEEE Robotics and Automation Letters*, vol. 5, pp. 3422–3429, Apr. 2020.

[30] J. K. Yim, E. K. Wang, and R. S. Fearing, "Drift-free Roll and Pitch Estimation for High-acceleration Hopping," in *2019 International Conference on Robotics and Automation (ICRA)*, (Montreal, QC, Canada), pp. 8986–8992, IEEE, May 2019.

[31] J. K. Yim and R. S. Fearing, "Precision Jumping Limits from Flight-phase Control in Salto-1P," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Madrid), pp. 2229–2236, IEEE, Oct. 2018.

[32] S. Burns and M. Woodward, "Design and Control of a High-Performance Hopping Robot," *IEEE Robotics and Automation Letters*, vol. 10, pp. 5641–5648, June 2025.

[33] J. Kang, Y. Wang, and X. Xiong, "Fast Decentralized State Estimation for Legged Robot Locomotion via EKF and MHE," May 2024. arXiv:2405.20567 [cs].

[34] M. M. Plecnik, D. W. Haldane, J. K. Yim, and R. S. Fearing, "Design exploration and kinematic tuning of a power modulating jumping monopod," *Journal of Mechanisms and Robotics*, vol. 9, no. 1, pp. 1–13, 2017.

[35] G. Cioffi, L. Bauersfeld, E. Kaufmann, and D. Scaramuzza, "Learned Inertial Odometry for Autonomous Drone Racing," Feb. 2023. arXiv:2210.15287 [cs].

[36] Y. Chen and N. O. Pérez-Arancibia, "Lyapunov-based controller synthesis and stability analysis for the execution of high-speed multi-flip quadrotor maneuvers," in *2017 American Control Conference (ACC)*, pp. 3599–3606, May 2017. ISSN: 2378-5861.

[37] S. Bouabdallah, P. Murrieri, and R. Siegwart, "Design and control of an indoor micro quadrotor," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 5, pp. 4393–4398 Vol.5, Apr. 2004. ISSN: 1050-4729.

[38] A. Mohsin and J. Abdulhady, "Comparing dynamic model and flight control of plus and cross quadcopter configurations," *FME Transactions*, vol. 50, no. 4, pp. 683–692, 2022.

[39] V. M. S. , "V5 camera specifications - Vantage documentation - Vicon Help."

[40] R. Armour, K. Paskins, A. Bowyer, J. Vincent, and W. Megill, "Jumping robots: A biomimetic solution to locomotion across rough terrain," *Bioinspiration and Biomimetics*, vol. 2, no. 3, 2007.

[41] Boston Dynamics, "Sandflea," 2013.

[42] M. Kovač, M. Schlegel, J.-C. Zufferey, and D. Floreano, "Steerable miniature jumping robot," *Autonomous Robots*, vol. 28, pp. 295–306, Apr. 2010.

[43] P. Fiorini, S. Hayati, M. Heverly, and J. Gensler, "A hopping robot for planetary exploration," in *1999 IEEE Aerospace Conference. Proceedings (Cat. No.99TH8403)*, (Snowmass at Aspen, CO, USA), pp. 153–158 vol.2, IEEE, 1999.

[44] J. Zhao, N. Xi, B. Gao, M. W. Mutka, and L. Xiao, "Design and testing of a controllable miniature jumping robot," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3346–3351, Oct. 2010. ISSN: 2153-0866.

[45] J. Zhao, N. Xi, B. Gao, M. W. Mutka, and L. Xiao, "Development of a controllable and continuous jumping robot," in *2011 IEEE International Conference on Robotics and Automation*, pp. 4614–4619, May 2011. ISSN: 1050-4729.

[46] J. Zhao, J. Xu, B. Gao, N. Xi, F. J. Cintron, M. W. Mutka, and L. Xiao, "MSU jumper: A single-motor-actuated miniature steerable jumping robot," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 602–614, 2013.

[47] B. Lambrecht, A. Horchler, and R. Quinn, "A Small, Insect-Inspired Robot that Runs and Jumps," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, (Barcelona, Spain), pp. 1240–1245, IEEE, 2005.

[48] A. M. Johnson and D. E. Koditschek, "Toward a vocabulary of legged leaping," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2568–2575, 2013. ISBN: 9781467356411.

[49] A. L. Brill, A. De, A. M. Johnson, and D. E. Koditschek, "Tail-assisted rigid and compliant legged leaping," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem, no. January, pp. 6304–6311, 2015. ISBN: 9781479999941.

[50] G. P. Jung, C. S. Casarez, S. P. Jung, R. S. Fearing, and K. J. Cho, "An integrated jumping-crawling robot using height-adjustable jumping module," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, pp. 4680–4685, 2016. Publisher: IEEE ISBN: 9781467380263.

[51] P. Fiorini and J. Burdick, "The Development of Hopping Capabilities for Small Robots," *Autonomous Robots*, vol. 14, pp. 239–254, Mar. 2003.

[52] S. A. Stoeter and N. Papanikolopoulos, "Autonomous stair-climbing with miniature jumping robots," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 35, no. 2, pp. 313–325, 2005.

[53] Y. Ho, S. Kyoung, T. Jin, J. H. Lee, J. Y. Choi, C. H. Yim, and D. H. Kim, "Mecianism and Control of a Jumping Robot," in *International Conference on Control, Automation and Systems 2007*, pp. 2499–2502, 2007.

[54] F. Li, G. Bonsignori, U. Scarfogliero, D. Chen, C. Stefanini, W. Liu, P. Dario, and X. Fu, "Jumping mini-robot with bio-inspired legs," in *2008 IEEE International Conference on Robotics and Biomimetics*, pp. 933–938, Feb. 2009.

[55] J. Zhao, R. Yang, N. Xi, B. Gao, X. Fan, M. W. Mutka, and L. Xiao, "Development of a miniature self-stabilization jumping robot," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2217–2222, Oct. 2009. ISSN: 2153-0866.

[56] U. Scarfogliero, C. Stefanini, and P. Dario, "Design and Development of the Long-Jumping " Grillo " Mini Robot," *IEEE International Conference on Robotics and Automation*, no. April, pp. 10–14, 2007.

[57] T. Ho and S. Lee, "A novel design of a robot that can jump and roll with a single actuator," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 908–913, Oct. 2012. ISSN: 2153-0866.

[58] J. Aguilar and D. I. Goldman, "Robophysical study of jumping dynamics on granular media," *Nature Physics*, vol. 12, pp. 278–283, Mar. 2016.

[59] A. Yamada, M. Watari, H. Mochiyama, and H. Fujimoto, "An asymmetric robotic catapult based on the closed elastica for jumping robot," in *2008 IEEE International Conference on Robotics and Automation*, pp. 232–237, May 2008. ISSN: 1050-4729.

[60] A. Yamada, H. Mameda, H. Mochiyama, and H. Fujimoto, "A compact jumping robot utilizing snap-through buckling with bend and twist," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 389–394, Oct. 2010. ISSN: 2153-0866.

[61] V. Zaitsev, O. Gvirsman, U. Ben Hanan, A. Weiss, A. Ayali, and G. Kosa, "A locust-inspired miniature jumping robot," *Bioinspiration & Biomimetics*, vol. 10, p. 066012, Nov. 2015.

[62] M. A. Woodward and M. Sitti, "Design of a miniature integrated multi-modal jumping and gliding robot," *IEEE International Conference on Intelligent Robots and Systems*, pp. 556–561, 2011. Publisher: IEEE ISBN: 9781612844541.

[63] M. Kovač, Wassim-Hraiz, O. Fauria, J.-C. Zufferey, and D. Floreano, "The EPFL jumpglider: A hybrid jumping and gliding robot with rigid or folding wings," in *2011 IEEE International Conference on Robotics and Biomimetics*, pp. 1503–1508, Dec. 2011.

[64] A. L. Desbiens, M. Pope, F. Berg, Z. E. Teoh, J. Lee, and M. Cutkosky, "Efficient jumpgliding: Theory and design considerations," in *2013 IEEE International Conference on Robotics and Automation*, (Karlsruhe, Germany), pp. 4451–4458, IEEE, May 2013.

[65] A. L. Desbiens, M. T. Pope, D. L. Christensen, E. W. Hawkes, and M. R. Cutkosky, "Design principles for efficient, repeated jumpgliding," *Bioinspiration & Biomimetics*, vol. 9, p. 025009, May 2014.

[66] A. Beck, V. Zaitsev, U. B. Hanan, G. Kosa, A. Ayali, and A. Weiss, "Jump stabilization and landing control by wing-spreading of a locust-inspired jumper," *Bioinspiration & Biomimetics*, vol. 12, p. 066006, Oct. 2017.

[67] M. A. Woodward and M. Sitti, "Tailored Magnetic Springs for Shape-Memory Alloy Actuated Mechanisms in Miniature Robots," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 589–601, 2019. Publisher: IEEE.

[68] E. Ambrose, W.-L. Ma, and A. D. Ames, "Towards the Unification of System Design and Motion Synthesis for High-Performance Hopping Robots," in *2021 IEEE International*

Conference on Robotics and Automation (ICRA), pp. 7073–7078, May 2021. ISSN: 2577-087X.

[69] N. Csomay-Shanklin, V. D. Dorobantu, and A. D. Ames, "Nonlinear Model Predictive Control of a 3D Hopping Robot: Leveraging Lie Group Integrators for Dynamically Stable Behaviors," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12106–12112, May 2023.

[70] O. Okubo, E. Nakano, and M. Handa, "Design of a jumping machine using self-energizing spring," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS '96*, vol. 1, pp. 186–191 vol.1, Nov. 1996.

[71] U. Scarfogliero, C. Stefanini, and P. Dario, "A bioinspired concept for high efficiency locomotion in micro robots: The jumping Robot Grillo," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2006, no. May, pp. 4037–4042, 2006. Publisher: IEEE ISBN: 0780395069.

[72] F. Guenther and F. Iida, "Energy-Efficient Monopod Running With a Large Payload Based on Open-Loop Parallel Elastic Actuation," *IEEE Transactions on Robotics*, vol. 33, pp. 102–113, Feb. 2017. Conference Name: IEEE Transactions on Robotics.

[73] B. A. Christensen, D. C. Lin, M. J. Schwaner, and C. P. McGowan, "Elastic energy storage across speeds during steady-state hopping of desert kangaroo rats (Dipodomys deserti)," *Journal of Experimental Biology*, vol. 225, no. 2, 2022.

[74] R. A. de Callafon, *Lagrange's Method application to the vibration analysis of a flexible structure.*

[75] J. Tatum, "13.4: The Lagrangian Equations of Motion," Jan. 2017.

[76] S. Caron, "Joint torques and Jacobian transpose."

[77] M. Kovač, M. Fuchs, A. Guignard, J. C. Zufferey, and D. Floreano, "A miniature 7g jumping robot," *Proceedings - IEEE International Conference on Robotics and Automation*, no. c, pp. 373–378, 2008. ISBN: 9781424416479.

[78] B. Shin, H.-Y. Kim, and K.-J. Cho, "Towards a biologically inspired small-scale water jumping robot," in *2008 2nd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 127–131, Oct. 2008. ISSN: 2155-1782.

[79] W. A. Churaman, A. P. Gerratt, and S. Bergbreiter, "First leaps toward jumping microrobots," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1680–1686, Sept. 2011. ISSN: 2153-0866.

[80] M. Noh, S. W. Kim, S. An, J. S. Koh, and K. J. Cho, "Flea-inspired catapult mechanism for miniature jumping robots," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1007–1018, 2012. Publisher: IEEE.

[81] W. A. Churaman, L. J. Currano, C. J. Morris, J. E. Rajkowski, and S. Bergbreiter, "The First Launch of an Autonomous Thrust-Driven Microrobot Using Nanoporous Energetic Silicon," *Journal of Microelectromechanical Systems*, vol. 21, pp. 198–205, Feb. 2012. Conference Name: Journal of Microelectromechanical Systems.

[82] J. S. Koh, S. P. Jung, M. Noh, S. W. Kim, and K. J. Cho, "Flea inspired catapult mechanism with active energy storage and release for small scale jumping robot," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 26–31, 2013. ISBN: 9781467356411.

[83] M. T. Tolley, R. F. Shepherd, M. Karpelson, N. W. Bartlett, K. C. Galloway, M. Wehner, R. Nunes, G. M. Whitesides, and R. J. Wood, "An untethered jumping soft robot," *IEEE International Conference on Intelligent Robots and Systems*, no. Iros, pp. 561–566, 2014. Publisher: IEEE ISBN: 9781479969340.

[84] N. W. Bartlett, M. T. Tolley, J. T. B. Overvelde, J. C. Weaver, B. Mosadegh, K. Bertoldi, G. M. Whitesides, and R. J. Wood, "A 3D-printed, functionally graded soft robot powered by combustion," *Science*, vol. 349, pp. 161–165, July 2015. Publisher: American Association for the Advancement of Science.

[85] Y. Pan, G. Tang, S. Liu, D. Mei, L. He, S. Zhu, and Y. Wang, "Tumro: A Tunable Multimodal Wheeled Jumping Robot Based on the Bionic Mechanism of Jumping Beetles," *Advanced Intelligent Systems*, vol. 6, no. 9, p. 2400024, 2024. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/aisy.202400024.

[86] M. A. Woodward and M. Sitti, "Universal Custom Complex Magnetic Spring Design Methodology," *IEEE Transactions on Magnetics*, vol. 54, pp. 1–13, Jan. 2018.

[87] M. A. Woodward and M. Sitti, "Morphological intelligence counters foot slipping in the desert locust and dynamic robots," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 115, no. 36, pp. E8358–E8367, 2018.

[88] J. Zhao, T. Zhao, N. Xi, M. W. Mutka, and L. Xiao, "MSU Tailbot: Controlling Aerial Maneuver of a Miniature-Tailed Jumping Robot," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 6, pp. 2903–2914, 2015.

[89] U. Scarfogliero, C. Stefanini, and P. Dario, "Design and Development of the Long-Jumping "Grillo" Mini Robot," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 467–472, Apr. 2007. ISSN: 1050-4729.

[90] D. W. Haldane, M. Plecnik, J. K. Yim, and R. S. Fearing, "A power modulating leg mechanism for monopedal hopping," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Daejeon, South Korea), pp. 4757–4764, IEEE, Oct. 2016.

[91] J. S. Lee, M. Plecnik, J.-H. Yang, and R. S. Fearing, "Self-Engaging Spined Gripper with Dynamic Penetration and Release for Steep Jumps," in *2018 IEEE International*

Conference on Robotics and Automation (ICRA), (Brisbane, QLD), pp. 1–8, IEEE, May 2018.

[92] Z. Wang, H. Zhao, S. Qiu, and Q. Gao, "Stance-Phase Detection for ZUPT-Aided Foot-Mounted Pedestrian Navigation System," *IEEE/ASME Transactions on Mechatronics*, vol. 20, pp. 3170–3181, Dec. 2015.

[93] A. Norrdine, Z. Kasmi, and J. Blankenbach, "Step Detection for ZUPT-Aided Inertial Pedestrian Navigation System Using Foot-Mounted Permanent Magnet," *IEEE Sensors Journal*, vol. 16, pp. 6766–6773, Sept. 2016.

[94] H. Ju, M. S. Lee, S. Y. Park, J. W. Song, and C. G. Park, "A pedestrian dead-reckoning system that considers the heel-strike and toe-off phases when using a foot-mounted IMU," *Measurement Science and Technology*, vol. 27, p. 015702, Jan. 2016.

[95] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Computer Graphics and Applications*, vol. 25, pp. 38–46, Nov. 2005.

[96] W. Liu, D. Caruso, E. Ilg, J. Dong, A. I. Mourikis, K. Daniilidis, V. Kumar, and J. Engel, "TLIO: Tight Learned Inertial Odometry," *IEEE Robotics and Automation Letters*, vol. 5, pp. 5653–5660, Oct. 2020. arXiv:2007.01867 [cs].

[97] S. Burns and M. Woodward, "Optimized Kalman Filter based State Estimation and Height Control in Hopping Robots," June 2025. arXiv:2408.11978 [cs].

[98] M. A. Woodward, "MultiMo-Bat: Biologically Inspired Integrated Multi-Modal Locomotion," *ProQuest Dissertations and Theses*, p. 208, 2017. ISBN: 9780355496734.