

Bio-inspired Robotics: A Kangaroo Rat Inspired Stunt Robot



Prepared by:

Josh Perry

PRRJOS001

Department of Electrical and Electronics Engineering

University of Cape Town

Prepared for:

Dr A. Patel

University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfillment of the academic requirements for a Bachelor of Science degree in Mechatronics

Key Words:

Bio-inspired Robotics, Control, Modelling, Mechatronics Design

October, 2015

Plagiarism Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that is is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it as their work or part thereof.

Signature of Author:

Josh Perry

Date 14 October 2015

Terms of Reference

Title

A Kangaroo Rat Inspired Stunt Robot

Description

Kangaroo rats have been shown to use their tails for acrobatic manoeuvres while airborne. Previous work at UCT has demonstrated the use of a tail to perform a rapid barrel roll with a stunt car. This project will investigate the use of a tail on a car driven off a ramp to perform a 360-degree yaw stunt while airborne.

Deliverables

- i. A mathematical model of the robot-tail system developed in Simulink.
- ii. Tail controller designs.
- iii. Mechanical design of a single degree of freedom tail.
- iv. Experimental testing of the robot.

Skills / Requirements:

Modelling, Simulink, Mechatronics Design, Control Systems.

Area

Bio-inspired robotics, Control

Acknowledgements

I would like to thank my parents, my father Graham and my mother Kathy, for giving me every opportunity I could ever have wished for in life. Not only have they funded my education, but they believed in me every step of the way, which is the primary reason for any success I have had. I will forever be indebted to you both, and this paper is dedicated to you two.

I would also like to thank my sister for the role she has played in developing my intellect, although her mind is completely different to mine, she has challenged me all my life. Thank you Zsa.

I would like to thank my supervisor Amir Patel, who through his own passion was an incredibly motivating force in the project. He was always enthusiastic about the project, and his desire to see all the AP chargers succeed was amazing to experience. I cannot thank him enough for the input to this project, and helping me believe it would be amazing.

I would like to thank Callen Fisher, my thesis mentor, for his amazing support during this project. He selflessly gave of his time to help me despite handing in his Msc in August. This project would not have been possible without his help with software design and the code he wrote for communications, both on board and for the application that logs data on the PC. Callen didn't turn me away once, and always helped me patiently. Callen's passion for research engineering is inspiring, and he motivated me to work hard on this project.

I would like to thank Sean Wood for his invaluable guidance during programing, his in depth knowledge of the STM and embedded systems was hugely beneficial to this project.

I would also like to thank the firm. To Nate, Kimmel, Kyle, Kev, Richie, Marc and Woot, thank you for keeping me sane during my four years in engineering at UCT. We went through the semesters with 8 courses, the late nights in white lab, and crazy test weeks together. You guys have made every day in engineering great, thank you.

I would like to thank Alon Bernitz, my engineering senior. Thanks for always being so keen to talk engineering with me, and even talking about control this year. I love our chats, and they have really helped me during my degree. Thank you for inspiring me with your passion for what you do, without it I wouldn't have finished my degree in the same way.

Finally, I would like to thank Robbie Frames. Thank you for being the best friend I could ever have asked for at university. From Smuts in first year till now, its been 5 of the greatest years, hugely thanks to you. Thank you for making my time at UCT such an experience, and always helping me stay balanced. Thank you for your support during this paper, everything from your help building the test rig to eating ice cream so I could use the tub. Thank for for getting excited when things were going well, and being there for me when things weren't.

Abstract

The process of looking to nature for solutions to engineering problems has become well established in recent times, biomimetic systems and bio-inspired robots are becoming increasingly prevalent. This project emulated this methodology, and incorporated various principles from natural phenomena in the designs. It has been postulated that flightless animals use their tails to achieve attitude control and aerial righting; one such example of this is the kangaroo rat, which uses its tail to rotate its body 180 degrees in the yaw plane when avoiding danger. The robotic concept designed for the project was inspired by the kangaroo rat. The aim of the project was to model, design and build a robot that was capable of using a tail to perform a 360-degree yaw stunt after hitting a jump.

Initially background research was conducted in order to understand how animals and robots use tails, especially for the purpose of aerial orientation. Thereafter modelling of all the systems was performed to evaluate the feasibility of the stunt, as well as select components. The hardware, software, mechanical structure and the control system for the robot was then designed. After the design was realized, the robot was put through a variety of tests. Initially it was tested on a test rig, to ensure safe system operation. Testing was then performed using a jump to briefly launch the robot into the air, allowing it to attempt the stunt.

Finally, the data and video footage of the all the tests was collected, reviewed and critically analyzed. From this analysis conclusions were drawn about the effectiveness of using tails for attitude control of robots, the success of, and the problems associated with the stunt. Recommendations were then made for future work associated with the principles of this project.

Table of Contents

1. Introduction	1
1.1 Background to the Research	1
1.2 Research Objectives	1
1.2.1 Aims of the Project.....	1
1.2.2 The Significance of the Study	2
1.3 Scope and Limitations.....	2
1.4 Plan of Development	3
2. Literature Review.....	5
2.1 Introduction.....	5
2.2 Biomimetics	6
2.2.1 History	6
2.2.2 Methodology.....	6
2.2.3 Ecological Implications.....	7
2.2.4 Current State of the Art	7
2.2.5 Future Attributes to Attain.....	9
2.2.6 Industrial Applications.....	9
2.3 Attitude Control and Aerial Orientation in Flightless Animals	12
2.3.1 Mammalian Aerial Righting.....	13
2.3.2 Reptilian Aerial Righting.....	13
2.3.3 The Potentially Hybrid Case of the Kangaroo Rat.....	15
2.3.4 Application for Robotics	16
2.4 Tails in Nature and Robotics.....	16
2.4.1 Tails in Nature	16
2.4.2 Tails in Robotics	16
2.4.3 Tail Design to Improve the Manoeuvrability of the MIT Cheetah Robot.....	17
2.4.4 Dima Robot	18
2.4.5 TAYLRoACH	19
2.4.6 Tailbot.....	20
2.4.7 FlipBot.....	22
2.4.8 Flexible and Rigid Tails.....	22
2.5 Conclusion.....	23
3. Theory Development	24
3.1 Lagrange Dynamics	24

3.2 Friction Models of DC Motors.....	24
3.2.1 Static Friction	25
3.2.2 Coulomb Friction	25
3.2.3 Viscous Friction	25
3.2.4 Integrated Friction Model	26
3.3 Controllability and Observability.....	27
3.3.1 Matrix Rank	27
3.3.2 Observability.....	27
3.3.3 Controllability.....	28
3.4 State Feedback Steady State Error.....	29
3.4.1 Steady State Error Illustration	29
3.4.2 Outer Loop Integrator State Feedback Design.....	30
3.5 State Observers.....	31
3.5.1 Observer Formulation	31
3.5.2 Observer Design	32
3.6 Actuator Limitations	33
3.6.1 Voltage Saturation	33
3.6.2 Motor Back EMF Constraint.....	34
3.7 Euler Angles and Rotation Matrices.....	35
4. Design Methodology and Experimental Procedure.....	37
4.1 System Requirements and Research.....	37
4.2 System Modelling, Design and Simulation	37
4.2.1 Modelling of the System	37
4.2.2 Simulation	37
4.2.3 Design.....	38
4.3 Model Verification and Controller Design.....	38
4.4 Testing and Experimentation	39
5. System Modelling and Simulation.....	40
5.1 Dynamic Modelling.....	40
5.1.1 The Robotic Tail and Platform System	40
5.1.2 Development of Equations of Motion.....	41
5.1.3 Conversion to Simulation Compatible Form	42
5.3 Analysis of the Dynamics.....	42
5.3.1 Significance of the Inertial Ratio	42
5.3.1 Significance of Coupling.....	42
5.4 The DC Motor Model.....	43
5.5 Synthesis of the Dynamic and Motor Models.....	44

5.6 State Space Representation	45
5.7 Uncontrolled Motor Simulation	46
5.7.1 Motor Options	46
5.7.2 Motor Torque Speed Characteristics	46
5.8 Uncontrolled Simulation	49
5.8.1 Simulink Model Description	49
5.8.2 Simulation Results	50
5.8.2 Simulation of Motor 1.....	51
5.8.3 Simulation of Motor 2.....	53
5.8.4 Simulation of Motor 3.....	55
5.9 Motor Selection and Motivation	57
6. System Design	58
6.1 Circuit Design	58
6.1.1 Launch Detection Circuit	58
6.1.2 Driver Circuit	59
6.1.3 Encoder Circuit.....	60
6.1.4 Communication Modules	60
6.1.5 Hardware Systems Block Diagram.....	61
6.2 Software Design	62
6.2.1 Software Requirements	62
6.2.2 Software Accreditation.....	62
6.2.3 Original MCU C Code	63
6.2.4 C Code Implementation	63
6.3 Microcontroller Selection	66
6.4 Mechanical Design	67
6.4.1 The RC Car.....	67
6.4.2 The Base Platform	68
6.4.3 Tail Design.....	69
6.4.4 Test Rig Design.....	70
6.5 Design Realization	71
7. Parameter Determination and Model Verification.....	73
7.1 Calculation of Moments of Inertia	73
7.1.1 Body Mass Model and Specifications	73
7.2.2 Moment of Inertia about I_{zz}	74
7.2.3 Moment of Inertia of the Tail:.....	74
7.2 Motor Parameter Determination.....	75
7.2.1 Measured Motor Parameters.....	75

7.2.2 Experimentally Determined Motor Parameters.....	75
7.3 Motor Friction Determination	76
7.3.1 Static Friction	76
7.3.2 Dynamic and Viscous Friction	76
7.4 Gearbox Efficiency.....	77
7.5 Summary of Parameters	77
7.6 Model Verification.....	78
7.6.1 Motor Step Tests.....	78
7.6.2 Step Data Analysis	78
7.6.3 Observations About the Step Data Outcome.....	79
7.6.4 Fully Integrated Model.....	80
7.6.6 Full System Step Tests	83
7.6.7 Step Data Analysis	86
8. Controller Design.....	88
8.1 Control Aims.....	88
8.2 Control Design Method Selection.....	89
8.3 State Space Model Population	89
8.2 State Controllability and Observability	90
8.3 Loop Decoupling	90
8.3.1 Decoupled Loop Control Models	91
8.3.2 Decoupled System Analysis	91
8.4 Integrated State Feedback Control Design for Loop One	92
8.4.1 Response Requirement and Pole Placement	92
8.4.2 Gain Calculations.....	93
8.4.3 Loop One Controller Response	93
8.4.4 Comments on the Controlled Response of Loop One	94
8.5 Integrated State Feedback Control for Loop Two.....	94
8.5.1 State Feedback Design	94
8.5.2 Observer Design	95
8.5.3 Loop Two Control Response	95
8.6 Combined System Response.....	97
8.7 Controller Discretization.....	99
8.7.1 Loop Two Controller Discretization and Simplification.....	99
8.7.2 Analysis of the Discrete Control System Performance	102
8.8 Controller Robustness Analysis	103
8.9 Final Comments on the Control System	103
9. Testing and Results	104

9.1 Test Rig Control Test.....	104
9.2 Creation of The Test Environment.....	106
9.2.1 Jump Conceptualization and Construction	106
9.2.2 Test Environment.....	107
9.3 Testing Prior to Actuated Jumps	107
9.3.2 Unactuated Jump Testing.....	107
9.4 Actuated Jump Tests.....	107
9.4.1 Video Footage and Documentation	108
9.4.2 Body Yaw Position, Velocity and Motor Velocity Data.....	111
9.4.3 Comments on the Data Presented	114
9.4.4 Pitch and Roll Data	115
9.5 Testing Problems	115
10. Discussion	116
10.1 Analysis of the Dynamics.....	116
10.1.1 Dynamic Coupling Relative Position Predictions	116
10.2 Analysis of the Control System Performance	118
10.2.1 Successful Control Performances	118
10.3 Analysis of Pitch and Roll Effects	120
10.3.1 The Dynamics of Pitch and Roll Effects.....	120
10.3.2 Analysis of Roll Effects	122
10.3.3 Analysis of Pitch Effects	123
10.3.4 Quantification of the Pitch Effect	123
10.4 Suggestion to Solve the Pitch Effect Using the Rotational Inertial Ratio	127
11. Conclusion and Recommendations for Future Work	128
11.1 Conclusion.....	128
11.2 Recommendations For Future Work.....	129
11.2.1 Generation of a 3D Model for the System and Two Degree of Freedom Tail.....	129
11.2.2 The Use of a More Appropriate Platform	129
11.2.3 Control System Improvements and Optimization	129
12. List of References	131
Appendix A:	137
Mathematic Proofs and Equation Derivations	137
A1: State Space Steady State Analysis	137
A1.1 State Feedback Configuration Steady State Resolution Mathematics	137
A1.2: State Space Integrated State Feedback Approach Mathematics.....	137

A2 Lagrange Dynamic Mathematics.....	138
A2.1 Derivation of the Dynamic Equations of Motion.....	138
A2.2 Resolution of Dynamic Equations to Simulation Compatible Form	138
A3 Pole Placement of Loop 2.....	139
A3.1 Loop 2 Pole Placement	139
A4 Derivation of the Differential Equation of Motion for Pitch Position During Damping Action	139
Appendix B:	140
Sensor Calibration.....	140
B1 Calibration Methodology.....	140
B1.1 Accelerometer Calibration	140
B1.2 Gyroscope Calibration	140
Appendix C:	143
Auxiliary figures.....	143
C1: Decoupled Step Test Simulation Graphics	143
C2: Motor Step Test Data	144

List of figures

Figure 1: Plan of development for the project report.....	4
Figure 2: Robotic whisker sensor system capable of generating tomographic mapping of its surroundings [12].....	8
Figure 3: The Japanese bullet train design was modified based on the Kingfisher bird, the image shows how the train's front carriage clearly imitates the bird's beak aerodynamic design [13]	9
Figure 4: Festo aerial aquatic robots: Top – AirRay, Middle – AirJelly, bottom - AirPenguin [15]	10
Figure 5: Festo gripper technologies, Top – a mechanical gripper modelled on an elephant trunk and, Bottom – one modelled on a chameleon tongue [16]	11
Figure 6: Festo WaveHandler conveyor system - which transports materials in a similar fashion to ocean waves [17].....	12
Figure 7: Festo BionicOpter, the 13 degree of freedom mechanical dragonfly [18]	12
Figure 8: Left - Four frames of H platyurus correcting its attitude using its tail during free fall. Right top (E) - graph illustrating initial position of the gecko and relative tail position. Right middle (F) - Plot of angular body position against tail position during free fall. Right bottom (G) – plot of body positon for tailless geckos [19].	14
Figure 9: Graphic representation of the effects of tail-to-body length ration on the ability of members of the lizard family to achieve aerial righting [19].....	14
Figure 10: Top left - Kangaroo rat mid flight adjusting its yaw position using its tail. Top right - Kangaroo rat using its tail as third balance appendage auxiliary to its back legs. Bottom - Kangaroo rat using its tail to achieve pitch control during flight	15
Figure 11: Each frame shows a different stage of the disturbance rejection test both with an unactuated (top), and actuated (bottom) tail [1]	18
Figure 12: Left - Cheetah swinging its tail to provide reactionary torque while corning [32] Right – Dima robot design to achieve rapid, high speed cornering [27].....	19
Figure 13: TAYLRoACH - the small legged robot that swings a tail in the yaw plane to execute turns[33]	20
Figure 14: Still frames of a lizard jumping from one surface to a surface with a different normal angle and undergoing two different pitch perturbations upon launch [6].....	20
Figure 15: Tailbot robot platform using its tail to mimic the pitch correcting action found in lizards performing a jump after pitch perturbation.....	21

Figure 16: The FlipBot robot performing a barrel roll stunt with tail actuation in the roll plane.	22
Figure 17: Left - Static friction velocity characteristic, Right – Coulomb friction velocity characteristic.....	25
Figure 18: Viscous friction against speed characteristic.....	26
Figure 19: Integrated friction model	26
Figure 20: State feedback loop.....	29
Figure 21: State feedback configuration with outer loop integrator	30
Figure 22: State feedback control loop including a state observer [22][23]	33
Figure 23: The characteristic of a saturation nonlinearity [45].....	33
Figure 24: Equivalent circuit model of a DC motor [47]	34
Figure 25: Simplified torque speed characteristic for a DC motor	35
Figure 26: Diagram to illustrate the transformation of angular planes with respect to the inertial frame.....	36
Figure 27: Body and tail planar representation showing the coordinate system selected to use in the development of equations of motion using Lagrange dynamics.....	40
Figure 28: Control loop of a DC motor.....	43
Figure 29: Augmented system control loop including the pseudo inertia concept developed in section 5.5 and the body position output signal.....	44
Figure 30: Stall torque and free running speed vs gear ratio for the HP Brushed Micro-metal Gearbox Motor (motor 1).....	47
Figure 31: Stall torque and speed free running speed vs gear ratio for motor 2.....	48
Figure 32: Stall torque and speed vs gear ratio for motor 3.....	48
Figure 33: Simulink model that simulates the body and tail's dynamic response to an input voltage of 6V on each motor, using the developed equations of motion and motor torque vs speed characteristics	49
Figure 34: Simulation prediction of motor 1 – body and tail angle against time and gear ratio....	51
Figure 35: Motor 1, Left – Slice taken at 180 degrees to illustrate optimal gear ratio, Right - Tail angle versus time at optimal gear ratio.....	52
Figure 36: Motor 1 angular impulse vs gear ratio at 0.14s.....	52
Figure 37: Simulation prediction of motor 2 – body and tail angle against time and gear ratio....	53
Figure 38: Motor 2, Left – Slice taken at 180 degrees to illustrate optimal gear ratio, Right - Tail angle versus time at optimal gear ratio.....	54
Figure 39: Motor 2 angular impulse vs gear ratio at 0.14s.....	54
Figure 40: Simulation of motor 3 – body and tail angle against time and gear ratio.....	55

Figure 41: Motor 3, Left – Slice taken at 180 degrees to illustrate optimal gear ratio, Right - Tail angle versus time at optimal gear ratio.....	56
Figure 42: Motor 3 angular impulse vs gear ratio at 0.14s.....	56
Figure 43: IR Launch detection circuit diagram.....	58
Figure 44: Driver circuit block diagram – the diagram illustrates the effective function of the L293NE IC [51].....	59
Figure 45: Left - Pololu 12 CPR hall effect magnetic shaft encoder [52], Right - XBEE communication module [53].....	60
Figure 46: Hardware systems block diagram showing all the hardware electronics interfacing with the MCU	61
Figure 47: Main software loop flow diagram showing how the stunt was initiated and how control was distributed during the stunt.....	64
Figure 48: Software interrupt 1 flow diagram showing the operation of the safety timer at different times during the stunt.....	64
Figure 49: Software interrupt 2 flow diagram – showing how the launch detection software operated	65
Figure 50: Left - STM on the INEMO PCB mount designed by Callen Fisher, Right - The populated PCB board	67
Figure 51: General view of SolidWorks model of the base platform designed for the robot	68
Figure 52: Left - Front view of SolidWorks model, Right – Top view of the SolidWorks model	68
Figure 53: Perspex base plate mounted on the car with Inemo board, driver circuit and battery	69
Figure 54: Concept diagram of the tail design.....	69
Figure 55: Diagram of the test rig concept	70
Figure 56: Left - Side view of the fabricated test rig, Right – Top view of the fabricated test rig	70
Figure 57: View of the fabricated test rig from a general angle.....	71
Figure 58: Side view of the TAYObot – fully constructed and integrated system showing the circuitry and battery placement	71
Figure 59: Top view of the TAYObot – fully constructed and integrated system.....	72
Figure 60: General front view of the TAYObot – fully constructed and integrated system	72
Figure 61: Mass distribution model of the body	73
Figure 62: Tail rotational inertia model	74
Figure 63: Graph of back EMF versus motor shaft speed graph	75
Figure 64: Graph of dynamic and viscous friction torque loading on the motor shaft and an interpolated trend line.....	77

Figure 65: Motor step test at 7V.....	79
Figure 66: Left – Body position and, Right – velocity step response for a 7.25V step input.....	81
Figure 67: Left – motor shaft velocity and, Right – current step response for a 7.25V step input	81
Figure 68: Shaft torque step response for a 7.25V step input.....	82
Figure 69: Robot mounted on the test rig for step tests.....	83
Figure 70: Z-axis gyroscope step test data of the system response from the test rig step test.....	84
Figure 71: Left – X-axis, and, Right – Y-axis gyroscope step test data of the system response from the test rig step test	84
Figure 72: Body yaw position response from the step test conducted on the test rig	85
Figure 73: Motor velocity (rad/s) data from the step test conducted on the test rig.....	85
Figure 74: Step test body position response, the data points show the time taken to rotate through 180 degrees	86
Figure 75: Simulink script used to perform a step test on the decoupled system	91
Figure 76: Integrated state feedback configuration for loop 1	92
Figure 77: Body position response in the isolated control loop one	93
Figure 78: Torque input commanded by the controller in loop one.....	94
Figure 79: Control loop two Simulink model with setpoint from loop one and output isolation (not connected to loop one's input)	96
Figure 80: Loop two control output (green) against loop one control input (blue) and time	96
Figure 81: Loop two control input (voltage) against time.....	97
Figure 82: Combined loop system output of body position against time.....	98
Figure 83: Discretized control system	100
Figure 84: Body position response against time for the final discrete control system.....	100
Figure 85: Torque input (blue) and output (green) responses against time for the final discrete control system.....	101
Figure 86: Control input (voltage) against time for the final discrete control system.....	101
Figure 87: Body position response settling time and overshoot percentage illustration	102
Figure 88: Robustness analysis using matrix element variation for the discrete control system.	103
Figure 89: Body position response to an input setpoint of 360 degrees for three controller tests on the test rig	105
Figure 90: Body angular velocity response to an input setpoint of 360 degrees for control tests on the test rig	105
Figure 91: Modified ramp configuration	106

Figure 92: Successful 360 degree yaw stunt jump sequence, taking a freeze frame at the launch, 90 degrees, 180 degrees, 270 degrees and 360 degrees	110
Figure 93: Body position against time for 7 arbitrary actuated jump tests, the launch occurs at 0.25 seconds and the landing occurs at approximately 0.6 seconds.	111
Figure 94: Body position against time for 3 landed jumps, the launch occurs at 0.25 seconds and the landing occurs at approximately 0.6 seconds.	112
Figure 95: Body velocity against time for 7 arbitrary actuated jumps, the launch occurs at 0.25 seconds and the landing occurs at approximately 0.6 seconds.	112
Figure 96: Body velocity against time for 3 successful jumps, the launch occurs at 0.25 seconds and the landing occurs at approximately 0.6 seconds.....	113
Figure 97: Motor velocity against time for 7 arbitrary actuated jump tests, the launch occurs at 0.25 seconds and the landing occurs at approximately 0.6 seconds.	113
Figure 98: Motor velocity against time for 3 successful jumps, the launch occurs at 0.25 seconds and the landing occurs at approximately 0.6 seconds.....	114
Figure 99: Test rig 360 degree stunt inspection of tail position, which is approximately 525 degrees	117
Figure 100: 360 Degree jump stunt inspection of tail position, which is approximately 555 degrees	117
Figure 101: Simulation prediction of the control system (black line) plotted with successful jumps (colourful lines).....	118
Figure 102: Body yaw velocity directly before launching	120
Figure 103: The dynamic analysis of pitch and roll effects that acted on the body, Left – is a top view of the robot and the two forces F_p and F_R are shown as well as the centre of mass, Middle – A side view of the robot is shown and the force F_p , Right – The front view of the robot is shown and the force F_R	121
Figure 104: Roll velocity against time for a successful jump, presented to show the mitigation of roll effects	122
Figure 105: Roll position against time for a successful jump, presented to show the mitigation of roll effects	122
Figure 106: Illustration of the effect of linear acceleration on pitch during launch	123
Figure 107: Illustration of the pitch force action on the body during the last quarter of the stunt	124
Figure 108: Top – Predicted pitch position and, Bottom – measured pitch position for the final portion of the stunt for a pitch effected crash jump	125

Figure 109: Left – Predicted and, Right – measured pitch rotational velocity for the final portion of the stunt for a pitch effect crash jump.....	126
Figure 110: Pitch torque due to angular velocity of the tail mass for the final portion of the stunt	126
Figure 111: Gyroscope calibration graphs mapping gyroscope readings to temperature for all 3 axes	141
Figure 112: Step test graphics of each state for the modified decoupled system	144

List of Tables

Table 1: A chronologic description of the plan of development of the project	3
Table 2: A link between natural elements and their synthetic counterparts for biomimetic design [4]	7
Table 3: Comparison of artificial and natural force/moment mechanisms and their ability to alter centre of mass [1]	17
Table 4: State variables of the system	45
Table 5: Pololu Metal Gearbox motors [49].....	46
Table 6: Simulation results for motor 1.....	52
Table 7: Simulation results for motor 2.....	54
Table 8: Simulation results for motor 3.....	56
Table 9: Selected motor specification	57
Table 10: L293NE driver circuit specifications [51].....	59
Table 11: Hardware requirements of the microcontroller.....	66
Table 12: Features of the STM32F103RE on the Inemo package.....	66
Table 13: Mass breakdown of the of the fully constructed and integrated system.....	73
Table 14: System Parameters.....	77
Table 15: Final System parameters.....	80
Table 16: Summary of important simulation data	82
Table 17: Loop 1 state feedback and integrator gains.....	93
Table 18: Loop two state feedback and integrator gains	95
Table 19: Motor data pertinent to the final portion of the stunt	124
Table 20: Gyroscope temperature dependency functions	141
Table 21: Motor test data for various input terminal voltages, motor shaft speed, output shaft speed terminal voltage and armature current are all shown	144

Glossary

PWM	– Pulse width modulation
IC	– Integrated circuit
EMF	– Electromotive force
RC	– Remote car
IR	– Infrared
GPIO	– General Purpose Input Output
UART	– Universal Asynchronous Receiver Transmitter
MCU	– Microcontroller Unit
CPR	– Counts Per Revolution
EMF	– Electromotive Force
SISO	– Single Input Single Output
MIMO	– Multi Input Multi Output
Conjugate pair	– A pair of imaginary numbers having a real product
Eigen value	– characteristic scalar value of a vector
Lipo cell	– Lithium Polymer cell
C.O.M	– Centre Of Mass
PD	– Proportional Derivative
R&D	– Research and Development
Tx	– Transmitter
Rx	– Receiver

1. Introduction

1.1 Background to the Research

In recent years a great interest has been shown in the field of biomimetics and bio-inspired design, which is the process of adapting principles and phenomena found in nature to human designs. This practise has been found to be particularly beneficial in robotics.

In the course of developing bio-inspired robotics, advantages have been associated with the concept of the tail for applications in technology. Studies have shown that tails can have diverse functionality in robotic systems, and one such example is its ability to aid aerial orientation and attitude control of the body to which it is appended [1]. This project will seek to analyse tails and robotic implementations of thereof as a mechanism to achieve aerial orientation and attitude control. If successful, the research may provide insight into how improvements can be made to various legged and terrestrial robots.

1.2 Research Objectives

1.2.1 Aims of the Project

The fundamental goal of this project is to model, design and build a bio-inspired robot capable of performing a 360-degree yaw stunt while airborne using a tail. The aims can be categorized chronologically as follows:

- i. Ascertain whether tails are the most effective and efficient tool available to robots for the purpose of attitude control during flight or gliding.
- ii. Develop a dynamic model of a robot concept fitted with a tail appendage.
- iii. Generate simulations of the model to verify the feasibility of the stunt, select optimal components and obtain theoretical data to compare to empirical data.
- iv. Design a control system to command the actuators in such a way that the stunt can be achieved in the requisite time.
- v. Design the mechanical structure of the robot to emulate the source of the bio-inspiration's features as closely as is functionally beneficial, as well as design all the hardware electronics and software for the system.

- vi. Physically build and test the robot.
- vii. Quantitatively analyse all aspects of the performance of the system and contrast it to simulations.

1.2.2 The Significance of the Study

The significance of this research may not be obvious when considered at face value, as performing a stunt with a robot seems frivolous. This is not the case at all, as the principles explored and developed in this study have the potential to benefit society in a meaningful way. Performing the stunt required deep understanding of the dynamics involved with the platform and tail combination, as well as precise control system performance. The insight gained through the study has added to the current knowledge base on aerial orientation and attitude control of flightless robots while airborne. Any additions to this body of knowledge may lead to improvements in search and rescue and space exploration robots, which are both required to circumnavigate uncertain terrain. The benefits of improving these technologies are obvious, and subsequently validate the significance of this project.

1.3 Scope and Limitations

This project will be limited to developing dynamic models in two dimensions, and as such, a one degree of freedom tail will be used in the system. The robot was made to hit a ramp, and during the flight following this, it attempted to perform a 360-degree yaw stunt. The scope included modelling and designing for the performance of the robot while airborne, it did not include the same processes for the operation of the robot on the ground – the events after a successful landing were not considered in this study. The project was also limited to investigate designs to be implemented on the RC car that was provided for this project by the University of Cape Town.

The two dimensional modelling entailed making the simplifying assumption that the tail joint would be fitted exactly at the robot's centre of mass. This was not possible, given the electronics that were required on-board the robot. As a consequence, the robot's three dimensional motion was not fully captured by the model; analysis of the resulting three dimensional motion was however broken into two dimensional components and evaluated. The effects of drag were also considered to fall outside the scope of this study, since their influence on a system with the experienced levels of velocity were negligible when compared to the forces that acted on the system.

1.4 Plan of Development

This report was broken down into seven distinct sections. Each section builds on the content presented before it, and prepares the reader for sections following it. The table below describes the plan of development chronologically, and Figure 1 presents it graphically.

Table 1: A chronologic description of the plan of development of the project

Chapter	Description	Content
2 - 4	Background research	A literature review was conducted to attain knowledge of the state of the art of bio-inspired robotics and current aerial orientation methods, and is presented in chapter 2. This is followed by a chapter describing the theoretical tools which were utilized in the project. Chapter 4 describes the methodology formulated and followed in conducting the project.
5	Modelling & simulation	In chapter 5 the report describes the dynamic modelling performed. The way in which these equations were used to simulate the system is discussed, and the simulation data itself is then shown. Deductions from the simulations are presented at the end of the chapter.
6	Design	Chapter 6 reports on all forms of design undertaken in the project, including electronic hardware, mechanical and software design.
7-8	Control system design	The report then describes the process followed in order to verify the validity of the models and theory developed in the project. Chapter 8 discusses all decisions made surrounding control system design and the method followed to design the final controller. It also presents simulations of the controller performance.
9 - 11	Testing, results & Interpretation	In Chapter 9 the report describes the testing environment, procedure and results obtained. This follows on to chapter 10, in which these results are critically evaluated. The project is summarized and concluded in chapter 11.
A-C	Appendices	The appendices report on auxiliary information used in the report, as well as show the mathematical steps taken to obtain various models and designs.

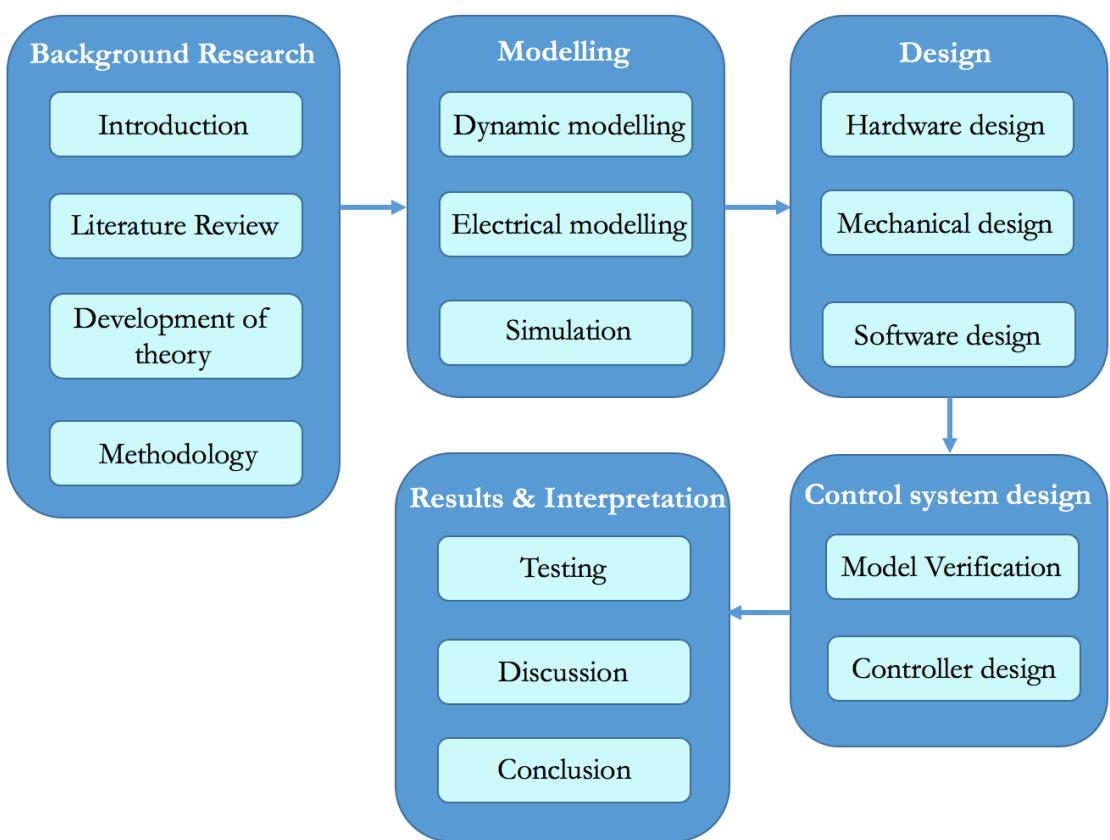


Figure 1: Plan of development for the project report

2. Literature Review

2.1 Introduction

Nature has conventionally been perceived to be in stark contrast to technology as the fields of study have previously been mutually exclusive, but that sentiment is changing. Nature has been advanced and developed over centuries by its own form of R&D, evolution [2]. Evolution has ensured that animals and plant life improve functionally each generation, albeit by minuscule increments (relative to average human lifespans). This process has been in place for centuries, and subsequently in the modern era, natural phenomena have become highly efficient [3]. Scientist and engineers have now started studying such phenomena with the intention of implementing applicable aspects in robotics and automation. The practice is still in its infancy, but early successes have generated increased interest as well as optimism that new technology can be derived from natural models [2][4].

Briggs *et al.* are quoted as stating, “The field of bio-inspired robotics continually mines nature’s abundant store of solutions to engineering challenges” [1]. This succinctly captures the promise and excitement currently surrounding the field of biomimetic robotics and bio-inspired robots.

This review of the literature documenting biomimetic robotics begins with a history of, and general discussion of the state of the art of biomimetics. It then narrows its focus to literature specifically relevant to this project, namely, attitude control and aerial orientation in flightless animals and the principles relevant to robotics, the use of tails in nature for this function, and tails currently used on robotic platforms. The review concludes by considering literature pertinent to this project that falls outside the scope of biomimetics.

2.2 Biomimetics

2.2.1 History

Nature has experimented with principles of material science, chemistry, control, sensors, physics and mechanics for 3.8 billion years [4]. Nature's inventions have, quite literally, survived a quality control test that has spanned millennia. Humans have only been involved in the process of design for what is effectively an evolutionary microsecond [5]. As a consequence, in many areas, natural processes are far superior to current technology in terms of efficiency, performance, robust operation and waste management [6]. It was thus natural for humans to look to nature for guidance and inspiration, which resulted in the field of biomimetics.

The inception of the study of biomimetics could be argued to have begun as early as Leonardo da Vinci's conceptualization of bird-like flying machines [7], but the term "biomimetics" was first used by Otto Schmidt in 1969, in his paper on useful and interesting biomimetic transforms [8]. The field gained support thorough the latter part of the 20th century, and was solidified by the publication of Janine Benyus' book, "Biomimicry: Innovation inspired by design" [9].

Biomimicry offers solutions in almost every area of human design, with applications in medicine, engineering, science and industry [5]. Biomimetic design is highly multidisciplinary, requiring deep understanding of actuators, control systems, sensors, anatomy and system of systems design [2]. It presents a complex yet rewarding challenge to designers.

2.2.2 Methodology

In order to effectively emulate natural systems, elements of the synthetic system must be linked to their natural counter parts. A common interpretation of these links is presented in *Table 2*.

In order to achieve high quality performance from biomimetic designs, the synthesized element should closely mimic the performance of its natural counterpart. Caution is recommended in the pursuit of perfect emulation, as this may not always be the most efficient artificial design realization. Robots do not have to perform all the functions that their living counterparts do, such as resist infection, rest or seek companionship [10].

Table 2: A link between natural elements and their synthetic counterparts for biomimetic design [4]

Natural Element	Corresponding synthetic element	Description
Body	System	System of systems, involving a variety of materials, structures, actuators, sensors and controllers.
Skeleton/bones	Structure and support	Materials for support structures should have similar operating points to their natural counterparts – stress levels, elasticity etc.
Brain	Computer/Processor	Modern processors are capable of computing and executing decisions much like the brain.
Intelligence	Artificial Intelligence	Artificial intelligence provides promise that synthetic systems will be able to learn and adapt like intelligent beings.
Senses	Sensors	Sensor technology has advanced to the point where almost all senses available to biological creatures are available to synthetic emulations.
Muscles	Actuators	Actuators of all kinds can perform the function of muscles for synthetic systems
Fat energy storage	Batteries	Various energy storage methods are available to biomimetic design, the most prevalent of which is batteries.

2.2.3 Ecological Implications

One of the most widely contested topics of the 21st century is sustainability, and the current level of it associated with economic and technological growth around the world [11]. The need has arisen to reduce the human footprint on the world, and biomimetic design may again be the solution. Not only are natural systems incredibly efficient, but they tend to positively effect the environment [5]. Design features of systems that operate with closed loop resource and reusable waste processing could be the inspiration to alter current harmful industrial systems [7].

2.2.4 Current State of the Art

Although the field of biomimetics can still be considered to be in its infancy, enormous advances have been achieved in a very short time. The current state of the art can best be described by interesting examples of how the field has improved technology.

(i) Actuator improvements through biomimetics

Mechanical actuators have improved recently through unconventional biological studies. The sensitivity fern (*Onoclea Sensibilis*) bends its leaves when touched, while sunflowers are able to track the movement of the sun, maximizing solar exposure. Studying the mechanisms used to provide such local control may yield new types of motors. Studies on caterpillars have resulted in the creation of linear actuators called ‘inchworm motors’; these motors demonstrate high levels of performance and have zero backlash effects [4].

Controlled adhesion is also an area of great interest in the field, and researchers are attempting to imitate the way in which beetles can achieve adjustable wet adhesion, and the way in which gecko’s exhibit this ability with dry surfaces [4].

Advances in polymer science have led to the development of artificial muscles. Certain polymers’ shape and size can change through electrical excitation; such a material is called an Electro-Activated Polymer (EAP). Artificial muscles could be implemented as mechanical actuators and it is likely that they could closely emulate biological actuation [2][4].

(ii) Sensors

Sensors have been widely modelled on the sensory systems of living organisms. An interesting new development in the field of instrumentation and sensing is the exploration of whiskers as a means of collision avoidance and environment mapping. Whiskers form a crucial part of the sensory system of rats, seals, and cats. A robotic whisker sensor has been developed that can create tomographic maps by means of analyzing the airflow through the system [12]. This system is shown in Figure 2.



Figure 2: Robotic whisker sensor system capable of generating tomographic mapping of its surroundings [12]

(iii) Large scale biomimetic systems

Large scale industrial biomimetic systems are being implemented world-wide. A prime example is found in the Japanese bullet train, called “Shinkansen”. The original design did not feature any biomimetic elements, and subsequently a large problem was experienced with the noise generated when exiting tunnels. The design was then modified based on the Kingfisher bird, using its optimal aerodynamic facial and beak structure [13]. The modified train structure is shown alongside a Kingfisher for comparison in Figure 3.



Figure 3: The Japanese bullet train design was modified based on the Kingfisher bird, the image shows how the train's front carriage clearly imitates the bird's beak aerodynamic design [13]

2.2.5 Future Attributes to Attain

Although great strides have been made in the field, biomimetics still has a lot to offer that is currently beyond our current capabilities. Natural attributes such as regeneration, real time camouflage adaption, chemical generation and storage of energy are highly desirable features that future research will attempt to synthesize [2].

2.2.6 Industrial Applications

No company has better illustrated the power of biomimetic design than the automation company, Festo. They have been wholly dedicated to the field and have spent billions of dollars on biomimetic research and development [14]. Their biomimetic division has produced some of the most brilliant robots seen to date, all with the hope of improving the efficiency and performance of various automation processes.

(i) Festo's aquatic to aerial adaptations

In the period between 2006 and 2009 Festo researched and built robots based on aquatic animals, but modified the designs to fly. The outcome of the project included what they call, the AirPenguin, AirJelly and the Air_ray. These robotic platforms are incredibly original and are shown in Figure 4. The research is said to have provided insight into precise, rapid control of flexible tripods – as was required by the 3D Fin Ray structure [15].

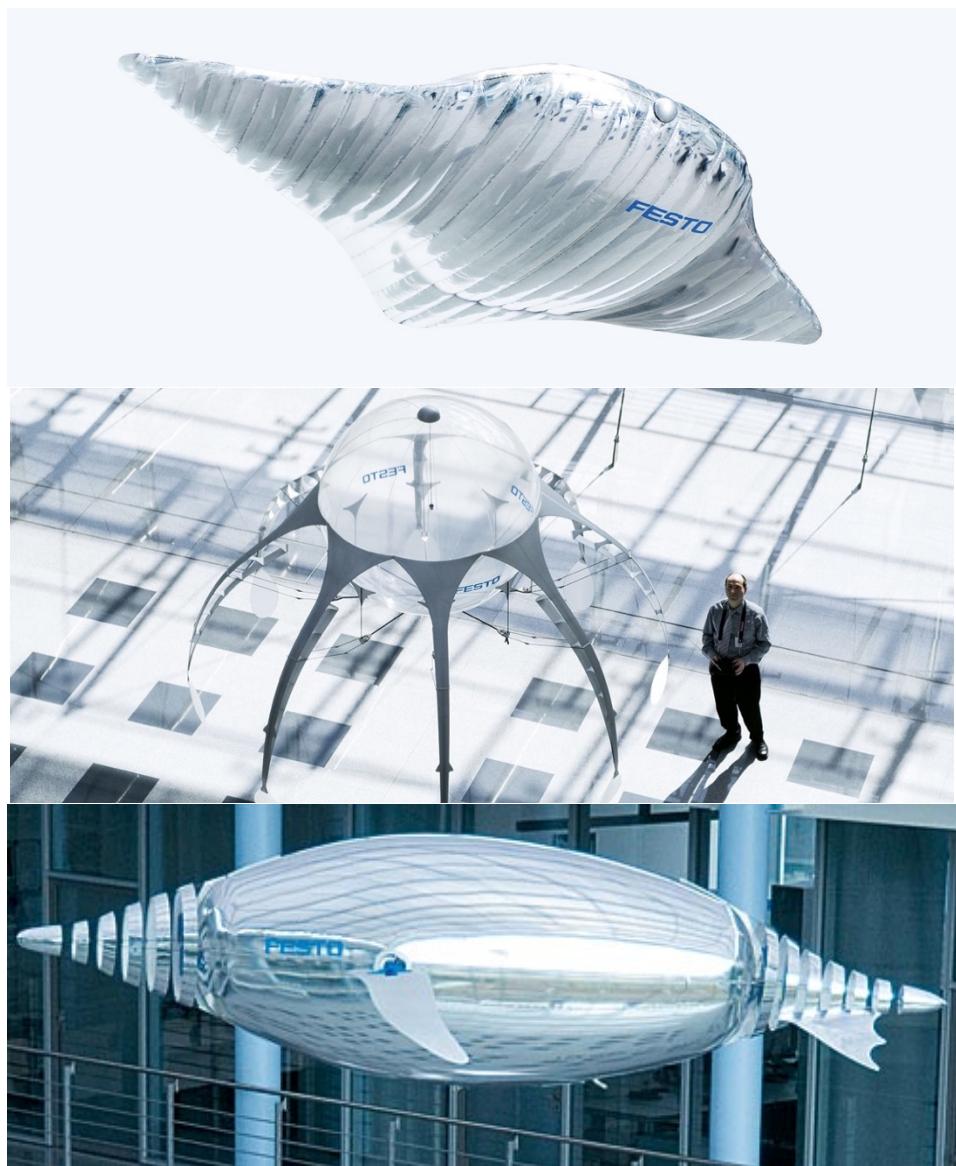


Figure 4: Festo aerial aquatic robots: Top – AirRay, Middle – AirJelly, bottom - AirPenguin [15]

(ii) Festo gripper mechanisms

Festo have also developed various gripper technologies based on different animals. The diversity in the technology is highlighted by the two grippers in Figure 5, one based on a chameleon tongue, and the other on an elephant trunk [16].

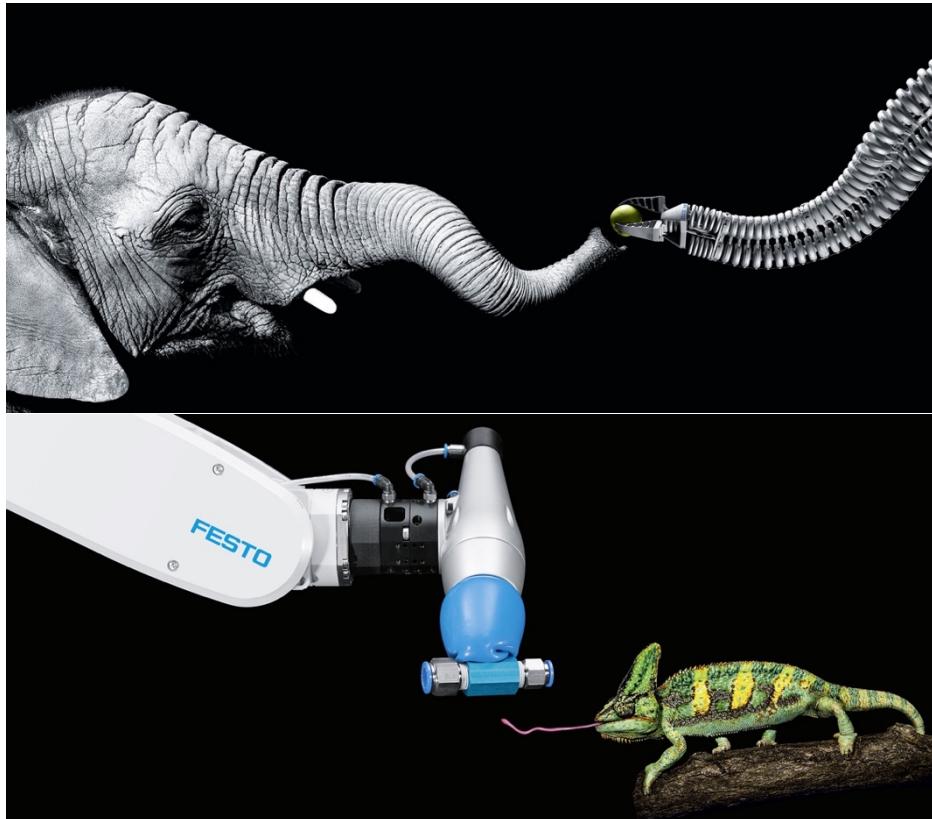


Figure 5: Festo gripper technologies, Top – a mechanical gripper modelled on an elephant trunk and, Bottom – one modelled on a chameleon tongue [16]

(ii) Innovation through observing natural physics

Festo's research is not limited to biology, they have also looked to physical phenomena for inspiration. They have created a new pneumatic conveyor with sorting capabilities, the "WaveHandler", which is depicted in Figure 6. The inspiration was said to have come from the manner in which ocean waves can transport a surfer riding the wave [17].

(iii) Festo's BionicOpter

One of Festo's self-proclaimed greatest achievements is the BionicOpter, a mechanical dragonfly. The robot can perform incredible aerial manoeuvres, operating with 13 degrees of freedom [18], an image of this robot is provided in Figure 7.

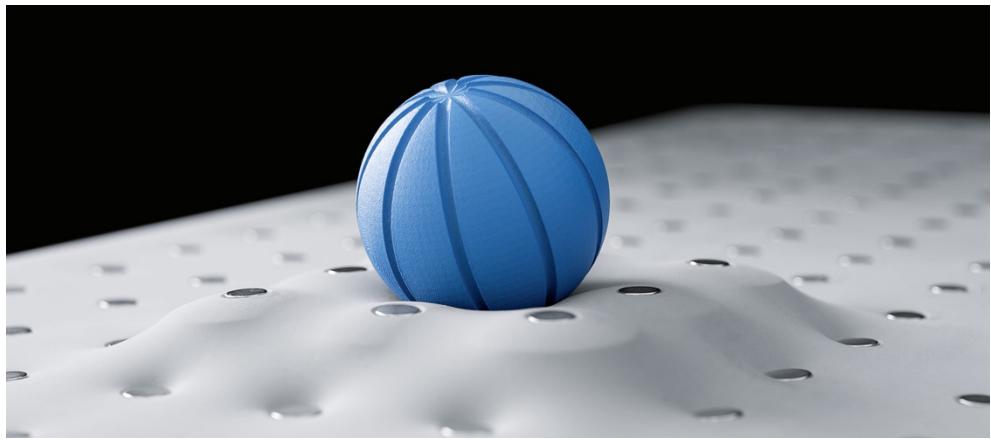


Figure 6: Festo *WaveHandler* conveyor system - which transports materials in a similar fashion to ocean waves

[17]



Figure 7: Festo *BionicOpter*, the 13 degree of freedom mechanical dragonfly [18]

2.3 Attitude Control and Aerial Orientation in Flightless Animals

There are many scenarios in which flightless animals become airborne for brief periods of time, either intentionally through a jumping or hopping action, or unintentionally as a result of falling. In either case, animals prone to such circumstances have adapted to handle them in ways that mitigate damage to the subject [19]. The mechanisms through which flightless animals control their attitude during flight varies between species, and a distinction can be made between mammalian aerial orientation methods and those attributed to reptilian species [19].

2.3.1 Mammalian Aerial Righting

The classic example is presented by the house cat. The question of how cats always seem to land on their feet despite their initial body orientation at the beginning of a fall has long been pondered by humans [20]. Research has showed, as is the case with most mammals, that cats bend and twist their torso in order to achieve the required re-orientation to land in the correct position. Studies in which tail-less cats were observed in free fall showed that cats depend very little on their tails in this process [19].

2.3.2 Reptilian Aerial Righting

Reptiles have been observed to achieve aerial righting differently, with a far greater dependency on their tails. A study was conducted by Jusufi *et al.* which considered the righting performance of the house gecko, *H. platyurus*. An experiment was conducted in which the gecko was dropped in a dorsoventrally downward posture (upside down), and its ability to correct its posture was analysed given a free fall time of 45 ms. The test was carried out 16 times, and the subject was able to right itself in all cases. Photographs of the test (shown in Figure 8) clearly show that the torso of the subject undergoes very little flexion during the fall, and the data plotted in the graph in the right middle position of the figure shows the correlation between tail position and body orientation. Geckos without tails were not able to achieve aerial righting, and only achieved in the best case half the requisite rotation [19].

The study went on to analyse the effects of an altered tail-to-body length ratio. Members of the lizard family, the Green Anole, *A. carolinensis*, have a similar morphology to *H. platyurus*, except that they have a larger tail-to-body length ratio. They were used as models to compare to the performance of the gecko. Figure 9 was generated by extrapolating the data from the previous test to simulate the performance of *A. carolinensis* in the same aerial righting task. The graphic shows improved ability to perform the correction with a greater tail-to-body length ratio [19].

Despite the high angular velocities reached by appendages in correcting action, wind tunnel testing has confirmed that aerodynamic drag effects were negligible in the experiments discussed above [19].

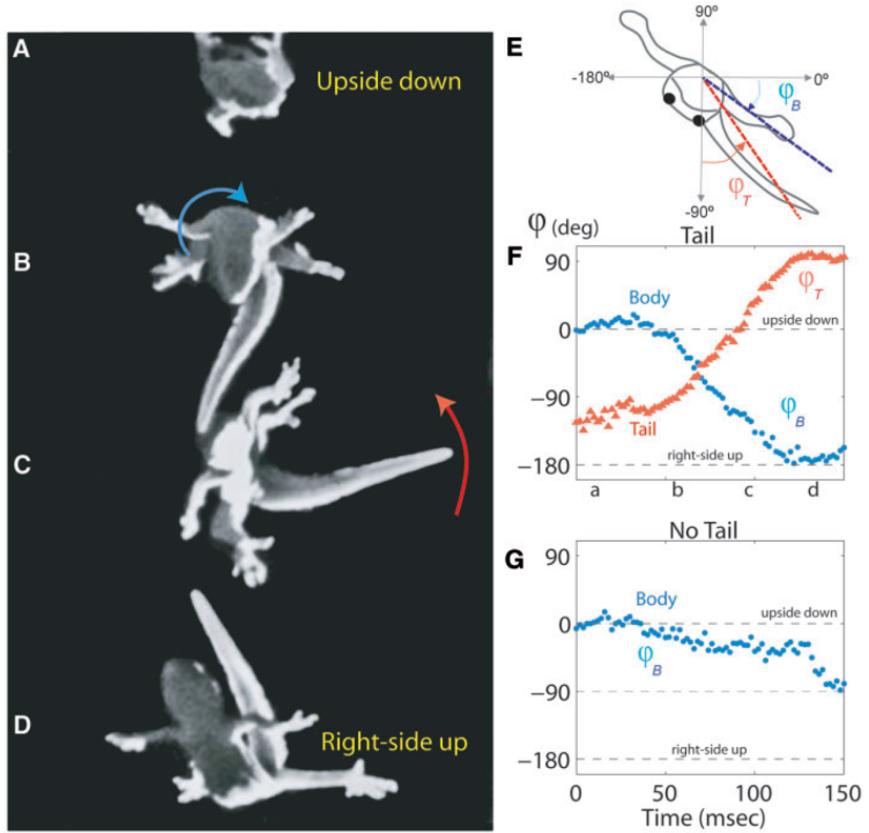


Figure 8: Left - Four frames of *H. platyurus* correcting its attitude using its tail during free fall. Right top (E) - graph illustrating initial position of the gecko and relative tail position. Right middle (F) - Plot of angular body position against tail position during free fall. Right bottom (G) – plot of body positon for tailless geckos [19].

Graph of righting roll angle achieved against tail-to-body length ratio

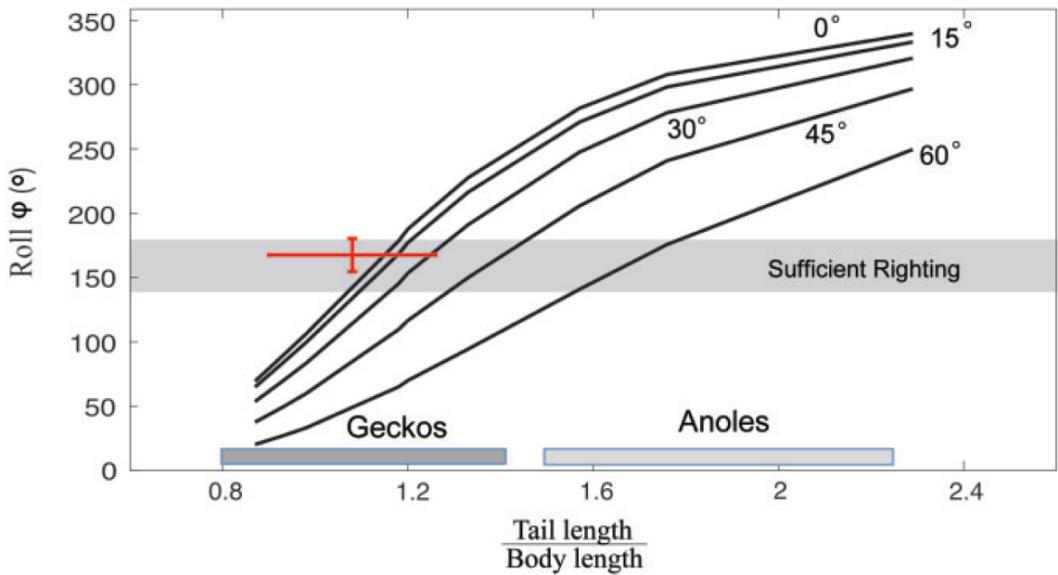


Figure 9: Graphic representation of the effects of tail-to-body length ration on the ability of members of the lizard family to achieve aerial righting [19]

2.3.3 The Potentially Hybrid Case of the Kangaroo Rat

The locomotion of Kangaroo rats was studied in detail by Bartholomew and Caswell in [21], and their research returned that these creatures are highly dependent on their tails (which are long and have large moments of inertia) in most forms of locomotion (bipedal, quadrupedal and hopping). Kangaroo rats are also able to alter their body position during flight using their tails; cases have been documented where the Kangaroo rat senses danger and uses its tail to perform a 180-degree yaw turn while mid flight [21], this scenario is illustrated in Figure 10. It is speculated that Kangaroo rats may use a combination of tail action and torso bending to achieve such aerial acrobatics, and if so they could be evolutionary leaders in aerial orientation methodology [19]. Research is yet to be conducted that confirms or denies this.

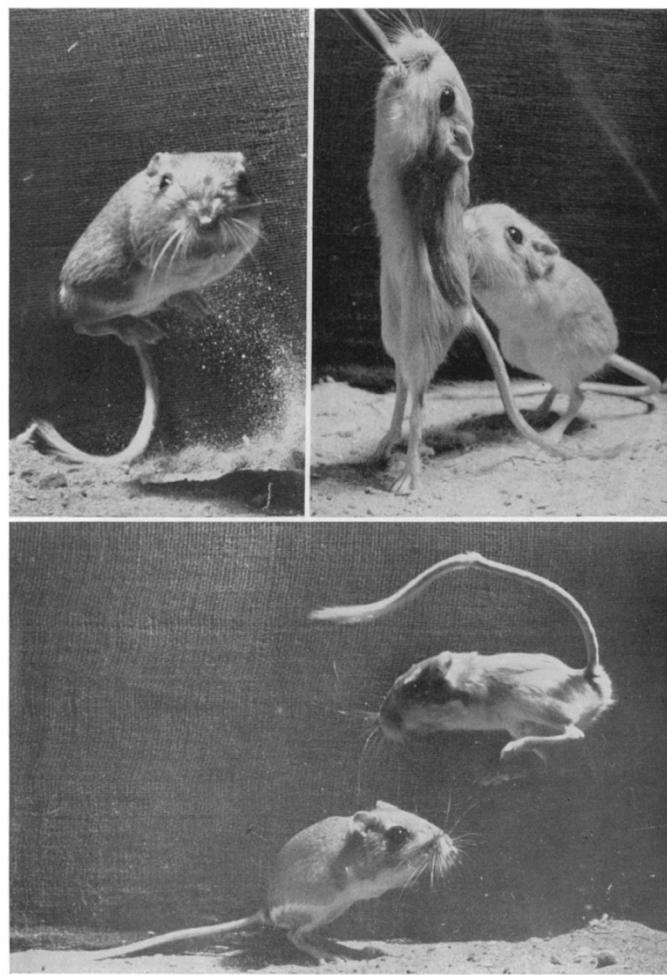


Figure 10: Top left - Kangaroo rat mid flight adjusting its yaw position using its tail. Top right - Kangaroo rat using its tail as third balance appendage auxiliary to its back legs. Bottom - Kangaroo rat using its tail to achieve pitch control during flight

2.3.4 Application for Robotics

The methods of attitude control and correction used by mammals and reptiles have been presented in this section. From this information, it is clear that it would be very difficult to design and fabricate robots capable of torso flexion, thus the orientation method adopted by mammals is not likely to be beneficial in robotics for some time [22]. The method used by reptiles presents a far more attractive option for robotics, as it would be fairly easy to fit an appendage onto a robotic platform and actuate it [6]. Consequently the study will narrow its focus to tails in nature and robotics.

2.4 Tails in Nature and Robotics

2.4.1 Tails in Nature

Although humans may not need tails in the modern era, having lost any remnants of tail morphology through evolution, many animals are still highly dependent on tails for a diverse set of functions [1]. Tails have been beneficial to balance, running, swimming, hopping, climbing, flight control and even thermoregulation for various creatures. Tails have been particularly useful to animals during locomotion, from flight to bipedal or quadrupedal motion [1].

2.4.2 Tails in Robotics

As a consequence of the utility seen in tails in nature, engineers and scientists have seen merit in attempting to emulate aspects of tail functionality on robotic platforms. Initially tails were first introduced in robotics for aesthetic purposes, but have recently become far more substantial functional units in designs [1].

In July 2015 the Robotics: Science and Systems (RSS) society held a conference in which experts in the field met to discuss the role played by tails in robotics [23]. It was shown that robotic tails can improve agility, enabling robots to navigate rough terrain such as rubble [24-26], they can aid robots in making sharp turns at high speeds [27-29] and tails can provide stabilization during dynamic disturbance rejection [1][30-31]. Tails have been implemented on platforms with diverse morphologies, ranging from flying, to legged and wheeled platforms. The bio-inspiration for such robots stems from dinosaurs, cheetahs, moths, kangaroos and lizards [23].

Tails clearly offer a wide range of functionality, but this study will focus on the inertial qualities of tails that provide the robotic platforms with a means to re-orientate itself or control its attitude.

2.4.3 Tail Design to Improve the Manoeuvrability of the MIT Cheetah Robot

The inspiration to implement a tail on the MIT cheetah robot came from video footage of how the cheetah swings its tail during locomotion, as well as biological studies of how other quadrupeds perform similar actions. It was hypothesized that cheetahs swing their tails while their hind legs are off the ground with the intention of correcting the orientation of their posteriors [1].

The aim of tail implementation was to improve dynamic stability and manoeuvrability of the platform, but the authors were cautious not just to copy nature without factual validation for the merits of the practise. A comparison of current engineering methods for providing similar functionality was first conducted [1], the results of which are shown in *Table 3*.

Table 3: Comparison of artificial and natural force/moment mechanisms and their ability to alter centre of mass [1]

	Mechanism	Force	Torque	COM
1	Tail	Yes	Yes	Yes
2	Reaction Wheel	No	Yes	No
3	Control moment gyroscope	No	Yes	No
4	Reaction mass	Yes	No	Yes
5	Thruster / Gas jet	Yes	No	No
6	Propeller / Rotor	Yes	Yes	No
7	Turbojet / Turbofan	Yes	Yes	No

Items 3, and 5 through 7 were not considered on the grounds of the complexity associated with their implementation. The reaction mass was also not considered as a consequence of its poor spatial efficiency. The result was a comparison between the reaction wheel and the tail. It was found that although the reaction wheel could be operated continuously, the tail has higher rotational inertial and thus can apply greater angular impulses. The study into using a tail for the premise outline previously was vilified by these deductions [1].

Figure 11 depicts the test that was carried out on the cheetah robot. A clay wrecking ball was made to impact the robots posterior with a force capable of knocking the robot over, and the tail was used to counter this disturbance. A state feedback controller was designed and used to perform the corrective action using the tail. The figure shows that in the actuated case (bottom set of

frames) the tail successfully stabilized the MIT Cheetah platform. The experiment illustrates the potential of the tail to improve the performances of legged robots [1].

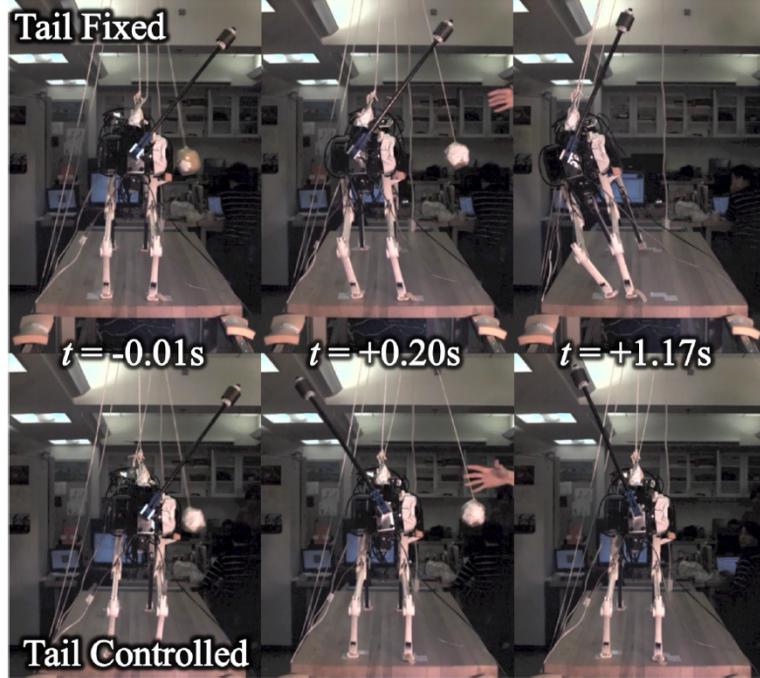


Figure 11: Each frame shows a different stage of the disturbance rejection test both with an unactuated (top), and actuated (bottom) tail [1]

2.4.4 Dima Robot

The cheetah was also the inspiration for the research conducted by Patel and Braae [27] into rapid turning at high speed for robots. Observations were made that the cheetah swings its tail in the roll axis while changing direction in pursuit of its prey, as shown by Figure 12. It was postulated that this action provides a torque that counters the centrifugal force experienced by the turning cheetah that would otherwise cause it to topple at such speeds [27].

The investigation aimed to recreate this counter torque action using a tail on a terrestrial robotic platform. It was hoped that the platform would be able to achieve rapid manoeuvrability as a result of the tail addition [27].

The Dima robot was constructed for such tests, as displayed in the right of Figure 12. The robot's performance during cornering was evaluated by imposing a 30 degree steering angle for 0.5 seconds at an initial steady linear velocity. Dima's ability to make the turn was then analysed with and without tail actuation [27]. The experiments show very positive results, in that without tail actuation the robot would roll and not complete the turn. For the actuated case the platform was

stabilized through the turn, and the robot was able to handle approximately 40% more lateral acceleration due to tail rotation [27].

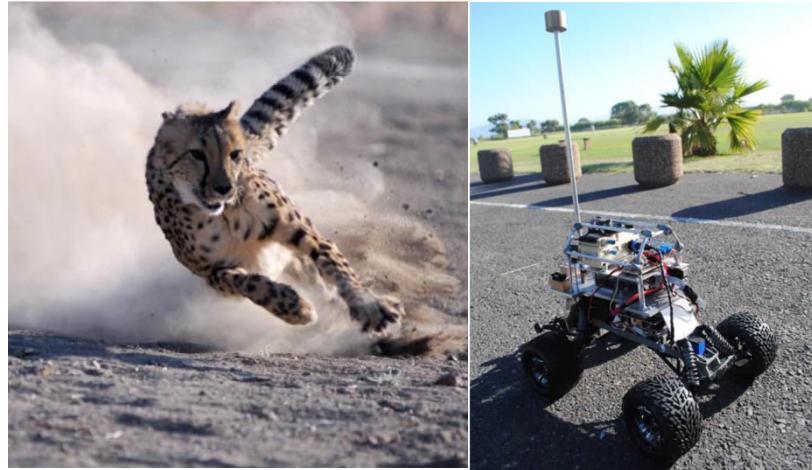


Figure 12: Left - Cheetah swinging its tail to provide reactionary torque while cornering [32] Right – Dima robot design to achieve rapid, high speed cornering [27]

2.4.5 TAYLRoACH

An investigation was also conducted by Kohut *et al.* into the effectiveness of a tail to provide an angular impulse to effect a turn for a legged robot. The TAYLRoACH (Tail Actuated Yaw Locomotion RoACH), depicted in Figure 13, is a small legged robot fitted with a tail that rotates in the yaw plane. This is another instance of the inertial qualities of a tail being used to create an angular impulse on a principal body [33].

An experiment was conducted in which the robot was made to run for two seconds with only the velocity of the legs controlled (without any position or heading control), thereafter the tail is actuated to effect a 90 degree turn [33].

Results showed that tail actuation proved to be a very effective means to alter the platform's yaw position, and achieve yaw angular rates up to 300 degrees per second. The system was however complicated by frictional effects, leg contact and other aspects that require consideration to create a more consistent system [33].

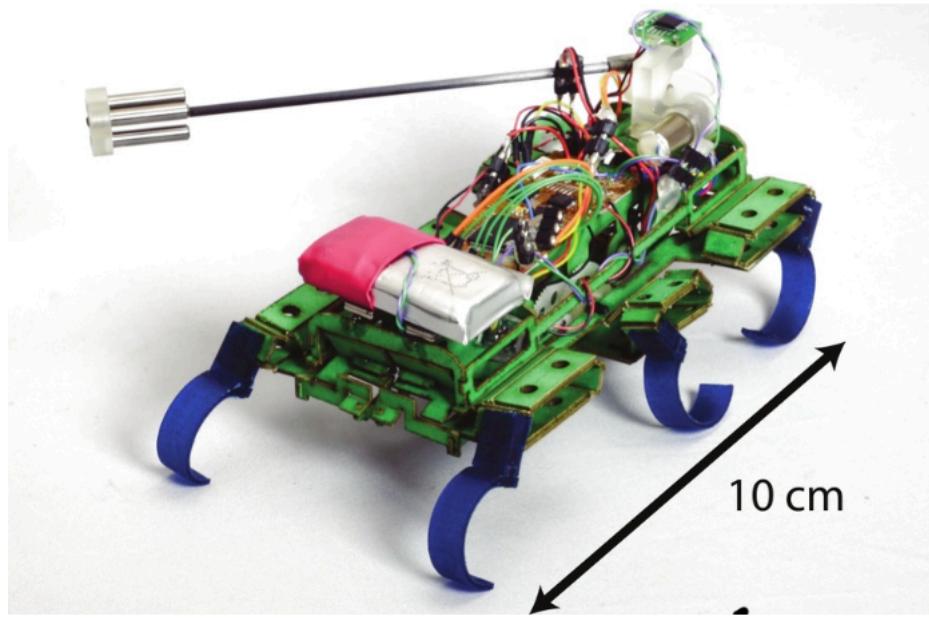


Figure 13: TAYLRoACH - the small legged robot that swings a tail in the yaw plane to execute turns[33]

2.4.6 Tailbot

A similar experiment to that conducted on the gecko's ability to aerially right themselves using their tails, discussed in section 2.3.2, was carried out on lizards by Libby *et al.* The method of tail inertia contributing an angular impulse to a principal body has been well developed in this text, and the lizard achieves attitude control in this way. In the first part of the experiment, the Red-headed Agama lizard was made to jump from one object to another, undergoing a pitch perturbation by means of variable traction on the launch object. Video footage of the lizard shows how it uses its tail to correct such a perturbation of its body pitch. Still frames of two perturbations are shown in Figure 14 [6].

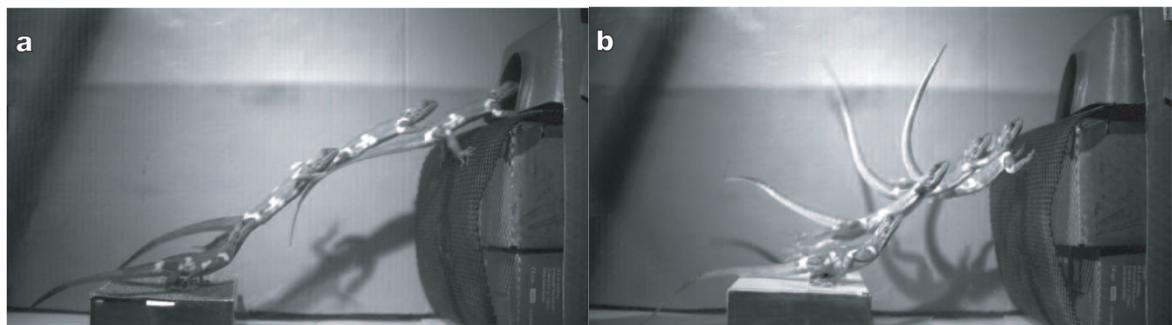


Figure 14: Still frames of a lizard jumping from one surface to a surface with a different normal angle and undergoing two different pitch perturbations upon launch [6]

An attempt was then made to synthesize this ability on a robotic platform, and to this end the Tailbot was built, emulating the size and weight of a lizard. This design is shown in Figure 15, and it is to be noted that it was the first robot with a specialized tail-like appendage to provide continuous attitude control. An experiment was set up which mimicked the perturbations imposed on the lizard, and PD feedback control was used to control the tail to mitigate the effects of the perturbation on the robot. The robot was launched with and without actuation. The actuated case is shown by the photograph sequence in the bottom of Figure 15, and the unactuated case above this [6]. It was observed that the tail provided accurate pitch control, and the movement of the robotic tail closely matched that of the lizard's tail, supporting the accuracy of the synthesis [6].

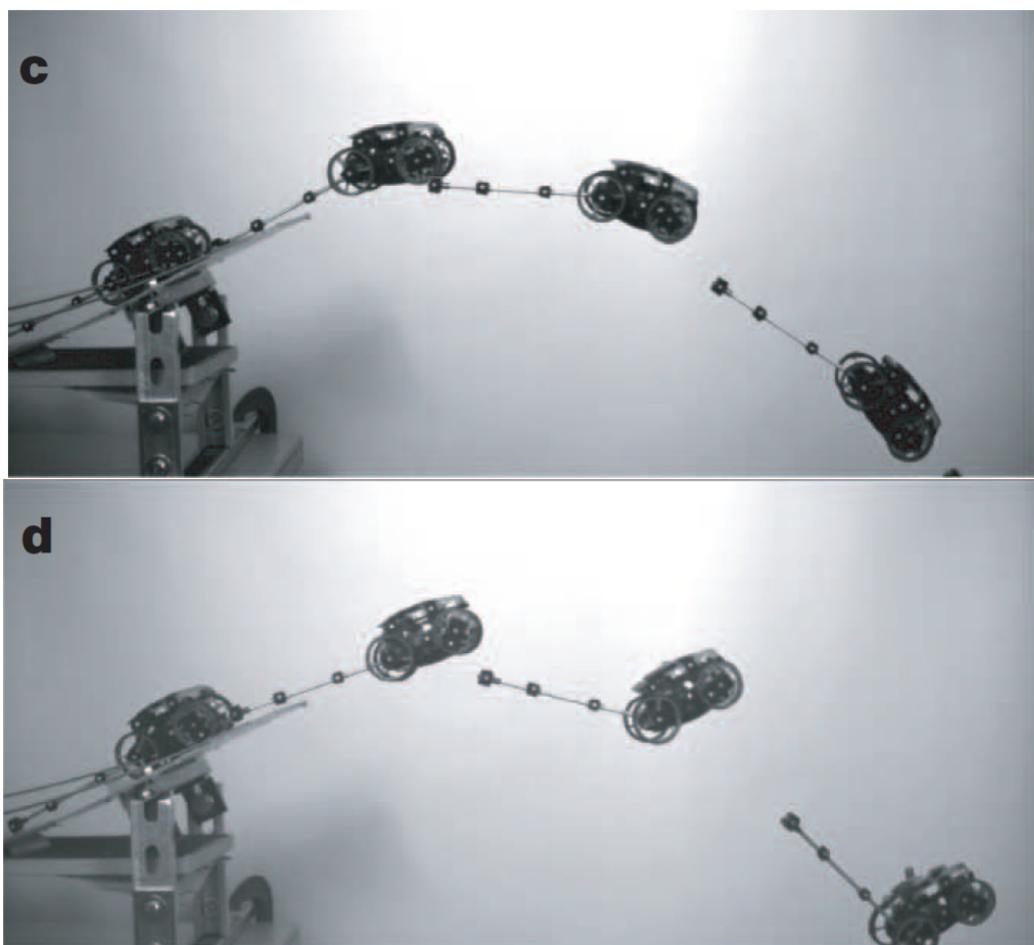


Figure 15: Tailbot robot platform using its tail to mimic the pitch correcting action found in lizards performing a jump after pitch perturbation

2.4.7 FlipBot

Robotic attitude control was taken a step further than perturbation correction by Fisher and Patel in [34], illustrating the ability of a tail-like appendage to provide sufficient angular impulse to perform a stunt. The Flipbot robot functioned on the same angular impulse theory, and used a tail rotating in the roll plane to perform a barrel roll off a ramp. The robot made use of a PI controller in conjunction with gyroscopic sensory feedback to perform the stunt. Figure 16 illustrates the robot performing said stunt [34].



Figure 16: The FlipBot robot performing a barrel roll stunt with tail actuation in the roll plane

2.4.8 Flexible and Rigid Tails

Tails found in nature on both mammals and reptiles are flexible, yet most of the tails implemented on robotic systems have been rigid. The main reason for this is to simplify the dynamics and control [1]. If a flexible link were to be used in place of a rigid link as a tail model, the complexity of the system would increase substantially. The dynamic modelling of the system would be complicated by the storage and loss of energy in the deformation of the link [35], and hence the control methods would need to be adapted to handle this energy storage. Adaptive control and neural networks have been used to deal with multiple flexible link systems [36], but this is outside the scope of this study. At this stage of bionic tail research, the potential advantages of a flexible tail are not considered a validation of the cost associated with the additional complexity, especially when the successes of implemented rigid tails are considered.

2.5 Conclusion

This literature review has illustrated the innovation achieved through biomimetic design, and hence supports the practice of looking to nature for solutions to problems in engineering, medicine and science. Current biomimetic designs in industrial and research engineering are performing well and have invoked greater interest in the field.

The literature reviewed also revealed that many flightless animals have shown the ability to re-orientate themselves during brief periods of flight. This attribute reduces the risk of injury to the animal, and is thus a desirable quality for robotic applications, which could reduce damage in many scenarios. Of the methods of attitude control used by animals, the reptilian use of tail inertia to generate angular impulses was judged to be most suitable to robotic applications.

Many robotic platforms have successfully altered their body position, resisted undesirable force and moment disturbances as well as performed stunts using tails. The research suggests that the optimal method to perform the stunt required by this project will be to use a rigid tail to provide an angular impulse on the body, and that the tail-to-body length ratio should be carefully considered and modelled. The current state of the art and research to date is very promising when considering the task of developing a robotic platform to perform a 360-degree aerial yaw stunt.

3. Theory Development

This chapter aims to explore all the theoretical tools that were required to complete the project. The key features of the background research conducted are presented here.

3.1 Lagrange Dynamics

The system was dynamically modelled using the Lagrange method. The Lagrangian is defined by equation 1. Where T is the kinetic energy of the system, and V is the potential energy of the system. Equation 2 is equated to the virtual work done by the system, and was evaluated term by term to develop the equations of motion for the system [36][37].

$$\mathcal{L} = T - V \quad (1)$$

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_j} \right) - \frac{\partial \mathcal{L}}{\partial q_j} + \frac{\partial V}{\partial q_j} = Q_j \quad (2)$$

The Lagrange method makes use of a generalized coordinate system, in which each coordinate is denoted by q_j . The selected coordinates must satisfy three criteria [37]:

- i. They must be complete – they must completely describe the position of all parts of the system.
- ii. They must be independent – they must be able to achieve full flexion while the other coordinates are fixed.
- iii. They must be holonomic – the number of degrees of freedom must equal the number of coordinates to completely describe the system.

3.2 Friction Models of DC Motors

Friction has been studied prolifically due to its effect on motor applications, specifically the DC motor [38]. In many applications, motors are required to perform precise position or speed control, and in such cases friction is often a large inhibiting factor [39]. Friction complicates the control of motors as it introduces nonlinearities into motor models, thus limiting the performance of common linear control designs.

Despite extensive study, there are a number of differing models characterizing friction as a nonlinear function of angular velocity [40]. This study has considered the popular model that combines Coulomb and viscous friction to generate a fully descriptive friction characteristic.

3.2.1 Static Friction

Static friction is the restrictive force that occurs between two dry, stationary surfaces in contact [39]. The magnitude of static frictional forces exactly matches the relative applied force between the objects, until that force exceeds a threshold (breakaway force), after which motions between the surfaces will develop and static friction ceases to operate on them [38]. The force characteristic of static friction is shown in the left of Figure 17.

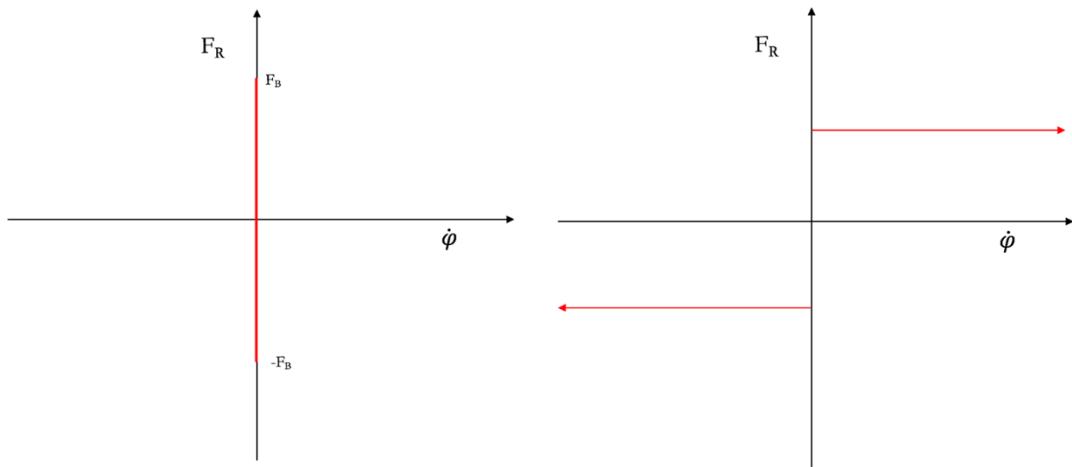


Figure 17: Left - Static friction velocity characteristic, Right – Coulomb friction velocity characteristic

3.2.2 Coulomb Friction

Coulomb friction, also known as dynamic friction, is present once a system is in motion, is less than static friction and is independent of velocity. The magnitude of Coulomb friction is dependent on the surfaces in contact [38]. Its characteristic is depicted in the right of Figure 17.

3.2.3 Viscous Friction

Viscous friction is dependent on angular velocity [38]. It has no effect at zero angular velocity (a zero value intercept), but the magnitude increases with angular velocity. Literature most commonly considers viscous friction as having a linear dependency on angular velocity, thus the coefficient B can be used to produce resistive torque versus angular velocity characteristic, as shown in Figure 18 [39].

$$F_R = B\dot{\phi} \quad (3)$$

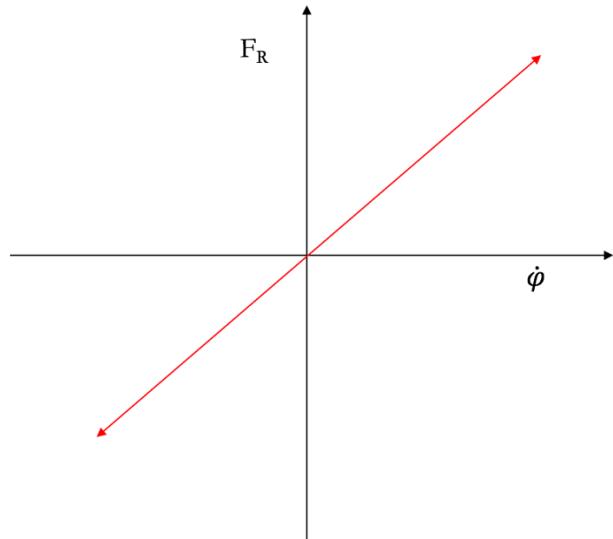


Figure 18: Viscous friction against speed characteristic

3.2.4 Integrated Friction Model

The different friction effects can be combined into a single highly nonlinear characteristic, as shown by Figure 19 [39].

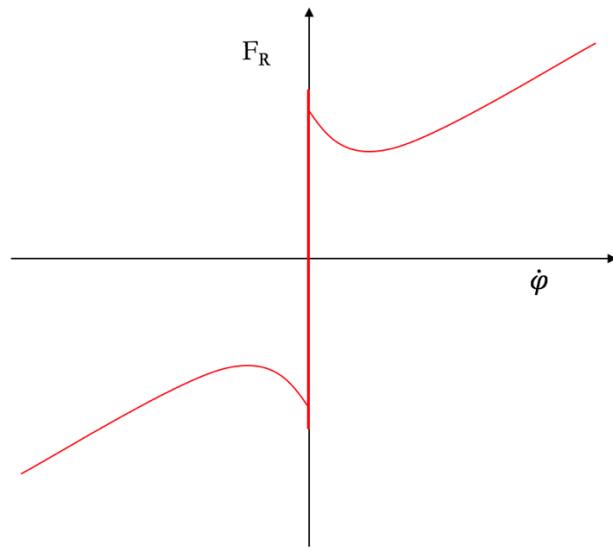


Figure 19: Integrated friction model

3.3 Controllability and Observability

When considering a system in terms of its state space representation, which is the characterization of a system's differential equations in matrix form, it is vital to assess the extent to which each state can be controlled and observed before controller design can occur. The knowledge of controllability and observability largely influenced the control design method used in the project.

3.3.1 Matrix Rank

The theory of matrix rank is used to analysis both controllability and observability. The rank of a matrix is an indication of the dimension of the space generated by the matrix, or more appropriately in this case, an indication of the number of linearly independent rows or columns. Full rank implies the space generated by the matrix is equal to its dimensionally [41].

3.3.2 Observability

Observability quite simply means that information about the states of a system can be accessed [42]. For the case of observability, it is sufficient to consider the free response of a system:

$$\dot{x}(t) = Ax(t) \quad (4)$$

$$y(t) = Cx(t) \quad (5)$$

Letting the initial state be $x(t_0)$ and unknown, equation (4) and (5) can be used to extrapolate through taking derivatives between the output and state vector:

$$\begin{aligned} y(t_0) &= Cx(t_0) \\ \dot{y}(t_0) &= C\dot{x}(t_0) = CAx(t_0) \\ \ddot{y}(t_0) &= C\ddot{x}(t_0) = CA^2x(t_0) \\ &\vdots \\ y^{n-1}(t_0) &= Cx^{n-1}(t_0) = CA^{n-1}x(t_0) \end{aligned}$$

Representing the expansion in matrix form:

$$\begin{bmatrix} y(t_0) \\ \dot{y}(t_0) \\ \ddot{y}(t_0) \\ \vdots \\ y^{n-1}(t_0) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} x(t_0) = O x(t_0) = Y(t) \quad (6)$$

C is a pxn matrix, (where p is likely to be one for this application), and **A** is an nxn matrix, **O**, the controllability matrix will be an (np)xn matrix. From linear algebra theory, it is clear that if the controllability matrix has full rank, the initial condition $x(t_0)$ can be uniquely determined [19].

3.3.3 Controllability

In the case of controllability, the forced response must be considered. Let the forced response of a system be defined by:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (7)$$

Using the same process as that in the observability discussion, taking derivatives gives:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ \ddot{x} &= A^2x + ABu + B\dot{u} \\ &\vdots \\ x^n &= A^n x + A^{n-1}Bu + A^{n-2}B\dot{u} + \cdots + Bu^{n-1} \end{aligned}$$

Re-arranging the subject of the formula to ensure the right side has only inputs.

$$x^n - A^n x = A^{n-1}Bu + A^{n-2}B\dot{u} + \cdots + Bu^{n-1}$$

Representing the set of equations in matrix form:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \vdots \\ x^n \end{bmatrix} - A^n x = [B \quad AB \quad A^2B \quad \cdots \quad A^{n-1}B] \begin{bmatrix} u^{n-1} \\ u^{n-2} \\ \vdots \\ u \end{bmatrix} = \mathcal{C} \begin{bmatrix} u^{n-1} \\ u^{n-2} \\ \vdots \\ u \end{bmatrix} \quad (8)$$

The controllability matrix, \mathcal{C} is defined as:

$$\mathcal{C} = [B \quad AB \quad A^2B \quad \cdots \quad A^{n-1}B] \quad (9)$$

Equation (8) is valid for all t in $[t_0, t_f]$, provided t_f is finite, if the controllability matrix is non-singular and the input vector will exist for all such time instances [42]. It can again be concluded that the system is controllable if, and only if, \mathcal{C} has full rank. In general (including multi-input systems), if the input has dimension r , and **A** is an $n \times n$ matrix, then \mathcal{C} will have dimension $n \times rn$, and if the rank of \mathcal{C} equal to n, the system is controllable [42].

The system under consideration in this project is a single input system, and hence the controllability can be assessed for this specific case, in which it is sufficient to examine whether \mathcal{C} is singular or not.

3.4 State Feedback Steady State Error

3.4.1 Steady State Error Illustration

It can be shown that state feedback control design exhibits high quality dynamic performance, but is unlikely to track step inputs to a system with zero steady state error [43]. The reason for this problem is best explained mathematically. Consider the following linear system in a state feedback configuration, illustrated by Figure 20.

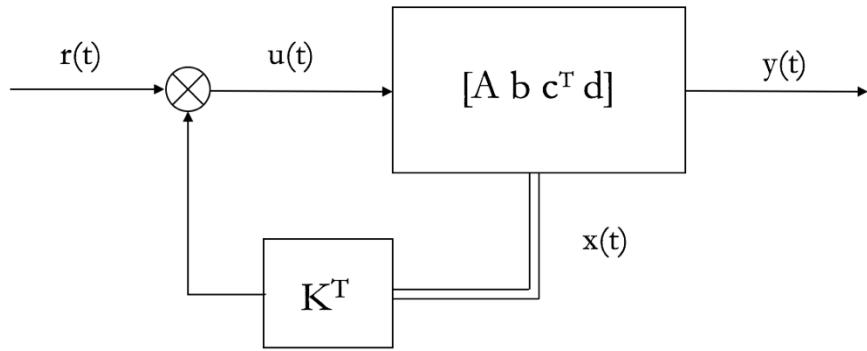


Figure 20: State feedback loop

The state differential and output equations are

$$\frac{d}{dt}x(t) = Ax(t) + bu(t) \quad (10)$$

$$y(t) = c^T x(t) + du(t) \quad (11)$$

Where x is a vector of length n , A is a $n \times n$ matrix, b is an $n \times m$ matrix and u is a vector of length m (this is the general case, it is common that u is a single input and thus b is a vector of length n). The output equation in the general form has c^T as a $p \times n$ matrix, and d as a $p \times m$ matrix.

The resolution to steady state conditions of a system in a state feedback configuration is provided in appendix A, section A1.1. For the output to track the input, it must occur that:

$$-c^T(A - bK^T)^{-1}b = 1$$

$$du = 0$$

This is possible, but improbable given the need to conform with the parameters set by the dynamic requirements of the design. The need arises to assert the correct type number on the system if it is to track set points with zero steady state error. A popular method of achieving this is by adding the appropriate number of integrators into the loop [43].

3.4.2 Outer Loop Integrator State Feedback Design

For step type set points (as applicable to this project), the new control configuration is shown in Figure 21.

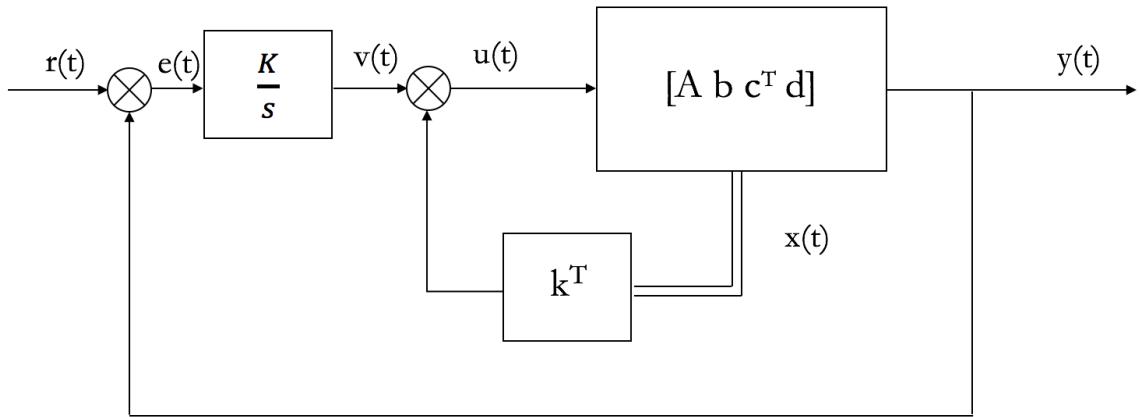


Figure 21: State feedback configuration with outer loop integrator

The integrator introduces a new state into the system, and a new state differential equation can be derived.

$$\frac{d}{dt} \begin{pmatrix} x(t) \\ w(t) \end{pmatrix} = \begin{bmatrix} A - bk^T & -BK \\ c^T & 0 \end{bmatrix} \begin{pmatrix} x(t) \\ w(t) \end{pmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(t) \quad (12)$$

This result is the factored form of the equations developed in Appendix A, section A1.2. The matrix applied to the new state vector $\begin{pmatrix} x(t) \\ w(t) \end{pmatrix}$ can be considered to be an ‘augmented’ matrix representing the closed loop system response to an input, r . Thus applying the steady state condition $\left(\frac{d}{dt} = 0\right)$:

$$\begin{bmatrix} 0 \\ r \end{bmatrix} = \begin{bmatrix} (A - bk^T)x - BKw \\ c^T x \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \quad (13)$$

Equation 13 shows that in steady state the output, y , is equal to the input r , hence the condition of zero steady state error is achieved.

3.5 State Observers

In many control system applications, certain system states are not available for measurements; this can occur due to various reasons, such as the expense of real time online instrumentation or to the difficulty of accessing the signal location [43]. In order to handle such scenarios, the theory of state estimation or observer design has been developed. The topic has been explored in depth as a consequence of its importance, and advanced, accurate methods of state estimation have been developed. A modern popular example is the Kalman Filter [44].

This project required formulation a state space system model in which not all states were available, and thus the theory of observer design was applicable. This analysis will consider the continuous time case, but a transformation to the discrete time case is simple and the same theoretical constructions apply [44].

3.5.1 Observer Formulation

The the system under consideration can be described by the same set of state space matrices outlined in 3.4, namely \mathbf{A} , \mathbf{b} , \mathbf{c}^T and \mathbf{d} . The state of the system at any time can be estimated using equations 14 and 15 below.

$$\frac{d}{dt}\hat{x}(t) = A\hat{x}(t) + bu(t) \quad (14)$$

$$\hat{y}(t) = c^T\hat{x}(t) + du(t) \quad (15)$$

Where \hat{x} represents an estimate of the system states, and \hat{y} is an estimate of the system output at the current time instance. This formulation is known as the real time simulator [44], and simply uses the dynamic model of the system. This is unlikely to be an accurate estimation, given modelling inaccuracies and system disturbances. In order to improve the accuracy of the estimator, the measured output of the system is compared to the estimated output, and this term is used to correct the state estimate [43][44]. This scenario is shown in equation 16.

$$\frac{d}{dt}\hat{x}(t) = A\hat{x}(t) + bu(t) + p(y(t) - \hat{y}(t)) \quad (16)$$

Where the p represents a weighting (or gain) vector applied to the error term in the state estimator equation. Equation 16 is a standard state observer formation [43][44]. The error of the state

estimator can be obtained by subtracting the estimated states from the ‘true’ states of the system, this operation yields the error vector, $q(t)$.

$$q(t) = x(t) - \hat{x}(t) \quad (17)$$

$$\frac{d}{dt}q(t) = Ax(t) + bu(t) - Ax(t) + bu(t) + pc^T x(t) - c^T x(t) \quad (18)$$

Equation 15 is used to make the substitution in the error term, and this allows the system to be factored into the following form [43]:

$$\frac{d}{dt}q(t) = [A - pc^T]q(t) \quad (19)$$

3.5.2 Observer Design

Equation 19 provides a model for the dynamics of the state estimation error, and these dynamics are governed by the eigenvalues of the $[A - pc^T]$ term [44] (in much the same way that a standard state feedback model is governed). If all the states are observable, the gain vector, p , can be adjusted to alter the dynamics of the error vector, q [44]. The characteristic equation that will define the system is:

$$\psi_c = \text{Det}(sI - A + pc^T) \quad (20)$$

In order for the system to act on accurate information, the observer error dynamics need to be sufficiently fast [43]. What is inferred by this statement is dependent on the system under consideration.

Figure 22 illustrates a typical control loop which incorporates a state observer. The block containing the p vector can be manipulated to achieve the desired dynamics.

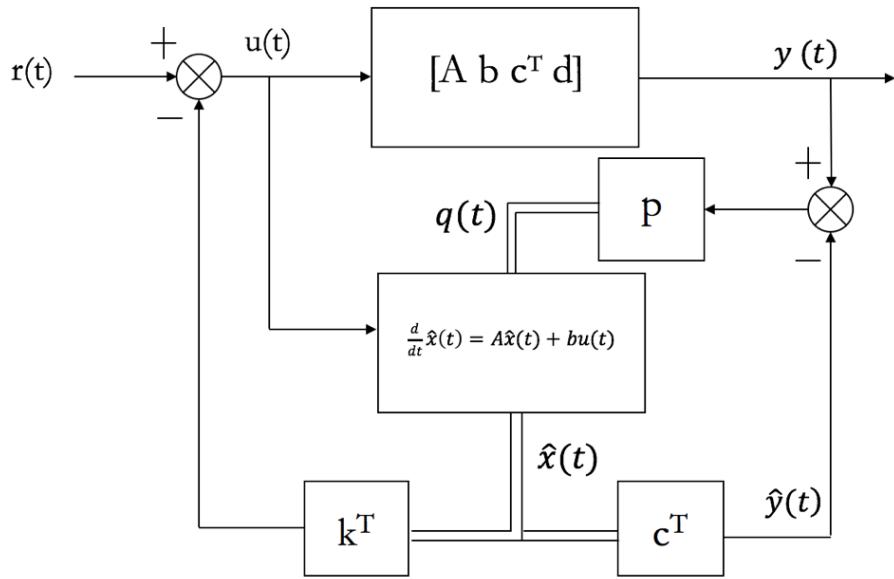


Figure 22: State feedback control loop including a state observer [22][23]

3.6 Actuator Limitations

Most real control systems will be inhibited by nonlinearities imposed by actuator limitations. The system created to perform the stunt suffered from such limitations.

3.6.1 Voltage Saturation

The control action applied to the system was in the form of a variable voltage supplied to a DC motor, and hence the first actuator limit was the maximum and minimum applicable voltages. This limitation took the form of a simple saturation nonlinearity, as shown by Figure 23.

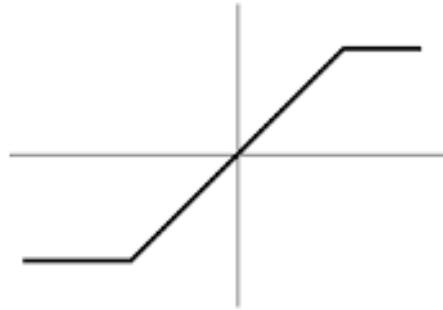


Figure 23: The characteristic of a saturation nonlinearity [45]

3.6.2 Motor Back EMF Constraint

A further actuator limitation will be imposed by the back EMF generated by the DC motor. The back EMF produced is proportional to the velocity of the motor, thus the maximum current attainable at each motor velocity is limited by this phenomenon in conjunction with the voltage saturation.

The DC motor was analysed using an equivalent circuit model. For the purpose of gaining insight into DC motor actuator limitations, it is acceptable to ignore the armature inductance (and with it the electrical dynamics) [46].

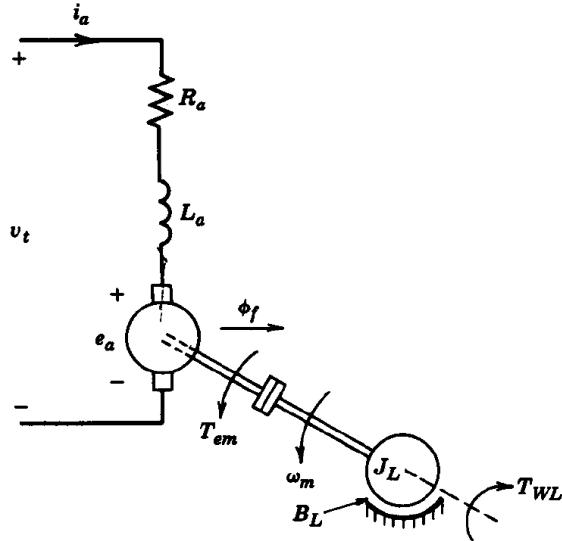


Figure 24: Equivalent circuit model of a DC motor [47]

The following equations describe the function of a DC motor:

$$e_a = k_e \varphi_{ag} \omega_m \quad (21)$$

$$T = k_t \varphi_{ag} i_a \quad (22)$$

$$i_a = \frac{v_t - e_a}{R_a} \quad (\text{where } L_a \text{ is assumed to be negligible}) \quad (23)$$

k_e is the speed constant

k_t is the torque constant

φ_{ag} is the airgap flux

ω_m is the motor speed in radians per second

Subbing 21 into 23, and then 23 into 22 produces equation 24:

$$T = \frac{k_t \varphi_{ag}}{R_a} (v_t - k_e \varphi_{ag} \omega_m) \quad (24)$$

Multiplying out and combining constant terms results in equation 25:

$$T = k_1 v_t - k_2 \omega_m \quad (25)$$

For a fixed terminal voltage, this equation illustrates the linear behaviour between torque and angular velocity of the motor shaft. Again it is reiterated that these equations are developed assuming that the motor electrical time constant $\left(\frac{L_a}{R_a}\right)$ is small enough to be neglected.

The resultant torque speed characteristic takes the form of a straight line graph, shown in Figure 25, with the maximum torque achievable for a fixed terminal voltage decreasing with angular velocity. This limits the control action that can be applied at each angular velocity of the motor.

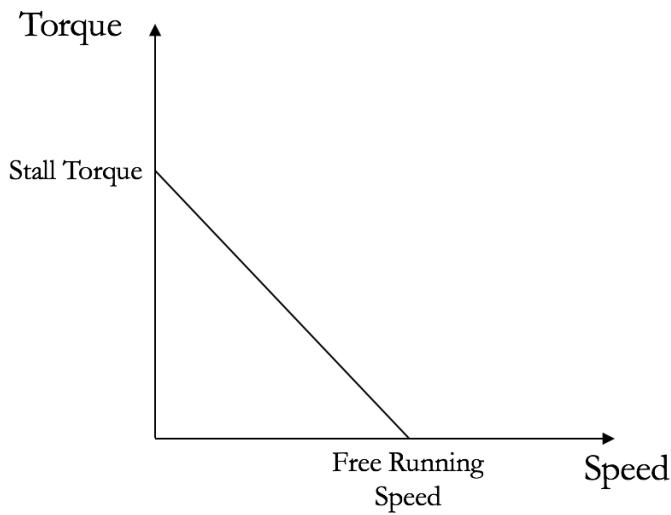


Figure 25: Simplified torque speed characteristic for a DC motor

3.7 Euler Angles and Rotation Matrices

Euler angles are a standard way of describing a rigid body's orientation in a three dimensional Euclidean space [48]. In many robotic applications, a robot platform will relay information about its position by means of on-board sensors; these sensor measure changes in the platforms position in terms of its own body's reference frame. In many cases it is required that a platform's position be described in terms of the inertial frame, and hence rotation matrices are required to perform this transformation. A popular method involves the use of the 3-2-1 Euler Angles yaw pitch roll sequence [48].

For this application, the resolution of angular position to the inertial frame was not judged necessary, or in anyway beneficial. The stunt called for the rotation of the body in one plane, thus ideally, the body's angular reference frame will not be altered from the inertial. If the body did

undergo two angular changes, for example if it rolled clockwise and yawed counter clockwise through 90 degrees, the body yaw plane would have become the inertial roll plane, and the body roll plane would have become the inertial pitch plane. This scenario is depicted in Figure 26.

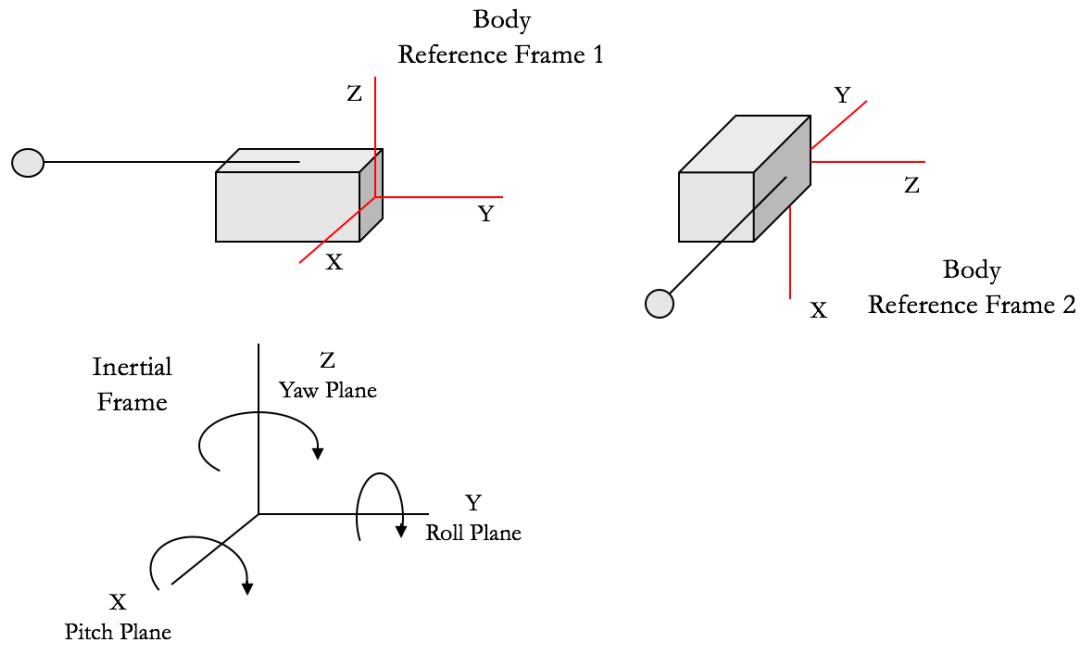


Figure 26: Diagram to illustrate the transformation of angular planes with respect to the inertial frame

In this hypothetical scenario, if the body reference frame had been resolved to the inertial frame, the control action would have requested the tail to move in the inertial yaw plane to complete the stunt. The inertial yaw plane has been transformed to the body pitch plane, and the control action would have requested the tail to pitch. Since the tail was a one dimensional link, this was not possible, and the resolution of references frames was not required or advantageous.

4. Design Methodology and Experimental Procedure

This chapter provides an overview of the research processes followed in order to achieve the requirements set out in both the terms of reference and introductory chapter. It explains the procedure followed in order to acquire simulation and empirical data and results.

4.1 System Requirements and Research

The requirements for the bio-inspired robotic system were set out in the introductory chapter. The ability of a robotic tail to provide aerial orientation and attitude control was to be investigated, which was conducted by means of a literature review, presented in chapter 2. It was further required to model, design, build and test a robot that was capable of using a tail to perform a 360-degree yaw stunt while airborne.

4.2 System Modelling, Design and Simulation

4.2.1 Modelling of the System

The first step in the research was to model the system in its entirety as accurately as possible. This project made use of the Lagrange method to model the body and tail system, as this is an energy method suited to closed system dynamics. Lagrange was used to derive the equations of motion describing the system. The electrical and mechanical aspects of the tail motor were modelled thereafter which resulted in the characteristic electrical and mechanical differential equations for the system.

4.2.2 Simulation

Matlab and Simulink were used to simulate the models that were derived in order to partially validate the models, select appropriate components, and gain insight into design constraints that will be imposed on the system. The simulation data was also compared to the empirical data collected during testing.

4.2.3 Design

In order to perform the stunt, the robotic platform needed to be designed and built. This called for hardware electronic design, mechanical design and software design. A modular design approach was adopted in order to ensure full system functionality post module integration.

The hardware electronic systems designed for the stunt included two driver circuits, one bidirectional and one unidirectional, a launch detection circuit, a communication module, power supply circuitry and microcontroller interfacing circuitry. The hardware design needed to be very compact, low cost and well suited to the application.

The platform used as the base plate for the robot needed to be designed to house all the circuitry, a two cell Lipo battery, and support the tail motor and tail in the correct position. The mechanical design was influenced by the minimal space available on the robot, and the platform needed to be able to withstand the torsional strain applied during the stunt. The tail used for the stunt was designed and fabricated based on optimal simulations, and specifications found in the literature. A test rig also was designed in order to perform tests on the platform in a safe environment..

The software written for the robot interfaced with all the circuitry, sensors, performed the control computations and control action, and sent data to the computer base station. It was critical that the software on-board the robot was designed robustly, with numerous fail safe functions in place. The software for the data logging application was designed to receive data from the car, display the data real time in the application window, as well as write the data to a file in order that analysis may be conducted on the data.

4.3 Model Verification and Controller Design

Once the system had been designed and completely constructed, system identification tests were conducted. These tests were used to validate the various derived models and the results were compared to previously generated simulations. The outcome of the accuracy of the modelling had a profound influence on the choice of design method used for the controller design.

The controller was designed using an appropriate method based on information provided by the system identification tests. The controller needed to settle fast enough to perform the stunt in the airtime achieved by the robot, it also needed to be exceptionally well damped, as overshoot could cause a crash, and the final steady state error needed to be very small otherwise the robot would crash on landing. The control system was simulated in Simulink.

4.4 Testing and Experimentation

Once all the requirements outlined above had been achieved, the system was then tested. The controller was first run on the test rig, which verified that the design met the requirements to perform the stunt. Provisional tests were run to ensure that the stunt would be performed safely, such as a launch detection test and an unactuated jump test. When it was ascertained that the jump procedure was operating safely, an actuated jump test was run. The test was performed numerous times in an attempt to attain a sufficient amount of data to ensure reliable results and analysis. Each stunt was filmed using a GoPro Hero 3 Black camera, set at the highest frame rate possible in order to visually document the stunts.

5. System Modelling and Simulation

This chapter develops the dynamic and electric equations that govern the system. It presents the models in diagrammatic form, as well as in state space matrices. It then goes on to simulate the models in the Matlab Simulink environment.

5.1 Dynamic Modelling

5.1.1 The Robotic Tail and Platform System

The body and tail system under consideration was modelled in the horizontal plane, and in two dimensions. This model limits the developed equations to act in the yaw plane alone. The pitch and roll effects of the tail actuation were considered negligible during the initial analysis, and subsequently explored separately. The projectile motion of the system was also considered separately, this treatment was valid as the rotational and projectile motion are decoupled.

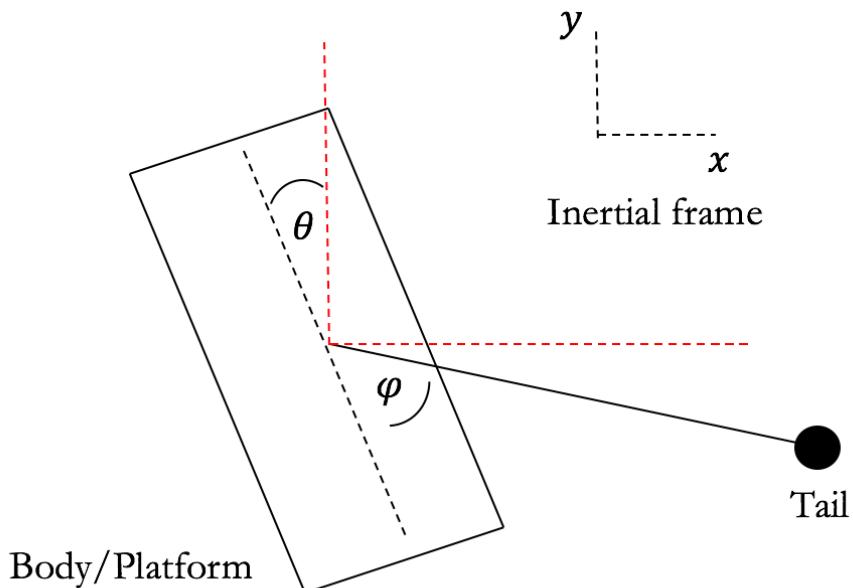


Figure 27: Body and tail planar representation showing the coordinate system selected to use in the development of equations of motion using Lagrange dynamics

The selected coordinates for the system under consideration were:

1. θ - the body angle relative to the inertial frame
2. φ - the tail angle relative to the body.

These coordinates successfully satisfied the three requirements of Lagrangian dynamics.

Summary of constraints:

- i. The system was constrained to act in two dimensions: the horizontal plane. This implied the system possessed no potential energy ($V = 0$).
- ii. Translational motion was ignored, as it is largely independent of the rotational motion of interest.
- iii. The projectile calculations were performed separately.
- iv. Friction at the tail joint was assumed to be negligible for development of the dynamic equations (it was later incorporated in motor modelling for controller design).
- v. The tail was assumed to be perfectly rigid.

5.1.2 Development of Equations of Motion

Kinetic energy of the system can be represented by equation 26, and the potential energy by equation 27:

$$T = \frac{1}{2}I_B\dot{\theta}^2 + \frac{1}{2}I_T(\dot{\theta} + \dot{\varphi})^2 \quad (26)$$

$$V = 0 \quad (27)$$

Where I_B is the rotational inertia of the body, I_T is that of the tail. The equations of motion were developed term by term using equation 2 and the Lagrange method. The mathematic steps are shown in appendix A, section A2.1. The deduced equations of motion are:

$$0 = I_B\ddot{\theta} + I_T(\ddot{\theta} + \ddot{\varphi}) \quad (28)$$

$$\tau = I_T(\ddot{\theta} + \ddot{\varphi}) \quad (29)$$

These equations were used to predict the dynamic response of the system, and simulations were built using Matlab's Simulink environment.

5.1.3 Conversion to Simulation Compatible Form

In order to simulate the dynamics, the equations were rearranged to isolate the variables corresponding to the generalized coordinates. A mass matrix can be used to achieve this isolation [35], but this was not required and variable substitution was used. Appendix A, section A2.2 shows the steps followed to arrive at the equations below.

$$\ddot{\theta} = -\frac{I_T \ddot{\phi}}{I_B + I_T} \quad (30)$$

$$\ddot{\phi} = \frac{1}{I_T} \tau - \ddot{\theta} \quad (31)$$

Equation 30 and 31 are in forms suitable for programmable simulation.

5.2 Analysis of the Dynamics

5.2.1 Significance of the Inertial Ratio

From the dynamic equations, it is clear that the ratio of body rotational inertia to tail inertia is an important quantity. The equations illustrate that this ratio will influence body acceleration in comparison to tail acceleration. In the hypothetical case in which body inertia is much less than tail inertia ($I_B \ll I_T$), the magnitude of the body acceleration will be equal to that of the tail, for all other cases the magnitude of body acceleration will be less than tail acceleration.

The inertial ratio is given by $\frac{I_B}{I_T}$. If adjustments need to be made to the amount of rotation the tail travels through, this quantity will need to be altered.

5.2.1 Significance of Coupling

In order to achieve the desired control action on the system, some important distinctions must be highlighted and made exceptionally clear. The torque generated by the motor is applied directly to the tail, but the effect on the angular accelerations of both the body and tail can be related directly to the torque generated by the motor using the dynamic equations 30 and 31, despite these elements being dynamically coupled. The back EMF produced in the motor is solely dependent on tail speed, and related through the motor constant K_e and gear ratio, hence this is decoupled from the body since the tail coordinate is given relative to the body. These distinctions make it possible to define the system to be controlled by a 4 state, state space model.

5.3 The DC Motor Model

The DC motor model was derived from the equivalent circuit illustrated by figure 9, in section 3.6.2. The system electrical differential equation is [47]:

$$v_t = iR + L \frac{di}{dt} + e_a \quad (32)$$

Where the variables are, terminal voltage, armature current, armature resistance, armature inductance, and back EMF respectively. This was used in conjunction with the mechanical differential equation [47]:

$$T_d = J_{eq} \frac{d}{dt} \dot{\phi}_m + B\phi_m + T_L \quad (33)$$

Where the variables are developed torque, equivalent rotational inertia, motor shaft angular velocity, viscous friction, and torque load respectively. These two equations were linked using the motor equations: [47]

$$e_a = K_e \dot{\phi}_m \quad (34)$$

$$T_d = K_t i \quad (35)$$

K_e and K_t are the motor constants, the determination of which will be discussed later. The differential equations can be represented in a closed control loop, which is shown in Figure 28

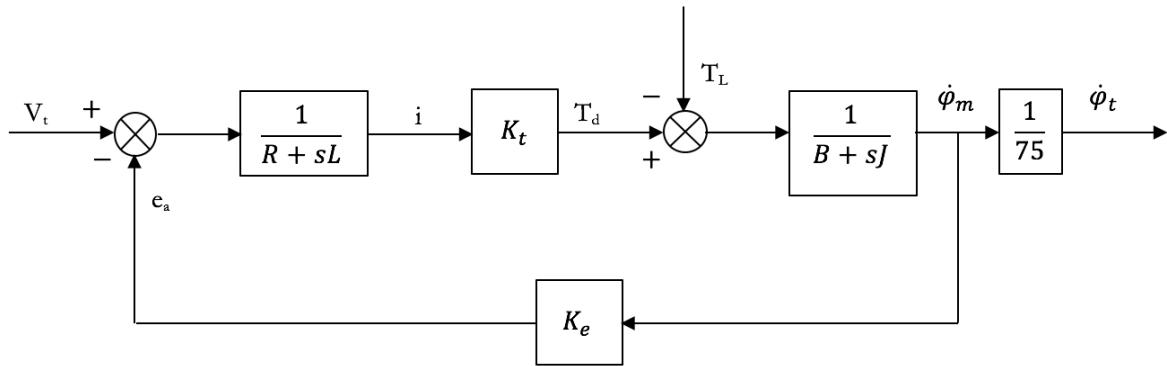


Figure 28: Control loop of a DC motor

It must be noted that signal feedback through the gain block, K_e , is the motor shaft speed, not the tail speed. The tail speed can easily be obtained by passing the shaft speed through a gain block representing the gear ratio. It is also noted that the shaft has no external torque loading, thus the second comparator was eliminated.

5.4 Synthesis of the Dynamic and Motor Models

Using equations (30) and (31), the angular acceleration of the body and that of the tail was related to the torque applied to the tail. By using mutual substitution and variable isolation; the mathematical steps are shown in appendix A, section A2.3. The following relations were derived:

$$\ddot{\theta} = - \left(\frac{\frac{1}{I_B + I_T}}{1 - \frac{I_T}{I_B + I_T}} \right) \tau \quad (36)$$

$$\ddot{\varphi} = \left(\frac{1}{1 - \frac{I_T}{I_B + I_T}} \right) \left(\frac{1}{I_T} \right) \tau \quad (37)$$

Equations 36 and 37 relate the respective element's angular accelerations to torques through constant terms (the body and tail inertia). Classical inertia, which is a constant, relates angular acceleration to torque in the same way, hence the dynamic coupling can be seen to create pseudo inertias for each element that characterize its response to a torque input. As a result, the coupling effect was incorporated into the control loop using this concept of pseudo inertia, and the body was added to the loop without altering the feedback. This configuration can be seen in Figure 29.

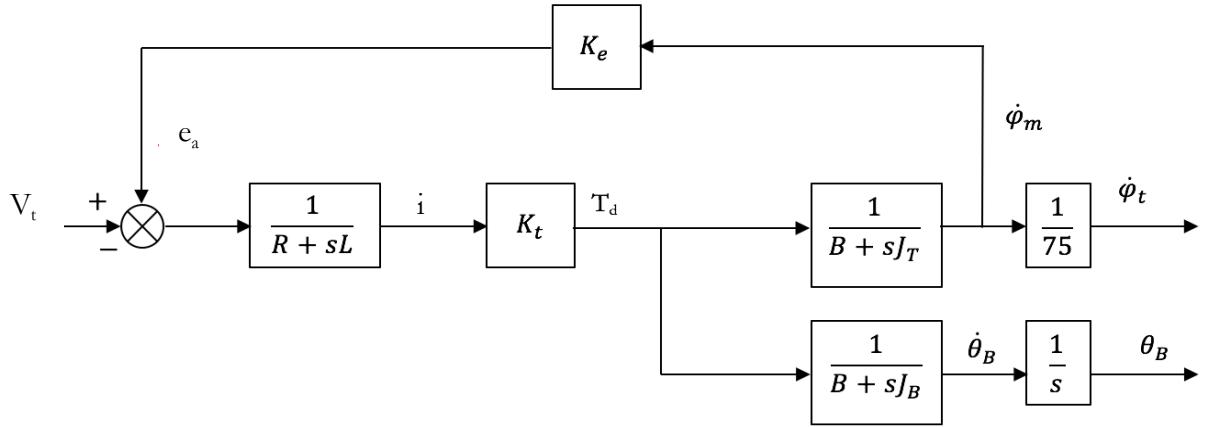


Figure 29: Augmented system control loop including the pseudo inertia concept developed in section 5.5 and the body position output signal

5.5 State Space Representation

The set of differential equations that describe the system can be formulated in a state space model.

The model will have four states: body position, θ , body velocity, $\dot{\theta}$, motor shaft velocity, $\dot{\phi}_m$, and DC motor armature current, i . These are summarized in *Table 4*.

Table 4: State variables of the system

State Name	Symbol
Body position	θ
Body velocity	$\dot{\theta}$
Motor shaft velocity	$\dot{\phi}_m$
Armature current	i

The state differential equations can be constructed in matrix form from equations 32, 33, 36 and 37:

$$\frac{d}{dt} \begin{bmatrix} \theta \\ \dot{\theta} \\ \dot{\phi}_m \\ i \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -B_o x_1 & 75\eta K_t x_1 \\ 0 & 0 & -5625 B_m x_2 & 5625 K_t x_2 \\ 0 & 0 & -K_e/L & -R/L \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \dot{\phi}_m \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/L \end{bmatrix} V_t \quad (38)$$

where

$$x_1 = -\left(\frac{\frac{1}{I_B+I_T}}{1-\frac{I_T}{I_B+I_T}}\right)$$

$$x_2 = \left(\frac{1}{1-\frac{I_T}{I_B+I_T}}\right) \left(\frac{1}{I_T}\right)$$

The torque developed on the motor shaft is related to the body through the gearbox, which has a ratio of 75:1, at an efficiency η . This pseudo inertia is related to the tail, and is matched to the load through the gear ratio of 75:1, hence the factoring term 5 625 (from $(\frac{n_m}{n_L})^2$) [47].

The output matrix can be manipulated to obtain any one of the states, or others, such as tail position, but the state of interest was the body position. The selected output matrix is shown below:

$$y = [1 \ 0 \ 0 \ 0] \begin{bmatrix} \theta \\ \dot{\theta} \\ \dot{\phi}_m \\ i \end{bmatrix}$$

5.6 Uncontrolled Motor Simulation

It was desirable that the 360-degree yaw stunt be performed in as little time as possible, as this would require a short flight time, and the shorter the flight time of the robot, the less damage the system was likely to sustain during testing. In order to achieve this, it was vital that a motor be selected that can generate a sufficient angular impulse over the desired operating range and time interval [1]. Angular impulse is defined by equation 39:

$$J = \int T dt \quad (39)$$

5.6.1 Motor Options

The motor and gearbox combination was primarily selected on its ability to provide sufficient angular impulse to achieve the stunt in as little time as possible, but consideration was also given to weight and cost. The motors identified for analysis were “Pololu Metal Gearmotors”, which come with gearboxes fitted to the output shaft. Three different motors were considered with the full range of gear ratios offered as displayed in Table 5.

Table 5: Pololu Metal Gearbox motors [49]

Name	Size (mm)	Weight (g)	Gear range	Rated speed (rpm)	Starting Torque
					(nm)
Micro HPCB	10x12x26	10	10 – 1000	3000 – 32	0.028 – 0.883
20Dx42L	20D x 41-47L	44	29 – 154	450 – 90	0.18 – 0.85
25Dx58L	25D x 48-58L	82 – 96	4.4 – 499	2220 – 12	0.035 – 2.118

5.6.2 Motor Torque Speed Characteristics

For the purpose of analysing which motor is best suited to the stunt and this particular robot platform, torque speed characteristics were developed for each motor. In this analysis the electrical motor constant ($\frac{L_a}{R_a}$) was neglected, but was accounted for during controller design.

From equation 25 in section 3.6.2, it can be seen that under the assumption made, torque and speed are linearly and inversely related. The torque – speed characteristic of a DC motor can thus be approximated by a straight line graph. It is intuitive to expect this profile to scale directly with gear ratio:

- (i) As the gear ratio increases, stall torque increases and free running speed decreases proportionally.
- (ii) As the gear ratio decreases, stall torque decreases and free running speed increases proportionally

This is not the case, as the relationships between gear ratio and stall torque and free running speed were indeed both monotonic, but nonlinear, with one exception shown in Figure 31, for the Pololu motors under consideration [49]. In order to accurately generate torque speed characteristics, each motors' product quoted stall torques and free running speeds were plotted against their corresponding gear ratios. This data was then used to interpolate the correct characteristics. The curves generated for motors 1 through 3 are shown in Figure 30 to Figure 32.

The stem plots (in blue) are the quoted gear ratios and corresponding stall torques and free running speeds. The green curves are interpolations of functions that can be used to generate appropriate characteristics for each motor at each gear ratio. The green curves were fitted using Matlab's curve-fit toolbox, and are quadratic, power, polynomial and linear interpolations.

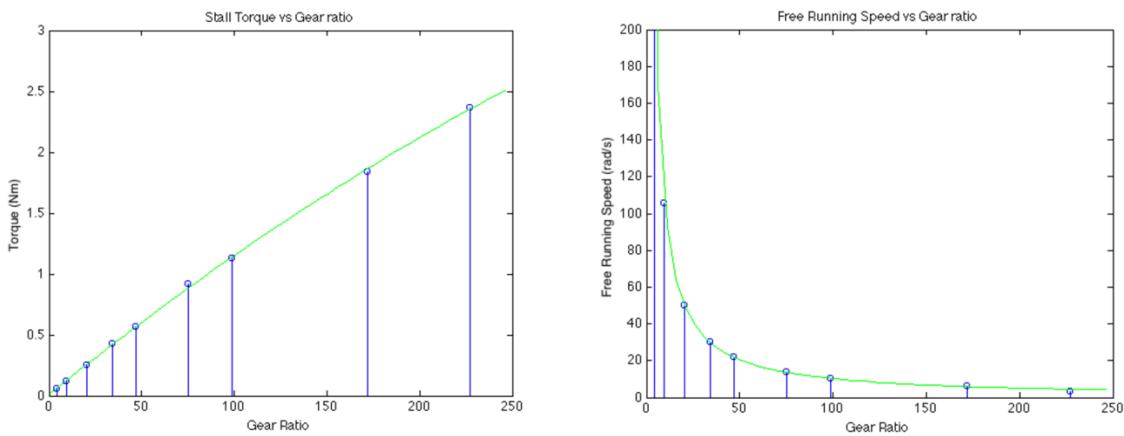


Figure 30: Stall torque and free running speed vs gear ratio for the HP Brushed Micro-metal Gearbox Motor (motor 1)

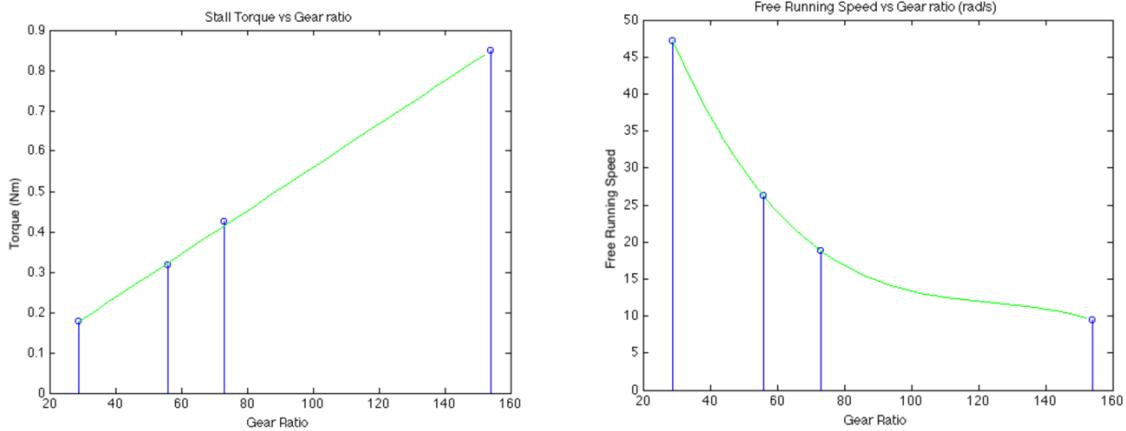


Figure 31: Stall torque and speed free running speed vs gear ratio for motor 2

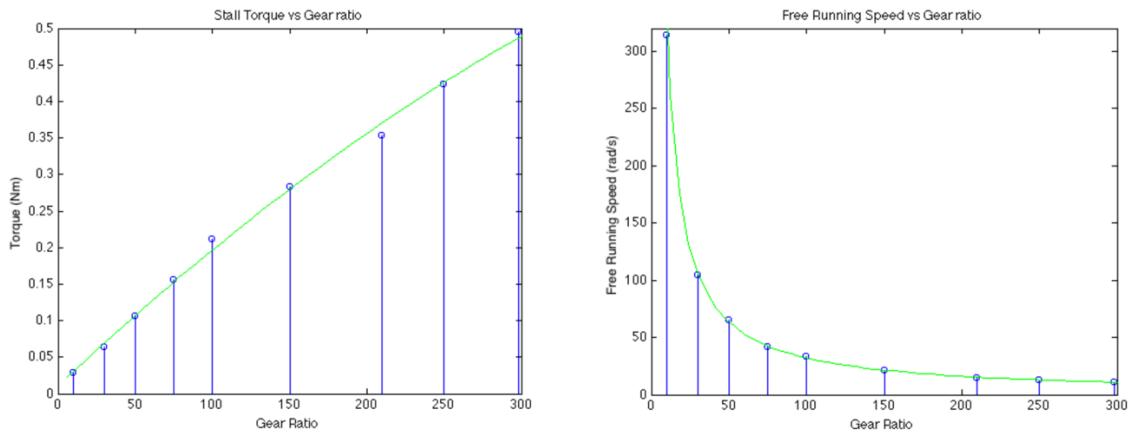


Figure 32: Stall torque and speed vs gear ratio for motor 3

With the knowledge of the stall torque and free running speed, the torque speed characteristic shown in figure 10 was generated accurately for each motor gearbox combination. The following equation was used to generate the relationship:

$$T = T_{stall} \left(1 - \frac{\omega_{current}}{\omega_{max}} \right) \quad (40)$$

5.7 Uncontrolled Simulation

In order to identify the ideal motor and gear ratio for the criteria defined earlier in this chapter, a simulation was run in Matlab's Simulink environment. The model was built using the equations of motion derived in this chapter, as well as the stall torque, free running speed versus gear ratio relations and corresponding torque – speed characteristics formulated above.

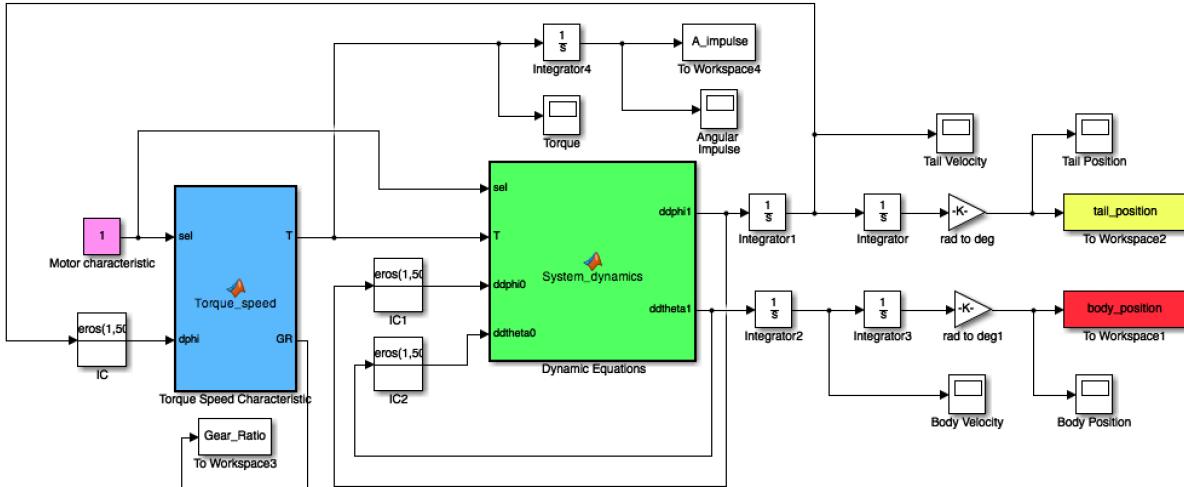


Figure 33: Simulink model that simulates the body and tail's dynamic response to an input voltage of 6V on each motor, using the developed equations of motion and motor torque vs speed characteristics

5.7.1 Simulink Model Description

(i) Input (pink block)

The model designed for the simulation takes one input: Motor characteristic. This variable is an integer between 1 and 3. The number corresponds to which motor profile is selected in the Torque Speed Characteristic function.

(ii) Function 1: Torque – Speed Characteristic (blue block)

This function applies the appropriate motor stall torque and free running speed versus gear ratio relationships. An individual torque speed characteristic (like that shown in Figure 25 and described by equation 40) is generated for every gear ratio in the range supplied by the manufacturer. The signal lines are thus all vectors, representing the temporal response to the different gear ratios in the range.

(iii) Function 2: Dynamic Equations (green block)

The second function implements the dynamic equations derived earlier in the chapter. The function takes torque applied to the tail as an input, as well as current body and tail accelerations, and outputs the recalculated body and tail accelerations.

(iv) General comments

The body and tail accelerations are initialized to zero for each gear ratio in the range. The output accelerations are integrated once to give body and tail velocity, and again to give position. A gain block is included to convert radians to degrees. The body position and tail matrices are sent to the workspace for manipulation (the red and yellow blocks respectively), as is the angular impulse matrix.

5.7.2 Simulation Results

Two three dimensional graphs are generated for each motor, mapping predicted body and tail angle (in degrees) against time and gear ratio. These graphs are shown in Figure 34, Figure 37, and Figure 40. They provide insight into what gear ratios provide the greatest angular impulse on the system for this specific inertial loading. It is interesting to note that the lower gear ratios produced greater body and tail rotation in less time, as a consequence of the small inertias of both elements.

A slice at 180 degrees was taken from each motor's body angle graph in order to evaluate the optimal gear ratio. The 180 degree mark was chosen, as these simulations involve no control action, and as a result the time to 360 degrees will be unrealistically fast; the measure of time to 180 degrees doubled is a better approximation to the controlled system. These graphs appear in the left of Figure 35, Figure 38 and Figure 41.

Tail angle against time graphs for the optimal gear ratio were also produced, for the purpose of predicting the relative rotation of the tail compared to the body – in order to ensure that it would be within acceptable limits. It would not be aesthetically desirable for the tail to make many rotations during the stunt. These graphs are shown in the right of the figures highlighted in the above paragraph.

Angular impulse curves were generated at 0.14s (approximate time to 180) for each motor to confirm the range of gear ratios generating a maximum angular impulse. These graphs are shown in Figure 36, Figure 39 and Figure 42.

A summary of important results is shown in a table at the end of each set of simulations.

5.7.2 Simulation of Motor 1

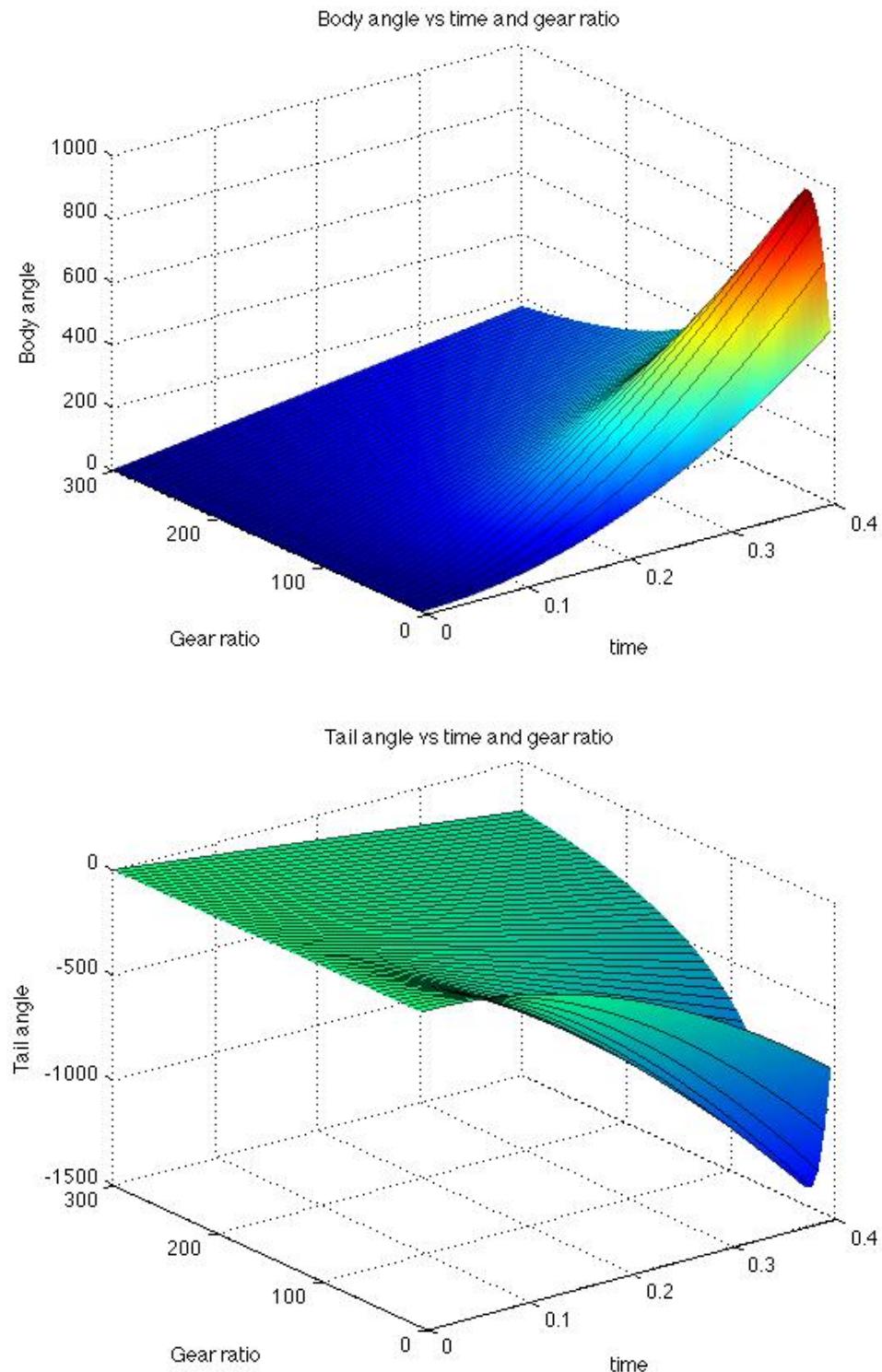


Figure 34: Simulation prediction of motor 1 – body and tail angle against time and gear ratio

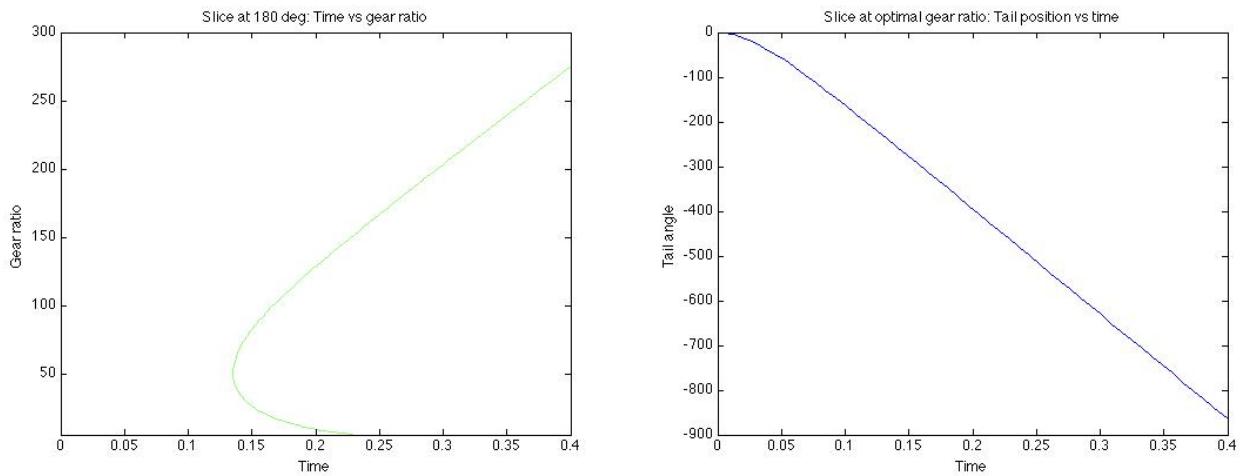


Figure 35: Motor 1, Left – Slice taken at 180 degrees to illustrate optimal gear ratio, Right - Tail angle versus time at optimal gear ratio

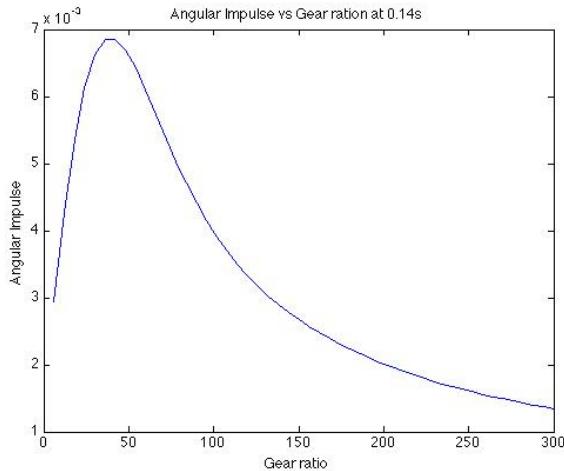


Figure 36: Motor 1 angular impulse vs gear ratio at 0.14s

The important deductions made from the first set of simulations are summarized in Table 6.

Table 6: Simulation results for motor 1

Aspect	Value
Optimal gear ratio	54:1
Time to 180 degrees	0.135s
Approximate predicted time to 360	0.27s
Tail angle rotation	-270 degrees

5.7.3 Simulation of Motor 2

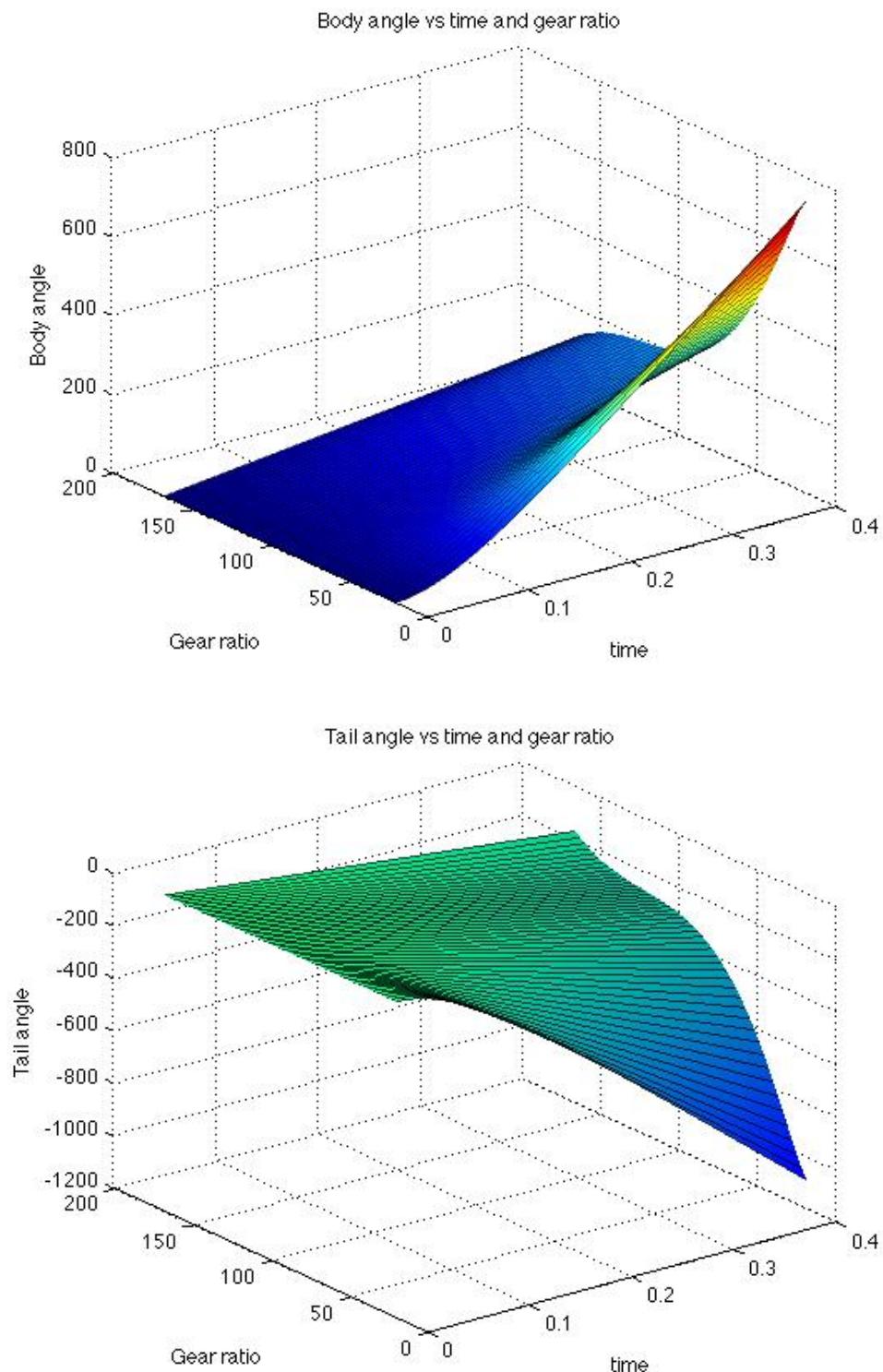


Figure 37: Simulation prediction of motor 2 – body and tail angle against time and gear ratio

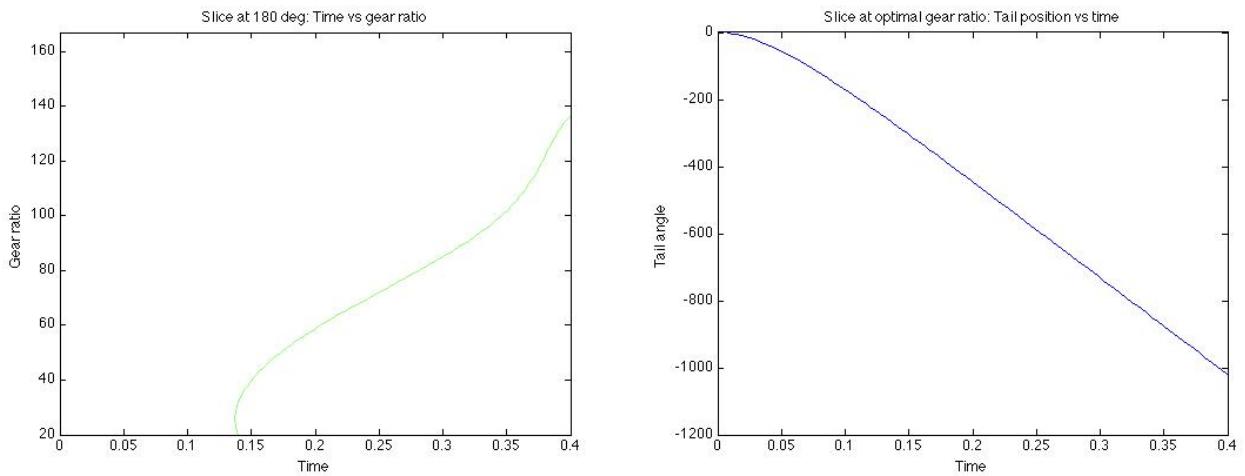


Figure 38: Motor 2, Left – Slice taken at 180 degrees to illustrate optimal gear ratio, Right - Tail angle versus time at optimal gear ratio

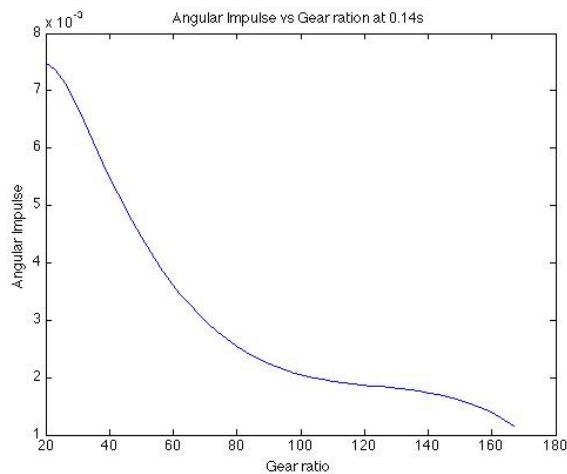


Figure 39: Motor 2 angular impulse vs gear ratio at 0.14s

The important deductions made from the second set of simulations are summarized Table 7.

Table 7: Simulation results for motor 2

Aspect	Value
Optimal gear ratio	26:1
Time to 180	0.137s
Approximate predicted time to 360	0.274s
Tail angle rotation	-270 degrees

5.7.4 Simulation of Motor 3

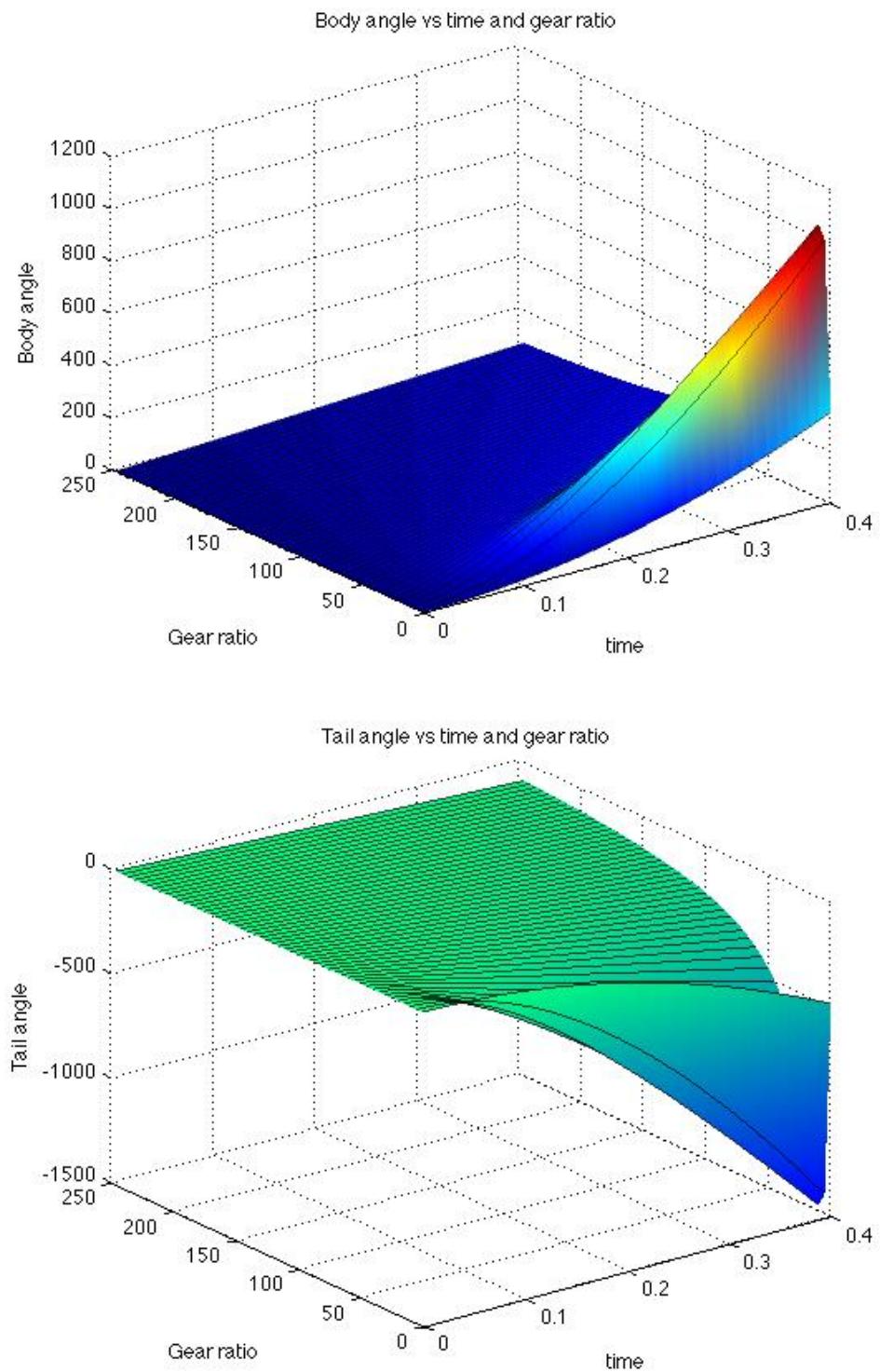


Figure 40: Simulation of motor 3 – body and tail angle against time and gear ratio

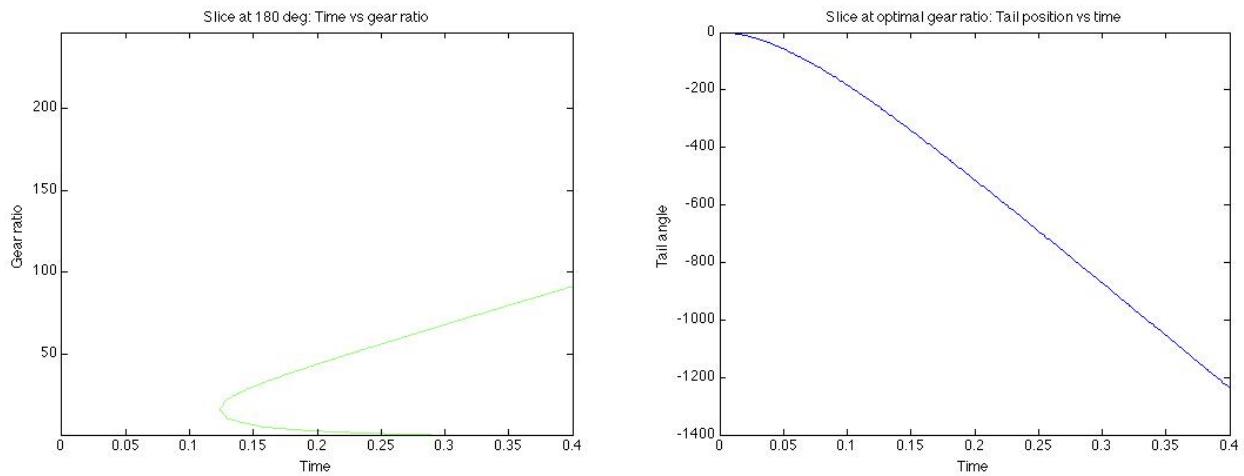


Figure 41: Motor 3, Left – Slice taken at 180 degrees to illustrate optimal gear ratio, Right - Tail angle versus time at optimal gear ratio

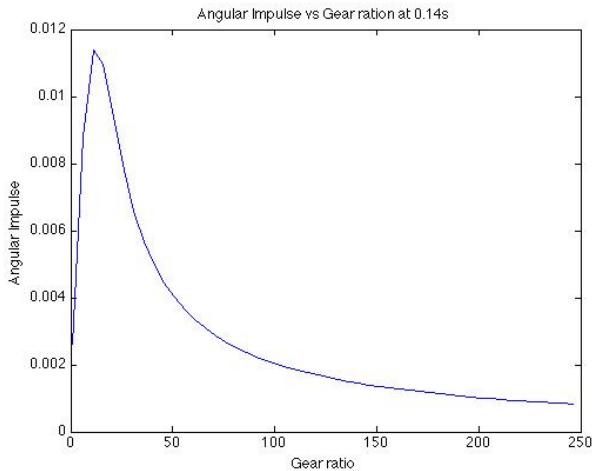


Figure 42: Motor 3 angular impulse vs gear ratio at 0.14s

The important deductions made from the third set of simulations are summarized

Table 8.

Table 8: Simulation results for motor 3

Aspect	Value
Optimal gear ratio	26:1
Time to 180	0.124s
Approximate predicted time to 360	0.248s
Tail angle rotation	-260 degrees

5.8 Motor Selection and Motivation

At each of their specific optimal gear ratios, the motor simulations predict that the motors will achieve the stunt in a reasonably similar time – the difference being 0.026s. Hence this was not a significant differentiating factor.

The first motor has a less severe slope of body rotation around the optimal ratio, and this made it the most robust choice. If operating conditions changed, or assumptions in the model affect the true operation more than anticipated, this motor was likely to perform better than its substitutes. The first motor was also the lightest, and had the lowest cost, which further endorsed it as the best choice.

The optimal gear ratio for this motor as predicted by the simulation is 54:1, the nearest commercial options to this were 50:1 (below) and 75:1 (above). Obviously 50:1 is closer to the optimal ratio, but 75:1 was selected. This decision was made as if the body or tail's rotational inertia increased, the optimal ratio shifted upward, and if less rotation of the tail was required, its rotational inertia would need to have been made larger. The specifications of the final motor selected are shown in *Table 9*.

Table 9: Selected motor specification

Motor type	High power carbon brushed micro metal gearbox motor
Gear ratio	75.81:1
Nominal voltage	6 V
Stall torque	0.155 Nm
Free running speed	400 rpm
Stall current	1600 mA
Nominal current	120 mA
Dimensions	10 x 12 x 26 mm
Mass	9.5 g
Shaft diameter	3 mm

6. System Design

This chapter describes all forms of design undertaken in the project, including hardware electronics, mechanical, and software designs. It also gives insight into the fabrication process where applicable.

6.1 Circuit Design

The platform requires four electronic systems to interface with the microcontroller in order to perform the stunt. They perform the functions of sensing when the platform leaves the ramp, a driver circuit to drive both the tail motor and rear motor, an encoder circuit to sense motor shaft velocity and a communication circuit.

6.1.1 Launch Detection Circuit

In order to actuate the tail as early as possible, it is required to sense when the platform leaves the ramp. This was achieved using an infrared distance sensing circuit. An infrared emitter diode drove a PWM signal through an npn transistor, and an IR receiver module received a reflected signal. The module used is the TSOP 38438, which operates at 38 kHz with embedded filters to cancel noise and stray ambient light. It outputs a binary high when not exposed to infrared and a low when exposed [50]. This output is compatible with a microcontroller and was fed directly into the on-board microcontroller. A potentiometer mounted in series with the emitter diode provided adjustable power and thus detection range. The circuit design is illustrated in Figure 43.

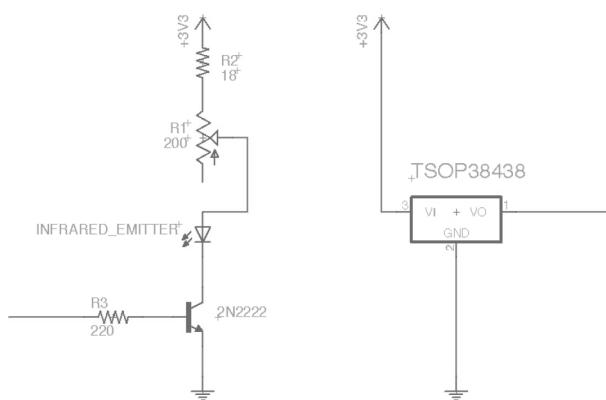


Figure 43: IR Launch detection circuit diagram

6.1.2 Driver Circuit

The tail motor required a bidirectional driver circuit, while the rear motor required only a unidirectional drive circuit. The tail motor was thus supplied by a standard H-bridge, and the rear motor by a half H bridge. In order to achieve the amplification in the small space required from the compact mechanical structure of the car, a package IC, the L293NE, was selected for the function. The L293NE's key specifications are displayed in *Table 10*.

Table 10: L293NE driver circuit specifications [51]

Specification	Value
Supply Voltage	2.3 V – 36 V
Peak Current	± 2 A
Continuous Current	± 1 A
Input/Output Logic level	2.3 V – 7 V

The driver circuit selected was perfectly matched to this application in terms of input and output power, voltage and current levels. Notably the peak output current was capable of supporting the stall current of the tail motor, which made maximum angular impulse possible at the first instance of actuation. Figure 44 shows a block diagram of the driver circuit.

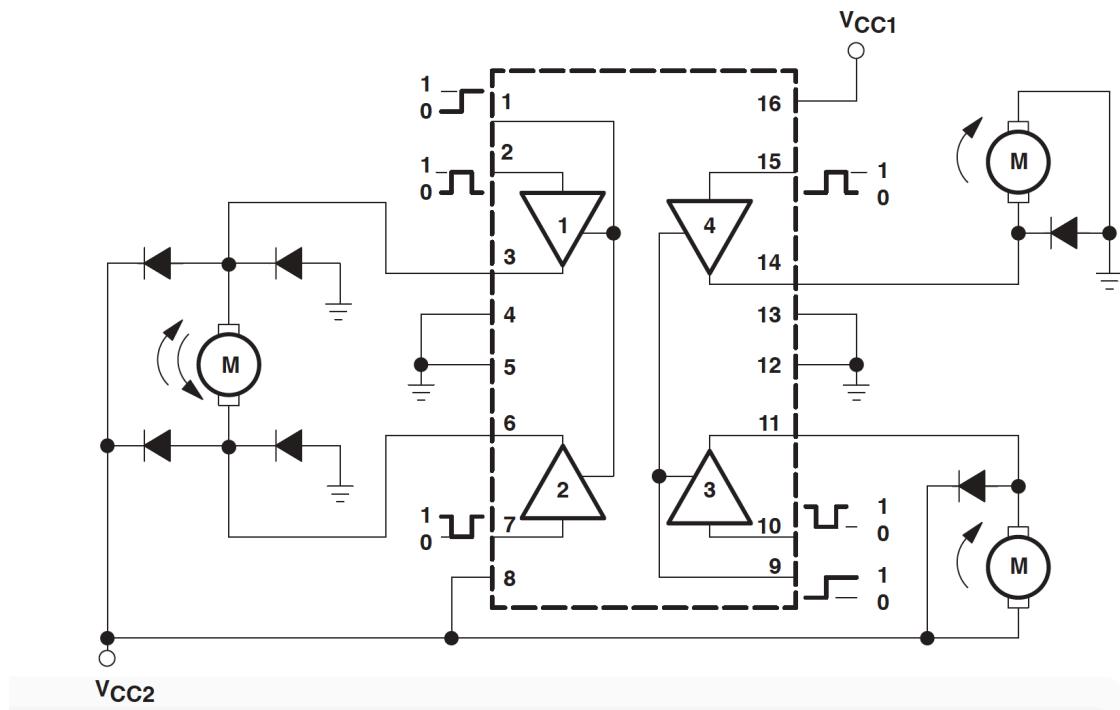


Figure 44: Driver circuit block diagram – the diagram illustrates the effective function of the L293NE IC [51]

6.1.3 Encoder Circuit

The motor encoder selected for this application was the magnetic, hall effect 12 CPR encoder manufactured by Pololu. The package information stated that the encoder was specifically designed for the selected motor, thus its performance was guaranteed. Beyond this, the encoder output was a 50% duty cycle PWM signal at a frequency proportional to the shaft speed that could directly interface with a microcontroller. An image of the encoder is provided in the left of Figure 45.

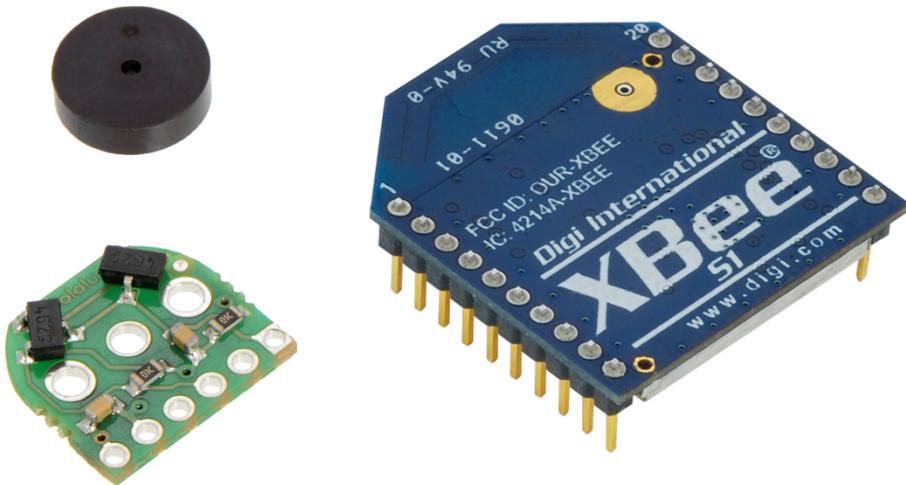


Figure 45: Left - Pololu 12 CPR hall effect magnetic shaft encoder [52], Right - XBEE communication module [53]

6.1.4 Communication Modules

Communication was required to link the platform with a base station (computer) for the purposes of data logging (used to capture the sensor readings). The application of the designed controllers was performed on-board the platform by the MCU, but the base station logged data for model verification and provided a facility to abort any operation in the case of malfunction.

The communication modules selected for the stunt were XBEE S2 RF modules. These modules operate at 2.4 GHz, at 0 dBm (1 mW) providing a line of sight range of 90 m, more than adequate for the application [54]. The module is shown in the right of Figure 45.

6.1.5 Hardware Systems Block Diagram

Figure 46 is an illustration of the hardware electronic systems and circuits designed for the robot. The diagram was a useful visual method of seeing how the systems interfaced and was an integral aid in the fabrication of all the electronic designs.

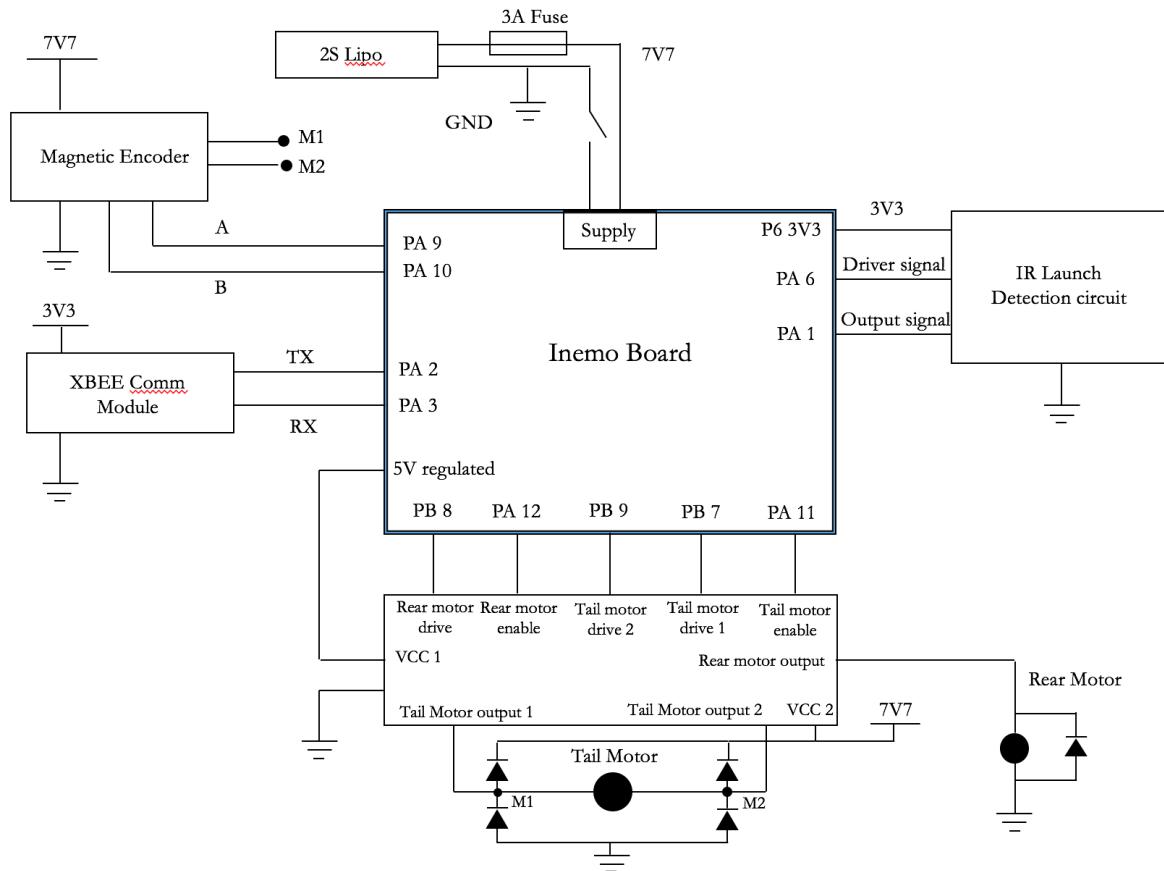


Figure 46: Hardware systems block diagram showing all the hardware electronics interfacing with the MCU

6.2 Software Design

The software required for this project took two forms, the C code implemented on the MCU on-board the robot, and the C# code used to create an application that could display and log data received from the car.

6.2.1 Software Requirements

The software that was written and used in this project was not very complex, but certain crucial requirements were set before software design was undertaken. The software needed to fulfil the following criteria:

- i. It must have robust fail safe systems in place to prevent any possible malfunction or exceptional circumstance from arising during testing.
- ii. It must be modular to ensure there is ease of use and that each module can be tested for functionality individually.
- iii. It must be efficient and burden the processor as minimally as possible.
- iv. It must be well structured and logical to ensure readability, and for effective distribution to future users and colleagues.

6.2.2 Software Accreditation

The C code that was run on the MCU on-board the robot was a hybrid of original code written for this project, and code written by Mr Callen Fisher. Source files from Mr Fisher's Msc [64]] that contain code that initializes and interfaces with all the Inemo sensors, as well as code that performs the communication, were used. Small adaptations were made to the setup of the sensors, and the type casting of the data read from the sensors.

The C# application used to display and log data during testing was derivative of numerous sources. The code was originally written for the Dima platform by Dr A Patel, and a team, of which large contributors were Justin Coetsier and James Gowans. Thereafter it was adapted by Ms Stacey Shield for her undergraduate final year project. It was again slightly manipulated for this project. The software provided by the various contributors mentioned above was invaluable in this project, as it saved time during the design phase.

6.2.3 Original MCU C Code

Original C code was written to interface with all the peripherals and perform the control action for the stunt. The code written called the MCU to perform the following functions:

- i. Implement and run a safety timer*.
- ii. Generate PWM signals for the motor drive circuits and the IR launch detect circuit.
- iii. Service a rising edge line interrupt for the launch detect circuit.
- iv. Perform input capture to acquire motor encoder data.
- v. Operate an indicator LED

* The safety time was given overriding interrupt priority and is used to deactivate the rear and tail motor drives in the case of any malfunction, such as a false launch detection scenario. The operation of the safety timer is explained in 6.2.4.

6.2.4 C Code Implementation

The most effective way to communicate the code operation is by means of a programming flow diagram. Such diagrams illustrate the flow of control in software; since the C code makes use of interrupts, which are essentially breaks in control flow, three different diagrams are used to explain the code.

Figure 47 illustrates the main loop operation, which performed all initializations, started transmitting data, and waited for the safety timer to count 15 seconds before starting the stunt. This gave ample time to ensure all systems are in working order. Figure 48 shows how the safety timer operated, and how it initiated the jump (by enabling the rear drive), and how it disabled the rear drive if the jump was not hit in reasonable time (if the robot missed the ramp).

Figure 49 explains how the launch detect interrupt functioned, and how it initiated tail actuation and started the controller action. It also reset the safety timer variable, which would, when a launch had been detected, time the controller and disable the control if the stunt had not been completed in a fair amount of time (0.8 seconds, given that most jumps were complete in approximately 0.4 seconds).

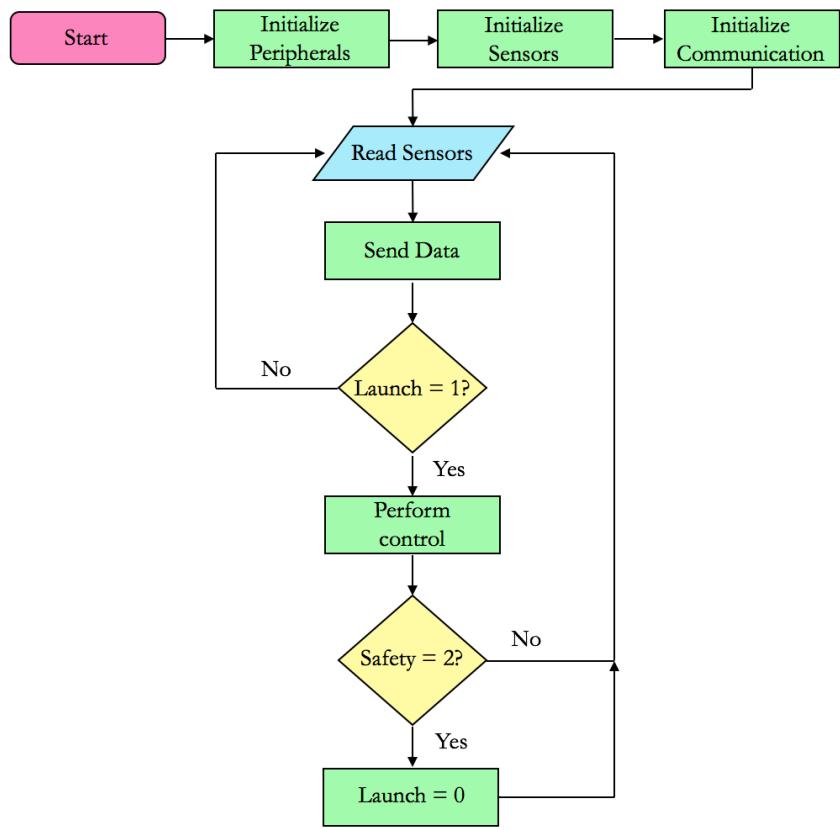


Figure 47: Main software loop flow diagram showing how the stunt was initiated and how control was distributed during the stunt

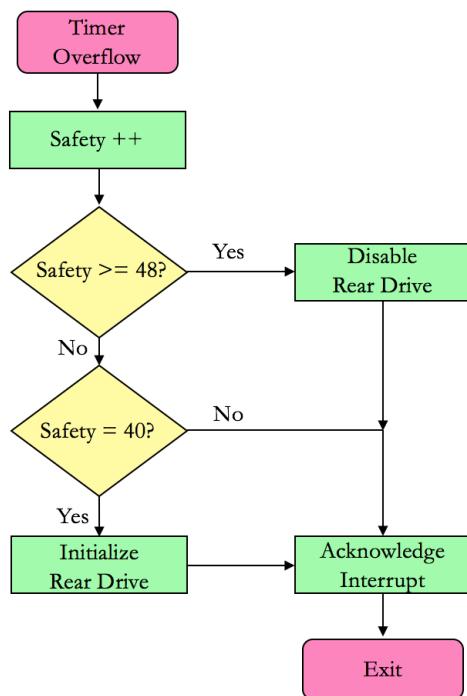


Figure 48: Software interrupt 1 flow diagram showing the operation of the safety timer at different times during the stunt

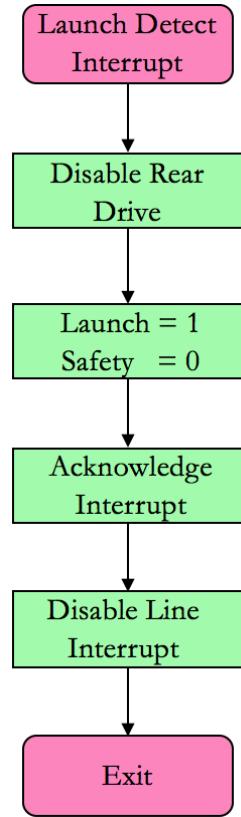


Figure 49: Software interrupt 2 flow diagram – showing how the launch detection software operated

6.3 Microcontroller Selection

The microcontroller needed to be matched to the needs of the hardware, having the correct number of peripherals; it also needed to be able to operate at a speed sufficient for the application and be able to perform the control calculations timeously. The needs of the hardware and software are summarized by the following Table 11.

Table 11: Hardware requirements of the microcontroller

Feature/Peripheral	Hardware Requirement
GPIO ports	7
General purpose timers	3
Channels per timer (minimum)	3
Advanced timers	1
UART communications (Tx and Rx)	1
Line Interrupts	1
Timer Interrupts	1

Based on these criteria, the microcontroller selected for the application was the high density performance line ARM-based microcontroller, the STM32F103RE. The selected package included an Inemo-M1 system on-board module which featured a 9-axis sensing system. The features of the integrated package are summarized in Table 12.

Table 12: Features of the STM32F103RE on the Inemo package

Feature	Description/Value
Supply	3.6 – 6 V
Sensors:	
	3-axis accelerometer
	3-axis gyroscope
	3-axis magnetometer
	Temperature sensor
Interfaces	CAN, UART, SPI, I ² C
ADC	8 channels
General purpose timers	4
Channels per timer	4
Advanced timers	2
USART modules	3
DMA controllers	2
External crystal oscillator	72 MHz

Table 12 shows that the selected microcontroller sensory package meets the requirements set by the hardware and software designs and is judged to operate fast enough for the application.

The part came as a surface mount package and was mounted on a PCB designed by Callen Fisher. The Altium PCB layout of the design is shown in Figure 50.

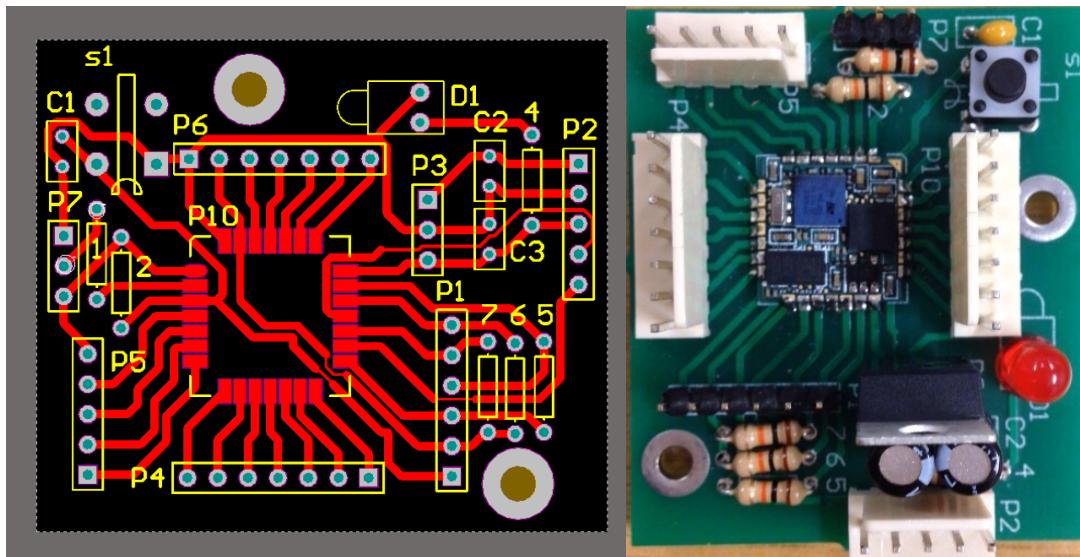


Figure 50: Left - STM on the INEMO PCB mount designed by Callen Fisher, Right - The populated PCB board

6.4 Mechanical Design

The mechanical design aspects of the project were three-fold. Initially the platform to be fitted to the car was designed; this needed to support all the hardware, the MCU PCB, the tail motor and tail, the battery and the rear drive motor. Secondly, a tail was designed, and thirdly a test rig was designed and constructed to provide a safe environment to test the system in various modes of operation.

6.4.1 The RC Car

The robotic platform conceptualized to complete the stunt was based on a car adapted to have a tail, similar to the Dima and flipbot systems [27][34]. The University of Cape Town provided an RC car for this purpose. This car was stripped entirely; all unnecessary components were removed to save weight and all that was retained was the rear motor and chassis.

6.4.2 The Base Platform

A reformed body was designed using SolidWorks that was suitable to mount the necessary components and hardware for the stunt onto the car. The part assembly is shown in Figure 51 and Figure 52.

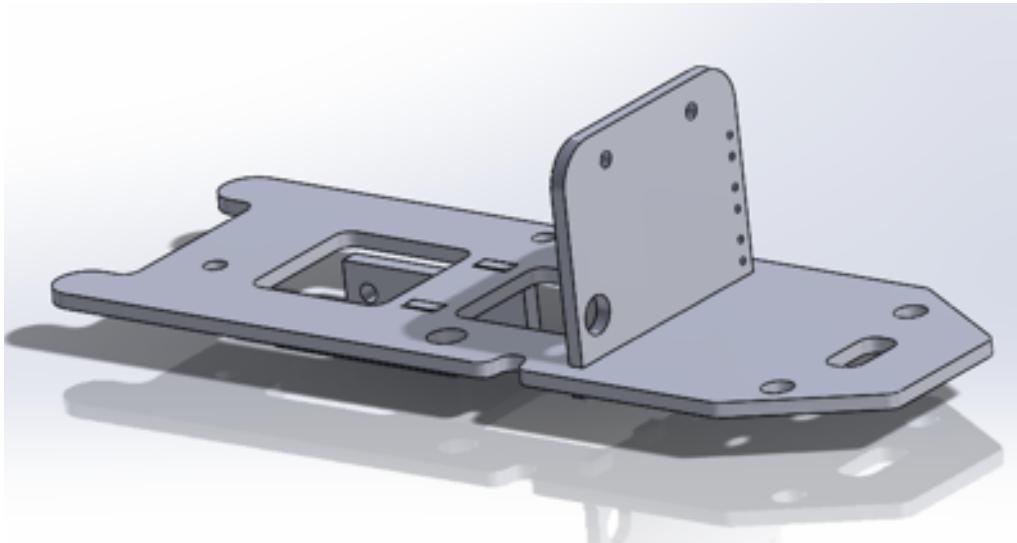


Figure 51: General view of SolidWorks model of the base platform designed for the robot

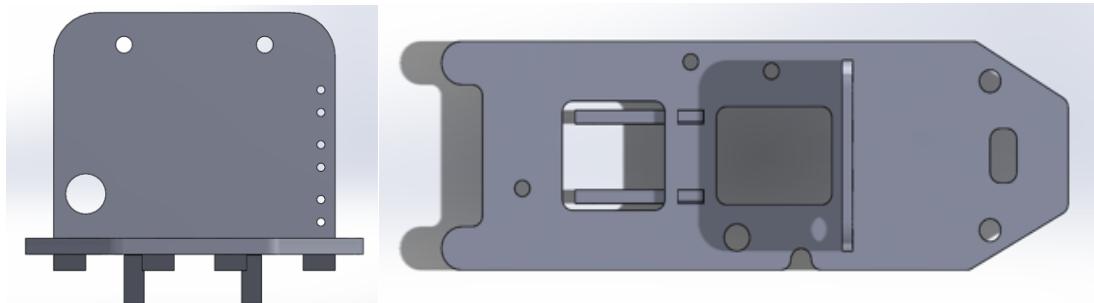


Figure 52: Left - Front view of SolidWorks model, Right – Top view of the SolidWorks model

Careful consideration was given to where the battery, fuse, switch, circuit boards, sensors and the motors were to be mounted. The primary factors affecting these decisions were firstly the point of rotation of the tail, desired to be as close to the centre of mass as possible, and secondly the position of the centre of mass, desired to be as spatially central as possible.

It was critical to minimize the weight of the platform; to achieve this, holes were cut into the base plate, reducing the mass of perspex of the plate. Other small holes were strategically placed for efficient wiring, fixing the plate to the chassis, as well as for switch and LED mounting. Consideration was given to the torsional strain that the body will be subjected to during the stunt.

The SolidWorks model was realized by laser cutting the template into 3 mm perspex. A plain view of the cut Perspex with only the Inemo and driver circuit in is shown in Figure 53.

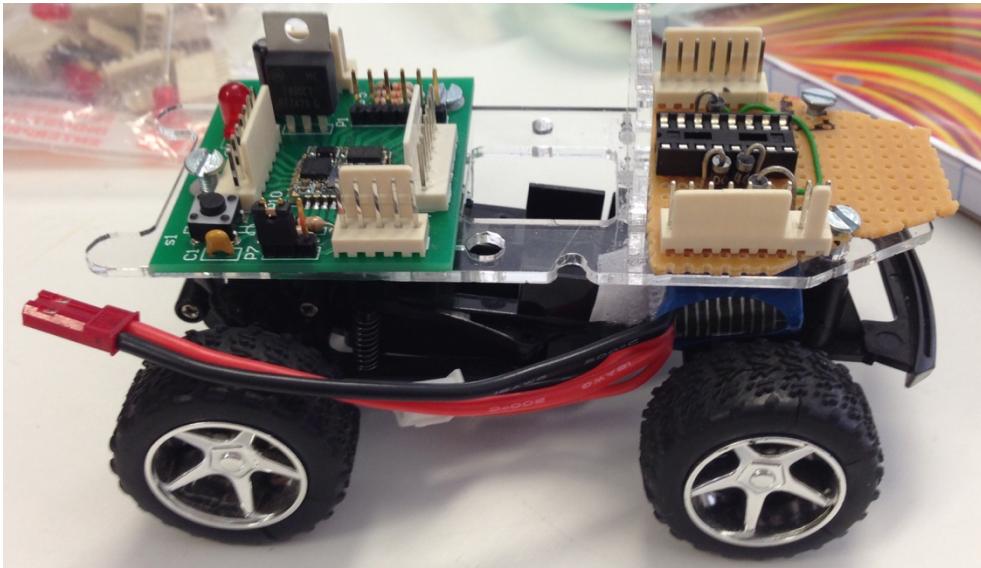


Figure 53: Perspex base plate mounted on the car with Inemo board, driver circuit and battery

6.4.3 Tail Design

The initial simulations were conducted with a tail design based on current literature. Subsequent to the initial simulations, the same model was utilized to analyse the effects of various tail lengths and masses. It was found that the simulation time to 180 degrees (the standard used in this project) decreased with increasing tail rotational inertia to a maximum operating point which corresponded to a tail with greater inertia than the standard one body length and 10% mass tail. This design was judged not to be feasible given that when hitting a ramp, long tails would hit the ground behind the robot, and the benefit gained by increasing tail length or mass was not substantial.

Thus the tail was designed based closely on standards set out in the literature, with the length slightly increased for the added performance, resulting in a length of 15 cm, and end point mass of 15 g. A model of this is shown in Figure 54. The tail was mounted onto the motor shaft using a 3D printed mount that was designed using SolidWorks.



Figure 54: Concept diagram of the tail design

6.4.4 Test Rig Design

The test rig envisioned for the project would provide a plate the could rotate in the horizontal plane. This would allow step tests to be performed on the system, as well as controller designs to be tested before jump tests. This safe testing environment would immensely increase the probability of the system performing a successful stunt. The concept test rig design had a rectangular, flat base plate to support the system under any moment generated by the car, a rod extending upward, and a bearing fitted to the rod supporting the rotating plate. This is depicted by Figure 54.

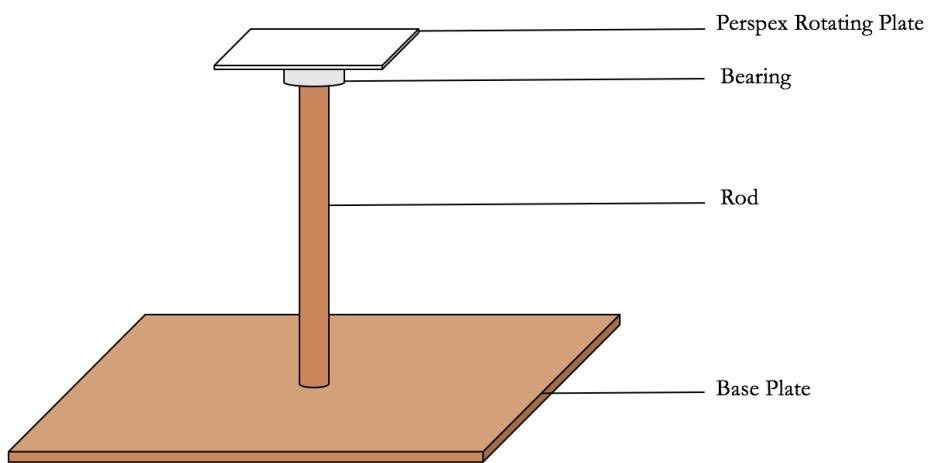


Figure 55: Diagram of the test rig concept

The test rig was fabricated using a wooden base, a dial rod, a high quality, low friction bearing, and a laser cut perspex plate. Photographs of the rig are shown in Figure 56 and Figure 57.

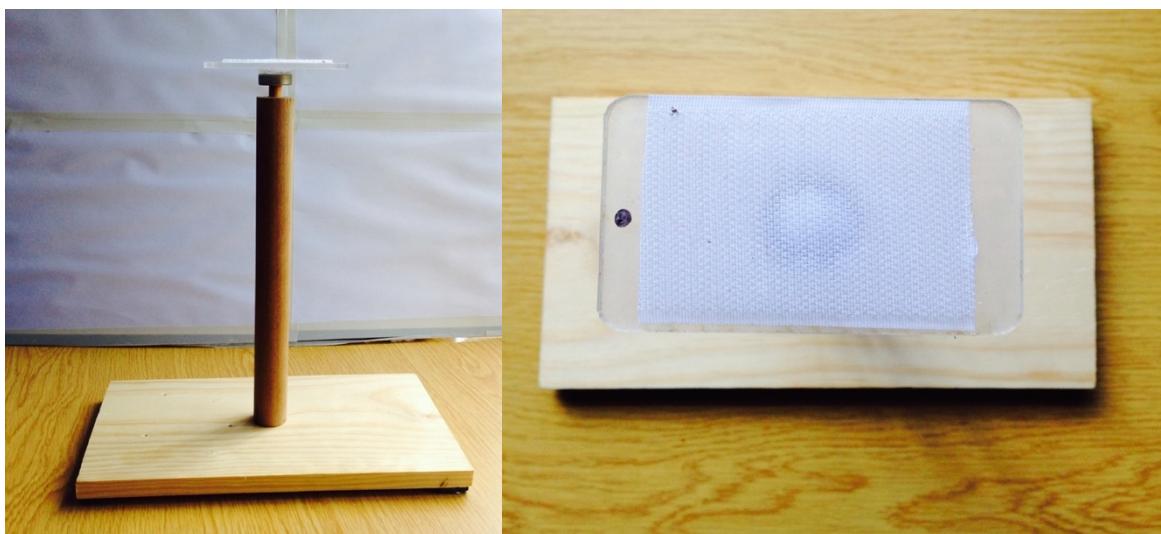


Figure 56: Left - Side view of the fabricated test rig, Right – Top view of the fabricated test rig



Figure 57: View of the fabricated test rig from a general angle

6.5 Design Realization

All the physical designs were constructed and fabricated in the UCT labs. The final product is compact, robust and modular. If any part should be damaged in testing, the system was designed and built in such a way that it can be relatively easily replaced (given fair time). Photographs of the robot are shown from various angles in Figure 58 to Figure 60.

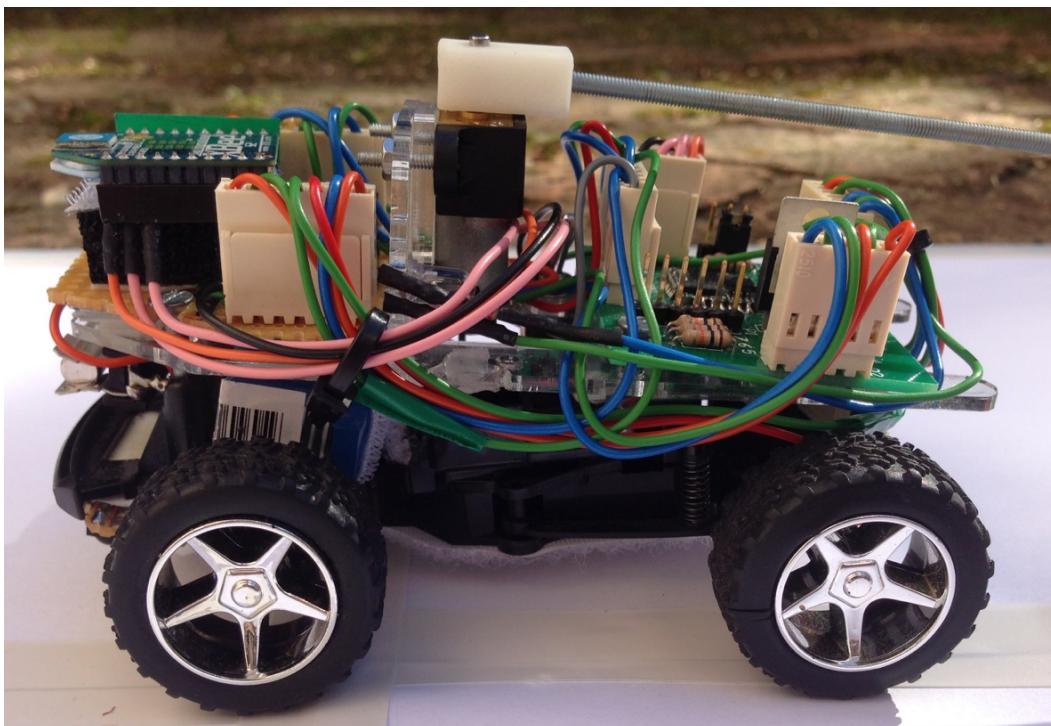


Figure 58: Side view of the TAYObot – fully constructed and integrated system showing the circuitry and battery placement

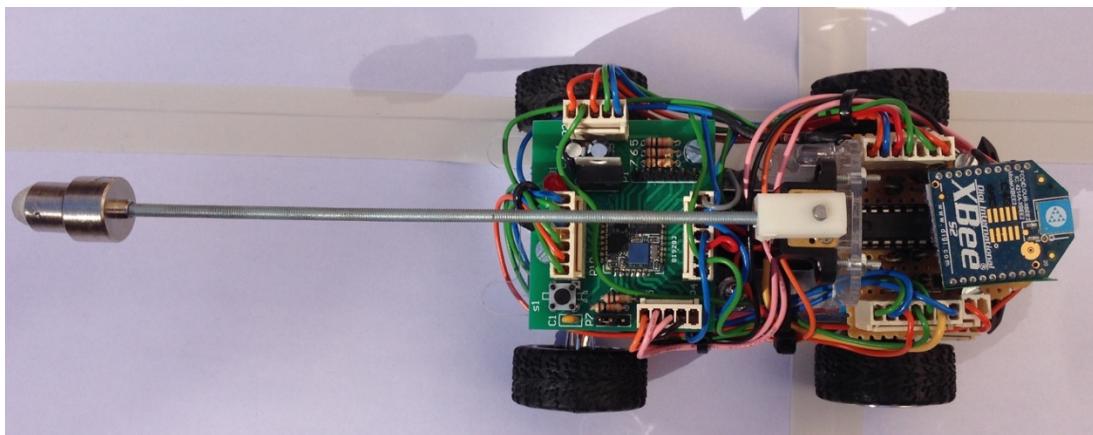


Figure 59: Top view of the TAYObot – fully constructed and integrated system

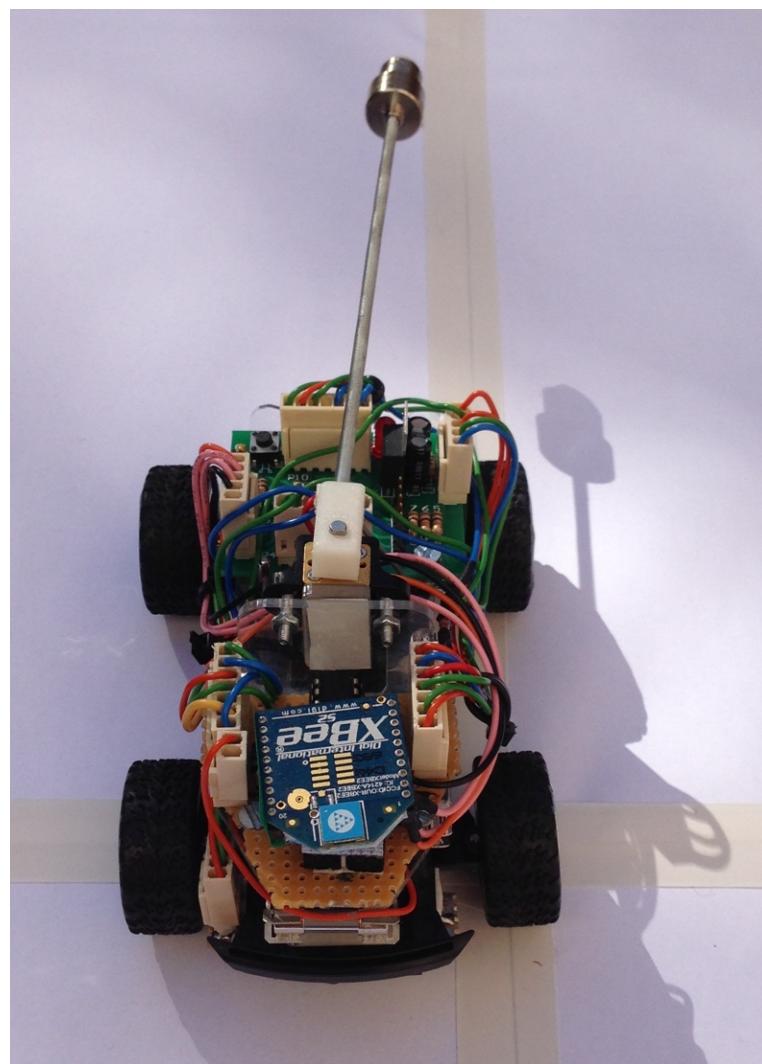


Figure 60: General front view of the TAYObot – fully constructed and integrated system

The robot was called the TAYObot (Tail Actuated Yaw Orientation bot), drawing inspiration from the TAYLRoACH [33].

7. Parameter Determination and Model Verification

This chapter explains the procedure followed in determining all the parameters required to populate the models derived in previous chapters. It then goes on to present step test data and compare the result to the simulations of the derived models in an attempt to validate the models.

7.1 Calculation of Moments of Inertia

7.1.1 Body Mass Model and Specifications

The moment of inertia for the body, I_B , and the tail, I_T , must be quantified. The body will be modelled as a uniform block, as displayed in Figure 61, and thereafter the moment of inertia reduced by an approximate reduction factor to account for the more centralized mass distribution. The key dimensions affecting the rotational inertia about the horizontal plane are the length (L) and width (W), the height has no effect in this plane. The mass used in the calculation is derived in *Table 13*.

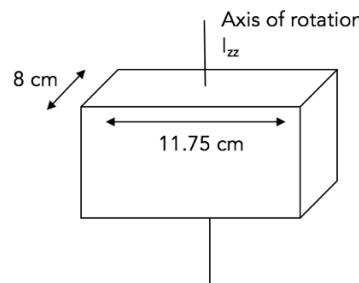


Figure 61: Mass distribution model of the body

Table 13: Mass breakdown of the of the fully constructed and integrated system

Item	Mass (g)
Inemo and communication boards	40
2S 350mAh Lipo battery	25
Motor	10
Platform	80
Total	155

7.2.2 Moment of Inertia about I_{zz}

The moment of inertial for a block about the vertical axis was calculated using:

$$I_B = \frac{m}{12} (L^2 + W^2) \quad [35] (41)$$

Subbing in values:

$$I_B = \frac{0.155}{12} (0.1175^2 + 0.08^2)$$

$$I_B = 2.61 \times 10^{-4} \text{ kg} \cdot \text{m}^2$$

Approximate reduction factor: 0.75 (based on positioning of significant masses). Therefore:

$$I'_B = 2.61 \times 10^{-4} \times 0.75$$

$$I'_B = 1.95 \times 10^{-4} \text{ kg} \cdot \text{m}^2$$

7.2.3 Moment of Inertia of the Tail:

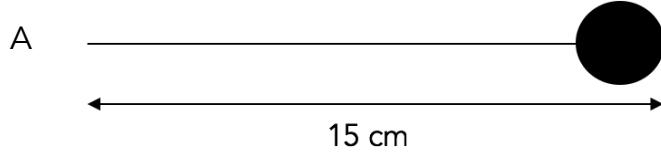


Figure 62: Tail rotational inertia model

Length: $R = 15 \text{ cm}$

Mass = 15 g

The moment of inertial of the tail was modelled as a point mass at the end of a massless link. The moment of inertia about point A is given by formula 42:

$$I_T = mR^2 \quad [56] (42)$$

Subbing in values:

$$I_T = 0.015 \times 0.15^2$$

$$I_T = 3.8 \times 10^{-4} \text{ kg} \cdot \text{m}^2$$

7.2 Motor Parameter Determination

Motor tests were conducted to determine the motor parameters of armature resistance, armature inductance, motor constants, K_e , and K_t , and static and viscous friction.

7.2.1 Measured Motor Parameters

Armature resistance and inductance were measured using a digital multimeter and LCR multimeter. The motor resistance was found to be 3.75Ω and the motor inductance was recorded as $570 \mu H$.

7.2.2 Experimentally Determined Motor Parameters

The voltage and torque constants can be calculated from equations 21 and 22 in conjunction with motor test data. The motor was driven directly off a variable voltage DC supply at different voltages, and the encoder output was displayed on an oscilloscope, providing the data in Table 21, which is shown in appendix C, section C2.

From Table 21 the back EMF can be calculated at each operating point using knowledge of the equivalent circuit. In steady state operation the voltage across the motor inductance is zero, and thus with knowledge of the current, armature resistance and terminal voltage, back EMF can be determined using equation 43.

$$e_a = V_t - iR_a \quad (43)$$

Graph of Back EMF versus motor shaft speed in rpm

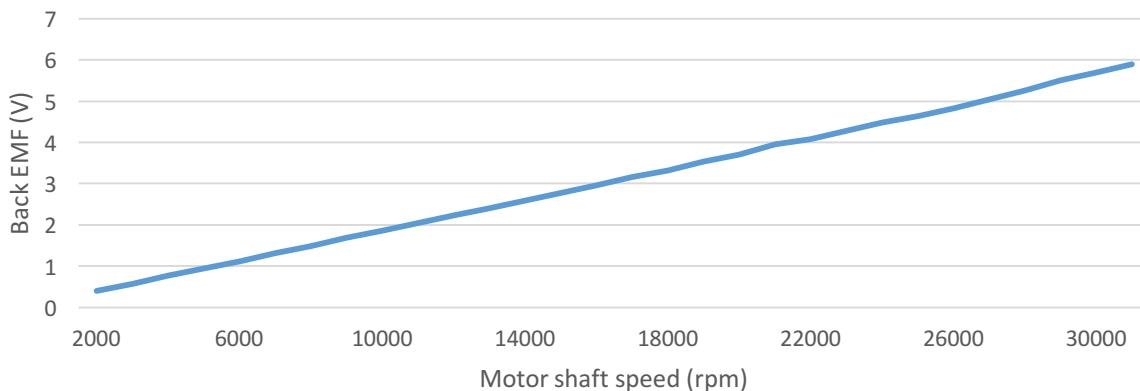


Figure 63: Graph of back EMF versus motor shaft speed graph

Subsequently if e_a is known for each speed, the motor voltage constant, K_e , can be determined.

$$K_e = \frac{e_a}{\omega_m} \quad (44)$$

Computing this ratio at each shaft speed gives a mean motor voltage constant of:

$$K_e = 1,78 \times 10^{-3} \text{ V/rad.s}^{-1}$$

In the case of a brushed DC motor, the motor voltage constant and the motor torque constant have the same magnitude (when expressed in SI units) [25].

$$K_t = K_e = 1,78 \times 10^{-3} \text{ Nm/A}$$

7.3 Motor Friction Determination

As discussed in chapter 3, the motor exhibits three types of friction, static, dynamic and viscous friction [17]. The friction specific to this motor can also be derived from the data in Table 21, as well as the torque constant.

7.3.1 Static Friction

Static friction can be determined from the torque required to just get the motor to turn, which is captured by the first data point. Resistive static frictional torque:

$$T_{sf} = K_t i \quad (45)$$

$$T_{sf} = 1.78 \times 10^{-3} \times 82 \times 10^{-3} = 1.46 \times 10^{-4} \text{ Nm}$$

This corresponds to a terminal voltage of 0.25 V, and obviously 0 rad.s⁻¹. Static friction was not used in the system models, since it would add complexity due to its nonlinear nature. Knowledge of its relative magnitude to motor stall torques gave an indication of the performance that could be expected from the system at angular velocities near zero.

7.3.2 Dynamic and Viscous Friction

The dynamic and viscous frictional effects can be determined from the same data set. Their effects will be approximated by a linear interpolation with an intercept set to zero to maintain linearity. The interpolation from Figure 64 gives the viscous frictional coefficient, B as 5×10^{-6} N.m.s.

Graph of motor torque against speed (blue) and zero intercept linear interpolation (red)

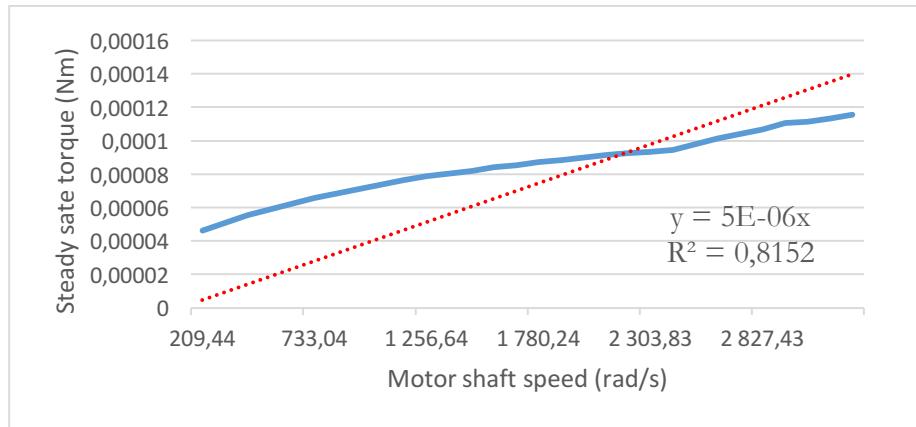


Figure 64: Graph of dynamic and viscous friction torque loading on the motor shaft and an interpolated trend line

7.4 Gearbox Efficiency

The gearbox efficiency was calculated from the motor parameters supplied by the manufacturer (stated in *Table 9*). The developed stall torque calculated using the stall current and torque constant compared to the rated stall torque provided the gearbox efficiency.

$$\eta_{gearbox} = \frac{T_{rated}}{T_{developed}} \quad (46)$$

$$\eta_{gearbox} = \frac{0.155}{NK_t i_{stall}} = \frac{0.155}{75 \times 1.78 \times 10^{-3} \times 1.6}$$

$$\eta_{gearbox} = 72.5\%$$

7.5 Summary of Parameters

A summary of the parameters of the system is provided in *Table 14*.

Table 14: System Parameters

Parameter	Symbol	Value	Unit
Body moment of inertia	I'_B	1.95×10^{-4}	kg.m^2
Tail moment of inertia	I_T	3.8×10^{-4}	kg.m^2
Armature resistance	R	3.75	Ω
Armature inductance	L	570	H
Motor constant	$K_e = K_t$	1.78×10^{-3}	$\text{V.rad}^{-1}.\text{s/Nm.A}^{-1}$
Motor shaft viscous friction	B	5×10^{-6}	Nm.s
Gearbox efficiency	η	72.5	%

7.6 Model Verification

In order to validate the theoretical and experimental parameters and models derived thus far, standard control step tests were conducted both on the motor (without the tail fitted), and the fully integrated robotic system on the testing rig.

7.6.1 Motor Step Tests

The motor was stepped at various voltages six times, and the encoder data was captured by the MCU and logged. The time constant and gain of each response was averaged to validate the motor viscous friction coefficient, B , as well as determine the shaft inertia (which was not provided by the manufacturer).

The motor model in isolation that maps torque to angular velocity exhibits a first order system response, and can be characterised by a first order transfer function, the form of which is shown in equation 47, if frictional nonlinearities are ignored. G represents the system gain, and τ the system time constant. Substituting the DC motor parameters into this form gives equation 48.

$$G(s) = \frac{G}{1+\tau s} \quad (47)$$

$$DC(s) = \frac{1/B}{1+s(\frac{Jm}{B})} \quad (48)$$

The linear DC motor transfer function – ignoring static friction – takes torque as an input and gives motor shaft angular velocity as an output.

7.6.2 Step Data Analysis

Six sets of data were obtained from the tests, and one such response for a 7V step is depicted in Figure 65. The time constant was measured as the average time taken to reach 63% of the final value. The gain was calculated by first determining steady state current, then steady state torque, and finally the quotient of angular velocity and torque; this value was also averaged using all the data.

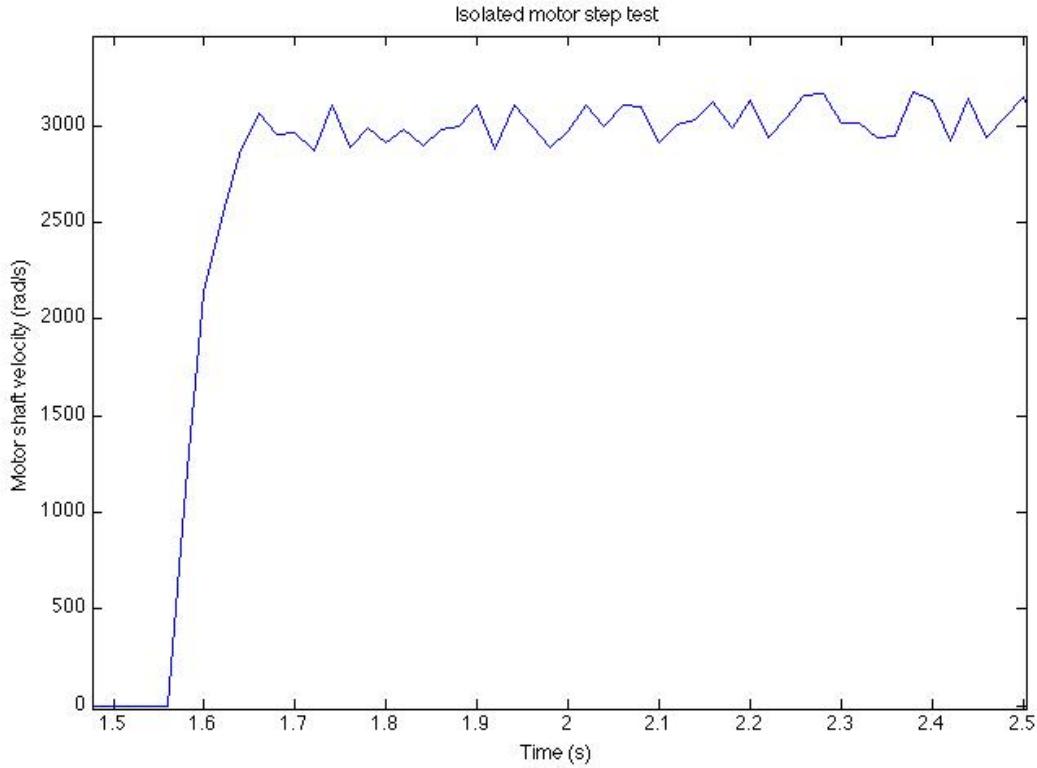


Figure 65: Motor step test at 7V

The gain calculation was performed using equation 43 and 22 as follows:

$$i = \frac{v_t - K_e \dot{\phi}}{R} \quad (49)$$

$$G = \frac{\dot{\phi}}{T} \quad (50)$$

The calculations resulted in a time constant of 0.0318 s, and a gain of $4.25 \times 10^6 \text{ rad.s}^{-1}/\text{Nm}$ for the system. These values were then converted to the parameters of the DC motor; the viscous friction coefficient, B, was calculated as $2.35 \times 10^{-7} \text{ Nm.s}$, and the shaft rotational inertia, J_m , was $7.48 \times 10^{-9} \text{ kg.m}^2$.

7.6.3 Observations About the Step Data Outcome

The gain value obtained from each step response varied quite largely with step amplitude. This was expected as a consequence of the nonlinear relationship between angular velocity and friction. The best linear interpolation has been presented above. The value attained from these tests for the viscous friction coefficient, B, was quite different than that determined from the steady state testing presented earlier in the chapter, again confirming the nonlinear nature of the measure.

The relatively crude linear approximation was justified by the aim of the project, which was position control of the body through the stunt, and not motor shaft position control. The focus of the project did not permit deep investigation into friction characterization and mitigation. The effects of friction were not expected to have a large adverse effect on the controller performance, and the controllers were intended to be designed to handle plant variation by means of robust control design.

7.6.4 Fully Integrated Model

The full system was characterized by the models derived in chapter 5, and these models were populated with the parameters determined in this chapter. Equation 38 and 39 showed the state differential equations describing the system. The final model was produced by substituting the parameters of *Table 15* into these state space matrices.

The factors differing in this table to *Table 14* are viscous friction, B (updated from the motor step tests), and motor shaft rotational inertia J_m (which has been added to referred tail inertia).

Table 15: Final System parameters

Parameter	Symbol	Value	Unit
Body inertia	I'_B	1.95×10^{-4}	$\text{kg} \cdot \text{m}^2$
Tail inertia	I_T	3.8×10^{-4}	$\text{kg} \cdot \text{m}^2$
Motor shaft inertia	J_m	7.48×10^{-4}	$\text{kg} \cdot \text{m}^2$
Armature resistance	R	3.75	Ω
Armature inductance	L	570	H
Motor constant	$K_e = K_t$	1.78×10^{-3}	$\text{V} \cdot \text{rad}^{-1} \cdot \text{s} / \text{Nm} \cdot \text{A}^{-1}$
Motor shaft viscous friction	B_m	2.35×10^{-7}	Nm.s
Gearbox efficiency	η	72.5	%

7.6.5 Final Model Simulations

A Matlab script was written to run step test simulations on the final system model. The simulated data could then be compared to physical step tests performed on the platform. Output functions of each state were obtained for a step input of 7.25V to be compared to the physical tests performed at the same step amplitude. These responses are shown in Figure 66 to Figure 68.

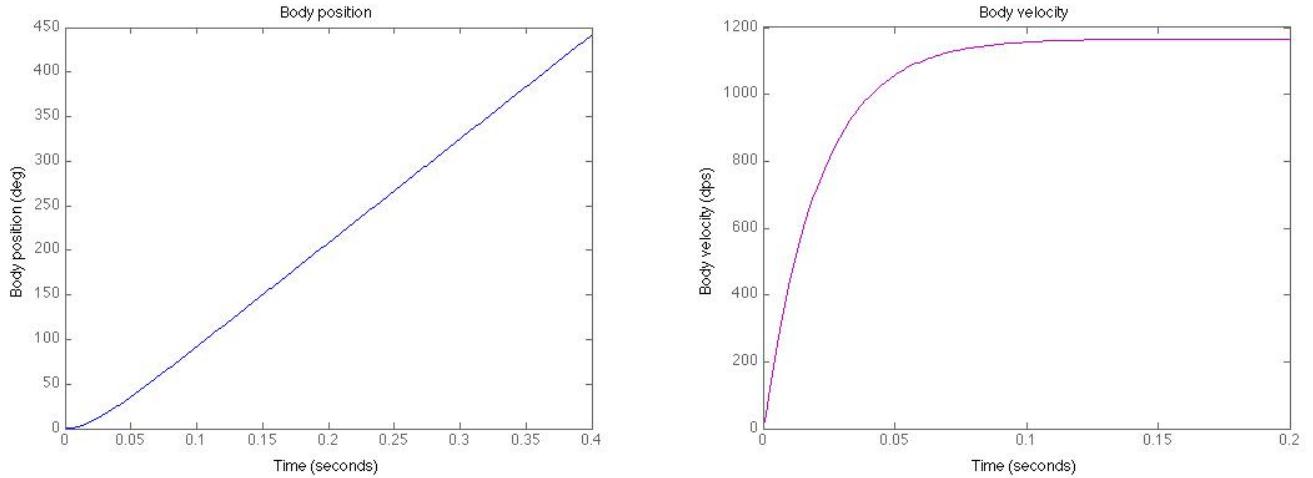


Figure 66: Left – Body position and, Right – velocity step response for a 7.25V step input

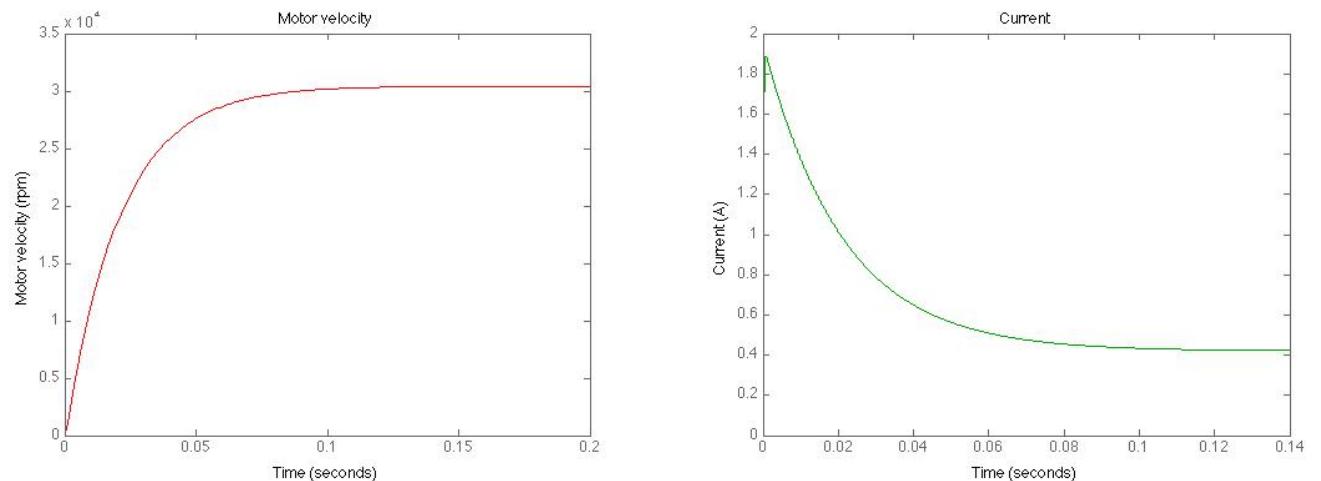


Figure 67: Left – motor shaft velocity and, Right – current step response for a 7.25V step input

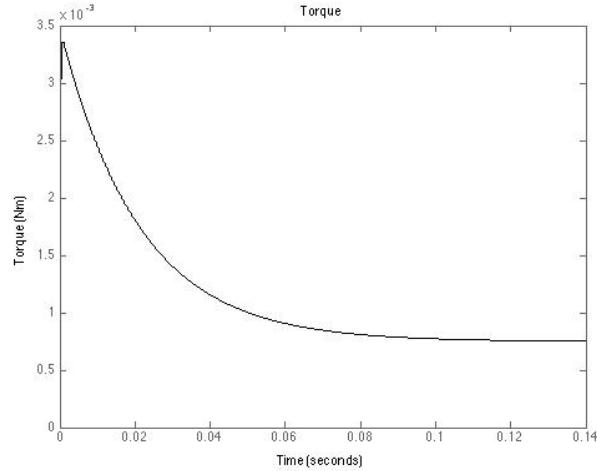


Figure 68: Shaft torque step response for a 7.25V step input

The noteworthy data from these step responses is summarized in Table 16.

Table 16: Summary of important simulation data

Data element	Value
Body position time to 180 degrees	0.176 s
Steady state body velocity	1170 dps
Steady state motor velocity	30 040 rpm
Stall current (initial value)	1.89 A

The time to 180 degrees was slightly slower than the model in chapter 5 predicts, but this is expected as the simulations in chapter 5 do not account for friction in the motor. The steady state velocity and stall current of this simulation matched scaled motor specifications (noting that the motor specifications are given at a nominal voltage of 6 V). The steady state body position can be compared to the gyroscope data obtained from the step tests conducted on the robot.

7.6.6 Full System Step Tests

The testing rig described in chapter 6 was used to perform a variety of tests on the system, including system step tests. The robot with the tail in place was fixed to the rig, and stepped, this setup is shown in Figure 69. The sensor readings were interpreted to verify the derived model.

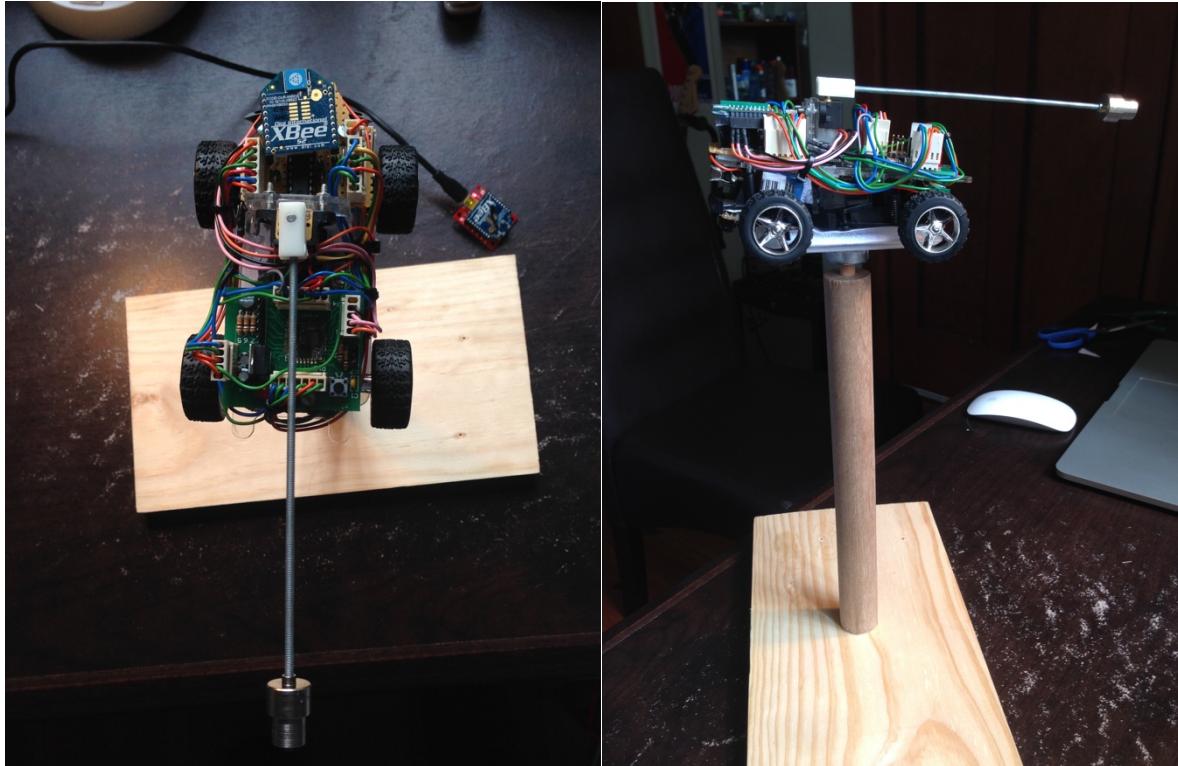


Figure 69: Robot mounted on the test rig for step tests

A step test was performed on the system, driving the tail motor with a PWM duty cycle of 90%, with a battery charge of 8.05V, thus implementing an input voltage step of 7.25V. The data was logged, and processed using a Matlab script. The test was performed 6 times with a high level repeatable accuracy. The tests were also video recorded; one sample of the video footage is on the disc attached and named “Full_system_step_test.mp4”.

The data obtained from the tests included the three gyroscope angular rate readings, the accelerometer readings, and motor encoder output. One set of data is presented from Figure 70 to Figure 73.

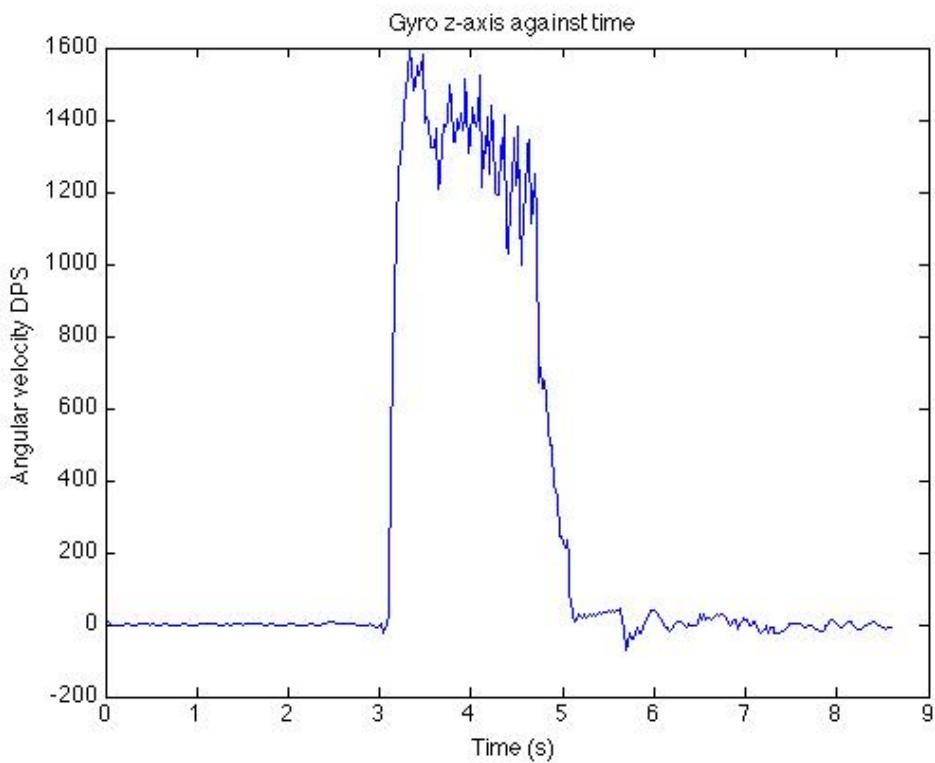


Figure 70: Z-axis gyroscope step test data of the system response from the test rig step test

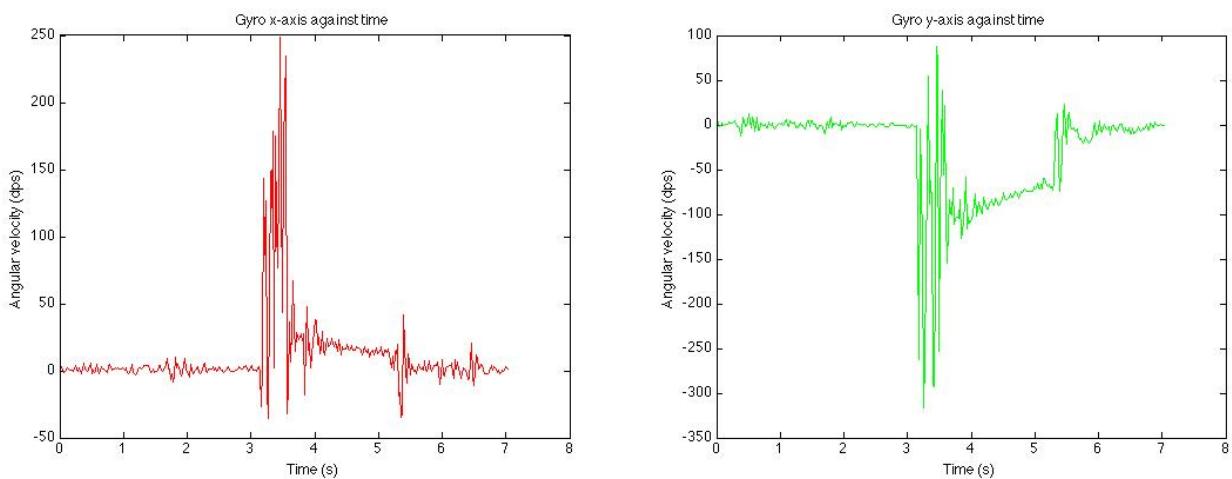


Figure 71: Left – X-axis, and, Right – Y-axis gyroscope step test data of the system response from the test rig step test

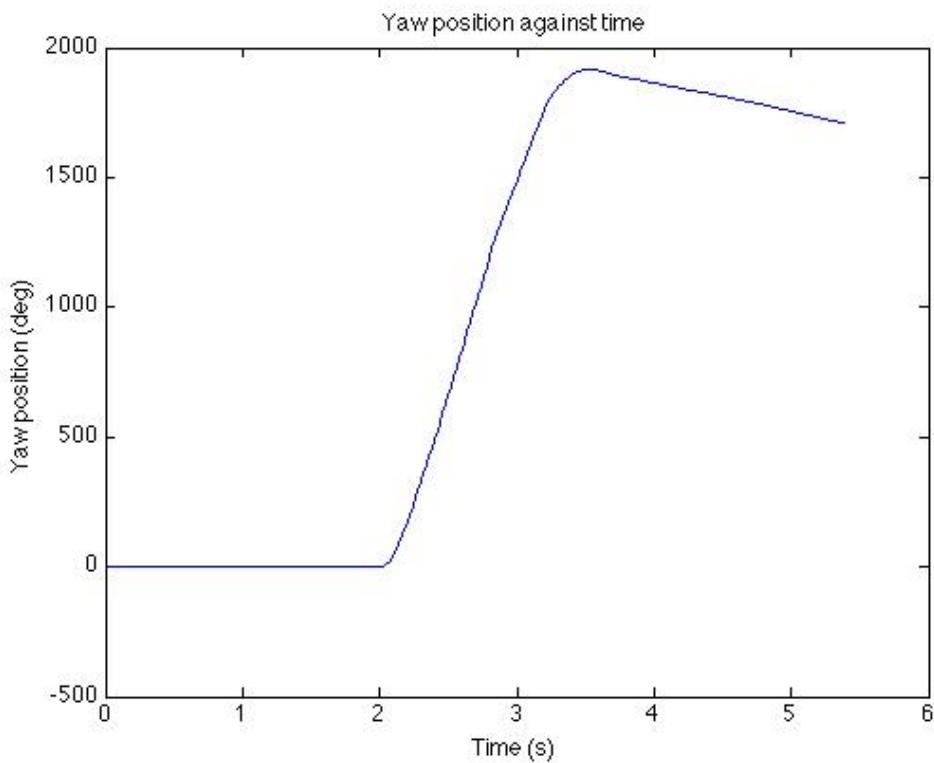


Figure 72: Body yaw position response from the step test conducted on the test rig

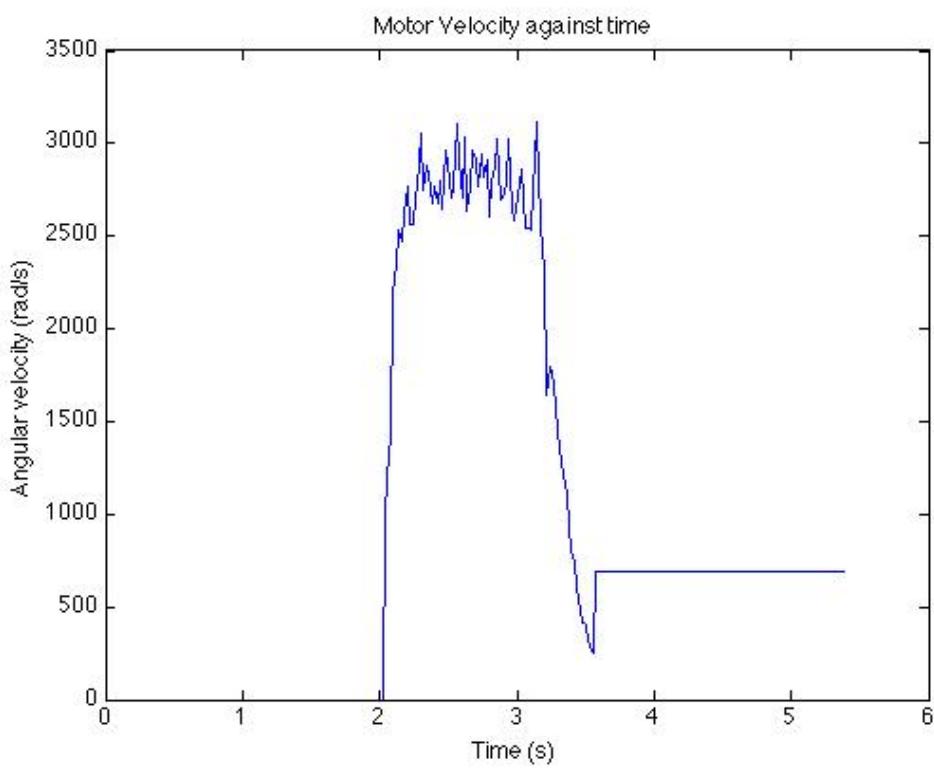


Figure 73: Motor velocity (rad/s) data from the step test conducted on the test rig

7.6.7 Step Data Analysis

The most important outcome of the step tests was model validation, and if each aspect of the model is considered, it can be seen that both the theoretical dynamic and circuit models, as well as their parameters derived in previous chapters are very accurate.

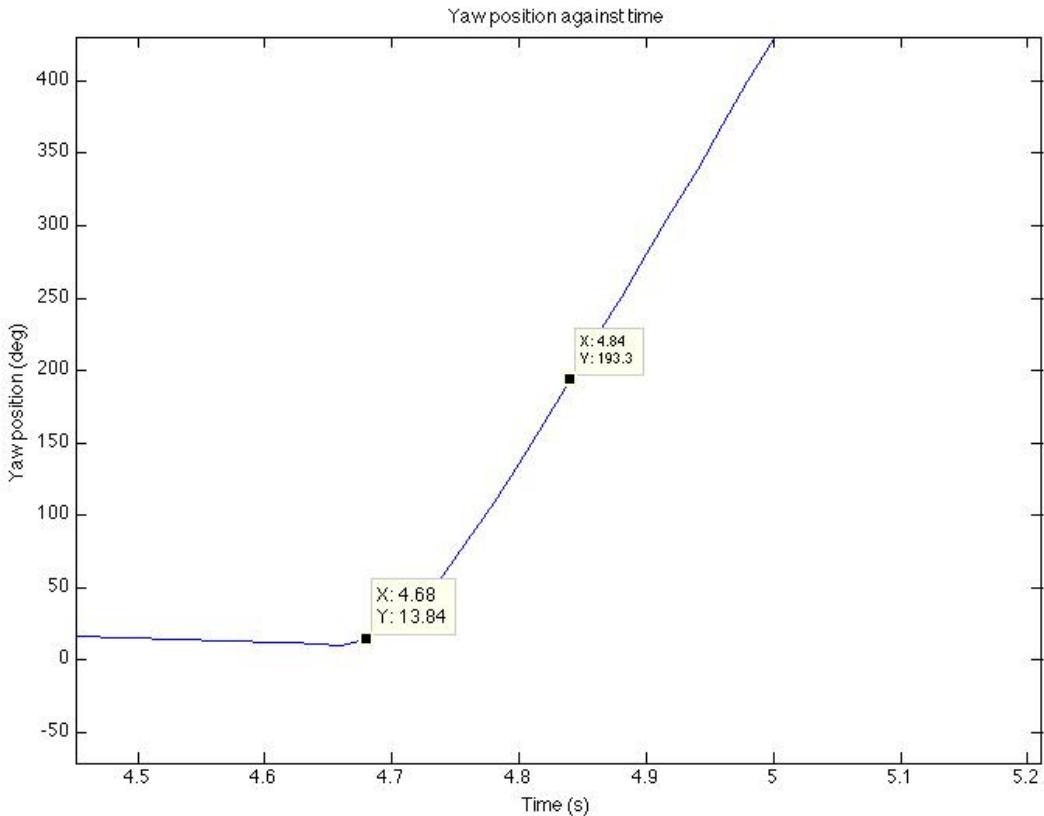


Figure 74: Step test body position response, the data points show the time taken to rotate through 180 degrees

Arguably the most significant aspect of the model was the mapping of input voltage to body position. Figure 74 shows that during the physical test, the platform reached 180 degrees (193.3 – 13.84) in 0.16 seconds. This is exceptionally close to the simulation prediction of 0.176 seconds (from Figure 66), and thus it can be concluded that this aspect of the model is accurate.

The motor velocity also followed the prediction relatively well, having a steady state velocity of approximately 3 000 rad/s (28 650 rpm), which is close to the predicted motor velocity of 3 040 rpm. The predicted body steady state velocity of 1170 dps was slightly lower than the average recorded in the step test, which was 1300 dps, but again this suggested a good model given the uncertainties in the sensors and assumptions in the modelling.

The gyroscope sensor data from the x-axis and the y-axis was shown because of the concern raised in chapter 6 about potential pitch and roll problems associated with the stunt. Even while fixed to the rig there were some severe pitch and roll angular rates experienced. These needed to be monitored during the stunt.

The data acquired from both the motor and full system step tests both aided in the improvement of, and proved the validity of the system models derived in this project. The models were judged to be sufficiently accurate to move forward to controller design.

8. Controller Design

This chapter explains the aims of the control system and discusses the motivation for the control methods selected. It goes on to describe the control design process that was followed, as well as show the simulated controllers and analysis thereof.

8.1 Control Aims

The control of the robot during this stunt needed to be exceptionally precise. The primary motivation for this was that when landing a yaw stunt, even if the body had accurate positioning upon landing, if it possessed even small amounts of angular velocity, the body would roll on landing. Having said that, if the positioning was slightly off, the linear momentum would also cause the body to roll, regardless of the state of angular velocity. The stunt also needed to be conducted in as little time as was feasible, since reduced flight time could be associated with reduced risk of damage to the robot. Originally it was stated that the controller would need to be highly robust and insensitive to modelling inaccuracy, but the results in chapter 7 suggest that this may not have been the case, since the model is fairly accurate and controller robustness could possibly be traded off for performance, obviously maintaining a sensible balance.

Considering all of the above, the following approximate requirements were set for the control design:

- i. The controller should have a settling time to the 2% band in less than 0.6 seconds.
- ii. The controller should have as small a steady state error as possible, but not more than 5% (approximately 20 degrees).
- iii. The controller should be well damped, with not more than 8% overshoot.
- iv. The controller should be realized in a digital form.
- v. The controller should be reasonably robust.
- vi. The control system output performance should receive 100% of the cost weighting, as input cost was not of concern for this application.

8.2 Control Design Method Selection

Classical control methods have been developed in depth over the past century, with most methods relying on frequency domain analysis [57]. Popular design methods involve using Root Locus plots, Nyquist diagrams, Bode plots and Nichols charts [58]. Most methods involve graphically analysing and shaping the bandwidth of a system to achieve certain control design criteria. The QFT method is particularly effective for such loop shaping. The strength of frequency design methods is that they provide an effective graphical illustration of controller stability, performance and robustness [59].

Control methods developed in the last five or so decades are considered to be modern control methods, and include state space, state feedback, observer design, Linear Quadratic Regulator (LQR) and neural network design methods [60]. Modern methods are far better equipped to handle MIMO systems, and through using information about the different states of the system, can exhibit high quality dynamic performance – achieving damped, fast responses. It is more difficult, however, to analyse stability margins and robustness using modern methods, it is possible, but it is more mathematically intensive [61].

With this information, the design decision was made to use state space or LQR methods (depending on the number of poles in the system). The reasons for this choice are that:

- i. The hardware on-board the robot offered access to 3 of the 4 system states (the motor armature current was not available).
- ii. The model is relatively accurate
- iii. These methods can achieve high quality dynamic performance, which was a requirement for this application.

8.3 State Space Model Population

The model to be controlled has been discussed at length, and can be numerically represented by four state space matrices that were presented in equation 38 and 39.

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0.0655 & -496.35 \\ 0 & 0 & -10.258 & 7.7695 \times 10^4 \\ 0 & 0 & -3.1228 & -6.5789 \times 10^3 \end{bmatrix} & B &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1.7544 \times 10^3 \end{bmatrix} \\ C &= [1 & 0 & 0 & 0] & D &= 0 \end{aligned}$$

8.2 State Controllability and Observability

Before proceeding with a state space control design, the degree of controllability and observability of the system states was ascertained. Both the controllability and observability matrices were constructed from the system matrices above, as explained in chapter 3, and Matlab was used to determine the matrix rank of each. The observability matrix had full rank (4), but the controllability matrix did not, it was of depleted rank (2). The implications of this were that in its original form, the system could not be moved from an arbitrary set of states to any desired states within in the phase plane [42]. The need thus arose to resolve this problem if state space design methods were to be used.

8.3 Loop Decoupling

With an understanding that the control system represented two coupled systems, the dynamic model of the body and tail, and the motor model (including both mechanical and electrical elements), the decision was made to decouple these systems. The motivation to decouple the loops was that each individual system may have full rank and thus be controllable. If this was in fact true, each loop could be controlled separately with its own controller. In order to perform the decoupling, the body and tail dynamics neglected friction effects in the DC motor. The validity of this required assumption is analysed later in the chapter.

The first new system represented the dynamic model of the body and tail, and is henceforth referred to as ‘loop one’. The second new system represented the motor with the tail loading, and is referred to as ‘loop two’. The output of the control action in loop one is a torque command; this then becomes the setpoint to the second loop, which is controlled to output a current corresponding to that torque; finally the output of loop two is fed to the input of loop one. This configuration is best explained graphically, and is illustrated by Figure 79. This system was feasible as a consequence of the motor electrical time constant being much faster than the dynamics of the body.

8.3.1 Decoupled Loop Control Models

i. Loop one:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad b = \begin{bmatrix} 0 \\ 5.12 \times 10^3 \end{bmatrix} \quad c^T = [1 \ 0] \quad d = 0$$

ii. Loop two:

$$A = \begin{bmatrix} -10.26 & 77.69 \times 10^3 \\ -3.123 & -65.79 \times 10^2 \end{bmatrix} \quad b = \begin{bmatrix} 0 \\ 17.54 \times 10^2 \end{bmatrix} \quad c^T = [0 \ 0.0967875] \quad d = 0$$

8.3.2 Decoupled System Analysis

The observability and controllability was checked again in order to ascertain whether the new system would be beneficial. The same Matlab script was used to assess the new systems, and it returned that both systems' observability and controllability matrices had full rank. This meant that if the decoupling did not alter the overall system model greatly, two controllers could be designed for each loop to achieve the unified control aims set out at the beginning of the chapter.

In order to assess whether the decoupling was acceptable, a Simulink model was developed (shown in Figure 75), and stepped with a step input of 7.25V. The resulting responses of each state are shown in appendix C, and were check against the those produced in chapter 7, section 7.6.5. It was concluded that the decoupling of the systems was an acceptable action for the purposes of controller design, since the graphs did not differ greatly in the operating region of the control action. The simulation did suggest, however, that the error would increase with time and increasing body velocity.

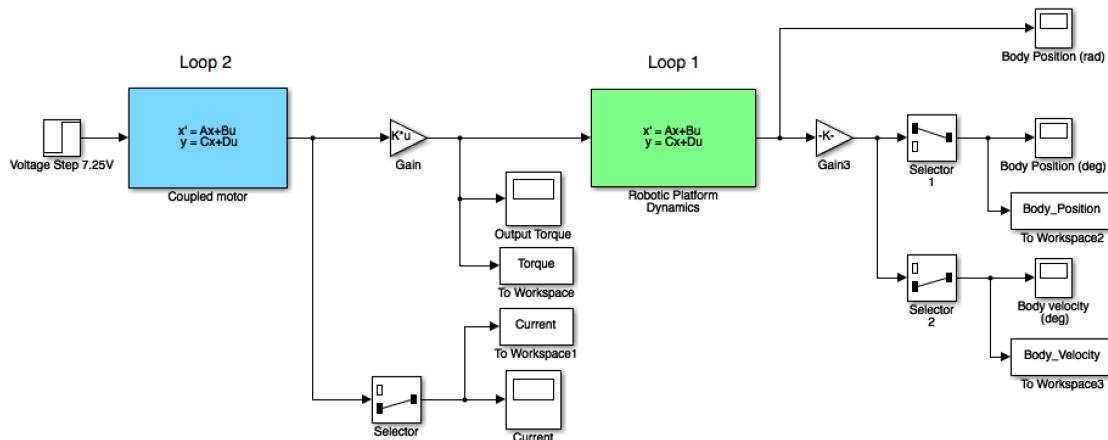


Figure 75: Simulink script used to perform a step test on the decoupled system

8.4 Integrated State Feedback Control Design for Loop One

The first design step taken was to design a state feedback controller with an outer loop integrator for the first loop. This loop represented the body dynamics, and thus the output of this loop was the output of the stunt, and needed to directly achieve the criteria outline in section 8.1. The loop form is shown in Figure 76.

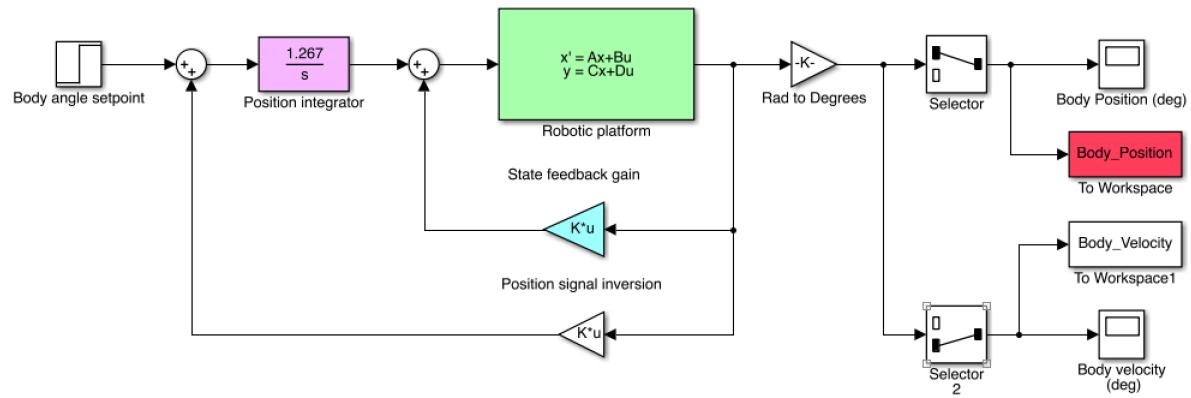


Figure 76: Integrated state feedback configuration for loop 1

8.4.1 Response Requirement and Pole Placement

In order to meet the requirements set out, the poles will be placed strategically using the state feedback gains. In order to achieve the following objectives:

- i. Settling to the 1% band in approximately 0.35 seconds
- ii. Less than 2% overshoot

The poles were placed such that the dominant time constant of the system was 0.067 seconds, and the system had a damping ratio of 0.8. This complies with a pole configuration of:

$$s_1 = -15 + j10$$

$$s_2 = -15 - j10$$

$$s_3 = -20$$

These poles correspond to the following characteristic equation:

$$\psi_c = s^3 + 50s^2 + 925s^2 + 6500 \quad (51)$$

8.4.2 Gain Calculations

The method of computing the gain vector was outlined in chapter 3, section 3.4. A Matlab script was written using the symbolic variable toolbox to compute the relevant matrix determinant, shown in equation 12 and presented in characteristic form here.

$$A' = \begin{bmatrix} A - bk^T & -BK \\ c^T & 0 \end{bmatrix}$$

$$\psi_c = \det(sI - A') \quad (52)$$

The coefficients of the characteristics equation, 52, were matched those of 51 using Matlab's solve function, and the resultant gains are shown in Table 17.

Table 17: Loop 1 state feedback and integrator gains

Gain	Value
Integrator	1.2675
Body position state	0.180375
Body velocity state	0.0097

8.4.3 Loop One Controller Response

The calculated gains were implemented in the model shown in figure 70 and simulated in the Simulink environment. The response of the body position against time and the torque input were subsequently analysed.

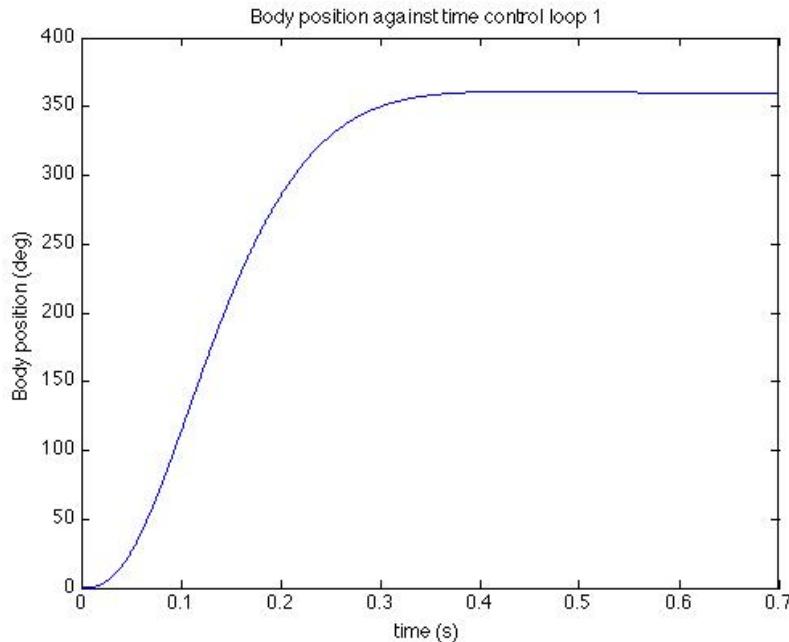


Figure 77: Body position response in the isolated control loop one

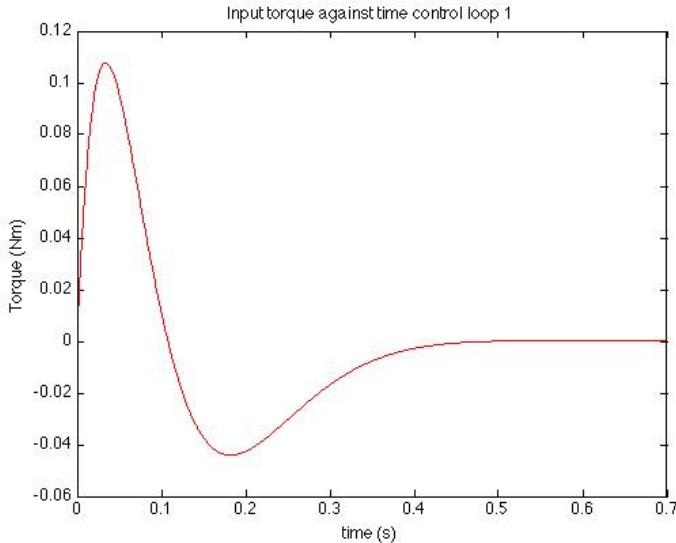


Figure 78: Torque input commanded by the controller in loop one

8.4.4 Comments on the Controlled Response of Loop One

Figure 77 shows that the body position response achieved the criteria set out for the control system in loop one. Figure 78 shows the torque input commanded by the system, which was satisfactory given that it did not exceed the actuator limitation set out in chapter 3, section 3.6.2.

8.5 Integrated State Feedback Control for Loop Two

8.5.1 State Feedback Design

A similar design procedure was followed for the second control loop, with the exception that the setpoint was supplied from torque input commanded by loop one. The controller was made to act at a similar speed to the electrical dynamics of the motor, which had a time constant of 0.152 ms. The details of the characteristic equation and pole placement are shown in appendix A section A3.1. The resultant gains calculated for the system (using the same Matlab script as before) are stated in Table 18.

Table 18: Loop two state feedback and integrator gains

Gain	Value
Integrator	2.08 x 10 ⁵
Motor velocity state	0.0049
Current state	0.8101

8.5.2 Observer Design

A current sensor that matched the operating conditions of this application was not available for this project, and thus an observer was required to observe the current state. A simple design approach was adopted, the theory behind which was described in chapter 3, section 3.5. The Eigen values of the observer characteristic equation were matched to poles at twice the speed of control loop two. Equation 20 is repeated here to illustrate the design procedure.

$$\psi_c = \text{Det}(sI - A + pc^T) \quad (20)$$

8.5.3 Loop Two Control Response

Before connecting loop two's output to loop one's input, its performance was analysed in isolation. This configuration is shown in Figure 79. In order to evaluate the performance of loop two, the output of loop two (in green) was plotted against the torque input commanded by loop one (which is the system setpoint, and displayed in blue) in Figure 80. The response undershot slightly at the peak of the input command, but in general the response looked fast enough and followed the input with relative integrity. The control input, which was a voltage command sent to the tail motor, was also analysed to ensure it was within reasonable bounds, since the voltage from a charged two cell lipo battery will saturate at approximately 8V. Figure 81 shows that the voltage is just within acceptable bounds.

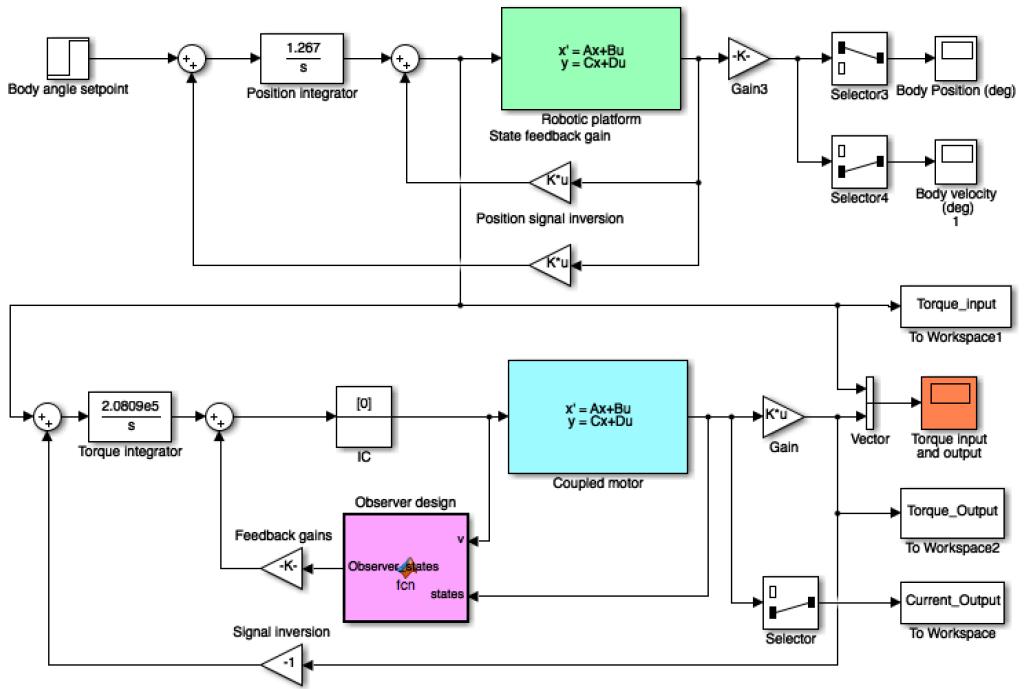


Figure 79: Control loop two Simulink model with setpoint from loop one and output isolation (not connected to loop one's input)

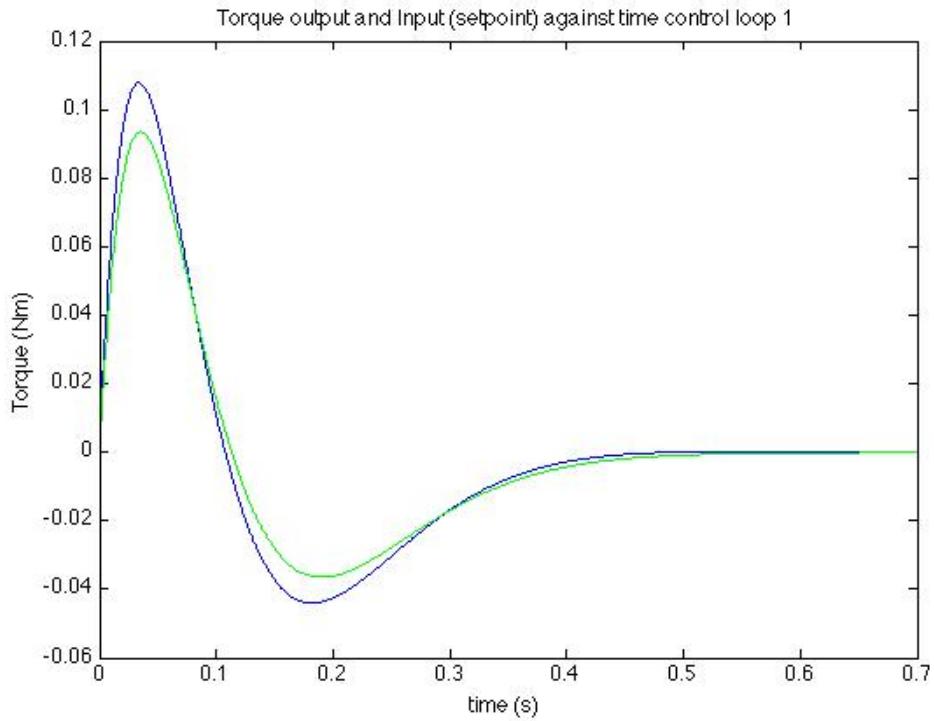


Figure 80: Loop two control output (green) against loop one control input (blue) and time

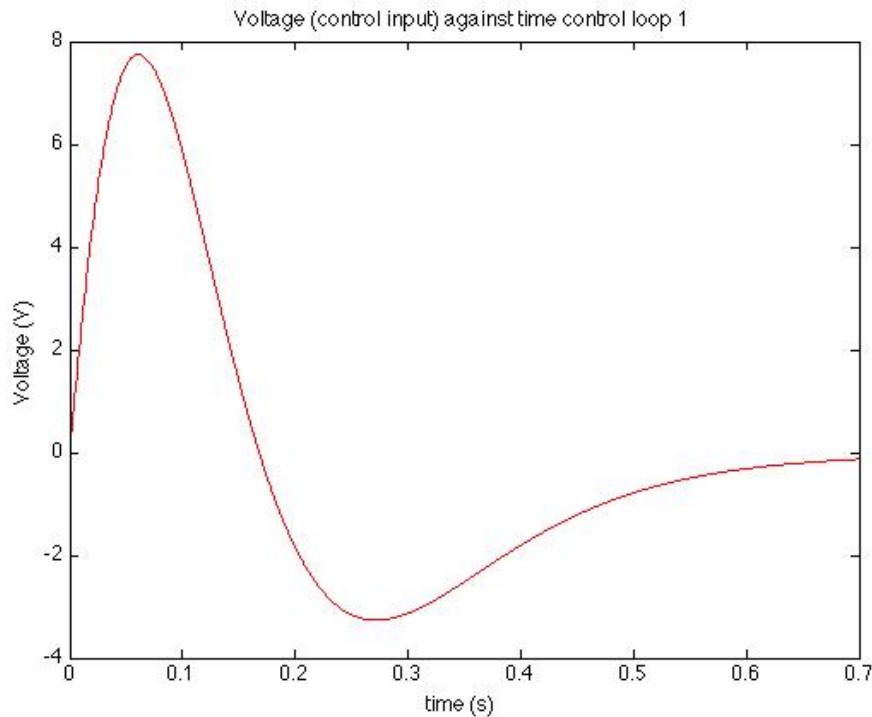


Figure 81: Loop two control input (voltage) against time

8.6 Combined System Response

The next step in the design was to connect the output of loop two to the state space model of loop one. This configuration is identical to that of Figure 79, simply with the above alteration performed, hence a diagram is not shown here.

Analysis of the signal outputs and inputs of the combined loops showed that they did not differ substantially from their isolated performances. As a result, only the body position is shown in Figure 82. The body position response of the combined system was slightly more damped, and slightly slower than the isolated system, both results could have been expected given the state of the input and output torque comparison graph (Figure 80). This simulation suggested that loop decoupling control design method was acceptable for this application.

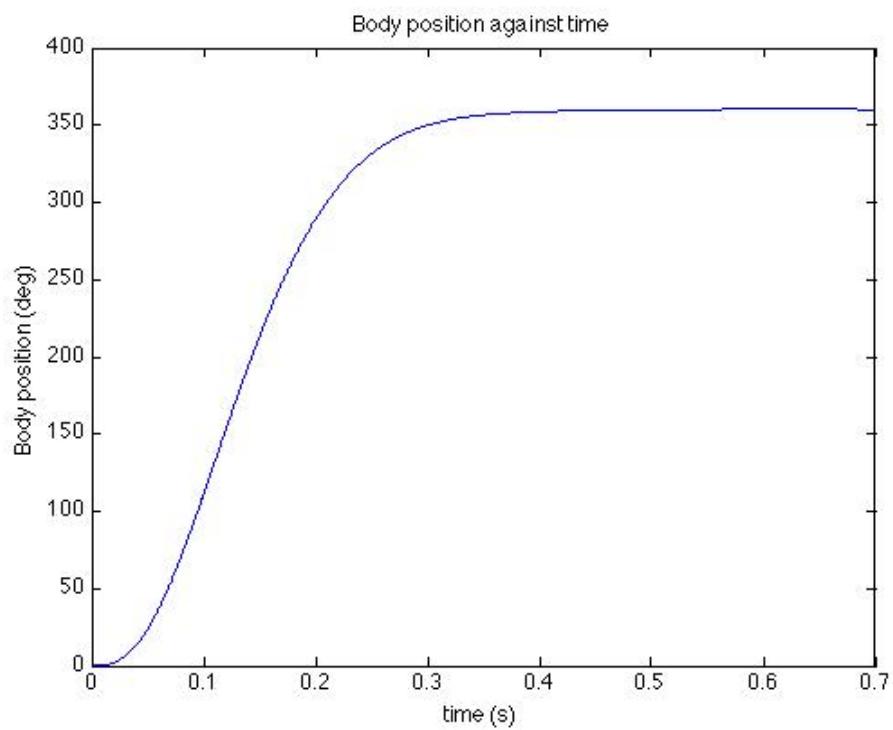


Figure 82: Combined loop system output of body position against time

8.7 Controller Discretization

The control for the stunt was performed by the on-board MCU, thus it was required that the controller be in a digital form. The design carried out in previous sections was performed in the continuous domain, and discrete implementation needed to be handled carefully. If a control loop runs 10 times faster than the fastest pole in a system, the phase lag introduced by sampling is assumed to be negligible [61]. This requirement meant that the loop needed to run at approximately 13.5 kHz, which seemed feasible given that the MCU processor could be run at a frequency of 72 MHz.

Unfortunately, all the communication between the sensors and XBEE modules, along with the interrupt calculation of the encoder frequency loaded the processor heavily, and the processor could not run the control loop at even close to the desired speed. The control of loop one was able to perform as required in the discrete domain, as it complied with the requirement set above. The dynamics in the motor loop however were far too fast for direct discretization, and hence the controller needed adaption.

8.7.1 Loop Two Controller Discretization and Simplification

In order to cope in the discrete domain, the dynamics of the second loop were considered to be infinitely fast, and the observer design was reduced to a real time estimator and very simple predictive controller, based on the state differential equations [44]. The outer loop integrator was also removed due to the phase lag it added to the loop, which in the discrete domain was not acceptable. The new system formulation is displayed in Figure 83.

In order to discretize the control system, both state space models were converted to a discrete form using Matlab. The new characteristic matrices were loaded into the Simulink model shown in Figure 83. The integrator in loop one was also discretized. This system formed the final model for control analysis, and hence the voltage saturation non-linearity was also added to the model to impose the physical limits of the real system on the simulation.

This model was used to assess the performance of the control system as it would run on-board the robot. The resulting signal responses to a position setpoint of 360 degrees are shown from Figure 84 to Figure 86.

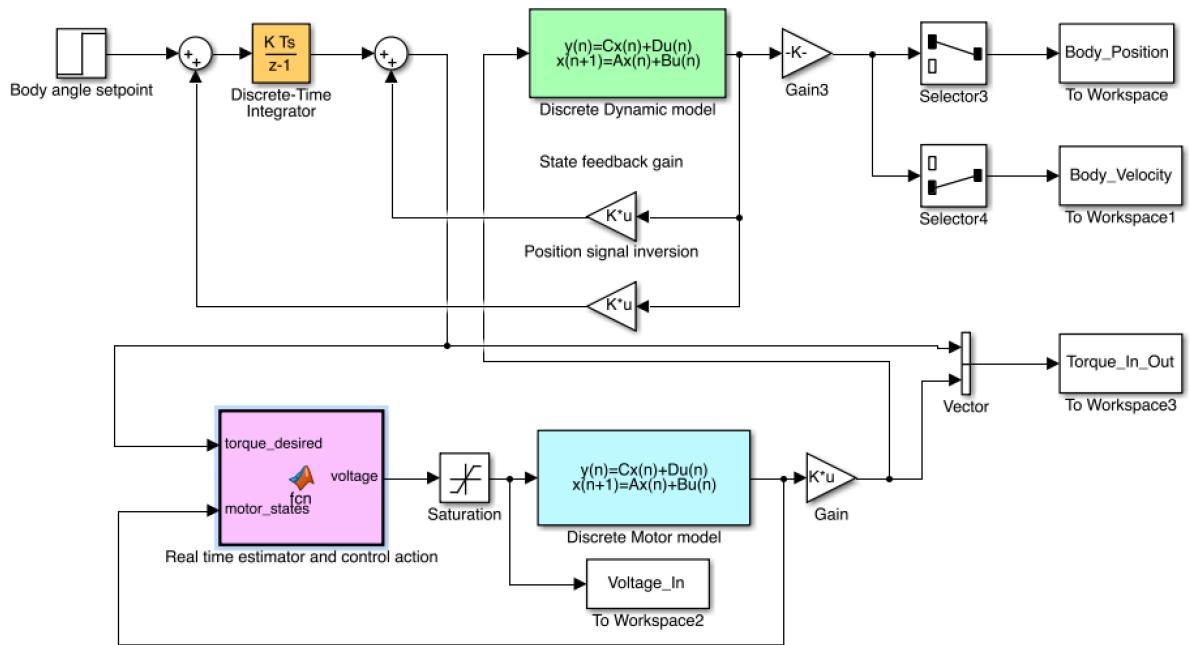


Figure 83: Discretized control system

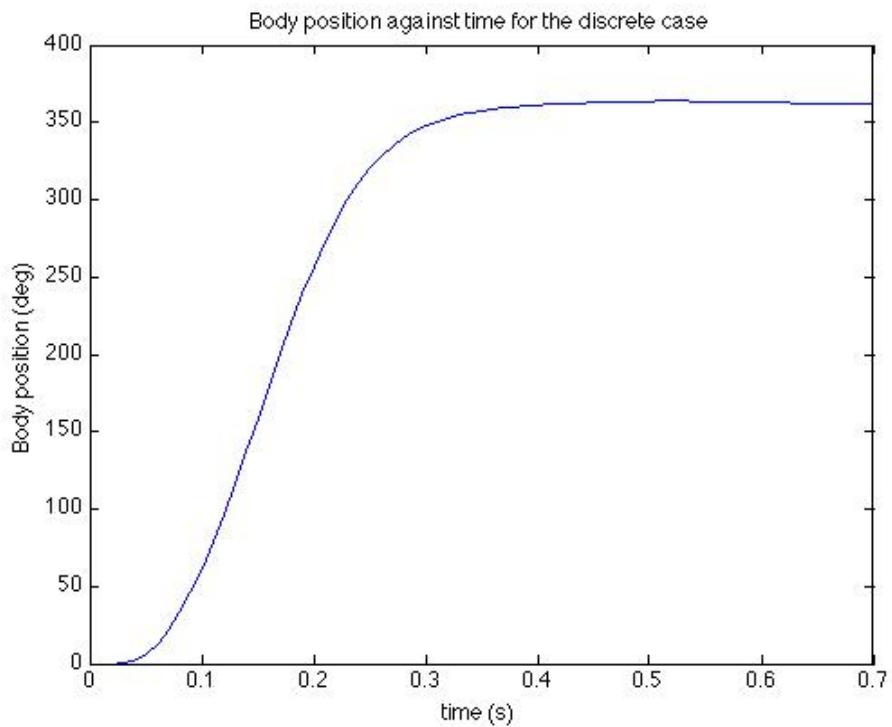


Figure 84: Body position response against time for the final discrete control system

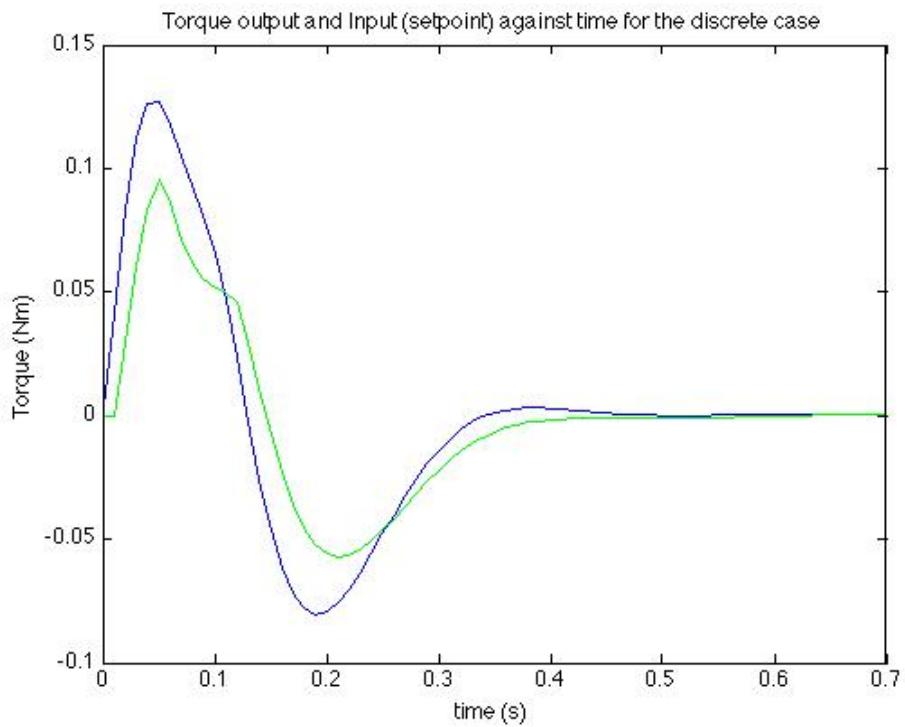


Figure 85: Torque input (blue) and output (green) responses against time for the final discrete control system

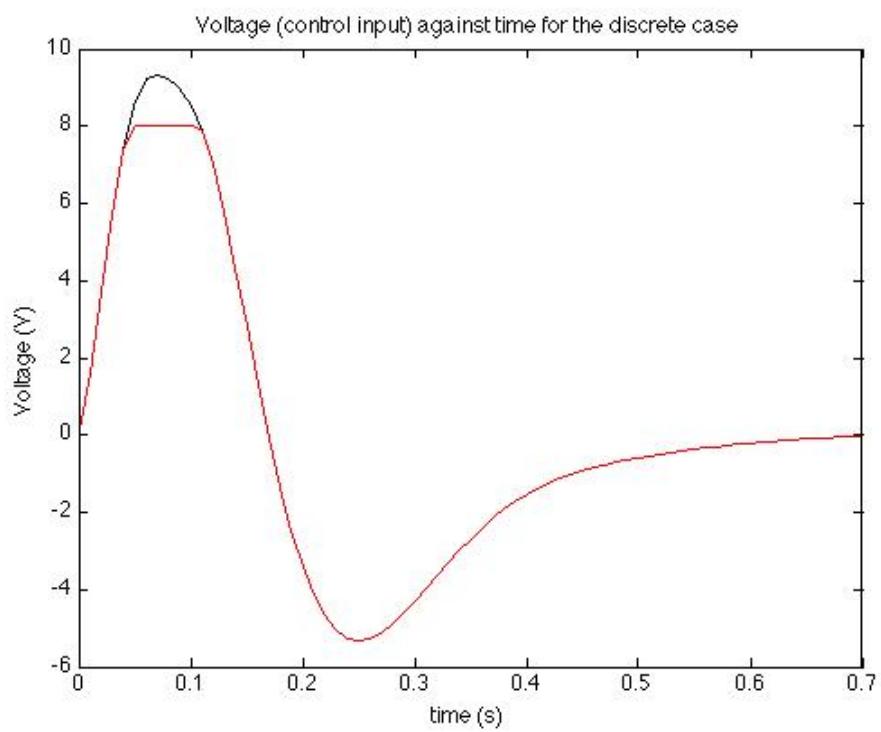


Figure 86: Control input (voltage) against time for the final discrete control system

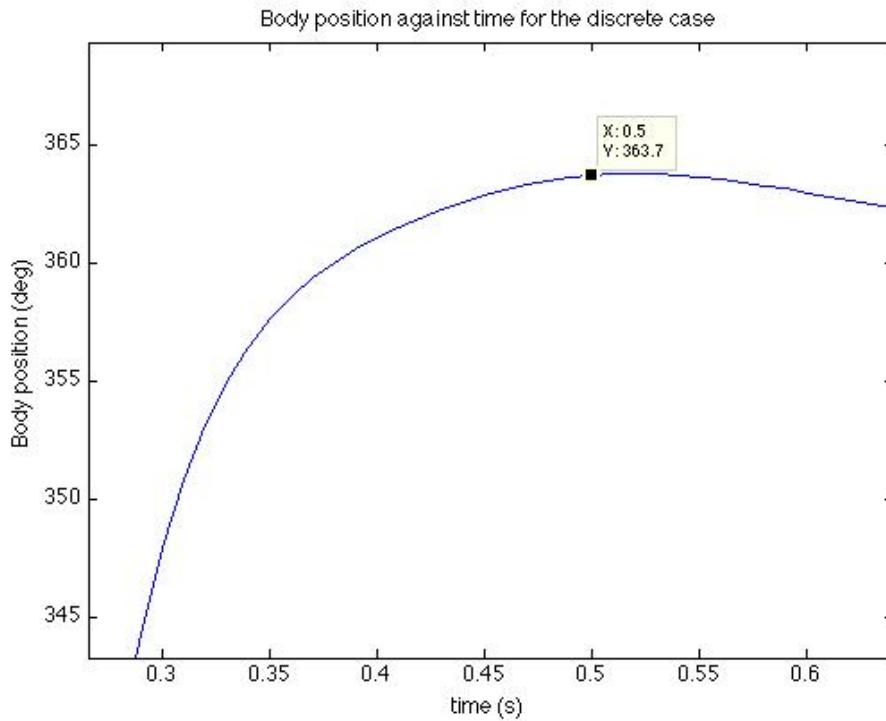


Figure 87: Body position response settling time and overshoot percentage illustration

8.7.2 Analysis of the Discrete Control System Performance

Figure 84 to Figure 86 show the responses of the important signals in the discrete control system. It can be concluded that the discretization of the control system made the system slower, settling in about 0.5s, and increased the overshoot. The discrete system overshoot was captured by the data point in Figure 87, and was about 1%. These effects can be attributed to the discretization, as well as the voltage saturation that occurred between 0.05 and 0.1 seconds (shown in Figure 86). Since the system contains an integrator, the final steady state error will be zero for the simulation, but since the application of the control system was not continual regulation of a system, but rather to control a dynamic action, the overshoot at 0.5 seconds was considered the final error of the control, since it was likely that the stunt would be completed around this time.

Despite the adverse effects of the discretization and the simplification in loop two, the control systems performance was not degraded severely, and the system clearly met the requirements set out at the beginning of the chapter. In summary, the settling time was less than 0.6 seconds, the error was less than 5% and the overshoot was less than 8%, thus all the aims of the control systems were satisfied.

8.8 Controller Robustness Analysis

For any physical implementation of a control system, the system must be checked for robustness, since there are always discrepancies between simulations and real system performance. The best tool for analysing stability margins, sensitivity and robustness of a linear system is frequency domain analysis [59]. It is possible to convert a state space model to a transfer function using matrix adjoints and determinants [61].

This is a laborious procedure, and a simplified approach was taken here, since the model was judged to be accurate in chapter 7. Elements in the discrete state space matrices were altered in the Simulink system model that corresponded to frictional effects and rotational inertias, which were considered to be most likely the largest sources of error. These quantities were varied up to 30%. The system response was then analysed, and it was seen that the system output did not vary substantially or become unstable, thus further robustness analysis was not conducted. Figure 88 below displays five various responses.

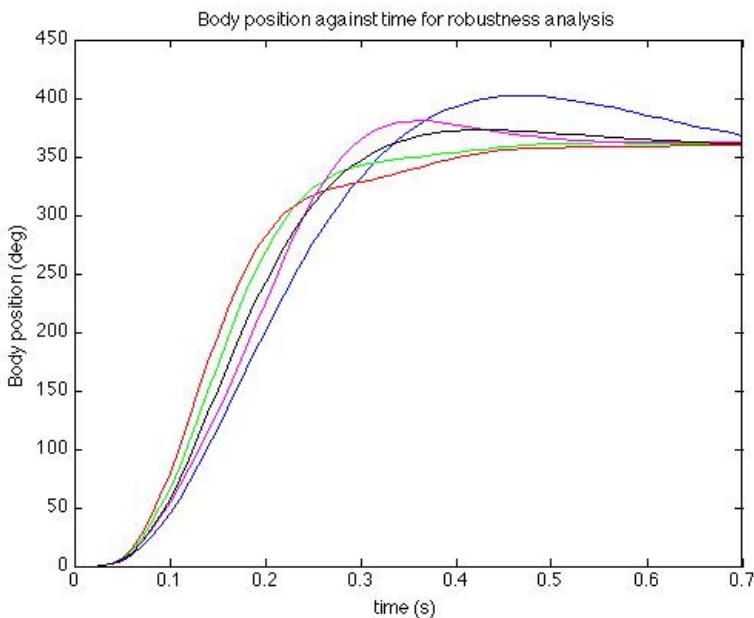


Figure 88: Robustness analysis using matrix element variation for the discrete control system

8.9 Final Comments on the Control System

The controller design procedure was a thorough process, and the various controller designs performed as required. It is regretful that aspects outside of control theory limited the implementation of the best possible system. Having said that, the design performed well enough in simulation to complete the stunt.

9. Testing and Results

This chapter explains the methods used to test the robot, including aspects of the design of the test, the realization of it and the conduction of various tests. The data acquired during the testing phase are then presented as the results of the tests.

9.1 Test Rig Control Test

Before the control system was implemented for a jump sequence, it was tested on the test rig. This gave invaluable insight into the control performance, as even though the models for the control system on and off the rig are not the same, it was possible to see how the controller performed in a safe environment. This also provided an opportunity to test all the fail safe systems in the code.

The controller was run numerous times on the rig, and data for three such test is shown here. The data for body position and body velocity for the three tests are grouped and displayed in Figure 89 and Figure 90. A video of the controller operating on the test rig is included on the disc provided, and the file is called, “Test_rig_control.m4v”.

The responses show that the controlled body position had a dynamic response similar to what was simulated, but that it had slightly more overshoot and steady state error. The details of the test rig data is explored more comprehensively in the discussion chapter to follow.

The discrepancies between the simulation and actual response were expected, as it was a real system with modelling inaccuracies and disturbances. The overall performance however was very satisfactory, and the decision to move forward to testing the system on a jump happened very early on during test rig testing.

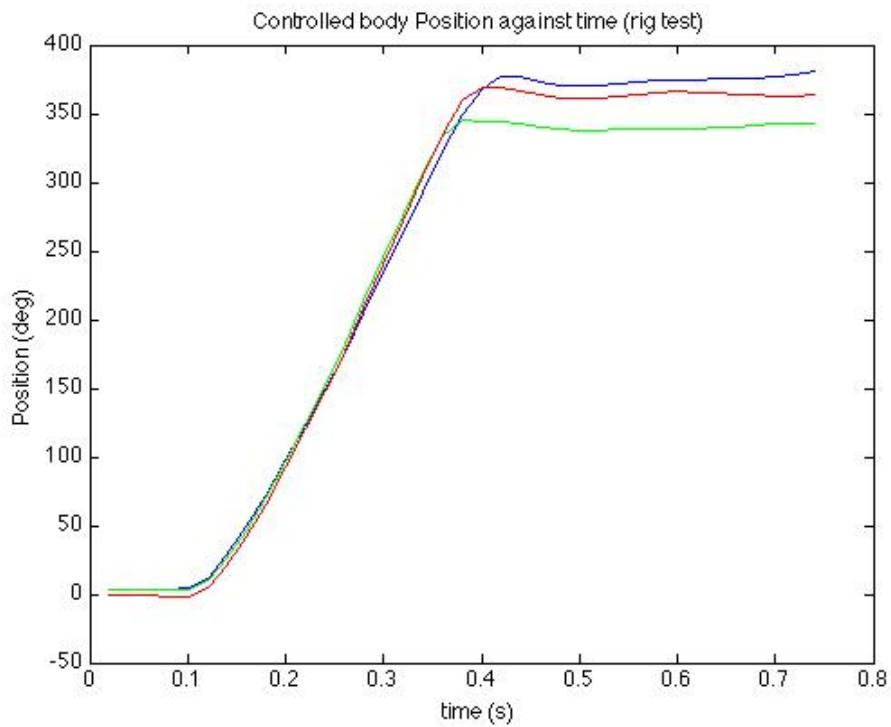


Figure 89: Body position response to an input setpoint of 360 degrees for three controller tests on the test rig

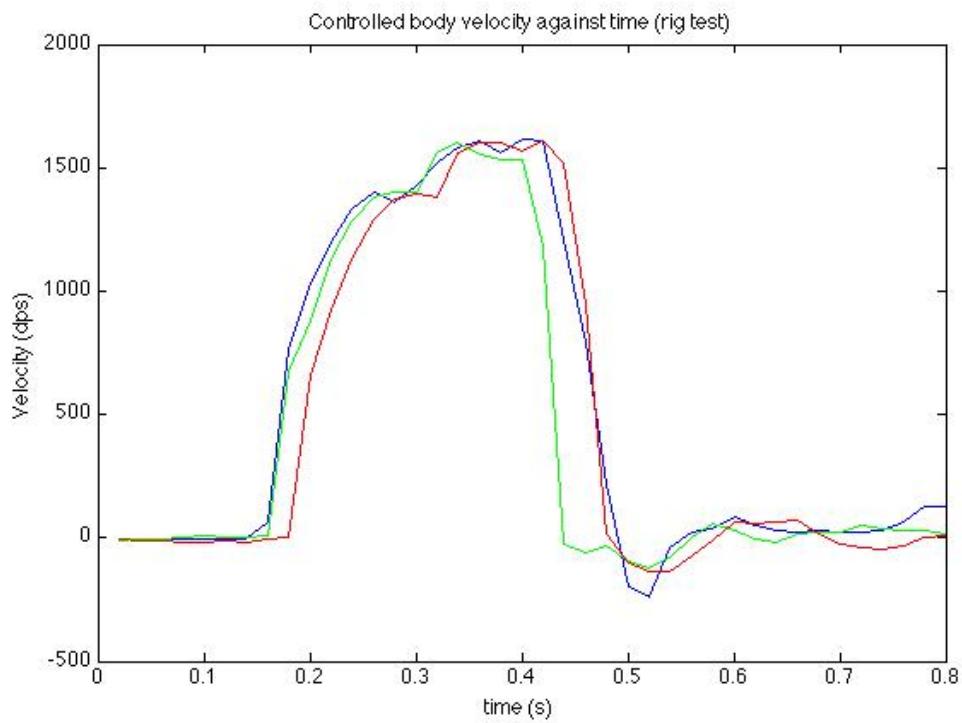


Figure 90: Body angular velocity response to an input setpoint of 360 degrees for control tests on the test rig

9.2 Creation of The Test Environment

9.2.1 Jump Conceptualization and Construction

Originally the stunt was intended to be performed in a similar environment to the FlipBot [34], with the robot driving up to a ramp, hitting the ramp and executing the stunt in the air thereafter. Unfortunately this was not possible, since the rear drive motor did not generate enough torque to drive the robot up even light gradients, let alone achieve airtime off a ramp. The improvisation was made to generate linear momentum for the robot by creating a raised, rectangular platform that allowed the robot to start on a downslope and curve upward toward the end of the jump. This concept is captured by the diagram in Figure 91.

The modified ramp was realized with a simple slab of Masonite, which worked well for the application; it was light, flexible and could be shaped into the correct ramp dimensions. Figure 91 is a diagram of how the ramp was structured. It was not possible to construct the ramp in a permanent form for a number of reasons; the physical size of the ramp platform meant that it would not be transportable if in a fixed form, and the launch angle required adjustment given the height of the protective landing area.

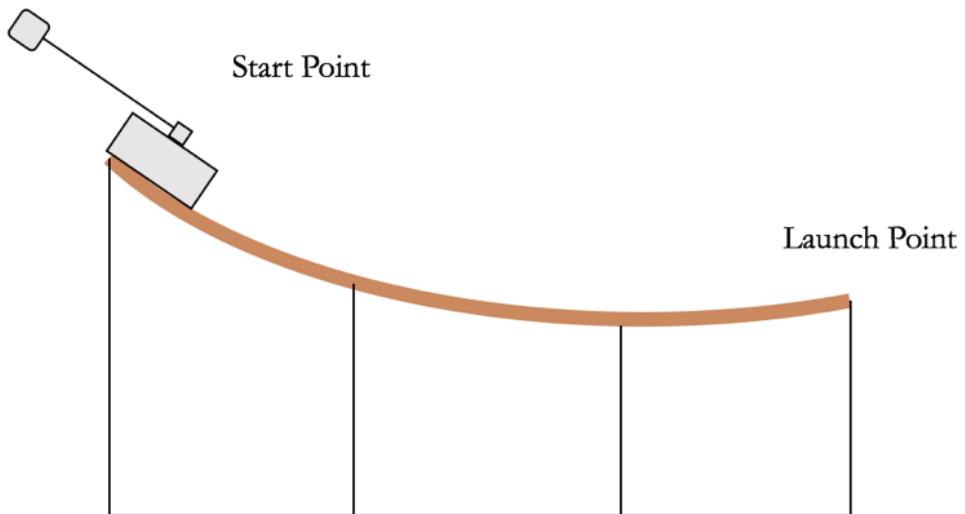


Figure 91: Modified ramp configuration

9.2.2 Test Environment

Apologies are made by the author about the environment in which the testing took place. Due to logistical and spatial constraints, it was not possible to conduct the test in a lab. It was also required to provide a protected landing area so that the robot did not damage itself. There was intent to move the stunt to a more professional environment for the purposes of capturing video footage, but at that moment in time the car suffered damage to the tail motor support bracket when it fell off the platform waiting to start a jump.

9.3 Testing Prior to Actuated Jumps

9.3.2 Unactuated Jump Testing

Before the control system was tested on a jump, the robot was made to hit the jump with the control system disabled. This test was used to analyse the flight time that the robot achieved on the ramp, and for the purposes of capturing video footage to contrast the actuated and unactuated cases. The footage shows that no yaw motion is achieved without tail actuation. The Z-axis accelerometer data was used to analyse flight time, as upon take off there is a negative spike and at landing there is an even larger negative spike. The ramp could then be shaped to the optimal setting to provide just enough flight time for the stunt (approximately 0.4 seconds). Video footage of the unactuated jump is documented on the attached disk and called, “Unactuated_jump.mp4”.

9.4 Actuated Jump Tests

Once all the above tests had been conducted safely, the system was ready to be put through an actuated jump sequence with the control system operational. The test was carried out numerous times, in the region of 30 to 40 jumps were completed with data successfully logged for approximately 20 jumps. In other cases the communication modules dropped signal at or before launch and the data link was broken before the start of the stunt.

A successful landing was classified as the robot landing on all four wheels, and staying in that position on the bounce; the robot would roll while driving thereafter because the steering was not controlled and always locked into a turn on the bounce. The robot was able to successfully

complete the stunt in the air many times, however it only landed the stunt three times. In the other cases the car pitched too drastically at the instant before landing.

The logged data included all the gyroscope and accelerometer readings, as well as the motor encoder data. The data of interest for analysis was obviously yaw position and yaw angular rate, captured by the z-axis gyroscope, the motor encoder data and the x-axis and y-axis gyroscope readings which relayed information about the pitch and roll angular rates and positions during the stunt.

9.4.1 Video Footage and Documentation

The jumps were recorded using a GoPro Hero 3 Black camera from the side angle. A video has been edited to contain general footage, various speeds, and freeze frames of different jumps. It is on the attached disc and called “Actuated_jump_tests.mp4”. One such example of a successful jump sequence is shown in Figure 92. These images are best viewed in conjunction with the video footage, and the reader is encouraged to consider both interchangeably.



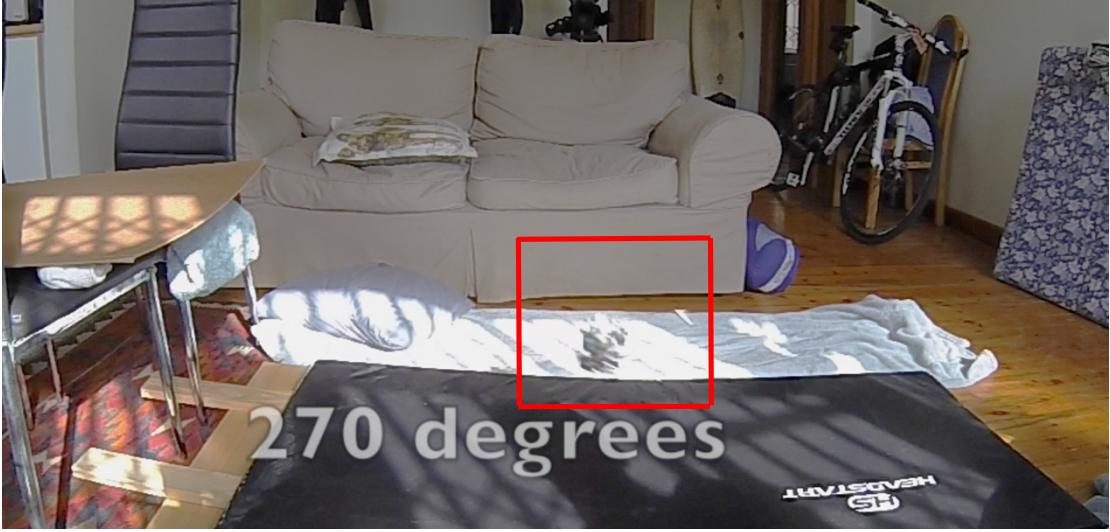
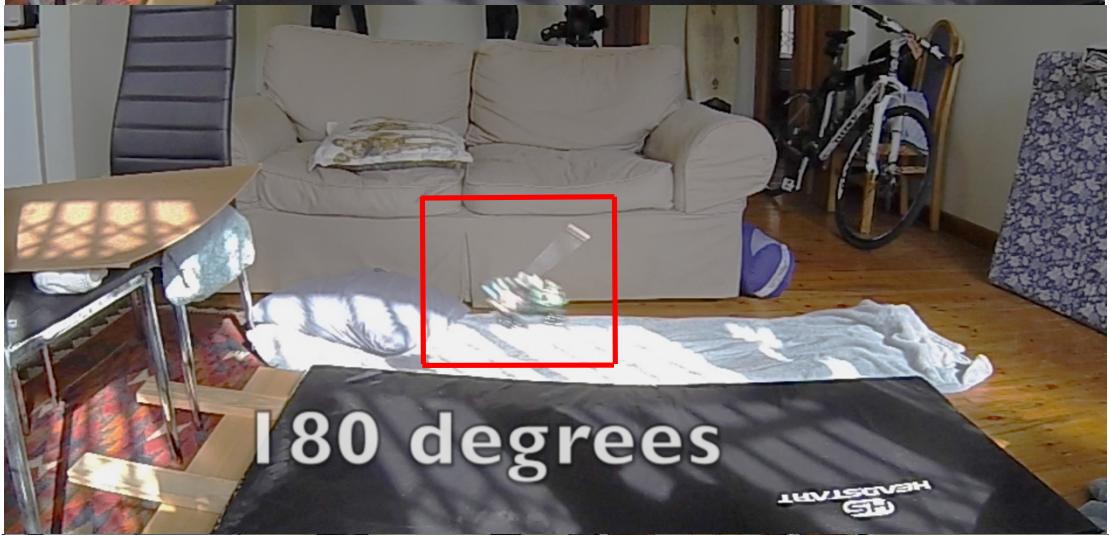
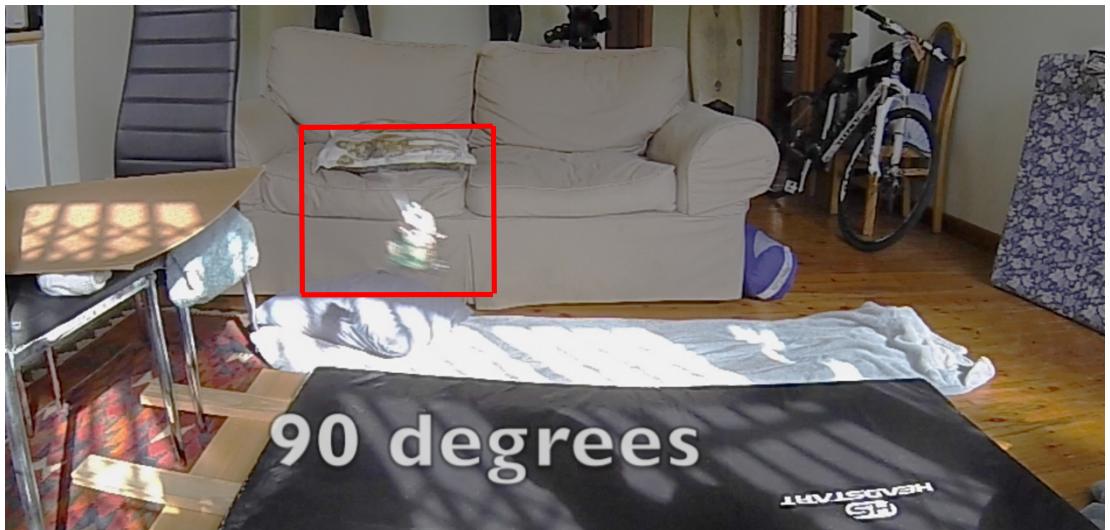




Figure 92: Successful 360 degree yaw stunt jump sequence, taking a freeze frame at the launch, 90 degrees, 180 degrees, 270 degrees and 360 degrees

9.4.2 Body Yaw Position, Velocity and Motor Velocity Data

Jump data has been characterized for body yaw position, body yaw velocity, motor velocity in 2 graphs for each element. The first graph of each shows 7 arbitrary jumps plotted together, and the second graph shows 3 successful jumps plotted together. The position graphs are shown in Figure 93 and Figure 94, the velocity graphs are shown in Figure 95 and Figure 96, and the motor encoder data is shown in Figure 97 and Figure 98.

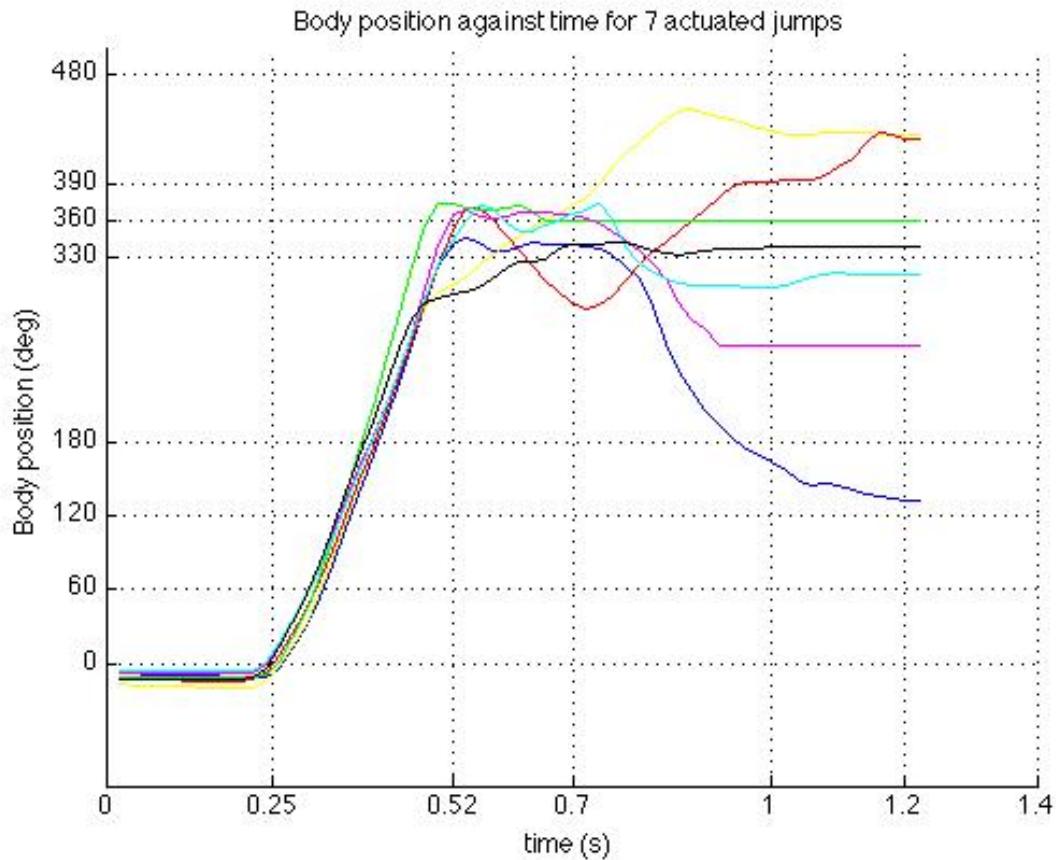


Figure 93: Body position against time for 7 arbitrary actuated jump tests, the launch occurs at 0.25 seconds and the landing occurs at approximately 0.6 seconds.

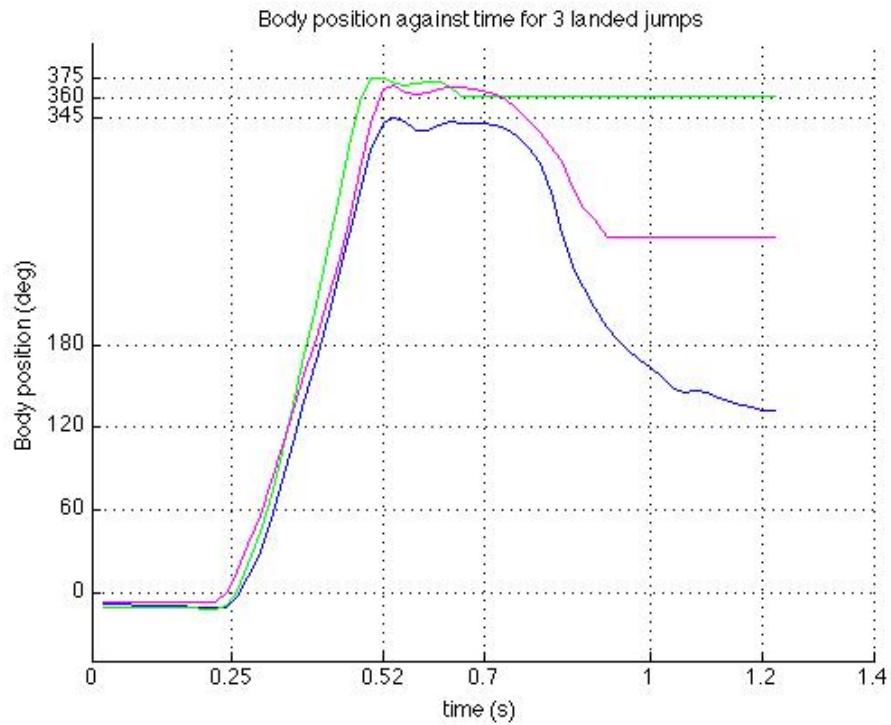


Figure 94: Body position against time for 3 landed jumps, the launch occurs at 0.25 seconds and the landing occurs at approximately 0.6 seconds.

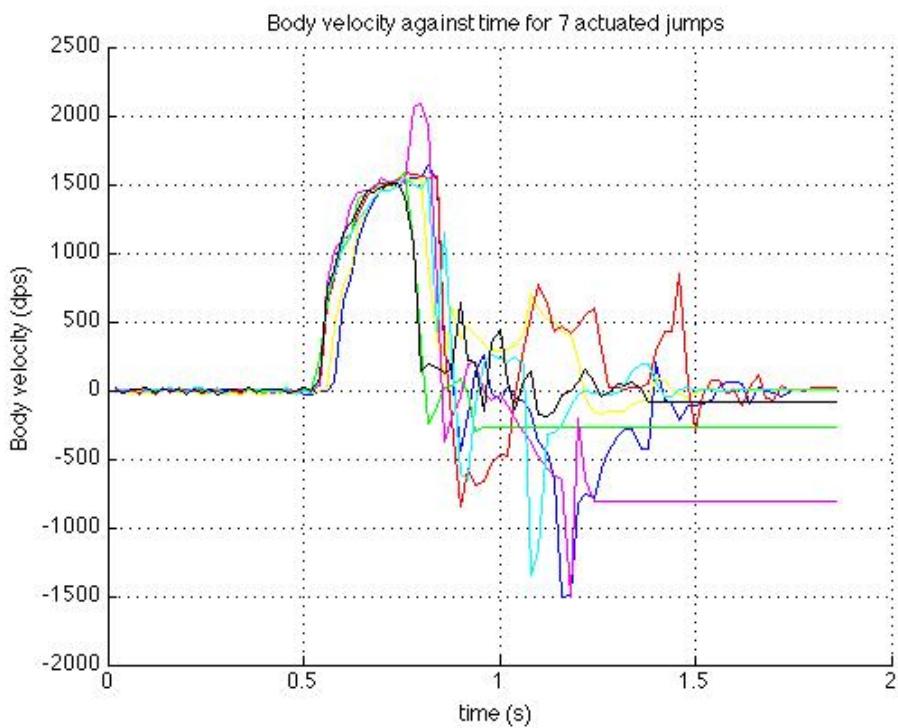


Figure 95: Body velocity against time for 7 arbitrary actuated jumps, the launch occurs at 0.25 seconds and the landing occurs at approximately 0.6 seconds.

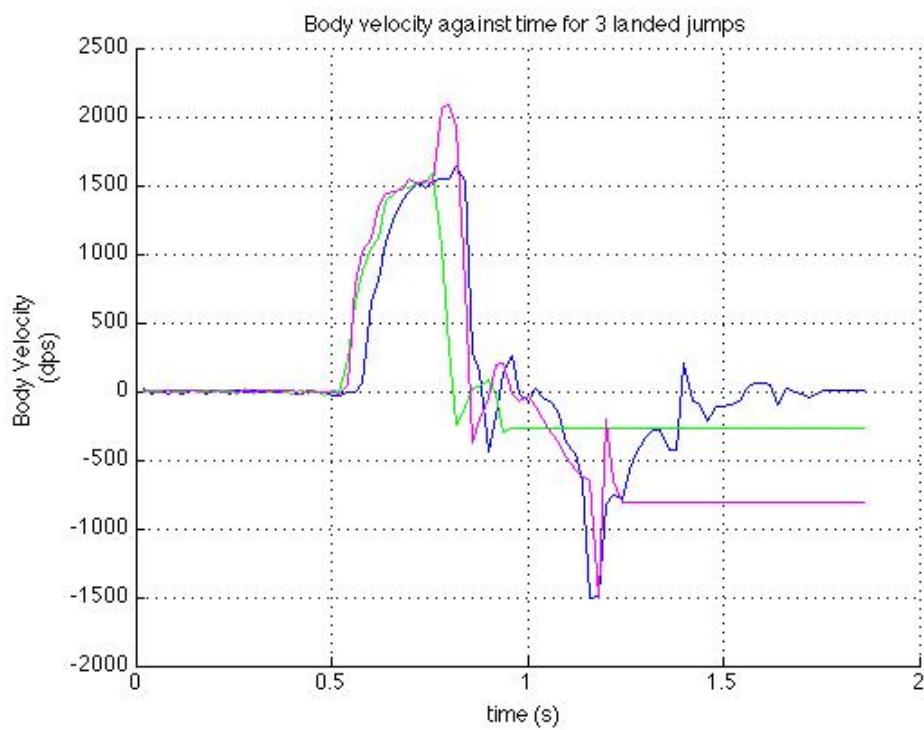


Figure 96: Body velocity against time for 3 successful jumps, the launch occurs at 0.25 seconds and the landing occurs at approximately 0.6 seconds.

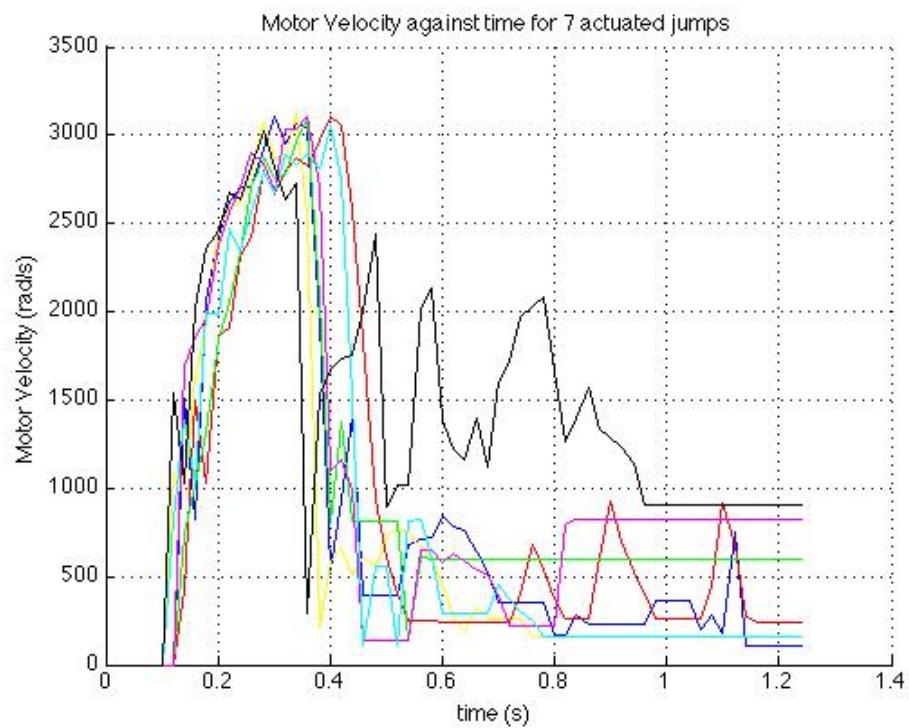


Figure 97: Motor velocity against time for 7 arbitrary actuated jump tests, the launch occurs at 0.25 seconds and the landing occurs at approximately 0.6 seconds.

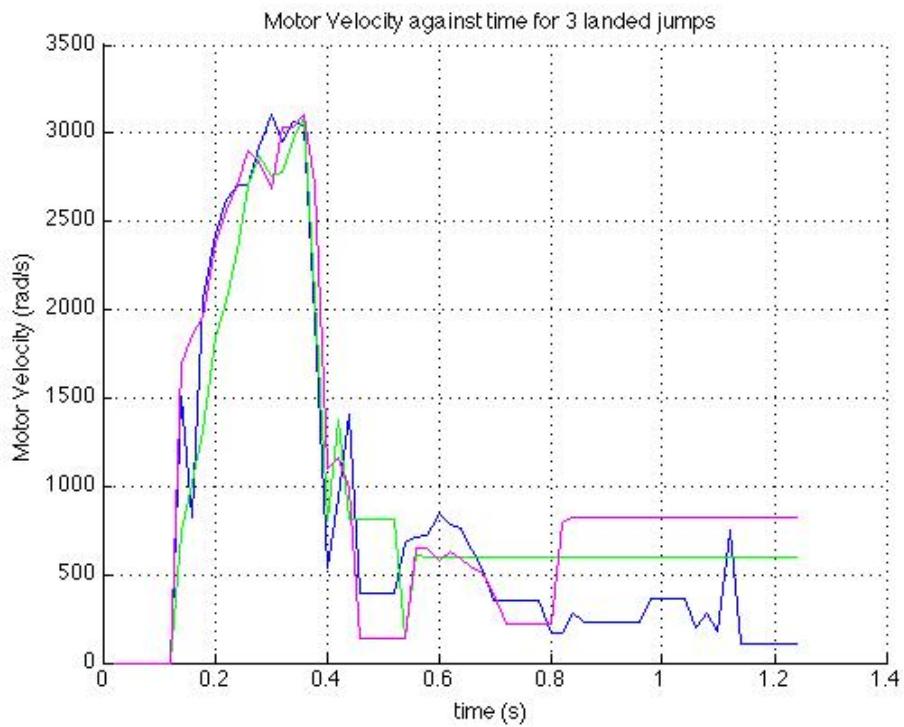


Figure 98: Motor velocity against time for 3 successful jumps, the launch occurs at 0.25 seconds and the landing occurs at approximately 0.6 seconds.

9.4.3 Comments on the Data Presented

Full analysis of the data presented above will follow in the discussion chapter, but at this point it must be mentioned that the time scale of the data does not reflect launch times accurately, and the reason for this is that the data processing was performed with the aim of capturing as much useful information in as small a graphical space as possible, thus the signals were mapped from different starting points against the same time vector. For example, it was useful to consider the z axis gyroscope readings before the launch, which gave an indication as to the route taken by the robot to the launch point, hence the body velocity graphs contain the most prelaunch data content. It must also be noted that the motor encoder data was transmitted as an unsigned integer, and hence always appears positive on the graphics, however the software in the control loop interpreted when this value became negative.

9.4.4 Pitch and Roll Data

Data was also captured from the x axis and y axis gyroscopes. This data was integrated to give pitch and roll positions of the robot during the stunt. It was analysed for selected cases, one in which the robot experienced pitch or roll effects causing it to not land the stunt, and one in which the robot managed to land the stunt. These results are presented in the discussion section.

9.5 Testing Problems

The RC car provided by UCT was originally intended for another project and a different stunt, thus it was not perfectly suited to the 360-degree yaw stunt of this project. This was epitomized by the fact that the rear motor could not supply sufficient torque to allow the car to hit an isolated, grounded ramp. Another substantial problem with the car was that it was not very robust, as the plastic wheel-base did not have strong reinforcement. This meant that the stunt had to be very carefully managed for fear of damaging the car. To this end, the testing environment imposed restrictions which ultimately resulted in the apologies from the author stated as previously in the chapter.

Despite the weak wheel base, the modifications to the car were built robustly, and consequentially the robot was able to endure numerous crashes on landing. Highly unfortunately, while I was setting the GoPro Hero 3 Black camera, the robot suffered damage to the tail motor perspex bracket after falling off the ramp before the initiation of a jump. The car was held back by insulation tape to prevent it from sliding down the large slope found at the start of the ramp; this tape gave way, and the robot fell off the side of the ramp onto the wooden floor upside down. This damage was unfortunate, for although sufficient data had been acquired, there was an intent to capture footage of the platform from multiple angles in a professional environment later that day.

Problems were also experienced during testing due to the ramp configuration. The steep slope at the start of the ramp required the car to be manually held before the start of the stunt. When the stunt was initiated, the car was not always released in exactly the same way, and hence the tests were not always performed in the perfectly controlled environment desired.

10. Discussion

This chapter evaluates the test data in terms of models and simulations that came before testing. It presents a critical analysis of the dynamic models, control systems and overall performance of the robotic platform designed and constructed to perform the 360-degree yaw stunt.

10.1 Analysis of the Dynamics

The dynamic modelling was undertaken at the start of chapter 5 and involved using the Lagrange method to develop equations of motion that would describe the body and tail system dynamically. Two methods of analysing the accuracy of this modelling are firstly, inspecting the prediction made about the dynamic coupling by considering the tail position relative to the body position, and secondly, to analyse the time constant of the dynamic differential equations. The second aspect will be discussed in the evaluation of the control system performance.

10.1.1 Dynamic Coupling Relative Position Predictions

The significance of the inertial ratio was first introduced in chapter 5, and it is readdressed now. The rotational inertial ratio for this system was quantified using equation 53 below.

$$I_{ratio} = \frac{I_B}{I_T} = \frac{1.95 \times 10^{-4}}{3.8 \times 10^{-4}} = 0.513 \quad (53)$$

From the dynamic equations, it can be seen that this is a representation of how much more the angular position of the tail will change compared to the body. As the rotational inertia of the body tends to zero, the inertial ratio tends to zero. For a hypothetical system in which body rotational inertia is zero, the body and tail would undergo the same change in angular position for a torque input to the system. The simulations predict that the inertial ratio of the robot constructed for this stunt (0.513) will cause the tail to undergo approximately 1.5 times the angular positional change of the body. This is illustrated by the simulation of motor 1, and Figure 34 and Figure 35, since for a 180 degree body rotation, the tail travels through 270 degrees.

This prediction was confirmed during both controlled rig tests and jump tests. Figure 99 and Figure 100 show the final position of the tail relative to the body; these images are again best considered in conjunction with the relevant video footage.



Figure 99: Test rig 360 degree stunt inspection of tail position, which is approximately 525 degrees



Figure 100: 360 Degree jump stunt inspection of tail position, which is approximately 555 degrees

The images and video footage confirm that the simulation prediction of tail angular positon relative to body angular position was reasonably accurate, as the tail travelled through approximately 540 degrees for a body position rotation of 360 degrees. It must be noted that the tail swings a bit further upon landing the stunt, and hence has travelled further than 540 degrees in Figure 100, but this is not attributed to aerial dynamics. This observation can be used to partially validate the dynamic modelling.

10.2 Analysis of the Control System Performance

The control system was tested exhaustively on the test rig, but the true test came during the jump sequences. In order to examine the control system's performance during the stunt, two cases were considered: instances of the controller performing well and the stunt being completed, and conversely examples of controller performance which resulted in a crash.

10.2.1 Successful Control Performances

In order to analyse the performance of the implemented controller, the data related to the three successful jumps presented in Figure 94 was plotted on the same graph as the simulation prediction of the control system response. This is shown in Figure 105.

(i) Dynamic performance

Figure 101 shows that the dynamic performance of the simulation (black line) closely matched the successful jumps. The jump was initiated at 0.25 seconds and completed at approximately 0.6 second. This is thus further support that the modelling of the system was accurate. It is evident that the real system (responses in cyan, magenta and green) was not as damped as the simulation, and this was to be expected given the decision made to neglect motor friction in the body dynamics.

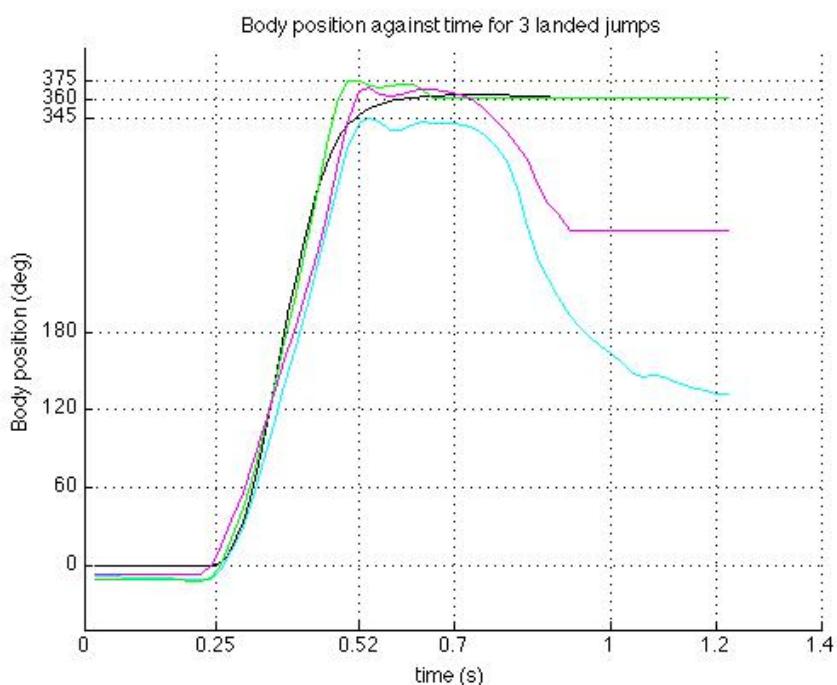


Figure 101: Simulation prediction of the control system (black line) plotted with successful jumps (colourful lines)

(ii) Final system error

The system almost always had a small amount of final error, in the successful cases it ranged from 4.2% to 1.2%. This can be explained, as even though the system contained an integrator, this integral action would only reduce final error to zero in a greater amount of time than was available during the stunt. This was the reason that the error at 0.5 seconds was considered to be a ‘final’ system error. The controller aim was to have not more than 5% error, corresponding to 20 degrees, and this requirement was met for the successful jumps. Other jumps shown by the black and yellow plots in Figure 93 suffered from final errors greater than were tolerable, and they did not land for this reason.

(iii) Precise control problem

Observation of the data and the controller performance on the test rig showed that the controller exhibited high quality dynamic performance almost every test, but it often had final error, and would subsequently struggle to correct itself to exactly 360 degrees. This can be attributed to the large static friction imposed on the system with the tail mount on the output motor shaft. Tests were not conducted with the tail on to quantify the static friction, as the controller would not include elements to compensate for this nonlinearity. With more time and resources the controller could be improved by adding static friction compensation, using methods such as adaptive control [40].

(iv) Effects of initial angular velocity.

The control system performance was also affected by the route taken by the car to the launch position. As stated previously there was no steering control on the robot, it was merely fixed to go as straight as possible, and thus the robot did not consistently hit the ramp with zero yaw angular velocity. This can be seen in the z axis gyroscope data shown in Figure 102, which is an inspection of the 7 jumps that feature in chapter 9. As can be seen from the graph in Figure 102, the body yaw velocity changed up to 30 dps at the instant of the launch. This meant that the control system states did not all start from zero. It would have been useful to impose such a disturbance on the system during the design phase and consider how the system would have performed under such conditions in simulation. These jumps also formed a set of better jumps, and more variation was seen in poor jump attempts. It can be deduced that nonzero initial yaw angular velocity was a contributor to failed jumps.

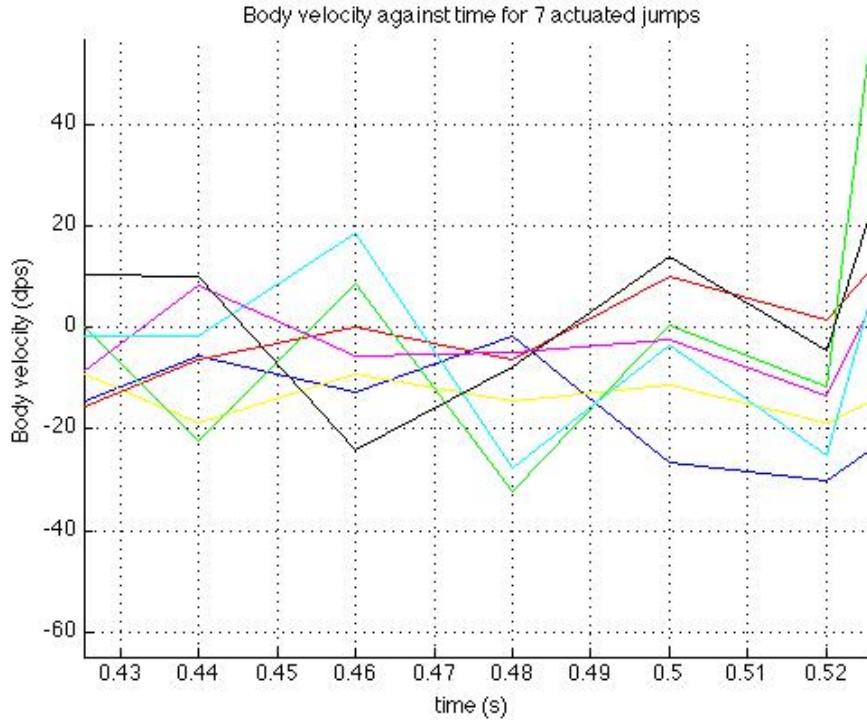


Figure 102: Body yaw velocity directly before launching

10.3 Analysis of Pitch and Roll Effects

During many of the jump tests, the robot completed the 360 degree stunt in the air, but subsequently crashed on landing due to excessive pitching or rolling of the body. The design made the best attempt to negate these effects by placing the tail joint as close to the centre of mass as possible, but physical constraints made this very difficult. The pitch and roll effects were expected and are discussed here.

10.3.1 The Dynamics of Pitch and Roll Effects

When a clockwise torque was applied to the tail, there are two reactive forces as well as a moment that acted on the body [37]. These forces are depicted in Figure 107. The force F_p was a result of the angular velocity of the tail, when the tail possessed angular velocity, it also underwent radial acceleration and the force that generated this radial acceleration acted at the tail joint. The tail experienced this force toward the tail joint (always), and the body experienced the reactionary force. This force is called F_p .

There was also a reactionary force experienced by the body that restricted the tail from generating linear velocity as a result of the force applied by the motor. This force is called F_R . These forces

acted interchangeably, causing pitch and roll depending on the tail position, velocity and motor torque. The simplifying assumption was made only to consider the effects at the end of the stunt, hence F_p only caused pitch effects and F_r only caused roll effects due to the tail being near the front of the car. Figure 103 offers 3 views of the body and illustrates the forces acting on it and their respective directions.

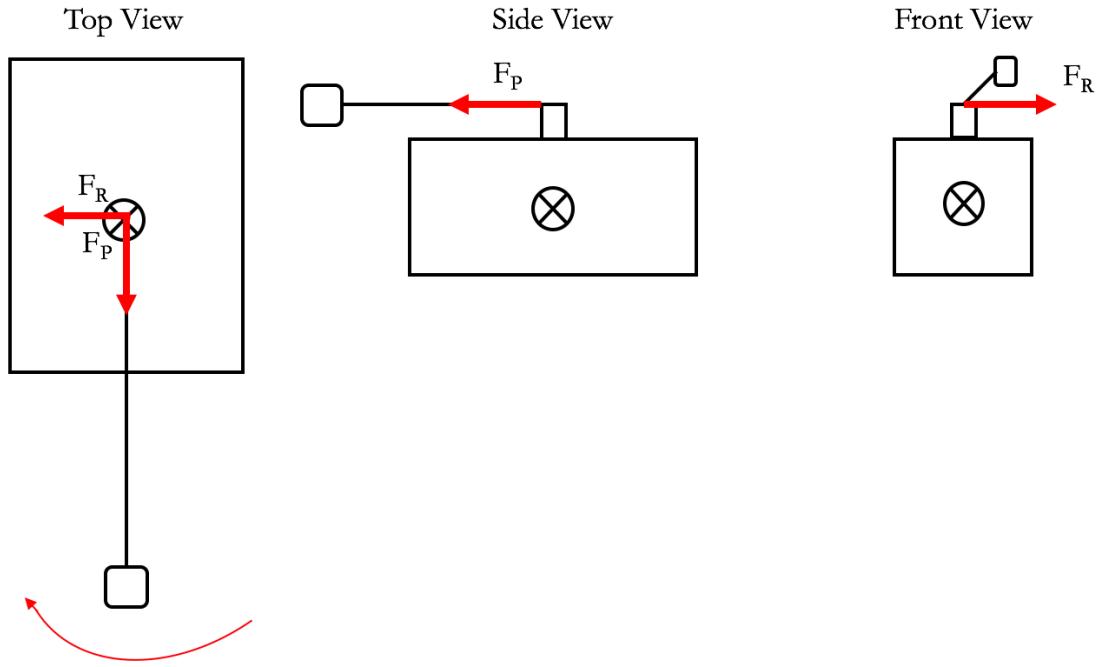


Figure 103: The dynamic analysis of pitch and roll effects that acted on the body, Left – is a top view of the robot and the two forces F_p and F_r are shown as well as the centre of mass, Middle – A side view of the robot is shown and the force F_p , Right – The front view of the robot is shown and the force F_r

During the stunt, the effects of these forces cancelled out through the first 270 degrees travelled by the body. This occurred because the tail travelled through roughly 360 degrees in that time, and the forces did the same. The net effect of the moments caused by the forces became zero.

The problems associated with the pitch and roll were experienced toward the end of the stunt. There are several videos that show that the body's attitude was perfect until right near the end of the stunt, where at the instant before landing the robot pitched downward and rolled clockwise, when viewed from the front. A video of this scenario is included on the attached disc and called "Pitch_and_Roll_Effects.mp4". These forces were unbalanced through the last quarter of the stunt, from a body angular position of 270 degrees through to 360 degrees, and they imposed moments on the body as a result of the tail joint not being positioned at the centre of mass.

10.3.2 Analysis of Roll Effects

The roll effects resultant of the force F_R explained above did not present a large problem to the stunt performance. Once the effects became evident, they were mitigated by curving the right side of the ramp (viewed from the front) upwards at the launch point. This meant that the robot began the stunt with a slight counter clockwise roll velocity. This initial velocity countered the large roll force experienced during the final portion of the stunt. Data acquired from a jump in which the roll effects were negated is shown in Figure 104 and Figure 105. The final roll positon is shown as -25 degrees, and corresponds to the robot just after turning right and before rolling due to the turn.

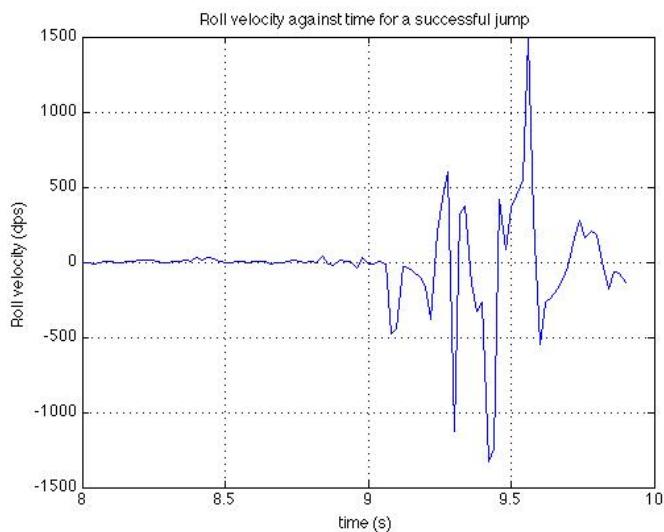


Figure 104: Roll velocity against time for a successful jump, presented to show the mitigation of roll effects

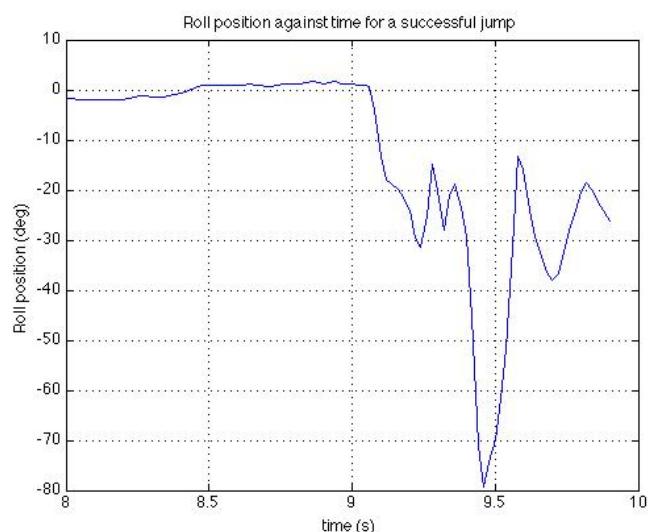


Figure 105: Roll position against time for a successful jump, presented to show the mitigation of roll effects

10.3.3 Analysis of Pitch Effects

The mitigation of the pitch effects experienced by the robot at the end of the stunt proved to be more challenging than for the roll case. The primary reason for this was the lack of power in the rear motor, since when the robot arrived at the launch zone, the rear motor was not generating sufficient linear acceleration required to cause an upward pitch upon take-off. If the gradient of the ramp was increased, the robot would develop even less acceleration, hence it would counterintuitively pitch downward even more.

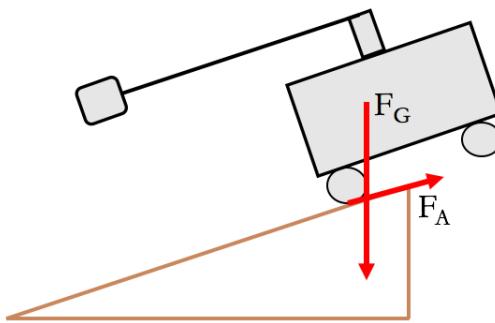


Figure 106: Illustration of the effect of linear acceleration on pitch during launch

The forces F_G and F_A shown in Figure 106 are responsible for defining the body pitch at launch. F_G is the force of gravity acting through the centre of mass of the combined body and tail system, and F_A is the force applied to the rear tires by the ground (reactionary frictional force). If the acceleration of the car is low, then F_G will dominate and the car will pitch clockwise.

The weakness of the rear motor meant that it was not possible to start the stunt with an adjusted pitch attitude, as was the case for the initial roll condition. Resultantly the pitch problem caused far more crashes than the roll problem.

10.3.4 Quantification of the Pitch Effect

An approximate quantification of the pitch force acting during the final portion of the stunt was obtained by considering the speed of the tail before damping action is initiated, and the time taken to bring the tail to a zero velocity (this occurs at the end of the stunt). The velocity of the tail during this deceleration can then be deduced, and consequently the pitch force, F_p , can be found. This analysis makes the assumption of a linear change in tail velocity, which is obviously not the case, but the approximation wasn't expected to skew the results substantially.

During a stunt when the car pitched sufficiently to cause a crash, the following information was extracted from the data (obtained from Figure 97) and presented in *Table 19*:

Table 19: Motor data pertinent to the final portion of the stunt

Data description	Value
Motor velocity pre-damping action	3100 rad.s ⁻¹
Tail velocity pre-damping action	41.33 rad.s ⁻¹
Time taken to zero tail velocity	0.05 s
Tail rotational inertia	3.8 x 10 ⁻⁴ kg.m ²

The robot's rotational inertia in the pitch plane was calculated with the same method as was used for the yaw plane, using equation 41:

$$I_p = \frac{mass}{12} (L^2 + H^2) = \frac{0.155}{12} (0.1175^2 + 0.07^2) = 2.416 \times 10^{-4} \text{ kg.m}^2 \quad (54)$$

The same reduction factor was applied for the same reason:

$$I_p = 0.75 \times 2.416 \times 10^{-4}$$

$$I_p = 1.812 \times 10^{-4}$$

The following equations (55 to 57) of rotational and linear motion were used to derive a differential equation for pitch position during damping action (the full derivation of which is shown in appendix A, section A4). The force causing the pitch and important distances for the calculations are shown in Figure 107.

$$a_{radial} = r\dot{\phi}^2 \quad (55)$$

$$F = ma \quad (56)$$

$$T = J \frac{d\phi}{dt} \quad (57)$$

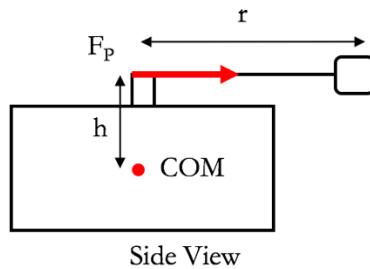


Figure 107: Illustration of the pitch force action on the body during the last quarter of the stunt

The pitch differential equation of motion is shown in equation 58.

$$\frac{d\dot{\theta}_p}{dt} = \frac{1}{I_p} m_t h r \dot{\phi}^2 \quad (58)$$

The constants in the equation represent body pitch plane rotational inertia, tail mass, perpendicular distance from tail joint to centre of mass, and tail length respectively. This equation was used in conjunction with the data in *Table 19* to quantify the pitch effect through the final portion of the stunt. A Matlab script was written to perform the double integration, and the resulting predicted pitch velocity, position and torque are shown in Figure 108 through Figure 110, along with empirical pitch position and velocity data collected from a pitch effect crash.

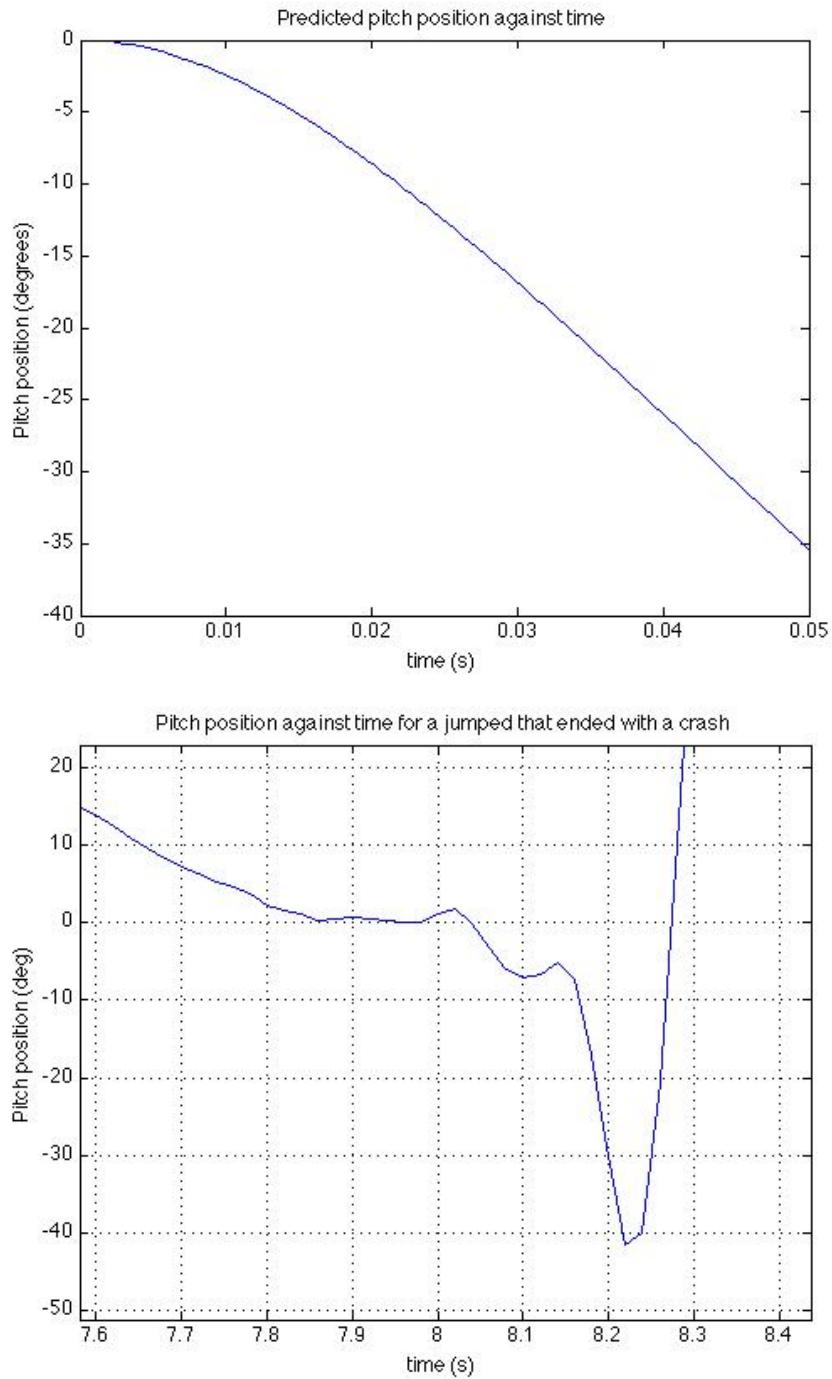


Figure 108: Top – Predicted pitch position and, Bottom – measured pitch position for the final portion of the stunt for a pitch effected crash jump

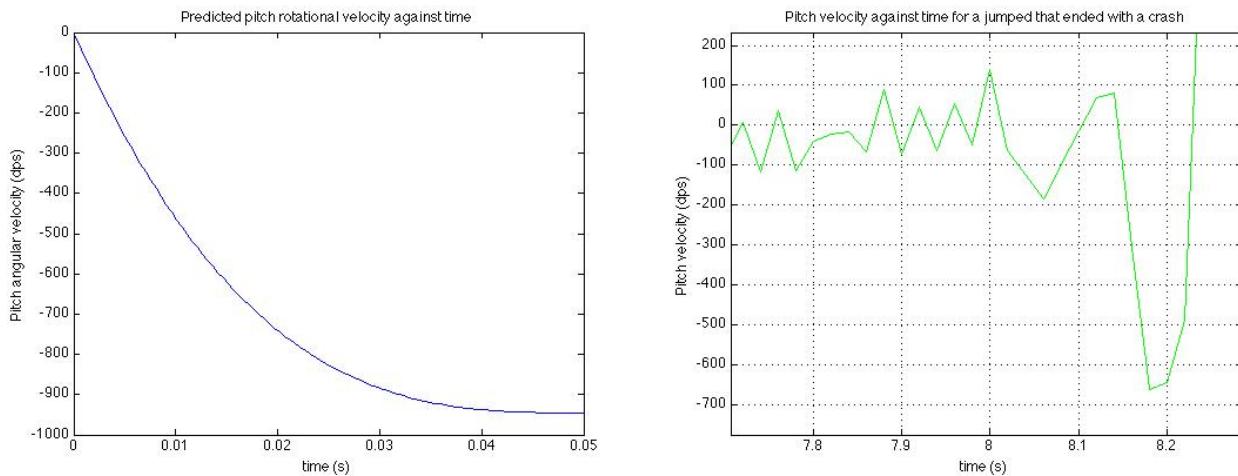


Figure 109: Left – Predicted and, Right – measured pitch rotational velocity for the final portion of the stunt for a pitch effect crash jump

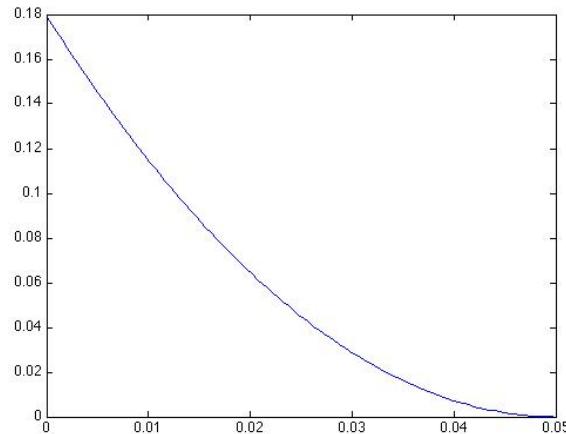


Figure 110: Pitch torque due to angular velocity of the tail mass for the final portion of the stunt

The second graph of Figure 108 shows a close view of the measured change in pitch position for the final part of the stunt. After landing, the robot pitched right back after bouncing, hence the upwards spike at time 8.22 seconds. The measured data shows that in the final part of the stunt, the robot pitched through approximately -45 degrees in the space of 0.05 seconds. The prediction in the first graph of Figure 108 suggested a pitch rotation of -35 degrees. This is slightly less than what was measured, but that was expected due to the assumption of a linear change of velocity. The prediction of the effects of the radial force due angular velocity of the tail mass were fairly well represented here.

The predicted and measured data for the pitch angular velocity, displayed in Figure 109, is also fairly accurate, although the measured data is far more volatile. The maximum change in velocity of the measured data was 800 dps, and the change predicted was 940 dps, which was also an accurate estimate.

The predicted torque acting on the robot in the pitch plane is shown in Figure 110, and this graphic illustrates the sizeable torque exerted on the platform. For context, it should be remembered that the motor stall torque is in the region of 0.16 Nm.

10.4 Suggestion to Solve the Pitch Effect Using the Rotational Inertial Ratio

Initially it was desired to keep the inertial ratio as low as possible, with the intention of keeping the tail rotation low for aesthetic purposes. However, after reviewing the test data and bearing the above pitch effect analysis in mind, this is likely to have been detrimental to the stunt performance.

The robot pitched forward during the last bit of the stunt because of where the tail was relative to the body during that portion of the stunt and the direction of torque applied by the motor. If the inertial ratio was higher, the tail would rotate even further than the current value of 540 degrees. If the ratio was adjusted such that the tail travelled through 720 degrees during the course of the stunt, the pitch and roll effects would cancel out during the second tail revolution, just as was seen during the first revolution during these stunts.

11. Conclusion and Recommendations for Future Work

11.1 Conclusion

The aim of this project was to model, design and build a bio-inspired robot to complete a 360-degree yaw stunt. Each facet of this process can be judged to have been carried out successfully, which ultimately culminated in the robot performing the stunt. The data collected during the study and the video footage captured of the testing supports these statements.

The theoretical models developed in the study were validated by data gathered in both the controlled and uncontrolled tests. The robot design and fabrication thereof endured rigorous testing, and proved to be exceptionally functional and robust. The integrated state feedback control method with a simple observer also exhibited high levels of performance, since the stunt was completed numerous times. This result ratifies the control design method selected.

The decision to model the dynamics assuming that the tail joint is at the centre of mass of the robot was necessary to reduce the complexity of the system, but did result in pitch and roll problems during the stunt. The roll effects were successfully nullified by test apparatus adjustment, but the pitch effects presented a problem that persisted, and could not be solved to make the stunt 100% repeatable. The pitch effects were subsequently quantified simply and relatively accurately.

This study has again shown the great advantages that can be obtained through bio-inspired robotic design, and more specifically the usefulness of a tail appendage on a robotic platform. This study has verified the ability of a robotic tail to provide attitude control and aerial orientation. The tail increased the manoeuvrability of the robot considerably – the yaw angular velocity achieved by the body consistently reached high values in the range of 1500 dps. The theory of attitude control and aerial orientation of robots using tails was explored in [6] and [34], but this is the first illustration of the principles in the yaw plane.

11.2 Recommendations For Future Work

11.2.1 Generation of a 3D Model for the System and Two Degree of Freedom Tail

The conclusions drawn from the study are clear that the optimal way to repeatedly perform the stunt would be for the tail to compensate for pitch and roll effects. In order to achieve this a tail mechanism must be design such that it has two degrees of freedom. A body coupled to such a tail design would need to be modelled in three dimensions, and would be substantially more complex than the models in this study. If a model could be developed and a system built in this configuration, it would be theoretically possible to perform the yaw stunt on every jump.

Beyond this, it would create the possibility to perform amazing aerial manoeuvres, including a 360-degree yaw and a backflip simultaneous.

11.2.2 The Use of a More Appropriate Platform

Fabrication and testing of the robot would have been greatly simplified by the use of a more appropriate platform. It is recommended that future work be conducted with a custom chassis, being robust enough to cope with impulses experienced during landing.

11.2.3 Control System Improvements and Optimization

Although the control system was judged to have perform well, improvements could be made. The controller did not compensate for frictional nonlinearities, and even smaller final errors could be attained if this is performed. Adaptive control systems have been shown to improve the management of friction in motors and could be implemented on the TAYObot [40].

The control system could also be enhanced by considering the effects of states having nonzero initial values. Control theory could be explored in search of optimal methods of coping with this.

The effect of the initial position of a tail and its ability to provide aerial righting was explored for the case of the Gecko in [19]. The initial position was shown to have a significant effect on the ability of the tail to provide the required rotation of the body [19]. Experiments of the effect of varying initial position of the tail in the robot in this study was not possible, as moving the tail to be partially perpendicular to the robot made the robot turn and it would have missed the ramp. The effects of initial tail position could be considered in future work.

12. List of References

- [1] Briggs, R, Lee, J, Haberland, M, Kim, S, "Tails in Biomimetic Design: Analysis, Simulation, and Experiment", *IEEE International conference on Intelligent Robots and Systems*, 2012.
- [2] Bar-Cohen, Y, Breazeal, C "Biologically inspired intelligent robots." *Smart Structures and Materials*, vol 1, pp. 14-20. International Society for Optics and Photonics, 2003.
- [3] LiveScience, "Evolution", September, 2015. [Online]. Available: <http://www.livescience.com/topics/evolution/>. [Accessed: Sept, 2015].
- [4] Bar Cogen, Y, "Biomimetics—using nature to inspire human innovation" *Bioinspiration and Biomimetics*, vol 1, no. 1, pp. 1-12, 2006.
- [5] Vanderbilt, T, "How Biomimicry is Inspiring Human Innovation", September, 2012. [Online]. Available: <http://www.smithsonianmag.com/science-nature/how-biomimicry-is-inspiring-human-innovation-17924040/?no-ist=&page=2>. [Accessed: 22 Jun, 2015].
- [6] Libby, T, Moore, TY, Chang-Siu, E, Li, D, Cohen, DJ, Jusufi, A, Full, RJ, "Tail-assisted pitch control in lizards, robots and dinosaurs." *Nature*, vol. 481, no. 7380, pp. 181-184, 2012.
- [7] "Biomimicry", June, 2010. [Online]. Available: <http://www.designboom.com/contemporary/biomimicry.html>. [Accessed: 10 Jul, 2015].
- [8] Schmitt, OH. "Some interesting and useful biomimetic transforms." *Third Int. Biophysics Congress*, vol. 1069, pp 197, 1969.
- [9] Benyus, Janine M. "Biomimicry: Innovation inspired by design", New York: William Morrow, 1997.
- [10] Erlhagen, W, Mukovskiy, A, Bicho, E, Panin, G, Kiss, C, Knoll, A, Van Schie, H, Bekkering, H, "Goal-directed imitation for robots: A bio-inspired approach to action understanding and skill learning." *Robotics and autonomous systems* vol 54, no. 5, pp. 353-360, 2006.
- [11] Martínez-Alier, J, Pascual, U, Vivien, FD, Zaccai, E, "Sustainable de-growth: Mapping the context, criticisms and future prospects of an emergent paradigm." *Ecological Economics*, vol 69, no. 9, pp. 1741-1747, 2010.

- [12] Sharpe, L, “Robotic whiskers could help robots navigate through dark and murky environments”, August, 2015. [Online]. Available: <http://www.popsci.com/robo-whiskers-could-help-robots-navigate-through-dark-and-murky-environments>. [Accessed: 9 Oct, 2015].
- [13] Zacaiah, B, “Biomimicry day 1”, July, 2013. [Online]. Available: <http://www.levinegabriella.com/exploringbiomimicry/ciid/2013/07/15/biomimicry-day-1-examples/>. [Accessed: 10 Oct, 2015].
- [14] Festo, “Scharnhausen Technology Plant”, June, 2015. [Online]. Available: <https://www.festo.com/group/en/cms/10967.htm>. [Accessed: 10 Oct, 2015].
- [15] Festo, “The AirPenguin”, Jun, 2009. [Online]. Available: https://www.festo.com/net/SupportPortal/Files/42070/airpenguin_en.pdf. [Accessed: 10 Oct, 2015].
- [16] Festo, “The Bionic Handling Assistant”, Dec, 2012. [Online]. Available: <https://www.festo.com/group/en/cms/10241.htm>. [Accessed: 10 Oct, 2015].
- [17] Festo, “The WavveHandler”, Jun, 2013. [Online]. Available: <https://www.festo.com/group/en/cms/10225.htm>. [Accessed 10 Oct, 2015].
- [18] Festo, “The BionicOpter”, June, 2013. [Online]. Available: <https://www.festo.com/group/en/cms/10224.htm>. [Accessed 10 Oct, 2015].
- [19] Jusufi, A, Zeng, Y, Full, RJ, Dudley, R, "Aerial righting reflexes in flightless animals." *Integrative and comparative biology*, vol. 51, no. 6, pp. 937-943, 2011.
- [20] Levi, M, “*Why Cats Land on Their Feet: And 76 Other Physical Paradoxes and Puzzles*”, Princeton University Press, 2012.
- [21] Bartholomew, GA, Caswell, HH, “Locomotion in Kangaroo rats and its adaptive significance”, *Journal of Mammalogy*, vol. 32, no. 2, pp. 155-169, 1951.
- [22] Pfeifer, R, Lungarella, M, Iida, F, “Self-Organization, Embodiment, and Biologically Inspired Robotics”, *Science*, vol. 318, pp. 1088-1093, 2007.
- [23] Johnson, A.M., Koditschek, D.E., “Robotic uses for tails”, July, 2015. [Online]. Available: <http://kodlab.seas.upenn.edu/TailsRSS2015/Main>. [Accessed: 20 Aug, 2015].

- [24] Chang-Siu, E, Libby, T, Tomizuka, M and Full, R.J., "A lizard-inspired active tail enables rapid maneuvers and dynamic stabilization in a terrestrial robot," in *Intelligent Robots and Systems*, San Francisco, CA, USA, 2011.
- [25] Johnson, A.M., Libby, T, Chang-Siu, E, Tomizuka, M, Full, R.J. and Koditschek, D.E., "Tail assisted dynamic self righting," in *Proceedings of the International Conference on Climbing and Walking Robots*, Baltimore, USA, pp. 611–620, 2012.
- [26] Zhao, J, Zhao, T, Xi, N, Cintron, F.J., Mutka, M.W. and Xiao, L, "Controlling aerial maneuvering of a miniature jumping robot using its tail," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, pp. 3802–3807, 2013.
- [27] Patel, A, Braae, M, "Rapid turning at high-speed: Inspirations from the cheetah's tail." In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 5506-5511, 2013.
- [28] Casarez, C, Penskiy, I, and Bergbreiter, S, "Using an inertial tail for rapid turns on a miniature legged robot," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, pp. 5469–5474, 2013.
- [29] Kohut, N, Pullin, A, Haldane, D, Zarrouk, D, and Fearing, R, "Precise dynamic turning of a 10 cm legged robot on a low friction surface using a tail," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, pp. 3299–3306, 2013.
- [30] Ackerman, J, Da, X, and Seipel, J, "Mobility of legged robot locomotion with elastically-suspended loads over rough terrain," in *Proceedings of the International Conference on Climbing and Walking Robots*, Baltimore, USA, July 2012.
- [31] Guan-Horng Liu, Hou-Yi Lin, Huai-Yu Lin, Shao-Tuan Chen, Pei-Chun Lin, "A Bio-Inspired Hopping Kangaroo Robot with an Active Tail," *Journal of Bionic Engineering*, vol. 11, no. 4, pp. 541–555, 2014.
- [32] National Geographic, "Cheetah (*Acinonyx jubatus*)", 2013. [Online]. Available: Cheetah Acinonyx. [Accessed: 20 Aug, 2015].

- [33] Kohut, N.J., Haldane, D.W., Zarrouk, D and Fearing, R.S., "Effect of inertial tail on yaw rate of 45 gram legged robot." In *Int. Conf. Climbing Walk. Robot. Support Technol. Mob. Mach*, pp. 157-164, 2012.
- [34] Callen Fisher, "A Lizard Inspired Stunt Robot", Bsc Hons thesis, University of Cape Town, Western Cape, 2013.
- [35] Garcia de Jalon, J, Bayo, E, "Kinematics and Dynamic Simulation of Multibody Systems. The Real – Time Challenge", Berlin: Springer-Verlag, 2009.
- [36] Yuh, J. "Application of discrete-time model reference adaptive control to a flexible single-link robot." *Journal of Robotic Systems* 4, no. 5, pp. 621-630, 1987.
- [37] Vandiver, J.K., Course 2.003SC, Engineering Dynamics, MIT Open Courseware, Topic, "Lagrange Dynamics", Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 2013.
- [38] Virgala, I, Frankovský, P, Kenderová, M, "Friction effect analysis of a DC motor", *American Journal of Mechanical Engineering*, vol 1, no 1, pp 1 – 5, 2013.
- [39] Armstrong, Brian, "Friction: Experimental determination, modeling and compensation." *Robotics and Automation, IEEE International Conference*, pp 1422 – 1428, 1988.
- [40] Canudas, C, Astrom, K, and Braun, K, "Adaptive friction compensation in DC-motor drives." *Robotics and Automation, IEEE Journal*, vol 3, no 6, pp 681-685, 1987.
- [41] Wolfram Alpha MathWorld, "Matrix Rank" July, 2015. [Online]. Available: <http://mathworld.wolfram.com/MatrixRank.html>. [Accessed: 1 Sept, 2015].
- [42] MIT Open Courseware, Course: Feedback Control Systems, "Controllability and state space features", Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 2010.
- [43] Braae, M, "Control theory for electrical engineers", Cape Town, South Africa: UCT Press (Pty) Ltd, 1996.

[44] Prof. Alan V. Oppenheim, Prof. George Verghese, Course 6.011, MIT Open Courseware, Topic: “State observers and state feedback”, Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 2010.

[45] Engineer on a disk, “Nonlinear systems”, June, 2010. [Online]. Available: <https://sites.google.com/site/engineeronadisk/dynamic-system-modelling-and-control/nonlinear>. [Accessed: 5 Oct, 2015].

[46] Industrial Devices, “Support, DC actuators”, August, 2015. [Online]. Available: <http://www.actuators-electric.co.uk/electric-linear-actuators-faqs/>. [Accessed: 22 Jul, 2015].

[47] Mohan, N, Underland, TM, Robbins, WP, “Power electronics, Converters, Applications, and Design”, Hoboken, NJ: John Wiley & Sons Inc, 2003.

[48] Diebel, J, "Representing attitude: Euler angles, unit quaternions, and rotation vectors." *Matrix*, vol. 58, pp. 15-16, 2006.

[49] Pololu Robotics & Electronics, “Pololu Metal Gearmotors”, June, 2015. [Online]. Available: <https://www.pololu.com/category/51/pololu-metal-gearmotors>. [Accessed: 28 Jun, 2015].

[50] Vishay Semiconductor, “IR Receiver Module For Remote Control System”, TSOP 38438, Datasheet, Oct 2012. [Revised 2015].

[51] Texas Instruments, “L293, L293D, quadruple half-H drivers”, L293, L293D Datasheet, Sept. 1986 [Revised 2004].

[52] Pololu Robotics & Electronics, “Pololu Motor Shaft Encoders”. [Online]. Available: <https://www.pololu.com/product/2598/pictures>. [Accessed: 28 July, 2015].

[53] Paralex Incorporated, “XBee 802.15.4 Low-Power Module, PCB Antenna”, 2015. [Online]. Available: <https://www.parallax.com/product/32404>. [Accessed 8 September 2015].

[54] Digi International Inc, “XBEE/XBEE Pro RF Module”, XBEE S2, Datasheet, Sept. 2009. [Revised 2009].

[55] St Electronics, “Inemo M-1”, Inemo M-1, Datasheet, Oct. 2013. [Revised 2013].

- [56] Hyper Physics, “Moments of Inertia”, June, 2015. [Online]. Available: <http://hyperphysics.phy-astr.gsu.edu/hbase/mi.html>. [Accessed: 30 Jun, 2015].
- [57] Lewis, FL, Applied Optimal Control and Estimation, “A Brief History of Feedback Control”, Electrical and Computer Engineering, Prentice Hall, Utah, 1992.
- [58] Braae, M, EEE 3069 W, Class lecture, Topic: “Frequency Domain control design methods”, Electrical Engineering, University of Cape Town, Cape Town, Western Cape, 2014.
- [59] Boje, E, Class lecture, EEE 4093 F, Topic: “Quantitative Feedback Theory control design methods”, Electrical Engineering, University of Cape Town, Cape Town, Western Cape, 2015.
- [60] Tsoeu, M, Class lecture, EEE 4093 F, Topic: “Modern Control Design Methods”, Electrical Engineering, University of Cape Town, Cape Town, Western Cape, 2015.
- [61] Williams, RL, Lawrence, A, Class lecture, Topic: “Linear State space control systems”, Hoboken, NJ: John Wiley & Sons Inc, 2007.
- [62] Coetser, J, “Low-cost Inertial Navigation System”, Bsc Hons thesis, University of Cape Town, Western Cape, 2013.
- [63] Reynolds, AM, "On the origin of bursts and heavy tails in animal dynamics." *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 2, pp. 245-249, 2011.

Appendix A:

Mathematic Proofs and Equation Derivations

A1: State Space Steady State Analysis

A1.1 State Feedback Configuration Steady State Resolution Mathematics

This section provides the mathematic resolution of the steady state conditions of a loop in state feedback. Adjusting the differential equation for the feedback configuration of the loop and subsequently substituting the result into equation 10 gives

$$\begin{aligned} u(t) &= r(t) - bK^T \\ \frac{d}{dt}x(t) &= (A - bK^T)x(t) + br(t) \end{aligned} \quad (\text{A1})$$

The steady state condition is that $\frac{d}{dt}$ be zero, and thus:

$$x_\infty = -(A - bK^T)^{-1}br \quad (\text{A2})$$

subbing this result into 11 gives:

$$y_\infty = -c^T(A - bK^T)^{-1}br + du \quad (\text{A3})$$

A1.2: State Space Integrated State Feedback Approach Mathematics

This section provides the mathematic proof to support argument made in section 3.4.2. Figure 21 shows that the error signal (or vector) can be set out as:

$$e(t) = r(t) - y(t) = r(t) - c^Tx(t) \quad (\text{A4})$$

Hence using equation A4 and equation 10:

$$\frac{d}{dt} \begin{pmatrix} x(t) \\ w(t) \end{pmatrix} = \begin{bmatrix} A & 0 \\ c^T & 0 \end{bmatrix} \begin{pmatrix} x(t) \\ w(t) \end{pmatrix} + \begin{bmatrix} b \\ 0 \end{bmatrix} u(t) - \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(t) \quad (\text{A5})$$

Incorporating the feedback

$$u(t) = -k^T x(t) + K w(t) = -[k^T \quad K] \begin{pmatrix} x(t) \\ w(t) \end{pmatrix} \quad (\text{A6})$$

Substituting (A6) into (A5)

$$\frac{d}{dt} \begin{pmatrix} x(t) \\ w(t) \end{pmatrix} = \begin{bmatrix} A & 0 \\ c^T & 0 \end{bmatrix} \begin{pmatrix} x(t) \\ w(t) \end{pmatrix} + \begin{bmatrix} b \\ 0 \end{bmatrix} [-k^T \quad -K] \begin{pmatrix} x(t) \\ w(t) \end{pmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(t) \quad (\text{A7})$$

The factorized form of equation A7 is shown in the text and used to motivate the argument made in section 3.4.2.

A2 Lagrange Dynamic Mathematics

A2.1 Derivation of the Dynamic Equations of Motion

The first equation of motion of the body was derived term by term using equation 26 and equation 2:

1. $\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\theta}} \right) = \frac{d}{dt} (I_B \dot{\theta} + I_T (\dot{\theta} + \dot{\varphi})) = I_B \ddot{\theta} + I_T (\ddot{\theta} + \ddot{\varphi}) \quad (\text{A8})$
2. $-\frac{\partial T}{\partial \theta} = 0$
3. $\frac{\partial V}{\partial \theta} = 0$
4. $Q_\theta = 0$

The second equation of motion was derived in the same manner:

1. $\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\varphi}} \right) = \frac{d}{dt} (I_T (\dot{\theta} + \dot{\varphi})) = I_T (\ddot{\theta} + \ddot{\varphi}) \quad (\text{A9})$
2. $-\frac{\partial T}{\partial \varphi} = 0$
3. $\frac{\partial V}{\partial \varphi} = 0$
4. $Q_\varphi = \tau \quad (\text{A10})$

A2.2 Resolution of Dynamic Equations to Simulation Compatible Form

This section provides the mathematic steps followed to isolate the acceleration variables of interest. Isolating in equation 28:

$$0 = I_B \ddot{\theta} + I_T \ddot{\theta} + I_T \ddot{\varphi}$$

Factorizing:

$$\ddot{\theta} = -\frac{I_T}{I_T + I_B} \ddot{\phi} \quad (\text{A11})$$

Isolation in equation 29:

$$\ddot{\phi} = \frac{1}{I_T} \tau - \ddot{\theta} \quad (\text{A12})$$

A3 Pole Placement of Loop 2

A3.1 Loop 2 Pole Placement

The pole settings for a settling time of 0.25ms and a damping ratio of 0.66 (chosen for speed since overshoot is acceptable in the fast loop) result in the following poles and characteristic equation:

$$s_1 = -4000 + j4500$$

$$s_2 = -4000 - j4500$$

$$s_3 = -100$$

The integrator pole is required to be slower as a consequence of the phase lag it introduces [61]. The corresponding gains were computed from the Eigen values of the augmented matrix A', as shown in equation 52. Matlab's symbolic toolbox was once again used in conjunction with the symbolic equation solver function to compute the desired gains.

A4 Derivation of the Differential Equation of Motion for Pitch Position During Damping Action

Substituting equation 55 into 56 gives:

$$F_p = m_t r \dot{\phi}^2$$

Using the fact that torque is the cross product of force and distance:

$$T_p = m_t h r \dot{\phi}^2$$

Substituting this result into 57 and rearranging:

$$\frac{d\dot{\theta}_p}{dt} = \frac{1}{I_p} m_t h r \dot{\phi}^2$$

Appendix B:

Sensor Calibration

The 360-degree yaw stunt relies directly on gyroscope readings, and the accuracy of the readings will have a profound effect on the control system performance. To this end, the gyroscopes on the Inemo board were calibrated before every jump test. The accelerometers were also calibrated, but less care was taken in this process as highly precise accelerometer readings were not required, they were merely used for timing purposes. The magnetometer could not be used due to interference from the motor's magnetic field.

B1 Calibration Methodology

B1.1 Accelerometer Calibration

In order to calibrate the accelerometers, the robot was mounted on a gimbal and data was logged with the robot in six positions: a positive and negative axis orientation for each of the three axes. The data was then compared to the nominal value of g (the acceleration due to gravity), and weighting factors for each axis were obtained. They were all in the region of 4.1 to 4.5.

B1.2 Gyroscope Calibration

Gyroscope readings typically vary with temperature [62]. For this reason there is a temperature sensor on the Inemo board to aid in the correction of this. In order to calibrate the gyroscopes, the robot was turned on and placed in the sun and data was logged, the purpose of this was to cause a temperature change in the sensor. The logged data was then run through a Matlab script written by Callen Fisher which generates a linear mapping of the gyroscope readings to temperature, as well as a zeroing constant term. The resultant dependencies are shown in Table 20.

Table 20: Gyroscope temperature dependency functions

Gyroscope axis	Temperature dependency
X axis	$x = 0.00017t - 11$
Y axis	$y = -0.000019t + 0.71$
Z axis	$z = -0.000057t - 0.17$

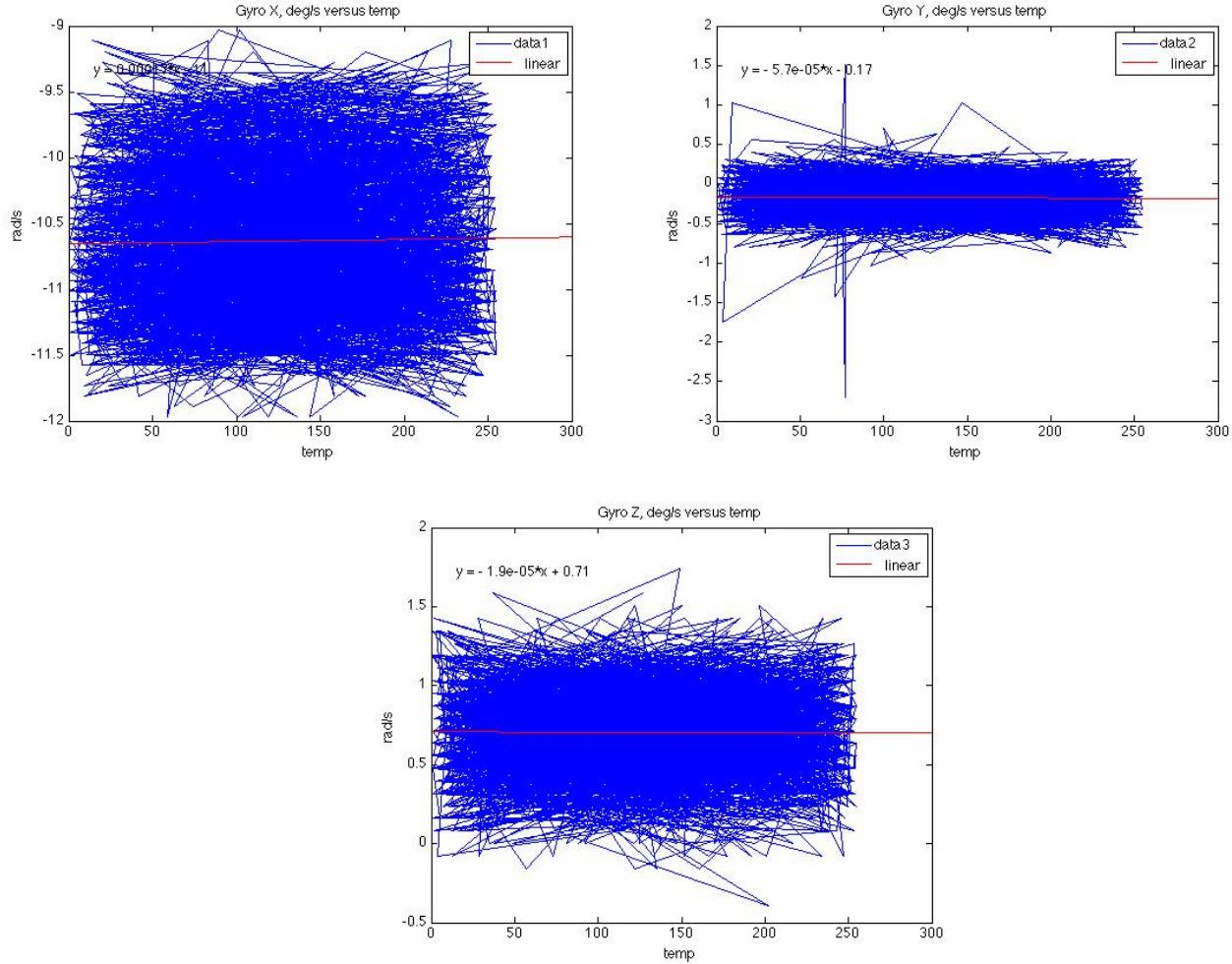


Figure 111: Gyroscope calibration graphs mapping gyroscope readings to temperature for all 3 axes

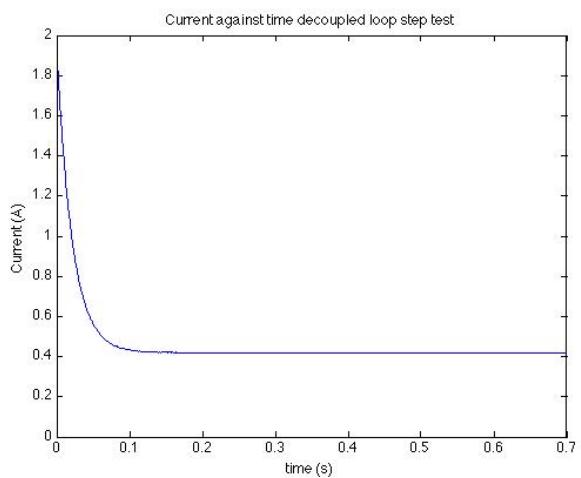
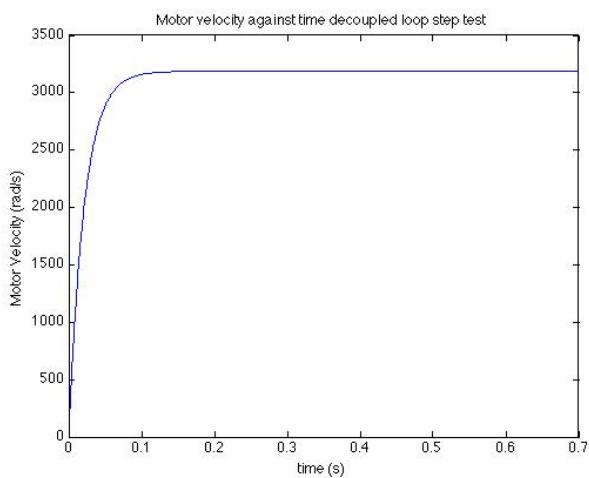
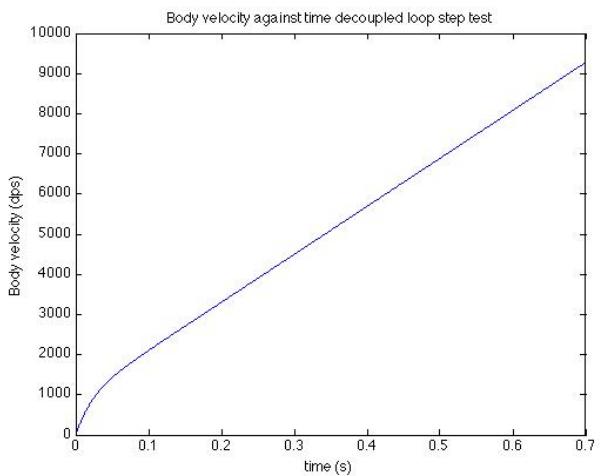
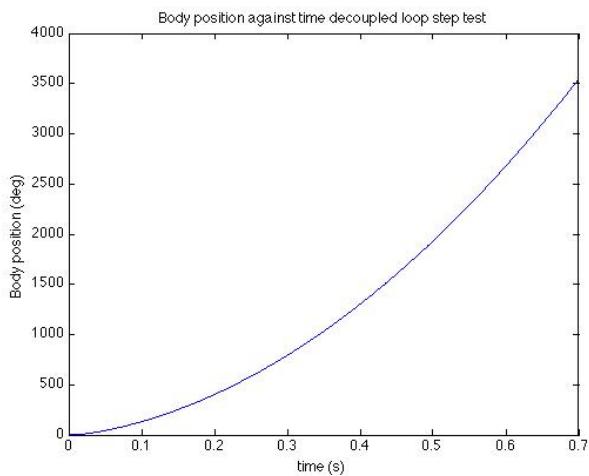
Figure 111 shows the calibration data that was logged and used to generate the equations in

Table 20. These equations were used to correct the gyroscope offset and temperature sensitivity.

Appendix C:

Auxiliary figures

C1: Decoupled Step Test Simulation Graphics



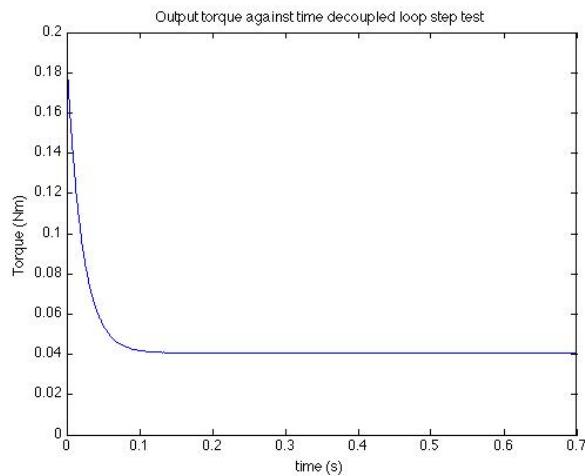


Figure 112: Step test graphics of each state for the modified decoupled system

C2: Motor Step Test Data

Table 21: Motor test data for various input terminal voltages, motor shaft speed, output shaft speed terminal voltage and armature current are all shown

Frequency (Hz)	Motor Shaft Speed (rpm)	Output Shaft Speed (rpm)	Voltage (V) _i (V)	Current (i_a) (mA)
0	0	0	0,25	82.0
100	2000	26,67	0,5	26.0
150	3000	40,00	0,68	28.5
200	4000	53,33	0,88	31.0
250	5000	66,67	1,06	33.0
300	6000	80,00	1,25	35.0
350	7000	93,33	1,45	37.0
400	8000	106,67	1,63	38.5
450	9000	120,00	1,83	40.0
500	10000	133,33	2,02	41.5
550	11000	146,67	2,21	43.0
600	12000	160,00	2,4	44.2
650	13000	173,33	2,57	45.0
700	14000	186,67	2,76	46.0
750	15000	200,00	2,955	47.2
800	16000	213,33	3,135	48.0

850	17000	226,67	3,34	49.0
900	18000	240,00	3,51	49.6
950	19000	253,33	3,72	50.5
1000	20000	266,67	3,9	51.4
1050	21000	280,00	4,15	52.0
1100	22000	293,33	4,275	52.5
1150	23000	306,67	4,48	53.0
1200	24000	320,00	4,685	55.0
1250	25000	333,33	4,85	57.0
1300	26000	346,67	5,04	58.5
1350	27000	360,00	5,26	60.0
1400	28000	373,33	5,49	62.0
1450	29000	386,67	5,74	62.5
1500	30000	400,00	5,93	63.5
1550	31000	413,33	6,14	65.0