



Faculty of Engineering and Materials Science
German University in Cairo

Design and Implementation of a Feedback System for One Legged Series Elastic Actuated (SEA) Robot

A thesis submitted in partial fulfilment of the requirements for the degree of
Bachelor of Science in Mechatronics Engineering

By
Ahmed Khaled Mady

Supervised by

Dr. Eng. Omar M. Shehata
Eng. Andrew Faried

July 31, 2022

This is to certify that:

- (i) the thesis comprises only my original work toward the Bachelor Degree of Science (B.Sc.) at the German University in Cairo (GUC),
- (ii) due acknowledgment has been made in the text to all other material used

Ahmed Khaled Mady

July 31, 2022

Acknowledgments

Working on this bachelor project was a very challenging task as the idea wasn't implemented several times before and resources were a little scarce. Yet, after much hard work and dedication, many ups and downs, I reached satisfying results and was able to stand up whenever I fell.

First, I would like to thank Allah for the Almighty, Gracious, and Most Merciful for giving me the strength to complete my work and research. Next, I would like to thank my family for tolerating my many busy days and handling me during my stress.

I would like to thank Dr. Omar Shehata for his support along the journey and for the motivation he was constantly giving and for this amazing chance he gave me to work with him and be a part of the MRS Team. I learned a lot from him and it was a pleasure working with him.

I would like to thank my colleagues, Youssef Al Mahdi, Hana Elzeky, and Mostafa Kashaf for your continuous support and help. It was fun working with you.

I would like to thank my friends, Amr Hassan and Marwan Sallam for tolerating my many questions and crying for help. You are one of the main reasons I could handle my stress. You always push me forward, encourage me, and make me discover new limits I didn't think I could reach. Thank you a lot.

Last but not least, I would like to thank Eng. Andrew Faried for his guidance and support throughout the semester. He exerted much effort with me and my colleagues and helped me manage many of my problems smoothly.

Abstract

A series elastic actuator can store energy in a compliant element such as a spring to facilitate locomotion and jumping abilities. Series Elastic Actuators (SEA) have unique features because of the elastic element, such as the capacity to store and release energy, tolerance to impact loads, and low mechanical output impedance. These advantages were tested using a feedback system consisting of a pulley system, belt, and an encoder to read deflection in the compliant element for series elastic actuator robotic legs. The Series Elastic Actuator (SEA), as a passive compliance actuator, was utilized to control the motion and force of a kangaroo and cheetah's robotic legs to provide a feedback system to compare the efficiency and stability. The feedback system was tested using a sinusoidal wave signal and PID force tracking was tested in simulations.

Contents

Acknowledgments	iii
Abstract	iv
List of Abbreviations	viii
List of Figures	ix
List of Symbols	xii
List of Tables	xiv
1 Introduction	1
1.1 Robotics	1
1.2 Industrial Robots	3
1.2.1 Applications	3
1.2.2 Benefits	4
1.2.3 Classification of the Robot	4
1.2.4 Industrial Robot Components	5
1.3 CoBots	5
1.3.1 The Benefits of CoBots	7
1.3.2 CoBots in Active Compliance	7
1.3.3 CoBots in Passive Compliance	8
1.4 Project Description	11
1.4.1 Problem Statement	11
1.4.2 Thesis Organization	11
2 Literature Review	12
2.1 Rigid vs Compliant Actuators	12
2.1.1 Rigid Actuators	12
2.1.2 Compliance Actuation	13

2.2	Series Elastic Actuator Background	15
2.2.1	Force-sensing Series Elastic Actuator (FSEA)	16
2.2.2	Reaction Force-Sensing Series Elastic Actuator (RFSEA)	17
2.2.3	Transmitted Force-Sensing Series Elastic Actuator (TFSEA)	18
2.3	Various SEA Control Techniques	19
3	Methodology	23
3.1	Mathematical Model	23
3.1.1	Spring Simulations	26
3.2	Force Control Loop Design	27
3.2.1	Motor RPM and Torque Relation	29
3.2.2	The choice of PID	30
3.2.3	Simulink Model	32
3.3	Force Control Hardware	34
3.3.1	The Actuator	34
3.3.2	Motor Driver	34
3.3.3	The Micro-controller	35
3.3.4	Power Supply	38
3.3.5	Rotary Encoder	40
3.3.6	Pulley System	43
3.3.6.1	Belt	43
3.3.6.2	Pulley	43
3.3.6.3	Pulley-Spring Relation	49
3.3.7	Circuit	51
3.4	Full Assembly	52
4	Results	54
4.1	Spring Simulations	54
4.2	Testing the Feedback System	56
4.3	Time Domain Specifications	58
4.4	Cheetah Robot	59
4.4.1	Tuning Choice	63
4.5	Kangaroo Robot	64
4.6	Force Tracking	68
5	Conclusion	71
6	Future Work	72
Appendix		73

List of Abbreviations

SEA	Series Elastic Actuators
RF-SEA	Reaction Force-Sensing Series Elastic Actuator
F-SEA	Force-sensing Series Elastic Actuator
TF-SEA	Transmitted Force-Sensing Series Elastic Actuator
UT-SEA	University of Texas Series Elastic Actuator
PID	Proportional-Integral-Derivative Controller
DOB	Disturbance Observer
RPM	Revolutions per minute
PWM	Pulse-Width Modulation
DIR	Direction signal

List of Figures

1.1	Mars 2020 Rover[1]	2
1.2	Industrial Robot in a Production Line[2]	3
1.3	Industrial Robot System: Major Components[3]	5
1.4	Cobots Empowering Humans in Manufacturing[4]	6
1.5	Bipedal Robot with a Compliant element,a spring, used for walking and running by Jonathan W. Hurst, Oregon State University[5]	9
2.1	This concept gives intrinsic capabilities (bandwidth, impacts, energy storage) over the joint stiffness range. However, two motors are required: one to control the equilibrium position and the second to control stiffness[6]	14
2.2	a. Model for a Series-Elastic Actuator b. Basic Components of a Series-Elastic Actuator[7]	15
2.3	Configuration of Force-sensing Series Elastic Actuator (F-SEA) [7]	17
2.4	Configurations of Reaction Force-Sensing Series Elastic Actuator (RF-SEA) (a) “motor reaction force” sensing type and (b) “gear reaction force” sensing type[7]	18
2.5	Proposed Control architecture by Pratt[8].	20
2.6	Proposed Control architecture by Paine, Q is a low-pass Butterworth filter[9].	21
2.7	The inner velocity loop in the velocity source SEA helps to overcome problems with non-linearities and stiction[10]	22
3.1	Model for RFSEA style actuator. The notations represent: F_m : motor force, F_o : output force, b_b : viscous backdriving friction, b_k : viscous spring friction, k : spring constant, x : spring deflection, m_k : lumped sprung mass, m_o : output mass, b_{eff} : lumped damping which equals $b_b + b_k$, and F_d : disturbance forces and forces which are difficult to model[11].	24
3.2	F_m is the force point, or motor force in our case, and r is the distance between the center of rotation, D is the diameter which equals $2r$ [12]	27

3.3	Motor current, speed, power, and efficiency characteristics compared to the torque. The 2 points, located on the same line, highlighted are the no load speed and stall torque. No load speed is on the y-axis: (0,5000) or (0,250) with considering the motor's gears, and the stall torque is located on the x-axis: (0.2646,0)[13]	29
3.4	PID Gains Representation[14]	30
3.5	Model for Cheetah leg	32
3.6	Model for Kangaroo leg	33
3.7	SG775125000-20K DC Geared Motor[13]	34
3.8	Cytron 10Amp 7V-30V DC Motor Driver[15]	35
3.9	Arduino UNO Pinout[16]	36
3.10	Arduino UNO R3 Board used[17]	37
3.11	Power supply 12V/10A[18]	39
3.12	Power source and its connections to the motor driver on Fritzing software	39
3.13	Encoder Disk Output square signals[19]	40
3.14	Encoder Disk Output Clockwise Rotation and Counter-clockwise rotation[19]	41
3.15	Rotary Encoder Pinout[20]	41
3.16	Rotary Encoder [20]	41
3.17	The encoder placed on pulley in the actuator	42
3.18	GT2 Timing Belt 2mm [21]	43
3.19	Holder for the pulley on SEA plates. [22]	44
3.20	Pulley attached on actuator plates [23]	44
3.21	Pulley for encoder[23]	44
3.24	Pulley system	45
3.22	Pulley System for spring deformation measurement designed on Solid works Software	46
3.23	Placement of the pulleys on the L-corners on the actuator front and middle plates	47
3.25	Pulley system assembled with belt	48
3.26	Pulley system assembled with belt and motor	49
3.27	Circuit Implemented showing motor, rotary encoder, power source, motor driver, and Arduino	51
3.28	Full Control Panel Assembled with actuator	52
3.29	Cheetah Robot Leg	53
3.30	Kangaroo Robot Leg	53
4.1	Closed-Loop step response for spring with stiffness 3088 N.m	55
4.2	Closed-Loop step response for spring with stiffness 4904 N.m	55
4.3	The waveform signal generated by Arduino	56
4.4	The output generated from the encoder output pins reading as spring deflection.	57
4.5	Cheetah Robot without Controller Simulink Model	59

4.6	Cheetah Robot without Controller Simulink Step plot	59
4.7	Model for Cheetah leg with controller added to be tuned	60
4.8	Cheetah Robot when tuning Controller on Simulink	60
4.9	Closed-Loop Step plot of the Cheetah Robot	61
4.10	Bode Plot for Cheetah Robot Leg, Frequency Response	62
4.11	Step Plot showing the new slower response vs the faster response	63
4.12	Kangaroo Robot without Controller Simulink Model	64
4.13	Kangaroo Robot without Controller Simulink Step plot	64
4.14	Model for Kangaroo leg with controller added to be tuned	65
4.15	Kangaroo Robot when tuning Controller on Simulink	65
4.16	Closed-Loop Step plot of the Kangaroo Robot	66
4.17	Bode Plot for Kangaroo Robot Leg, Frequency Response	67
4.18	Open-Loop Response for exponential chirp signal. The yellow signal is the input signal and the blue is the output signal.	68
4.19	Closed-Loop Response for exponential chirp signal. The yellow signal is the input signal and the blue is the output signal.	69
4.20	Link providing the videos of testing both the feedback system and the PID controllers on hardware.	70

List of Symbols

$\ddot{\theta}_m$	Acceleration of the Motor
η	drive-train efficiency
ω	Motor's RPM
ω_n	No Load Speed
τ_L	Load torque
τ_m	Motor torque
τ_s	Motor's Stall Torque
τ_{des}	Desired Torque
θ_m	Motor angle
a	acceleration from Newton's 2nd Law
b_b	Viscous Backdriving Friction
b_k	Viscous Spring Friction
b_{eff}	Lumped Damping which equals $b_b + b_k$
$e(t)$	error signal
$e[k - 1]$	previous error signal
$e[k]$	error signal in discrete
F_d	Disturbance Forces and Forces which are difficult to model
F_k	Measured Spring Force
F_m	Motor Force
F_o	Output Force
F_{des}	Desired Force Profile

F_{meas}	Measured Spring Force
i_m	Motor Current
J_E	Actuator's own effective internal inertia
k	Spring Stiffness Constant
K_d	PID Derivative Gain
K_i	PID Integral Gain
K_p	PID Proportional Gain
l	Ball Screw lead
m_o	Actuator's Output
M_p	Peak Overshoot
N_p	Pulley Reduction
P_n	UT-SEA Dynamics
$r(t)$	set-point or desired value
T	Sampling time
T_d	Derivative Time constant or Derivative Time
T_i	Integral Time constant or Integral Time
t_P	Peak Time
t_r	Rise Time
t_s	Settling Time
$u(t)$	PID correction signal
$u[k]$	PID correction signal in discrete
v	velocity
x	Spring Deflection
$y(t)$	output value
rpm	Revolutions per minute

List of Tables

Chapter 1

Introduction

1.1 Robotics

Robotics is a field that combines science, engineering, and technology to create machines that replace (or imitate) human actions. A robot is a programmable mechanism that can assist people or emulate human actions. Robots were originally designed to perform repetitive activities (such as making vehicles on an assembly line), but they have since evolved to accomplish duties such as fighting fires, cleaning homes, and aiding with extremely complex procedures. Each robot has a different amount of autonomy, ranging from fully-autonomous bots that perform tasks without any external influences to human-controlled bots that conduct tasks under full human direction.

The breadth of what is called robotics expands as technology advances. In 2005, 90 percent of all robots were working in auto manufacturers, building cars. These robots mostly comprise of mechanical arms that are tasked for welding or screwing on specific automotive parts. Today's definition of robotics has grown and expanded to include the development, creation, and deployment of bots that explore Earth's harshest environments, robots that aid law enforcement, and even robots that assist in practically every aspect of healthcare[24].

While the field of robotics is growing, there are some features that all robots share:

1. All robots have some type of mechanical structure. A robot's mechanical element aids it in completing duties in the environment for which it was created. The wheels on the Mars 2020 Rover, for example, are independently powered and composed of titanium tubing to help it grip the hard terrain on Mars in Figure 1.1.



Figure 1.1: Mars 2020 Rover[1]

2. Electrical components are required by robots to manage and power the machines. A huge majority of robots, in essence, require an electric current (for example, from a battery).
3. At least some amount of computer programming is present in robots. A robot would be little more than a piece of simple hardware if it didn't have a set of instructions directing it what to do. A program offers a robot the capacity to understand when and how to do a task.

Robots will become smarter, more flexible, more energy efficient in the near future as artificial intelligence and software progresses. They'll also be a key focus in smart manufacturing, where they'll tackle increasingly challenging difficulties and contribute to the security of global supply networks.

Human labour cannot match the speed and accuracy provided by robots. Robots can also save money by lowering operational expenses, reducing scrap, and being adaptable to future developments. The capabilities of robotics have only grown over time, while costs have decreased. Major robot manufacturers are continually improving their robots to improve payload capacity, accuracy, reach and range of motion, speed and acceleration, connectivity with external equipment, safety features, and operational expenses. Robots are now user-friendly, intelligent, and cost-effective[24].

Robotics Research Groups shaped the Robotics industry by advancing the field of robotics through contributions in machine vision, computer graphics, AI engineering tools, computer languages, autonomous robots, advanced factory applications, field robotics, tactical mobile robots, and pipeline robots. Robotics forward then is used in advancements in assembly, parts feeding, parcel handling, and machine vision are all examples of factory applications. Field robotics, in contrast to manufacturing, entails robotic applications in highly unstructured environments such as reconnaissance, surveillance, and explosive ordnance disposal. Tactical mobile robots are being developed for unstructured environments in both military and commercial purposes, similar to field robotics, to enhance human capabilities such as searching

through wreckage following disasters (earthquakes, bombed buildings, etc.)[24].

1.2 Industrial Robots

An industrial robot is a versatile, automatically controlled, re-programmable manipulator with three or more axes that can be fixed or mobile for use in industrial automation applications. An example of an industrial robot is shown in Figure 1.2. Industrial robots are used as material handling devices (moving work components between machines) or in additive (assembly, welding, gluing, painting, etc.) or subtractive (milling, cutting, grinding, de-burring, polishing, etc.) production processes.

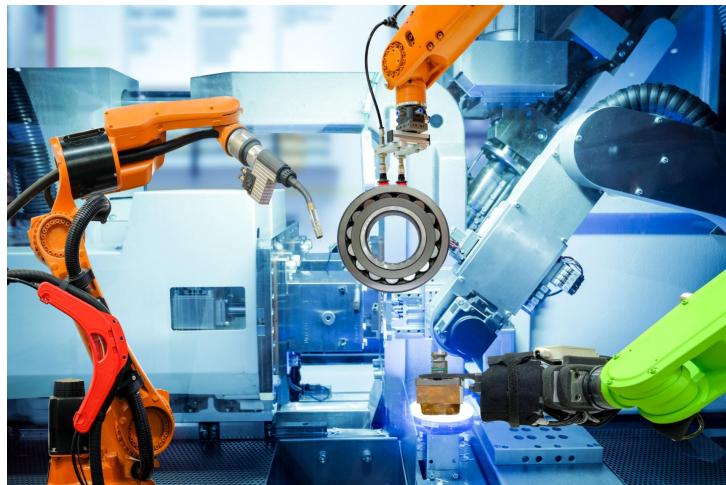


Figure 1.2: Industrial Robot in a Production Line[2]

1.2.1 Applications

Industrial robots are used in a variety of applications. These include:

- Handling: Capable of manipulating products as diverse as car doors to eggs, industrial robots are fast and powerful as well as sensitive. Applications include pick and place from work lines to packaging.
- Sealing and gluing: To apply sealant or glue, a robot follows a path accurately with good control over speed while maintaining a consistent bead of the adhesive substrate.
- Spraying: Due to the volatile and hazardous nature of solvent-based paints and coatings, robots are used in spray applications to minimize human contact.
- Welding: Used for spot welding, robots produce precise welds, as well as control parameters such as power, wire feed and gas flow.

- Industrial robots are used in many industries, including: Aerospace, Automotive, Beverage, Computers, Consumer goods, E-Commerce, Electronics, Food, Grocery, Hardware, Healthcare, Manufacturing, Medical products, Pharmaceutical, Quality control and inspection, Retail, and Warehousing and distribution.

1.2.2 Benefits

Industrial robots have a number of advantages:

- Accuracy: Robotic palletizers are programmed to put loads correctly.
- Flexibility: Robotic systems can be repurposed for different tasks, and end effectors can be swapped out to accommodate various load types.
- Lower labour costs - Automated pallet construction relieves worker strain and allows operators to focus on other tasks.
- Quiet operation: Servo-based robotic palletizers produce low noise levels during operation.
- Reduced product damage: Product damage is reduced – Package and product damage can be avoided with gentle handling.
- Speed: The systems can boost productivity by up to 50% [24]

1.2.3 Classification of the Robot

Armored robots, SCARA robots, PUMA robots, DELTA robots, KUKA robots, and Cartesian coordinate robots are the most frequent robot configurations. Most sorts of robots would be classified as robotic arms in the context of general robotics. Robots have several levels of autonomy:

- Some robots are programmed to repeat specific tasks over and over again (repetitive behaviours) with a high degree of accuracy and consistency. The direction, acceleration, velocity, deceleration, and distance of a series of coordinated motions are determined by programmed procedures.
- Other robots have a lot more flexibility when it comes to the orientation of the thing they're working on, or even the task that has to be done on the object itself, which the robot may need to distinguish. Robots, for example, frequently include machine vision sub-systems that operate as their "eyes" and are coupled to sophisticated computers or controllers for more accurate guiding. Artificial intelligence, or at least what passes for it, is becoming a more crucial role in today's industrial robots.

1.2.4 Industrial Robot Components

Manipulator, Controller, and Tooling are the three essential components of an industrial robot.

- Manipulator: It comprises of the robot's base and arm, as well as the power source, which might be electrical, hydraulic, or pneumatic. The manipulator is a gadget that allows movement in several degrees of freedom.
- Controller: The robot controller and the robot's multi-axis mechanical structure give the robot its flexibility. The capacity to reprogram the robot controller allows the robot to execute a broad variety of tasks.
- Tooling: Tooling is what allows the robot to do a certain task. End-effectors and tooling are sometimes used interchangeably, however the latter has a more limited definition that refers to end-of-arm fixtures that grip, lift, or turn. Tooling, on the other hand, has a broader definition that includes power tools such as drills and grinders, as well as paint and welding guns. There are six fundamental movements or degrees of freedom that allow the robot to move the end-effectors through the desired motion sequence. Three arm and body motions and three gripper motions make up the six motions.

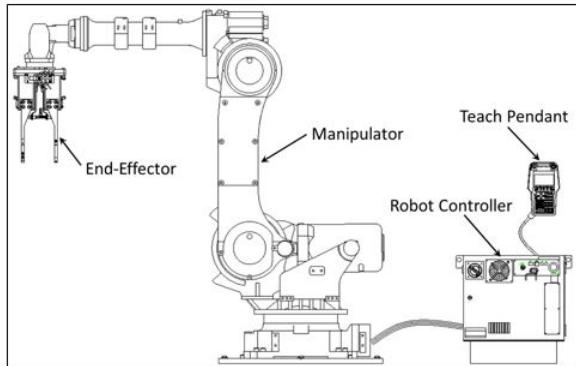


Figure 1.3: Industrial Robot System: Major Components[3]

1.3 CoBots

In many industries, physical robots are in high demand, particularly for duties that are described as the 4-D's:"dirty, dangerous, dear (or expensive), and dull" (or Demeaning). These robots do duties that would otherwise be performed by humans on a daily basis in manufacturing, warehouses, health care, and other settings, with mixed results. Industrial robots, on the other hand, must frequently be removed from actual human interaction in order to perform reliably without inflicting bodily injury to humans. This entails working in completely human-free zones or in cages that prohibit unintentional human interaction. Alternatively, if they are free to explore the globe, their strength and capability are limited, preventing them from causing harm.

Physical robots' application and power are, however, limited by this constraint. Companies who want to automate and enable more of their business processes that need physical human labour must find ways to expand the contact between robots and people without jeopardising their safety.

To operate properly, the industrial robots that come to mind when you think of physical bots operating in 4-Ds environments require isolation from humans. However, "collaborative robots," sometimes known as "cobots," are designed to work alongside and in close proximity to humans to complete tasks. Cobots, unlike their more isolated competitors, are designed to engage physically with humans in a shared office.

Humans would provide the power to move the machines, while cobots would give the control and steering to precisely arrange goods. In this approach, people were safe since they had control over the robot's power while still benefiting from the machine's assisting skills as shown in Figure 1.4. Cobots augment and expand human capabilities with super strength, precision, and data capabilities, allowing them to do more and deliver more value to the enterprise.



Figure 1.4: Cobots Empowering Humans in Manufacturing[4]

Traditional industrial robots are not trained the same way as cobots. Many cobots are trained by people controlling the arms and training by example, rather than being programmed to a precise set of steps using programming tools. Humans move the bot about physically, with the cobot remembering the steps and possibly even the final aim of what is being done, and then repeating and optimising those processes to get increasingly better results.

It's apparent that collaborative robotic solutions can now intelligently automate a variety of automation and intelligence jobs that previously couldn't be addressed by dangerous to deal with robots. If hardcore robotic adopters like automotive and industrial solutions can find applications for cobots, then surely the softer manufacturing, logistics, supply chain, warehouse,

and even even retail and consumer industries can. More crucially, the cobot's entire concept is based on its capacity to cooperate and work in close proximity to humans. These cobots are designed to help humans be more effective, efficient, and enhanced in their work. Indeed, if traditional robotics was centred on the four D's, cobots are likely to be focused on efficiency, effectiveness, and enhancement.

1.3.1 The Benefits of CoBots

Cobots have a few unique features that make automation possible for a wider range of businesses. Check out the benefits of cobots below.

- Compact: Cobots are small, compact robots and can therefore be used almost anywhere in a production process without taking up too much space.
- Installing and Programming: A cobot is easy to install by anyone and simple to program. With handy apps and software for smartphones and desktops, a cobot is operational in no time.
- Flexible: A cobot can easily learn new operations and is therefore able to work in different places in the production process.
- Mobile: Cobots are not heavy and easy to move around. Mounted on a mobile work-bench, they can easily perform new tasks at a different location within a company.
- Consistent and Precise: Cobots always perform actions in the same way with exactly the same force. This ensures equal quality and accurately placed parts.
- Positive Effect on Employees: Employees are spared monotonous or dangerous actions and are enabled to develop themselves by doing more creative work.
- Reduction of Production Costs: Through the use of cobots, processes are streamlined and production goes up. Ultimately, this leads to a better bottom line.

1.3.2 CoBots in Active Compliance

A strong force control is necessary for specific jobs, such as accurate load placement and human joint actuation as in rigid actuators. Humans are better at controlling force than ordinary robots and stiff actuators. Actuators are preferred to be as rigid as feasible in traditional robotic applications to allow for more precise position movements. Humans, on the other hand, have no trouble contacting hard surfaces and performing precise procedures which is referred to as compliance actuation.

When compared to standard robotic actuators, muscle has a superior power-to-weight ratio, force-to-weight ratio, compliance, and control, which are the key challenges to developing

robots that can match the mobility, safety, and energy efficiency of humans and other animals. The compliance or springlike behaviour exhibited in biological systems is one of these systems' major distinctions. Although such compliant actuators are likely less appropriate for traditional position-controlled applications, they provide significant benefits in a variety of unique applications, including safe human–robot interaction and the design of legged robots [25].

The most important necessity for cobots is safety. The goal of the safety assurance is to prevent any collisions between a robot and a human, an obstruction, or another moving item. To be employed in real-time cobot system controls, the probable collision must be quantifiably analysed.

- Safety-rated monitored stop (SRMS): In a safety-rated monitored stop (SRMS), a cobot stops running when a human interacts with it. The interaction between humans is monitored using the defined safety standards. As a result, the collaborative work is not completed simultaneously by the robot and the human. Picking and putting, inspection, human-assisted assembly operations, and robot-assisted placements of large goods are all common cobot applications in manufacturing[26]..
- Hand guiding (HG):In hand guiding, a human guides the cobot by hand contacts; in other words, the cobot is programmed on the fly by manual teaching. To this end, a task of a cobot was defined as a set of skills, and the control program was the collection of sub-programs generated by kinesthetic teaching[26].
- Speed and separation monitoring (SSM): In speed and separation monitoring (SSM), the distance and speed of a human relative to the cobot determines the motion direction and speed of the cobot. The cobot is placed in a shared workspace. The shared workspace is divided based on the safety levels, and safety levels are determined by the distance of the cobot from the human. The cobot runs at a full speed if no human is within the green zone, the cobot runs at a reduced speed when a human is approaching the robot (in the yellow zone), and the cobot stops when the human reaches the red zone[26].
- In power and force limiting (PFL), the contact force or power is restricted to avoid a potential injury to human; it is implemented by limiting the driving forces of joints at the design phase of cobot. If the event of contact is detected, either the motor brakes of cobot are activated[26].

1.3.3 CoBots in Passive Compliance

The main advantage of legged locomotion over wheeled locomotion is that legs have the capability of climbing rougher terrain than wheeled or tracked vehicles. Unfortunately, this ideal is often not achieved in reality, especially for the current generation of bipedal humanoid robots. Many walking controller implementations for humanoid robots assume perfectly flat surfaces,

and even a slight deviation in the floor can lead to serious instabilities in these controllers. One of the most difficult problems in this field is the robust balance of the walk, not only on even floor but on surfaces with irregularities and slopes too. Therefore, compliant elements are introduced in biomimetic robots. Biomimetic robotics is the study of biologically inspired robots, based on principles and ideas taken from living organisms. An example of a biomimetic robotic can be a humanoid robot inspired from a human.

In a human, the biceps femoris, semitendinosus and semimembranosus make up the hamstring muscles on the back of the thigh. These muscles work together to extend the thigh and flex the knee. While a human is walking on an incline, the hamstrings primarily work to extend the thigh on the leading leg as the back leg is moved to the front. In robots, muscles are similar to the complaint element in the system. The human leg can sense the incline or the irregularities and, therefore, compensate in the balance of the rest of the body motion. In robots, springs or elastic bands can be used for energy storage and, in turn, supply the necessary energy needed to balance the bipedal robot on different grounds whether irregular or normal as can be shown in Figure 1.5 [27].

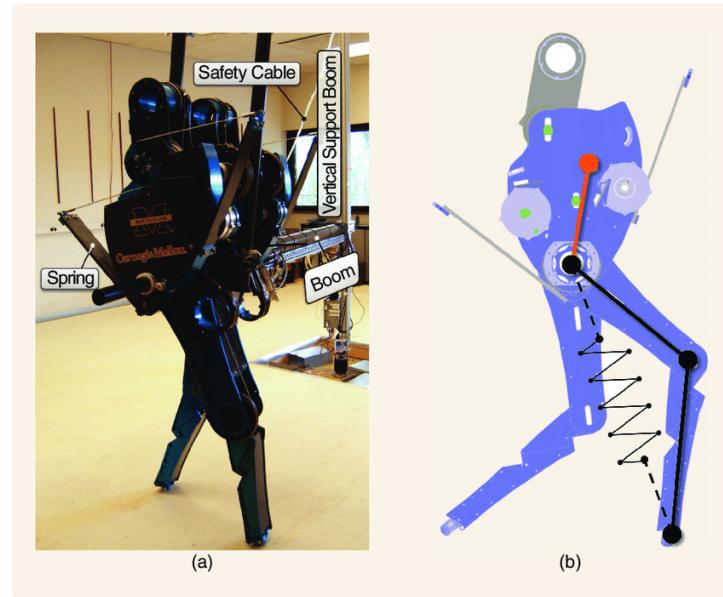


Figure 1.5: Bipedal Robot with a Compliant element,a spring, used for walking and running by Jonathan W. Hurst, Oregon State University[5]

Cobots are used in prosthetics because of their low weight, low energy consumption, safety and controllability[28]. Legged prosthetics in animals or humans may need a replacement for muscles that are used for locomotion or jumping routines. Compliant elements are installed in prosthetics to aid in the balance, locomotion, jumping, and other operations that are helpful in daily activities or routines[28]. One of these routines, jumping, is presented thoroughly in this paper.

The target is to compare the efficiency and the stability of two biomimiced robotic legs which are the cheetah and the kangaroo. As a significant type of locomotion with the reputation of high mobility, jumping motion can support animals which live in forests, jungles or marshes to pass over obstacles in complex environments, escape natural predators and obtain sustenance. Compared with walk and crawl, jump has advantages of high energy density, efficient obstacle negotiation, rapid terrain transition and so on. A robot via jump may move through different heights and irregular terrain, and further may realize more convenient freight transportation, patient care, disaster relief and rescue, interstellar exploration, etc[29].

1.4 Project Description

1.4.1 Problem Statement

A series elastic actuator has the ability to store energy in a compliant element such as a spring to facilitate locomotion and jumping abilities. Series Elastic Actuators (SEA) has unique features because of the elastic element, such as the capacity to store and release energy, tolerance to impact loads, and low mechanical output impedance. The Series Elastic Actuator (SEA), as a passive compliance actuator, will be utilised to control the motion of a kangaroo and cheetah robotic legs in order to provide a feedback system to compare the efficiency and stability of each.

1.4.2 Thesis Organization

- Chapter 1: Introduction to Robotics and specially Industrial Robots and their applications. CoBots are introduced and how they are helpful and very useful in industrial applications and engineering with an overview of what is and the different types of SEA.
- Chapter 2: A literature review is presented which summarizes the knowledge of a particular area or field of study, it also evaluates what research and design has been done, what still needs to be simulated/applied and why all of this is important to the industry.
- Chapter 3: The mathematical model is derived along with the design of the force feedback control loop implemented and the hardware components implemented.
- Chapter 4: Shows the results obtained from the simulations done and hardware testing.
- Chapter 5: The Conclusion of the thesis.
- Chapter 6: Future work advice and Recommendation.

Chapter 2

Literature Review

2.1 Rigid vs Compliant Actuators

”The stiffer the better” is a traditional assumption of excellent design when it comes to the mechanical interaction between motors and loads. Several writers have already investigated the function of interface compliance in stabilising force management during contact transitions and strategies for managing inevitably flexible structures (such as those found in space). However, with the exception of systems where energy storage is critical (such as the legs of a hopping robot) and some passive hand mechanisms, few people recommended that elasticity be included in general-purpose robotic actuators[8].

2.1.1 Rigid Actuators

The ”stiffer is better” rule of thumb in robotics was born out of the fact that higher stiffness enhances position-control accuracy, stability, and bandwidth. A stiff actuator is a mechanical device that can move to a precise location or follow a predetermined path. Whatever external pressures are applied to the actuator (within the device’s force limitations), once a position is attained, it will remain there[8].

A stiffer actuator has larger bandwidth force control and accurate position control. Increased interface stiffness reduces end-point position error under load disturbances when either open-loop positioning or co-located feedback is employed. Increased stiffness decreases required adjustments in response to load changes and enhances the resonance frequency of the motor inertia and interface compliance in non-co-located feedback systems (where the position sensor is located on the load side of the interface). As a consequence, the position control feedback loop’s bandwidth may be increased without affecting stability[30].

Force control, on the other hand, is challenging because it generates huge forces from tiny displacements, and a minor position error might result in a significant force error. As a result, the sensor must be extremely accurate and have a high resolution. Furthermore, in a non-structured environment, a stiffer actuator might cause accidents with people or injuries[30].

Friction, stick-slip, backlash, and reflected inertia via the transmission, cogging in motors, and pressure drop in hydraulic circuits, among other factors, prevent most common robot actuation systems from creating exact force on the robotic joints. Despite these non-linearities and disturbances in force readings, certain robots are well aligned for position or trajectory control, as the robot's mass and actuators operate as a low-pass filter for the force on position output[8].

Even with position-controlled systems, however, using a rigid interface comes at a cost. Because most electric motors have a low torque density, they can only achieve high power density at high speeds. Gear reduction is required to accelerate or sustain high loads. Gears, on the other hand, cause friction, backlash, torque ripple, and noise. The usage of N:1 gearing also creates a N^2 increase in reflected inertia, resulting in substantially larger pressures on the gear teeth when shock loads are applied. Peak torque is generally limited by the load rating of the gearing rather than the motor in light-weight actuators, and gearing failures due to shock are not unusual. Finally, when unexpected contact occurs, the higher reflected inertia and high back-drive friction of gear trains can cause environmental harm. Because the drawbacks of employing gears are so significant, industrial robots can benefit from direct driving. However, the power density and force density of direct drive are still insufficient for mobile robots[30].

2.1.2 Compliance Actuation

The needed range of compliance, as well as the actuator's torque requirement, is dependent on the application. Most controllable stiffness actuators are made up of two traditional stiff actuators that may be simply dimensioned for the desired torque. Depending on the supplied external force, a compliant actuator will tolerate departures from its own equilibrium position. A compliant actuator's equilibrium position is defined as the point where the actuator creates zero force or torque. Because stiff actuators do not have this idea, it is introduced particularly for compliant actuators[31]

Traditional actuators, including as hydraulic, pneumatic, and electric actuators, are employed in the majority of robotic and automated mechanical systems as compliant actuators. New substances have recently been developed as a result of developments in material technology, allowing for the construction of structurally robust articulated mechanisms that are small and lightweight. Shapememory alloys (SMAs), electrorheological fluids (ERFs), electrostrictive and magnetostrictive materials (including piezoelectric substances), and electroactive polymers are examples of such materials that can be utilised to construct innovative actuators. Because

their present working speeds are quite low, with reaction times in the tens of seconds, the utility of these novel materials as compliant actuators is not clear[31].

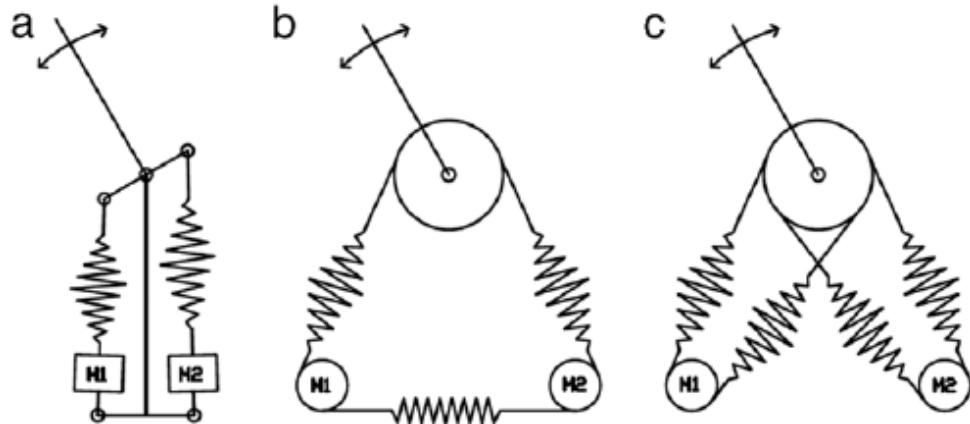


Figure 2.1: This concept gives intrinsic capabilities (bandwidth, impacts, energy storage) over the joint stiffness range. However, two motors are required: one to control the equilibrium position and the second to control stiffness[6]

2.2 Series Elastic Actuator Background

Series Elastic Actuator (SEA) is a type of Variable Impedance Actuator (VIA) which was first introduced in 1995 and has been currently recognized in the robotics field as an actuator system to implement a high-performance torque control. Unlike traditional actuation, which uses stiff gears to send force/torque to the load that is proportionate to the motor torque and current flowing through it, the torque transferred to the end load by a SEA is meant to be proportionate to the spring deformation, which is a major component of a SEA. In other words, due to the existence of the spring, the SEA transforms the torque production problem from a motor current decision problem to a spring deformation choice problem. Aside from its force/torque producing capability, SEA is recognised for its complex mechanism, which consists of a motor, gears, and a compliant component, the structure of which varies depending on the SEA configuration[32].

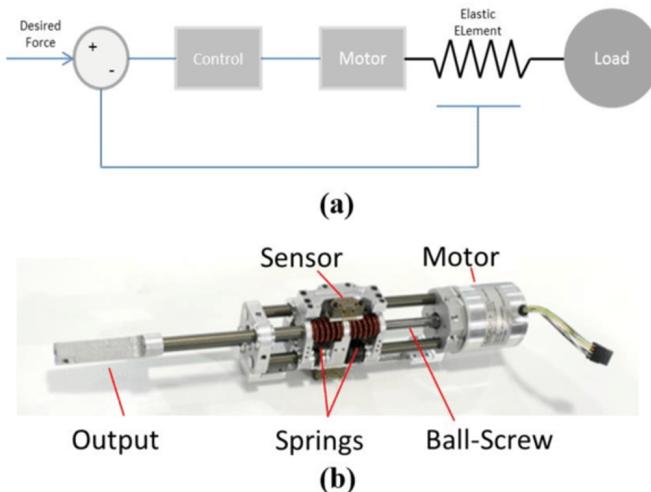


Figure 2.2: a. Model for a Series-Elastic Actuator b. Basic Components of a Series-Elastic Actuator[7]

Low-pass filtering shock loads is one of the results of series elasticity, which reduces peak gear forces significantly. Although the same low-pass filter that spreads out a shock impulse back driving the actuator also spreads out the actuator's output, which is an engineering trade-off rather than one-sided reduction. When the right amount of interface elasticity is used, shock tolerance may be significantly improved while yet keeping a modest motion bandwidth[8].

The stability criteria corresponds to a minimal load inertia that is most easily given by the robot structure's unavoidable mass. Because negative masses do not exist in natural surroundings, stability is ensured when in touch with any object. Finally, the option of energy storage

is provided by series elasticity. Such energy storage can considerably improve efficiency in legged mobility. Efficiencies can be gained by introducing elasticity within the actuator package, despite the fact that the elasticity is concealed from the higher level control system[8].

When force is adjusted, series elasticity necessitates elastic deformation. This additional motion may contribute to the load's movement in either a positive or negative way. In other words, the interface elasticity can enhance or reduce bandwidth based on the relative amplitude and phase of the load's force and motion wave-forms. When impedance control is employed when the intended impedance is close to the mechanical compliance of the interface elasticity, less motor motion is required, and bandwidth is enhanced, compared to when the interface is rigid. The compliant component (usually spring) is incorporated in SEA to realize high-fidelity force control. The location of the spring, however, can differ (any place among the power source, the transmission and the load) leading to various configurations of SEA[8].

SEA has been used in a variety of applications including quadruped robots, biped robots, dual arm robots, and wearable robots, demonstrating that it is a viable actuator system. SEA is classified into three types according to the relative position of the spring with regard to the gear:

- Reaction Force-Sensing Series Elastic Actuator (RFSEA)
- Force-sensing Series Elastic Actuator (FSEA)
- Transmitted Force-Sensing Series Elastic Actuator (TF-SEA)

Force-sensing Series Elastic Actuator (FSEA) locates the spring after the transmission gear, Reaction Force-sensing Series Elastic Actuator (RFSEA) locates the spring before the transmission gear, and Transmitted Force-sensing Series Elastic Actuator (TFSEA) locates the spring inside the transmission gear.

2.2.1 Force-sensing Series Elastic Actuator (FSEA)

FSEA is a SEA that includes a motor, a reduction gear, a spring, and a load in that sequence such that the spring can directly measure the force generated by the load. Many SEA designs have used this form, which was proposed as the structure of the first SEA. Figure 2.3 shows the FSEA design in detail, with the motor stator connected to ground to supply absolute force to the transmission, and the transmission's amplified force driving spring deformation to create spring force/torque. In other words, the spring torque is the force/torque production of SEA, which can be regulated by the motor torque[33].

It's worth noting that external forces from the load side can also have a direct impact on spring deformation. The spring displacement should be measured for the force measurement. The

motor angle and the load side angle are monitored using sensors such as encoders in most FSEA applications, and the spring deformation is calculated based on the two data. After that, multiply the spring deformation by the stiffness coefficient to get the force[33].

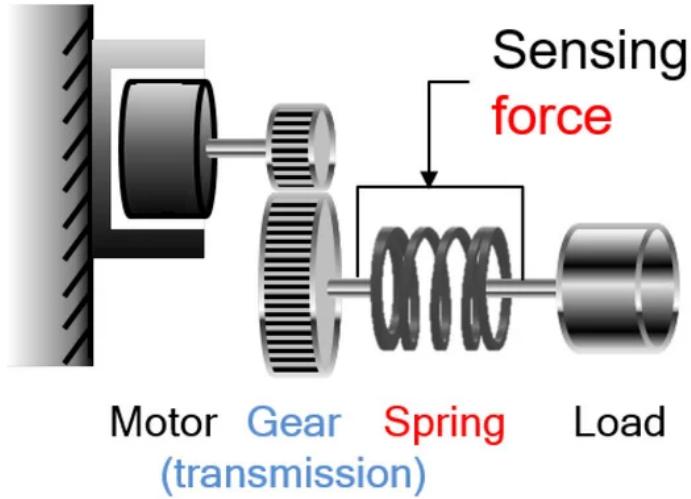


Figure 2.3: Configuration of F-SEA [7]

2.2.2 Reaction Force-Sensing Series Elastic Actuator (RFSEA)

RFSEA locates the spring before the transmission, which can be divided into two types as illustrated as shown in Figure 1.8. The spring can be positioned between the motor stator and the ground. The motor creates a relative torque between the stator and the rotor, which is increased by the gearbox and supplied directly to the load. The spring deformation in RFSEA in Figure 1.8(a) is proportional to the motor's response force with respect to the ground. The motor and the spring can both include position sensors[9].

Another form of RFSEA is shown in Figure 2.4(b), in which the spring is positioned between the motor rotor and the gearbox. The direct motor torque and the decreased external torque are measured by spring deflection in this situation. Because the response force that happens before transmission can be measured during driving, both scenarios may be called RFSEA. The dynamic properties of two kinds, however, differ: in Figure 2.4(a), the inertia of the motor stator should be considered since it is not fixed like in Figure 2.4(b) and creates inertial force/torque[9].

The RFSEA includes the UT-SEA, which is the University of Texas SEA, which uses a DC SERVO Motor as a drive, and ball-screw and a pulley reduction for transmission, a stiff spring for compliance. The UT-SEA is implemented in this paper as it compact and force measurement is easier.

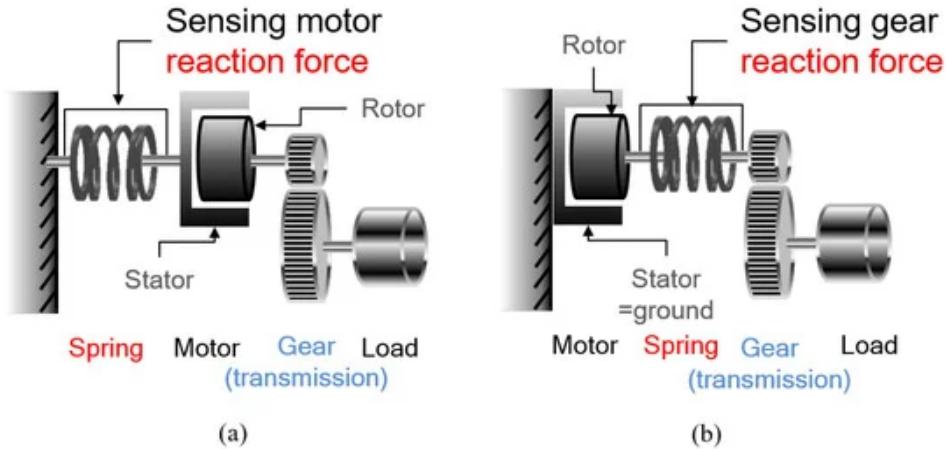


Figure 2.4: Configurations of RF-SEA (a) “motor reaction force” sensing type and (b) “gear reaction force” sensing type[7]

2.2.3 Transmitted Force-Sensing Series Elastic Actuator (TFSEA)

TFSEA is defined as SEA which places a spring inside the transmission. TFSEA's configuration has the spring located between the transmission gears, so that it can measure the transmitting torque inside the gears as one type. Another where a differential gear such as a planetary gear and Harmonic Drive is utilized as the compliant component. The motor torque is transmitted to the load, through the differential gear in this configuration. The spring is attached to the gear housing that is connected to the differential gear, so that the torque transmitting the gear can be measured by the spring. The position sensors can be attached to the motor and spring in this configuration[33].

2.3 Various SEA Control Techniques

This chapter is to introduce the innovative projects that were taken as a reference in the development of this thesis. In this section, previous control techniques are presented along with trends and patterns along with some gaps. The work presented is sorted by relevance and method of operation.

In order for a robotic manipulator to be useful in the real world, it must be able to interact with it in a safe and controlled manner. The forces between the robot and the environment must be controlled so that neither the robot nor its end-effector are damaged as a result of normal operation or unexpected collisions. Some actions, such as walking, running, and jumping, necessitate force control between the robotic end effector and its surroundings. The elasticity has the effect of making the force control easier, as larger deformations of the robot structure are needed to exert the same forces as a stiff robot. The control action reduces the reflected inertia, and the elasticity low pass filters shock loads, protecting the gearbox from damage[8].

The spring also filters the output of the actuator, limiting the bandwidth that can be achieved. Low bandwidth control is sufficient for human-like tasks. Introducing series elasticity also makes stable force control easier to achieve. The motor's force feedback loop can operate well at low frequencies, so neither the motor nor the load inertia can resonate. At high frequencies, where the feedback loop no longer operates well, the system behaves like a spring, which is passive and so stable. To prevent light loads resonating on the spring a minimum mass is required, which is easily provided by the unavoidable mass of the arm. The use of series elastic actuators should improve the performance of robots performing human-like tasks. The robot structure will be less stiff, and the force control will be less noisy, more accurate, and stable[8].

To provide accurate force control on the series elastic actuator many aspects are to be measured and analyzed. One of which is motor saturation. If the frequency drops the V/Hz goes up. This means that the motor needs a larger magnetic circuit. Without it, the magnetic circuit can be overloaded. This is called saturation and it leads to a rapid increase in current draw and a corresponding large increase in temperature, a motor's chief enemy[25].

The control architecture used by Pratt [8] contains both feedforward and feed-back paths. The feed-forward terms attempt to fully compensate for all of the following three terms. The first is the force applied through the elasticity to the load. The second, is the force required to accelerate the motor's mass in order to change the deformation of the elasticity. The third, is the force required to accelerate the motor's mass so as to track motion of the load. Feedback to compensate for modeling errors is accomplished by an ordinary Proportional-Integral-Derivative Controller (PID) loop, operating on force error. In a real system a problem arises, the actuator will take on the natural impedance of the series elasticity at sufficiently high frequencies. Thus,

a light load mass may resonate with the series elasticity. To avoid this problem, a minimum mass was placed on the load which lowers the resonant frequency to where the control loop operates well, at a low frequency. At this low frequency, the impedance of the series elasticity disappears from the overall impedance (which is very low), and resonance cannot occur. Because of saturation limit-cycle concerns or sampling rate limitations (if the control system is implemented digitally), the bandwidth of the PID system may be limited by design. However, at mid-range frequencies, the feed-forward components of the control system still operate unfettered, and performance is limited only by the capabilities of the motor[8].

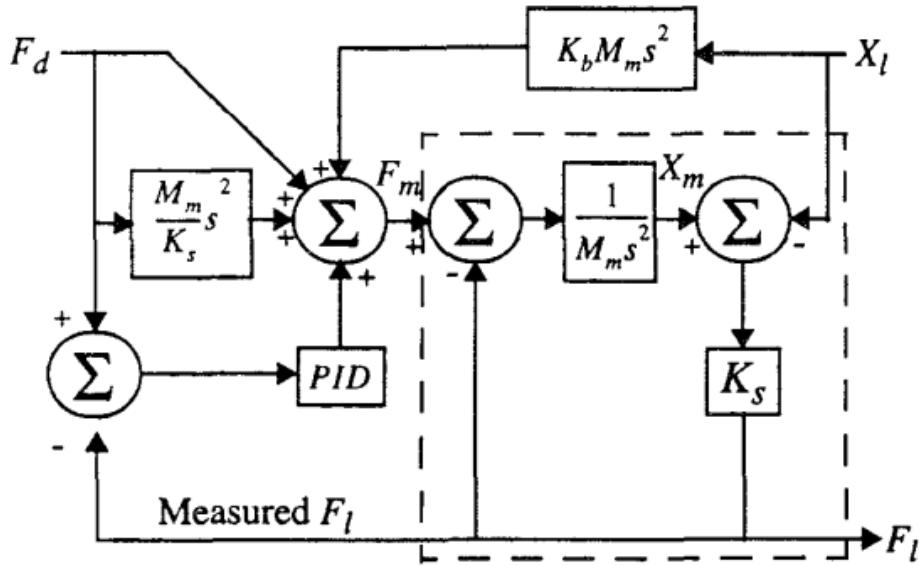


Figure 2.5: Proposed Control architecture by Pratt[8].

Another approach was implemented by Paine [32] using a PID and a Disturbance Observer(DOB), which will be explained later in this context, which was designed using a model obtained from experimental system identification. The goal of force control is to make the measured actuator output force (F_o) track a desired force profile (F_{des}) using motor current (i_m) as the plant input and spring deflection (x) as the output. However because the University of Texas Series Elastic Actuator (UT-SEA) implemented is an RF-SEA style actuator (F_o) cannot safely track (F_{des}) for all frequencies. Instead, the force control approach regulates spring force (F_k). This decision sacrifices force tracking of (F_o) near resonant frequencies but guarantees safe and oscillation free force tracking at all frequencies. Due to stability limitations, PID gains can only be increased up to certain values. To improve the force tracking performance further and to remove steady-state error another control approach was required, a Disturbance Observer (DOB). A DOB may be used to:

1. measure and compensate for error from disturbances
2. reduce the effect of plant modeling error. To use a DOB, a nominal plant model is

required

To obtain an accurate representation of the model, the actuator's output was fixed output (m_o) to ground to match the high output impedance model, and perform system identification of the model using an exponential chirp signal for (F_{des}). The frequency response of the magnitude and phase of F_k / F_{des} identified a second-order system, which was modeled as the a mass–spring–damper. With (k) measured before the actuator is assembled, the only unknowns were the sprung mass and effective damping[32].

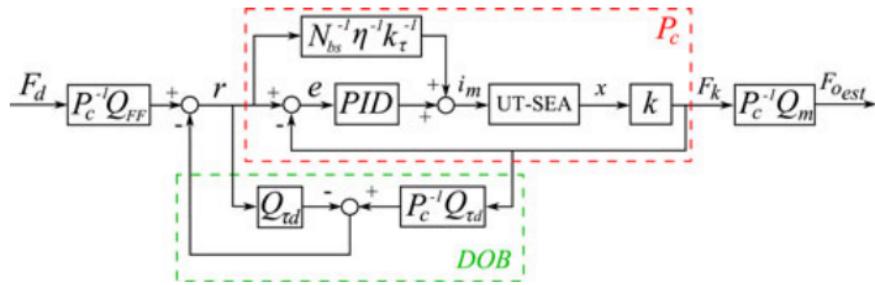


Figure 2.6: Proposed Control architecture by Paine, Q is a low-pass Butterworth filter[9].

In Brooks [10], the spring in this model was a beam with a cross-shaped cross section. Deflection in the beam was measured using strain gauges. The motor was controlled using current control as the input to the motor, making the motor an effective torque source. The compensation scheme used both feedback from the strain gauges, and feedforward from the desired torque input to calculate a desired current for the motor. While the implementation demonstrated many of the desirable characteristics of a SEA, there were a number of undesirable characteristics in the design. Backlash in the gearbox introduced some undesirable and unpredictable resonances in the closed loop response, and friction effects limited the effectiveness in providing large force bandwidth. In Robinson [10], the motor is to be controlled as a torque source. The effects of friction and backlash are better quantified, and some guidelines for spring selection are introduced.

Wyeth [10] used Williamson's [10] the authors note potential for improvement in the electronic design of the system. The motor was treating as a velocity source rather than as a torque source. The reason that this idea becomes attractive is that a tight velocity control loop on the motor can overcome some of the undesirable effects of the motor and the gearbox. Velocity control is also more straightforward from an implementation perspective, unlike current control which is generally considered challenging. An encoder provides velocity feedback to the motor, creating a tight loop for controlling the motor and gearbox. The velocity controller is tuned with no load attached, assuming that the spring decouples any high-frequency torque disturbances on the SEA output, and that a well tuned velocity controller should be able to deal with

low-frequency torque disturbances. The motor can be viewed as an effective velocity source with this tight velocity control loop in place, simplifying the torque control design which can be shown in Figure 2.7. The need to eliminate steady state error requires the introduction of two poles at the origin, and points to second-order PI compensation, using two cascaded PI compensates in the feedback path[10].

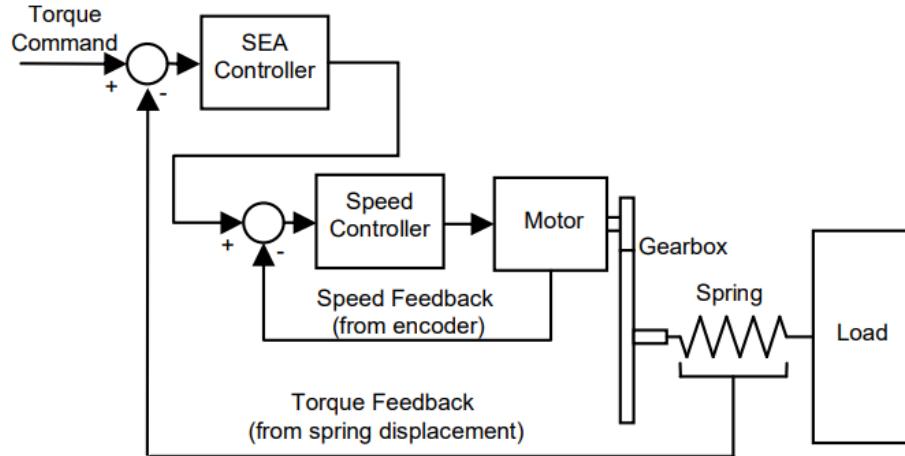


Figure 2.7: The inner velocity loop in the velocity source SEA helps to overcome problems with non-linearities and stiction[10]

Chapter 3

Methodology

3.1 Mathematical Model

In this section, the SEA model is presented. This model uses the implemented in hardware actuator UT-SEA, the University of Texas Series Elastic Actuator.

Figure 3.1 shows a simple model for the UT-SEA. In the RFSEA model, generalized motor force (F_m) is generated between the output mass (m_o) and the lumped sprung mass (m_k) which includes the rotor inertia, the gearbox reduction, and transmission inertia. (m_k) includes the mass of the actuator housing and motor, including the rotor mass.

A high-output impedance model is useful for simplifying the force controller design problem as will be shown later in the next chapters. It assumes that the actuator output is rigidly connected to an infinite mass, which cannot be moved. For the high-output impedance models, the sprung mass feels a summation of forces from 1) the motor (F_m); 2) the spring (F_k); 3) lumped viscous friction ($F_{b_{eff}}$) equation (3.1):

$$b_{eff} = b_b + b_k \quad (3.1)$$

and 4) from other disturbances that are difficult to model (F_d) such as the torque ripple from commutation, the torque ripple from the gearbox due to teeth engaging and disengaging, backlash, and various forms of friction such as stiction, and coulomb friction. In a RFSEAs, the spring acts as the force sensor.

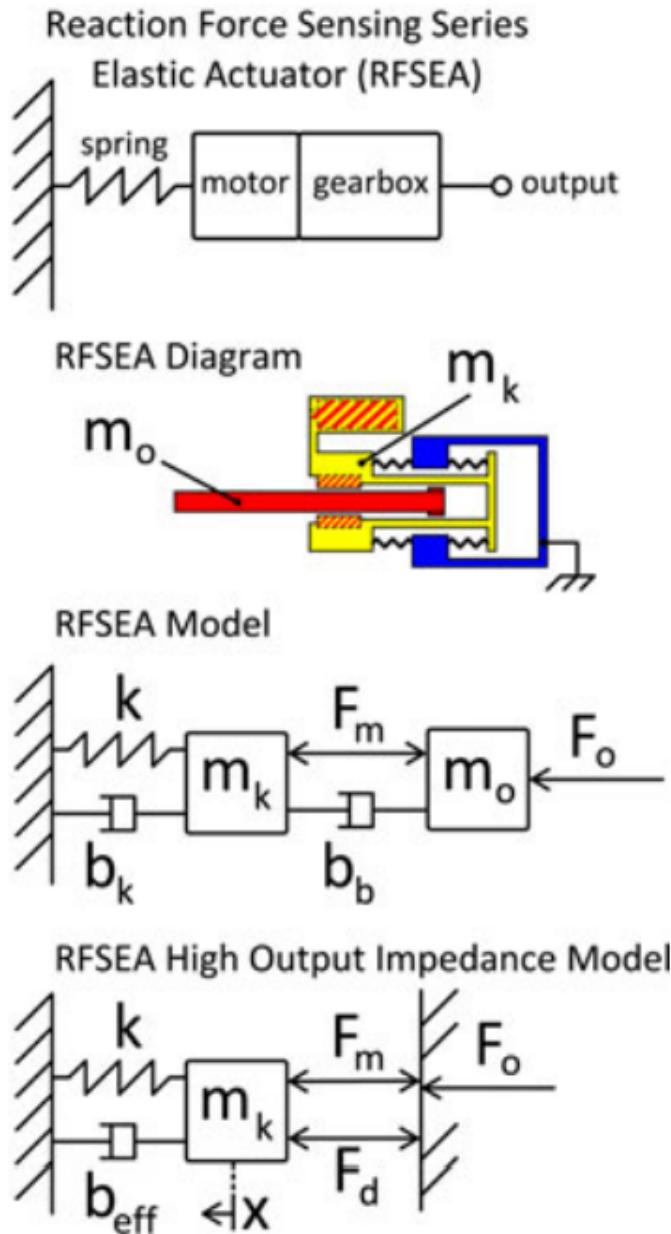


Figure 3.1: Model for RFSEA style actuator. The notations represent: F_m : motor force, F_o : output force, b_b : viscous backdriving friction, b_k : viscous spring friction, k : spring constant, x : spring deflection, m_k : lumped sprung mass, m_o : output mass, b_{eff} : lumped damping which equals $b_b + b_k$, and F_d : disturbance forces and forces which are difficult to model[11].

Newton's second law states that the summation of forces exerted on a body is equal the mass of the body times its acceleration, the second derivative of position.

$$\sum F = m\ddot{x} \quad (3.2)$$

According to the high output impedance model in 3.1 and Newton's 2nd law:

$$\begin{aligned} F_o &= F_m + F_d \\ &= F_{m_k} + F_{b_{eff}} + F_k \end{aligned} \quad (3.3)$$

It is known that:

$$\begin{aligned} F_{m_k} &= ma \\ &= m_k\ddot{x} \end{aligned} \quad (3.4)$$

and

$$\begin{aligned} F_{b_{eff}} &= b_{eff}v \\ &= b_{eff}\dot{x} \end{aligned} \quad (3.5)$$

and

$$F_k = kx \quad (3.6)$$

Combining 3.2, 3.3, 3.4, and 3.5 together in time domain will give:

$$F_o = m_k\ddot{x} + b_{eff}\dot{x} + kx \quad (3.7)$$

Applying Laplace Transform on 3.7 gives:

$$\mathcal{L}\{F_o\} = \mathcal{L}\{m_k\ddot{x} + b_{eff}\dot{x} + kx\} \quad (3.8)$$

$$F_o(s) = X(s)[m_k s^2 + b_{eff} s + k] \quad (3.9)$$

Making $X(s)$ the output and F_{des} as the input yields:

$$P_n = \frac{X_{meas}(s)}{F_{des}(s)} = \frac{1}{s^2 m_k + s b_{eff} + k} \quad (3.10)$$

For the Cheetah Robot Leg:

$$P_n = \frac{1}{7.9s^2 + 77.348s + 10806} \quad (3.11)$$

For the Kangaroo Robot Leg:

$$P_n = \frac{1}{7.3s^2 + 77.348s + 10806} \quad (3.12)$$

3.1.1 Spring Simulations

For the hardware implementation and the spring selection, 3 springs were considered: a first with stiffness 1544 N.m (combined stiffness of 2 springs 3008 N.m), a second with stiffness 2452 N.m (combined stiffness of 2 springs 4904 N.m), and a third with stiffness 5403 N.m (combined stiffness of 2 springs 10806 N.m).

3.2 Force Control Loop Design

The goal of force control is to make measured actuator output force track a desired force profile using motor torque (τ_m) as the plant input and measured spring deflection (X_{meas}) as the output. The motor force is calculated by using 3.13. F is the force point, or motor force in our case, and r is the distance between the center of rotation, D is the diameter which equals $2r$ as shown in Figure 3.2.

$$\tau_m = F_m r = F_m \frac{D}{2} \quad (3.13)$$

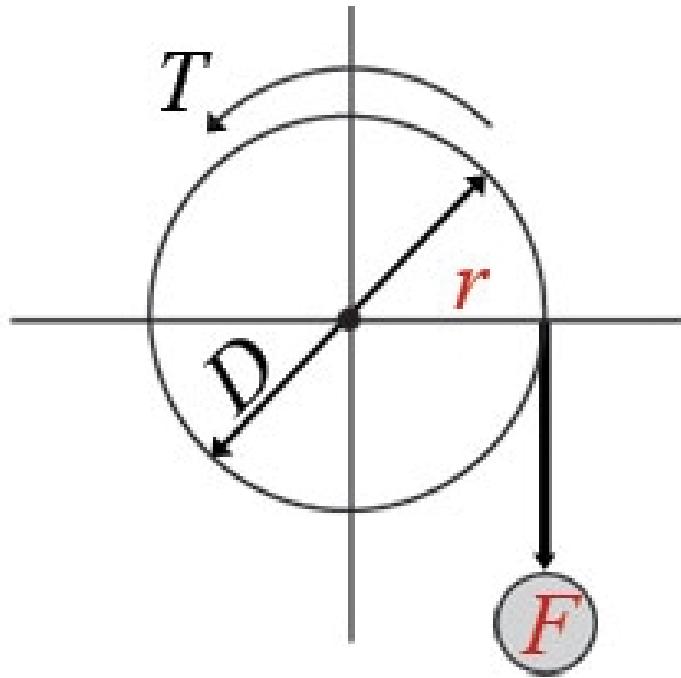


Figure 3.2: F_m is the force point, or motor force in our case, and r is the distance between the center of rotation, D is the diameter which equals $2r$ [12]

The compensator consists of a feedback term. The feedback attempts to keep the difference between desired force (F_{des}) or (F_m) and measured spring force (F_{kmeas}) small. (F_{des}) is calculated using (τ_{des}) multiplied by the force amplification due to the speed reduction of the gearbox (N_{bs}). For the UT-SEA, the speed reduction results from a pulley reduction (N_p) and a ball screw, which is parameterized by drive-train efficiency (η) and ball screw lead(l). The speed reduction defines the relation between actuator force (F) and motor torque (τ).

$$N_{bs} = \frac{F}{\tau} = \frac{2\pi N_p \eta}{l} \quad (3.14)$$

Measured spring force is calculated using spring deflection and Hooke's Law:

$$F_{kmeas} = kx_{measured} \quad (3.15)$$

For this application, the plant is the closed-loop transfer function ($P_n N_{bs}$) created by the PID controller acting on the physical actuator (PN_{bs}). Dividing the model's transfer function (which represents F_{kmeas}/F_{des} by k yields the X_{meas}/F_{des} transfer function:

$$PN_{bs} = \frac{X_{meas}(s)F_{des}(s)}{F_{des}(s)\tau_{des}} = \frac{N_{bs}}{s^2m_k + sb_{eff} + k} \quad (3.16)$$

Combining 3.14 and 3.16 fully characterizes the dynamics of the plant from τ_{des} to X_{meas} :

$$P_n N_{bs} = \frac{X_{meas}(s)}{F_{des}(s)} \frac{F_{des}(s)}{\tau_{des}(s)} = \frac{X_{meas}(s)}{\tau_{des}(s)} \quad (3.17)$$

For the Cheetah Robot Leg:

$$N_{bs} = \frac{F}{\tau} = \frac{2\pi(\frac{6}{4})0.8}{5 * 10^{-3}} = 1256.64 \quad (3.18)$$

$$PN_{bs} = \frac{1256.64}{7.9s^2 + 77.348s + 10930.4} \quad (3.19)$$

For the Kangaroo Robot Leg:

$$N_{bs} = \frac{F}{\tau} = \frac{2\pi(\frac{6}{4})0.8}{5 * 10^{-3}} = 1256.64 \quad (3.20)$$

$$PN_{bs} = \frac{1256.64}{7.3s^2 + 77.348s + 10930.4} \quad (3.21)$$

3.2.1 Motor RPM and Torque Relation

In order to relate the force control problem to motor motion, it is required to relate the motor force by the motor speed or rpm. This will be done by first converting the force to torque. This conversion is done by the ball screw reduction in Equation 3.14, given that $N_{bs} = 1256.64$. (F_{kmeas}) is the force which will be measured in hardware, which will be shown later on.

$$\tau = \frac{F}{N_{bs}} = \frac{F_{kmeas}}{1256.64} \quad (3.22)$$

The motor used in both the kangaroo and the cheetah legs is the SG-77125000-20K DC Geared motor. According to Figure 3.3 and Equation 3.23, the relationship between motor torque (τ_m) and the motor's speed (ω) is given. (τ_s) is the motor stall torque, the torque produced by a mechanical device whose output rotational speed is zero. It may also mean the torque load that causes the output rotational speed of a device to become zero, i.e., to cause stalling. (ω_n) is the no load speed of the motor, the speed that the DC motor will turn when nothing is attached to its shaft. This is why it is called no load. The DC motor isn't loaded with an object. When a DC motor has nothing attached to its shaft, it is able to operate at its highest maximum speed. According to the motor characteristics, the motor has a stall torque of 0.2646 N.m and a no load speed of 5000 Revolutions per minute (RPM), and the motor has a gear reduction of $\frac{1}{20}$, then the no load speed is equal to $2500/20$ which is 250rpm.

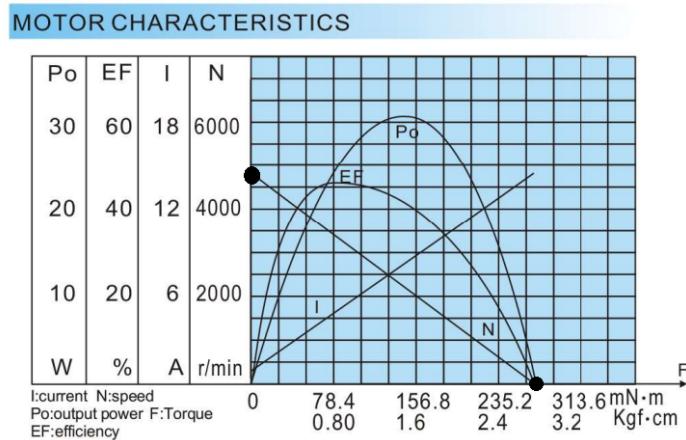


Figure 3.3: Motor current, speed, power, and efficiency characteristics compared to the torque. The 2 points, located on the same line, highlighted are the no load speed and stall torque. No load speed is on the y-axis: (0,5000) or (0,250) with considering the motor's gears, and the stall torque is located on the x-axis: (0.2646,0)[13]

$$\tau_m = \tau_s - \omega \frac{\tau_s}{\omega_n} \quad (3.23)$$

Rearranging to make the rpm, (ω) , subject yields:

$$\omega = \frac{\omega_n}{\tau_s} [\tau_m - \tau_s] \quad (3.24)$$

Substituting with the motor's stall torque and no load speed into Equation 3.24 gives:

$$\omega = \frac{250}{0.2646} [\tau_m - 0.2646] \quad (3.25)$$

$$\omega = 944.82\tau_m - 250 \quad (3.26)$$

3.2.2 The choice of PID

A proportional–integral–derivative controller (PID controller or three-term controller) is a control loop mechanism employing feedback that is widely used in industrial control systems and a variety of other applications requiring continuously modulated control. A PID controller continuously calculates an error value $e(t)$ as the difference between a desired setpoint (SP) and a measured process variable (PV) and applies a correction based on proportional, integral, and derivative terms (denoted P, I, and D respectively), hence the name.

The distinguishing feature of the PID controller is the ability to use the three control terms of proportional, integral and derivative influence on the controller output to apply accurate and optimal control.

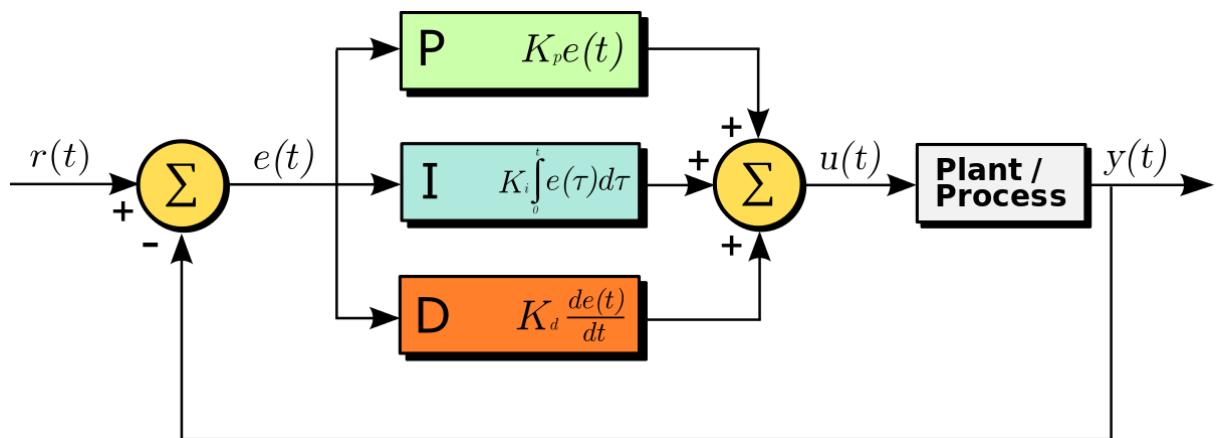


Figure 3.4: PID Gains Representation[14]

Figure 3.4 shows the principles of how these terms are generated and applied. It shows a

PID controller, which continuously calculates an error value $e(t)$ as the difference between a desired setpoint $SP = r(t)$ and a measured process variable $PV = y(t)$:

$$e(t) = r(t) - y(t) \quad (3.27)$$

and applies a correction based on proportional, integral, and derivative terms. The controller attempts to minimize the error over time by adjustment of a control variable $u(t)$, such as the opening of a control valve, to a new value determined by a weighted sum of the control terms[34].

- P Term is proportional to the current value of the $SP - PV$ error $e(t)$. For example, if the error is large and positive, the control output will be proportionately large and positive, taking into account the gain factor "K". Using proportional control alone will generally result in an error between the set-point and the actual process value because it requires an error to generate the proportional response. The controller cannot adjust the system unless there is an error present.
- I Term: accounts for past values of the $SP - PV$ error and integrates them over time to produce the I term. For example, if there is a residual $SP - PV$ error after the application of proportional control, the integral term seeks to eliminate the residual error by adding a control effect due to the historic cumulative value of the error. When the error is eliminated, the integral term will cease to grow. This will result in the proportional effect diminishing as the error decreases, but this is compensated for by the growing integral effect.
- D Term: is a best estimate of the future trend of the $SP - PV$ error, based on its current rate of change. It is sometimes called "anticipatory control", as it is effectively seeking to reduce the effect of the $SP - PV$ error by exerting a control influence generated by the rate of error change. The more rapid the change, the greater the controlling or damping effect.

The PID Controller in mathematics is represented by:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (3.28)$$

where K_p , K_I , and K_D are all non-negative, denote the coefficients for the proportional, integral, and derivative terms respectively (sometimes denoted P, I, and D). Now studying the controller in the continuous or analog domain makes it easier to realize what is going on. But most controllers these days are implemented digitally or with micro-controller like Arduino in software. So it is needed to implement this PID controller on the Arduino. I had to convert it

convert it to the discrete time or digital domain. This transform is relatively easy which makes the choice of PID an optimal controller for my purpose. The model in discrete form[34]:

$$u[k] = K_p(e[k] + \frac{T}{T_i} \sum_{j=0}^k e[j] + \frac{T_d}{T}(e[k] - e[k-1])) \quad (3.29)$$

where $u[k]$ is the correction signal in discrete, K_p is the proportional gain, $e[k]$ is the error signal in discrete, and $e[k-1]$ is the previous error signal. T is the sampling time, refers to the rate at which a discrete system samples its inputs. Given that T_i is Integral Time constant or Integral Time, and T_d is Derivative Time constant or Derivative Time:

$$K_i = K_p \frac{1}{T_i} \quad (3.30)$$

and

$$K_d = K_p T_d \quad (3.31)$$

3.2.3 Simulink Model

In this section, the dynamic model of the force control is presented. This model is simulated on Math-work's SIMULINK which makes designing the controller and tuning it easy. The model includes a reference input as a step function, the output by the rotary encoder(shown in next section) as a feedback, a scope to compare the input desired and output measured, and the model presented in the last chapter, and the PID controller.

For the Cheetah Robot Leg:

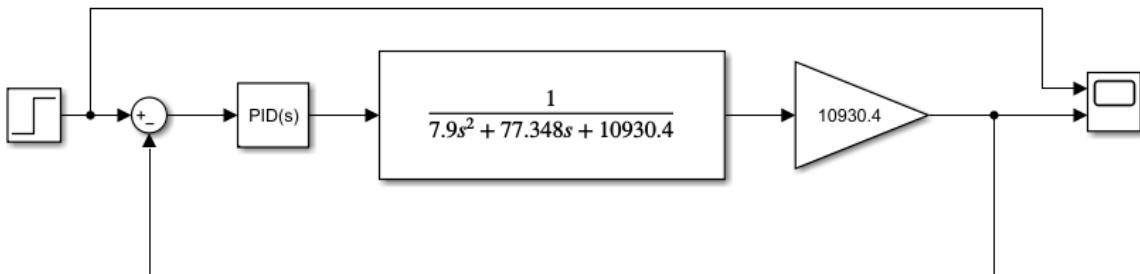


Figure 3.5: Model for Cheetah leg

For the Kangaroo Robot Leg:

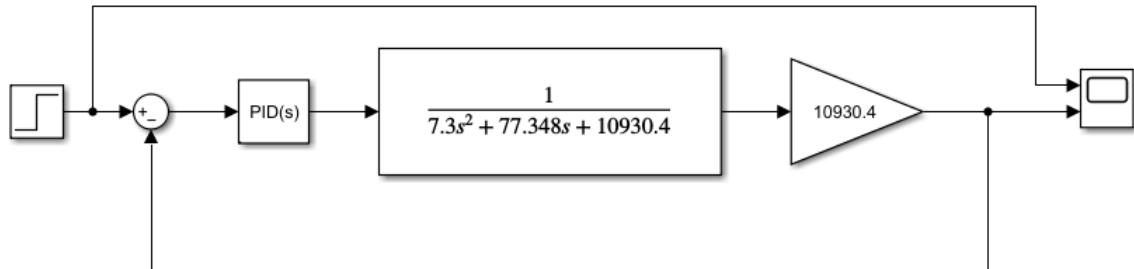


Figure 3.6: Model for Kangaroo leg

3.3 Force Control Hardware

In this section, all the used hardware elements and how was it installed and calibrated to achieve the cheetah and kangaroo legs' jump. The hardware is divided to the actuator and its driver, the controller, the sensors and the end-effector.

3.3.1 The Actuator

The motor used is the SG775125000-20K DC Geared Motor which will be made and controlled to act as a servo driver, meaning that it will rotate a few turns on the clockwise direction, and another few turns on the counter-clockwise direction. These few turns will be estimated based on trial and error. Figure 3.7 shows the motor used.



Figure 3.7: SG775125000-20K DC Geared Motor[13]

3.3.2 Motor Driver

The motor driver used was Cytron 10Amp 7V-30V DC Motor Driver. This motor driver was selected due to the motor's current requirements. This driver's advantages are:

- Bi-directional control for DC motor.
- Support motor voltage ranges from 7V to 30V.
- Maximum current up to 10A continuous and 30A peak (10 second). 3.3V and 5V logic level input.
- Solid state components provide faster response time and eliminate the wear and tear of mechanical relay.
- Fully NMOS H-Bridge for better efficiency and no heat sink is required.

- Speed control PWM frequency up to 10KHz.
- Support both locked-antiphase and sign-magnitude Pulse-Width Modulation (PWM) operation

The driver is controlled by a micro-controller, giving the PWM and DIR signals. The PWM signal is the pulse-width modulation with determines the speed of the motor based on the voltage the motor receives. The Direction signal (DIR) signal determines the rotation of the motor. It can be pulled LOW or HIGH and each determines a speed of rotation. In the robot, HIGH drives the motor counterclockwise, and LOW drives the motor clockwise.



Figure 3.8: Cytron 10Amp 7V-30V DC Motor Driver[15]

3.3.3 The Micro-controller

The micro-controller used is Arduino UNO REV3. The Arduino UNO is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started[35].

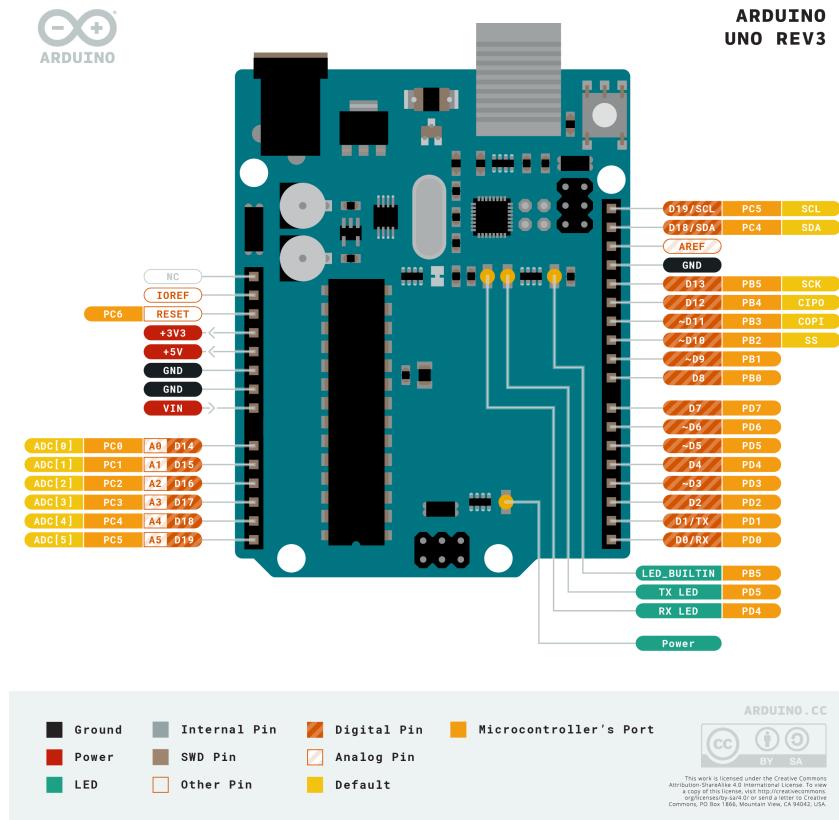


Figure 3.9: Arduino UNO Pinout[16]

Arduino UNO contains[35]:

- Microcontroller: Microchip ATmega328P[7]
- Operating Voltage: 5 Volts
- Input Voltage: 7 to 20 Volts
- Digital I/O Pins: 14 (of which 6 can provide PWM output)
- PWM Pins: 6 (Pin numbers 3, 5, 6, 9, 10 and 11)
- UART: 1
- I2C: 1
- SPI: 1
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA

- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Length: 68.6 mm
- Width: 53.4 mm
- Weight: 25 g
- ICSP Header: Yes
- Power Sources: DC Power Jack and USB Port



Figure 3.10: Arduino UNO R3 Board used[[17](#)]

3.3.4 Power Supply

A power supply is an electrical device that supplies electric power to an electrical load. The main purpose of a power supply is to convert electric current from a source to the correct voltage, current, and frequency to power the load. As a result, power supplies are sometimes referred to as electric power converters. Some power supplies are separate standalone pieces of equipment, while others are built into the load appliances that they power. Examples of the latter include power supplies found in desktop computers and consumer electronics devices. Other functions that power supplies may perform include limiting the current drawn by the load to safe levels, shutting off the current in the event of an electrical fault, power conditioning to prevent electronic noise or voltage surges on the input from reaching the load, power-factor correction, and storing energy so it can continue to power the load in the event of a temporary interruption in the source power (uninterruptible power supply).

All power supplies have a power input connection, which receives energy in the form of electric current from a source, and one or more power output or rail connections that deliver current to the load. The source power may come from the electric power grid, such as an electrical outlet, energy storage devices such as batteries or fuel cells, generators or alternators, solar power converters, or another power supply. The input and output are usually hardwired circuit connections, though some power supplies employ wireless energy transfer to power their loads without wired connections. Some power supplies have other types of inputs and outputs as well, for functions such as external monitoring and control.

In my circuit, a 12 Volts, 10 Ampere power supply was used according to the motor's requirement. Figure 3.12 shows the connection of the power supply to the motor driver which powers the motor. =V and -v connect to the motor driver, and ground along with the L port are connected to the 220V supply outlet.



Figure 3.11: Power supply 12V/10A[18]

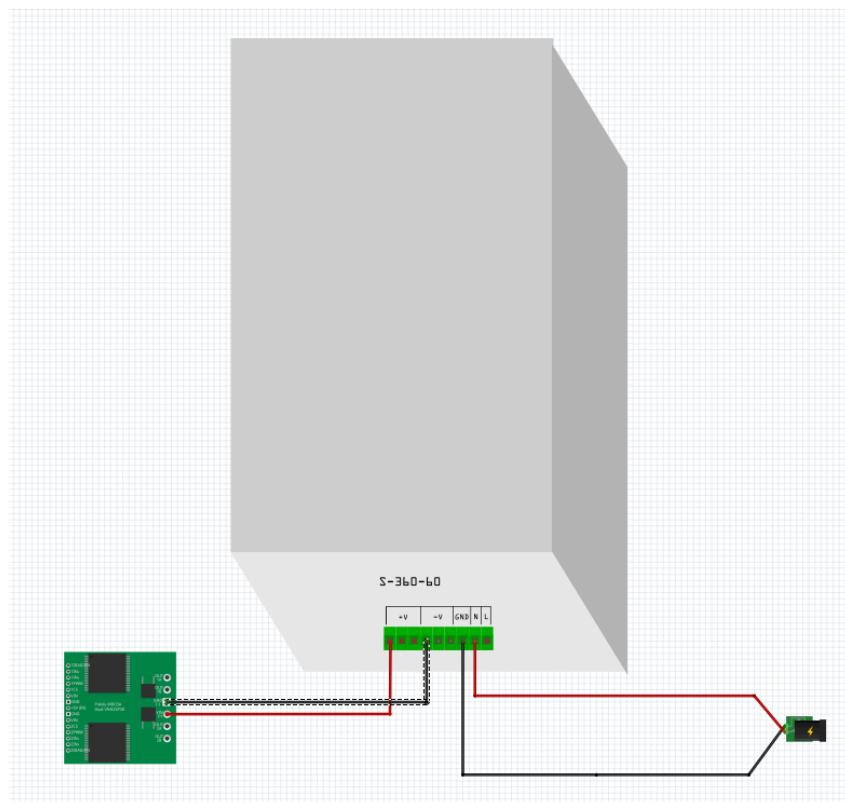


Figure 3.12: Power source and its connections to the motor driver on Fritzing software

3.3.5 Rotary Encoder

A rotary encoder is a type of position sensor which is used for determining the angular position of a rotating shaft. It generates an electrical signal, either analog or digital, according to the rotational movement. The rotary encoder is also known as quadrature encoder or relative rotary encoder and its output is a series of square wave pulses. In our robot, the rotary encoder was used on a pulley system to measure the actuator's spring deflection for the force control loop as will be shown later in this chapter. The rotary encoder used was the KY-040 encoder.

The encoder has a disk with evenly spaced contact zones that are connected to the common pin C and two other separate contact pins A and B, as illustrated in Figure 3.13.

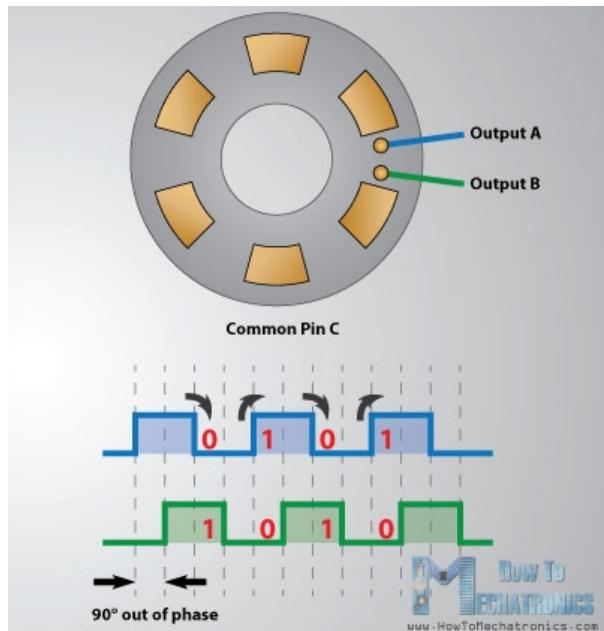


Figure 3.13: Encoder Disk Output square signals[19]

When the disk will start rotating step by step, the pins A and B will start making contact with the common pin and the two square wave output signals will be generated accordingly. Any of the two outputs can be used for determining the rotated position if the pulses of the signal is counted. However, if the rotation direction is to be determined as well, both signals need to be considered at the same time. It can be noticed that the two output signals are displaced at 90 degrees out of phase from each other. If the encoder is rotating clockwise the output A will be ahead of output B as shown in Figure 3.14.

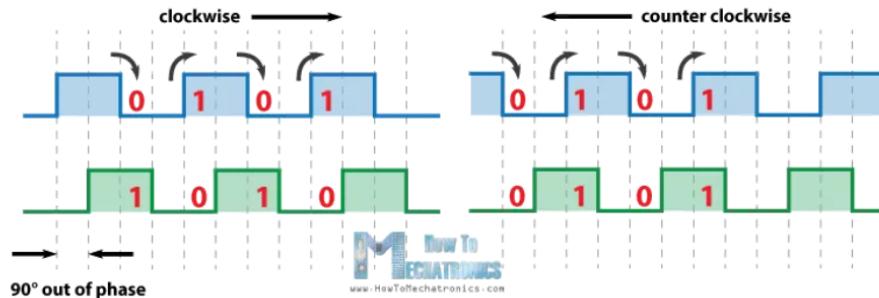


Figure 3.14: Encoder Disk Output Clockwise Rotation and Counter-clockwise rotation[19]

So when counting the steps each time the signal changes, from High to Low or from Low to High, it can be noticed that the two output signals have opposite values. Vice versa, if the encoder is rotating counter clockwise, the output signals have equal values. So considering this, we can easily program our controller to read the encoder position and the rotation direction. Figure 3.15 shows the pinout of the rotary encoder. 30 counts equals 1 full cycle.

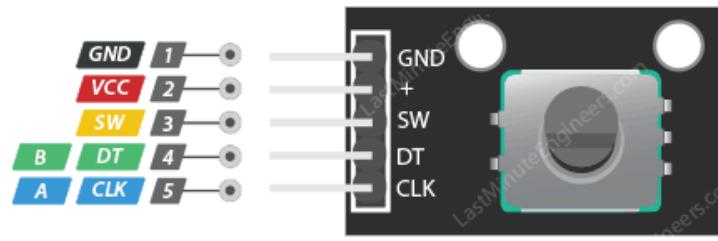


Figure 3.15: Rotary Encoder Pinout[20]

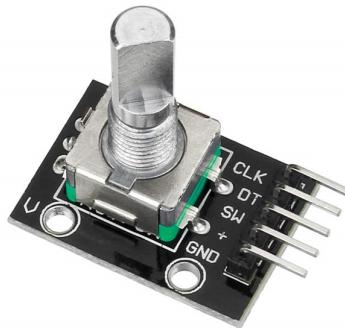


Figure 3.16: Rotary Encoder [20]

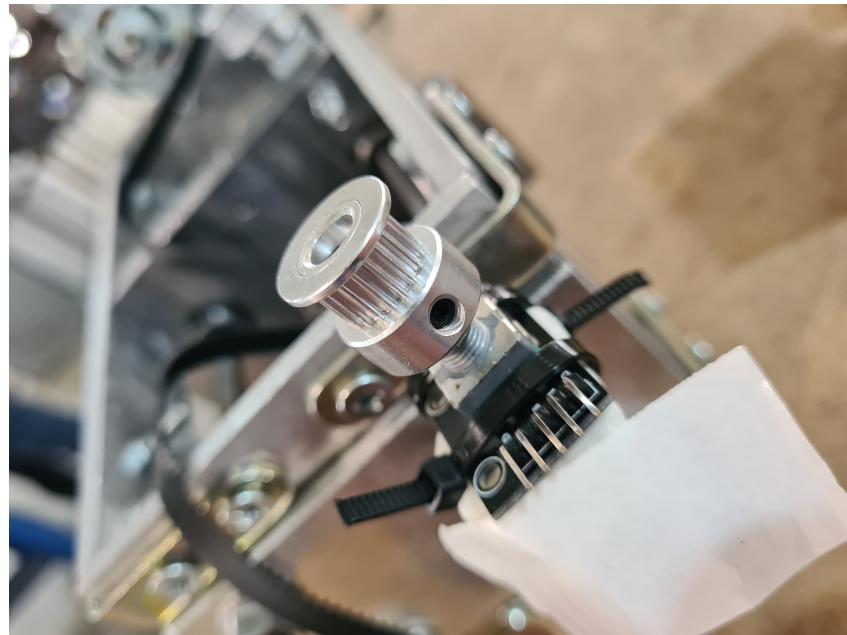


Figure 3.17: The encoder placed on pulley in the actuator

3.3.6 Pulley System

3.3.6.1 Belt

To measure spring deflection, a timing belt and a pulley system were used. GT2 49.4cm Timing Belt was used, which has a 2mm pitch and a 6mm width. Timing belts are a great way to transfer rotational motion into linear motion and these GT2 belts are excellent for the task. They have a special profile with rounded teeth which reduces slippage. Material is Neoprene Rubber, Fiberglass Reinforced.

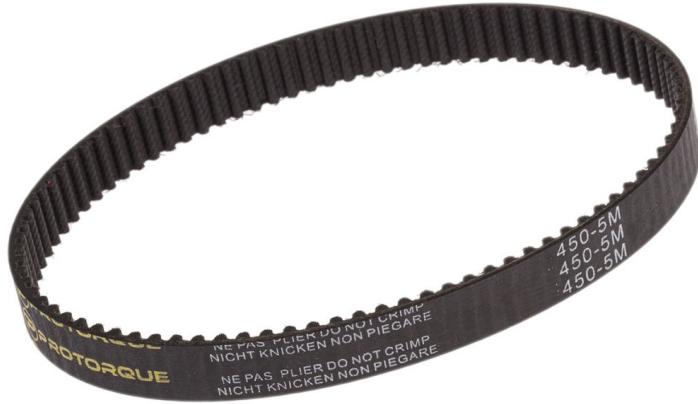


Figure 3.18: GT2 Timing Belt 2mm [21]

3.3.6.2 Pulley

Installed on each plate holder of the actuator is a L-shaped holder(Figure 3.19) and a bolt of 5mm thickness in order to hold the pulley. The pulley used is a 36 Teeth-5mm Inner Diameter-6mm Width shown in Figure 3.20. The rotary encoder is placed on a 20 Teeth-8mm Inner Diameter-6mm Width, as the shaft of the encoder is 6.35 mm as shown in 3.21.



Figure 3.19: Holder for the pulley on SEA plates. [22]



Figure 3.20: Pulley attached on actuator plates [23]



Figure 3.21: Pulley for encoder[23]

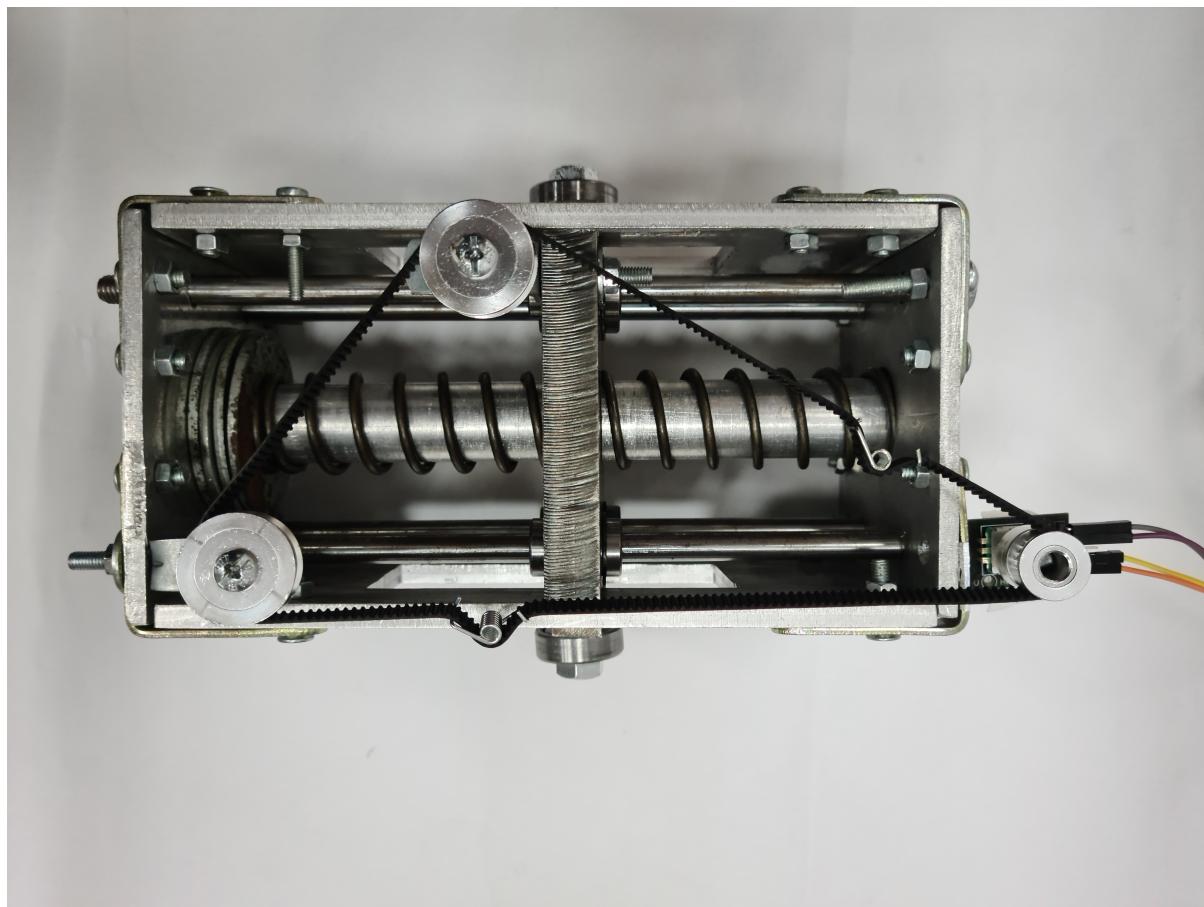


Figure 3.24: Pulley system

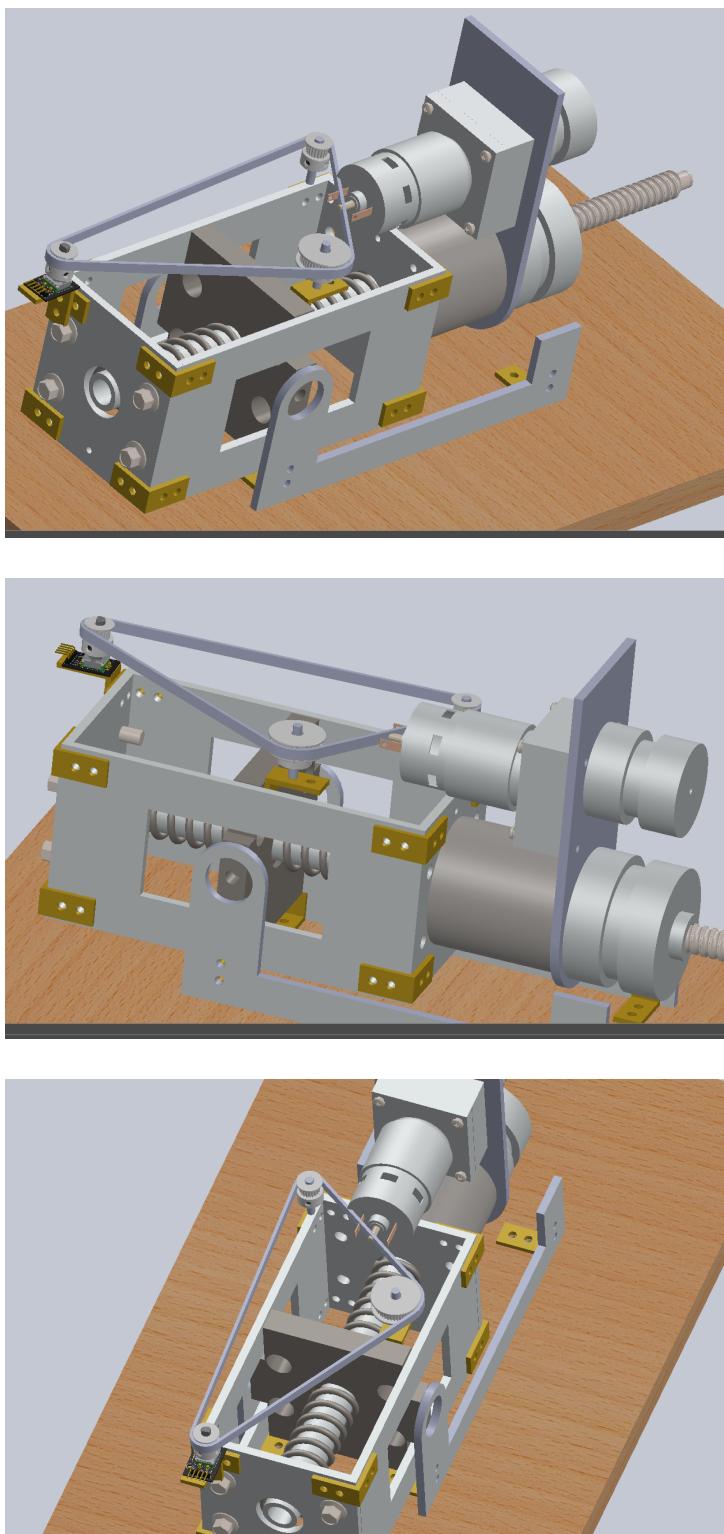


Figure 3.22: Pulley System for spring deformation measurement designed on Solid works Software

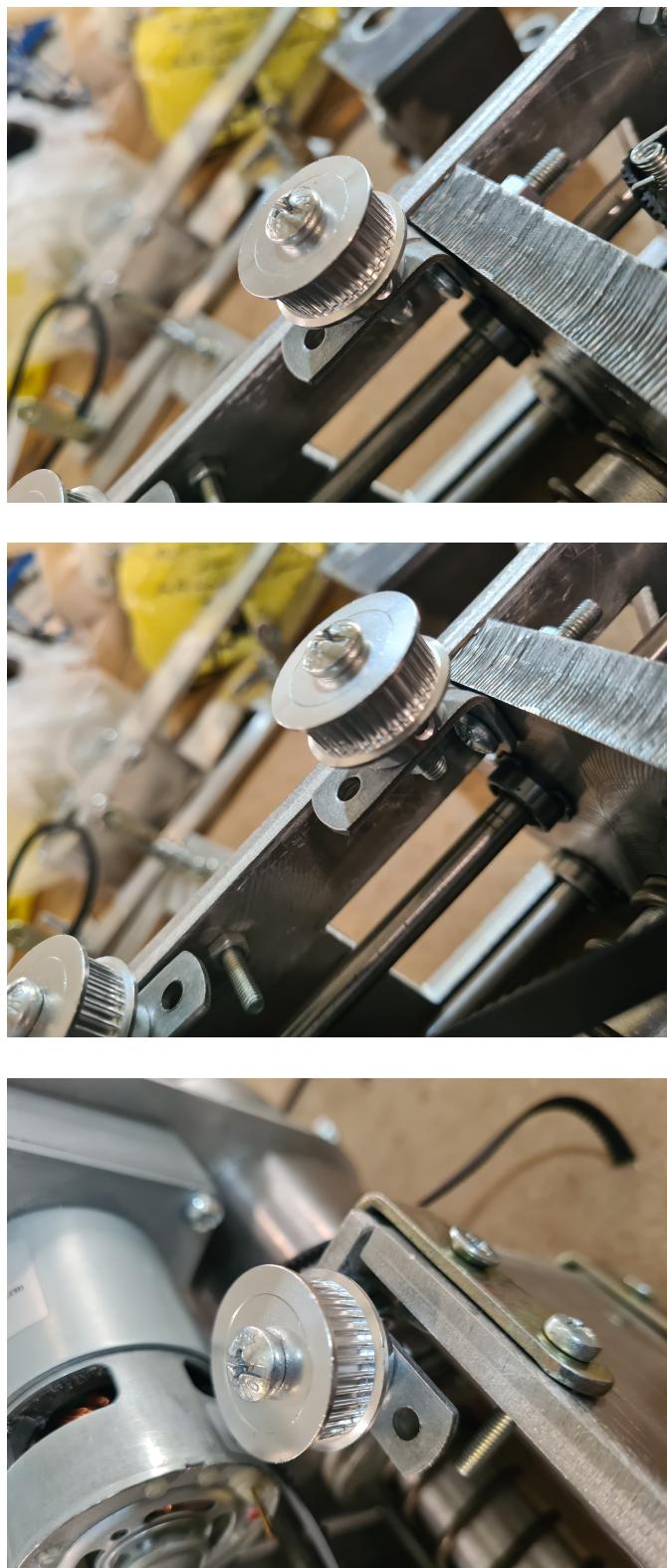


Figure 3.23: Placement of the pulleys on the L-corners on the actuator front and middle plates

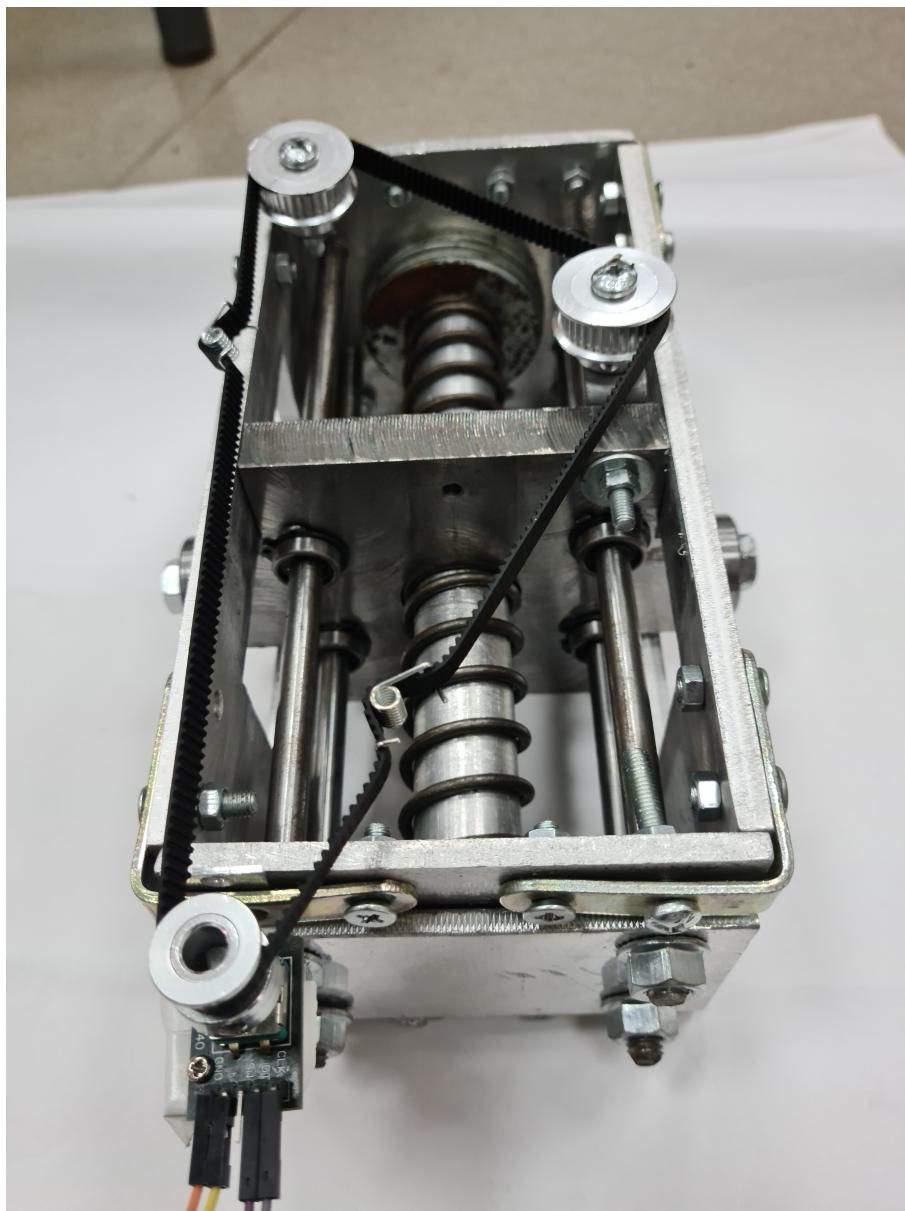


Figure 3.25: Pulley system assembled with belt

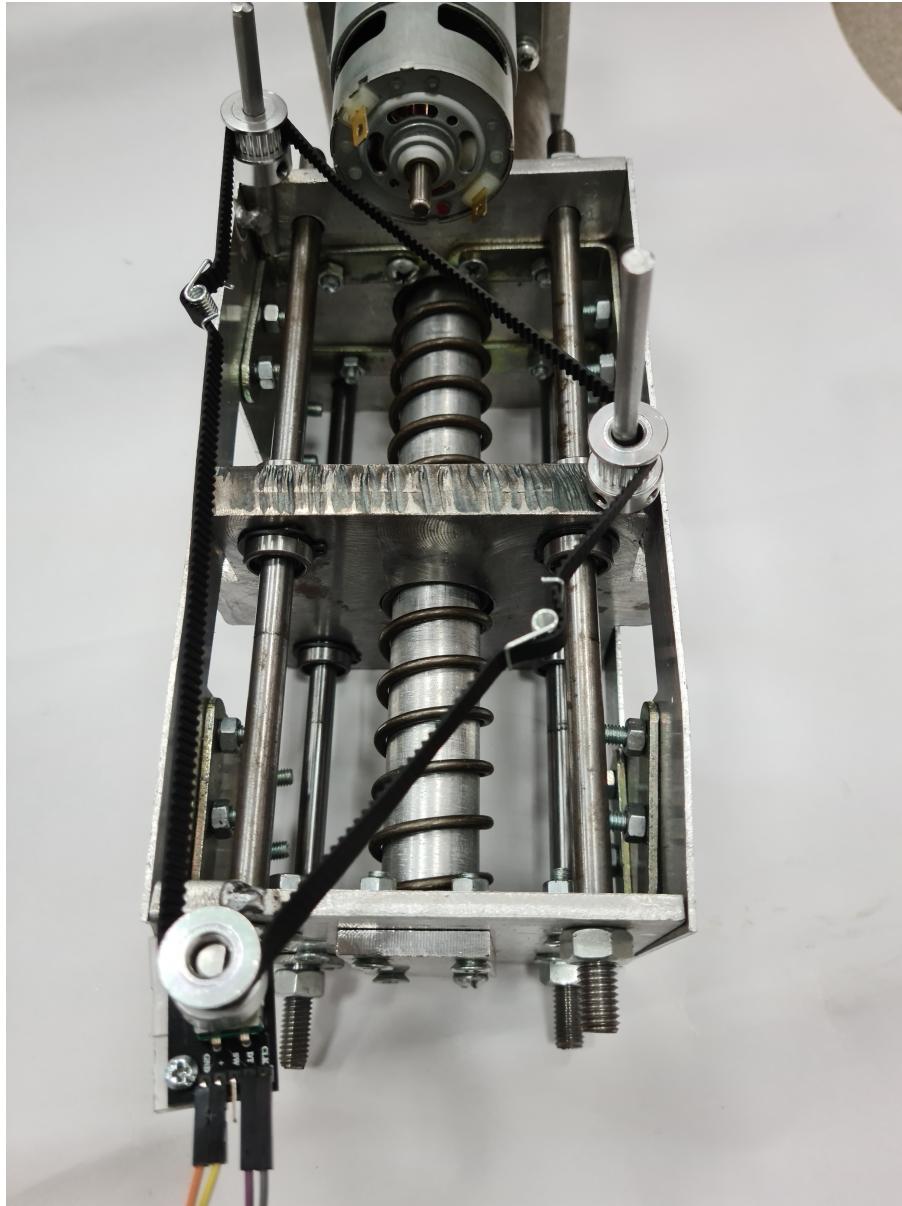


Figure 3.26: Pulley system assembled with belt and motor

3.3.6.3 Pulley-Spring Relation

The encoder ticks output is related to the the deformation of the spring. For the Kangaroo robot, every 12 ticks represent 1.8cm deformation. So whenever the encoder outputs ticks, a spring deformation by this amount shown in Equation 3.32.

$$\text{deformation} = \frac{1.8}{12} * \text{ticks} \quad (3.32)$$

For the Cheetah robot, every 11 ticks represent 1.8cm deformation. So whenever the encoder outputs ticks, a spring deformation by this amount shown in Equation 3.33.

$$\text{deformation} = \frac{1.8}{11} * \text{ticks} \quad (3.33)$$

3.3.7 Circuit

In this section, the electrical circuit implemented on Fritzing Software. It is color coded and shown below:

Red	+ voltage or Vin
Black	Ground
Blue	PWM Signal
Yellow	DIR Signal
Purple	Rotary Encoder Outputs

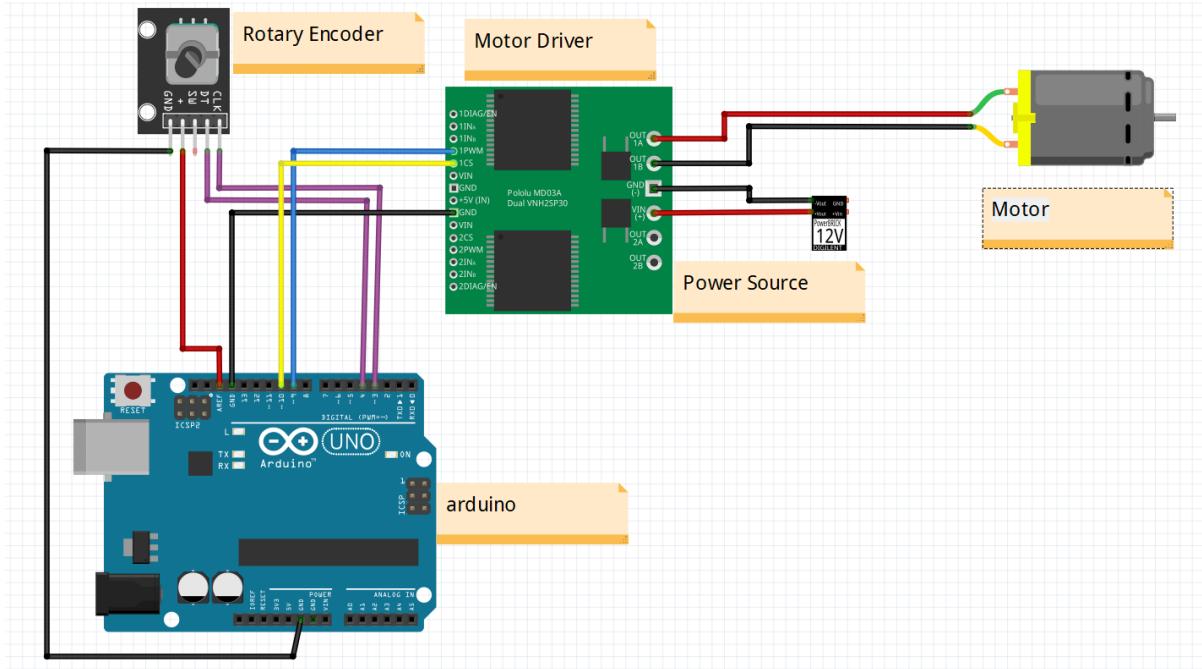


Figure 3.27: Circuit Implemented showing motor, rotary encoder, power source, motor driver, and Arduino

The motor driver receives the power from a 12 V source. It is connected to the motor from OUT1 AND OUT2 pins. The motor driver is controlled by 2 signals, the PWM signal and DIR signal given from pins 9 and 10 respectively. The rotary encoder has 4 pins, a vcc pin and a ground pin connected to the Arduino, and 2 output pins connected to pins 3 and 4 in Arduino. The common ground is established by connecting the ground pin in the motor driver to the Arduino ground pin. The Arduino is powered by 5V output from the PC.

3.4 Full Assembly

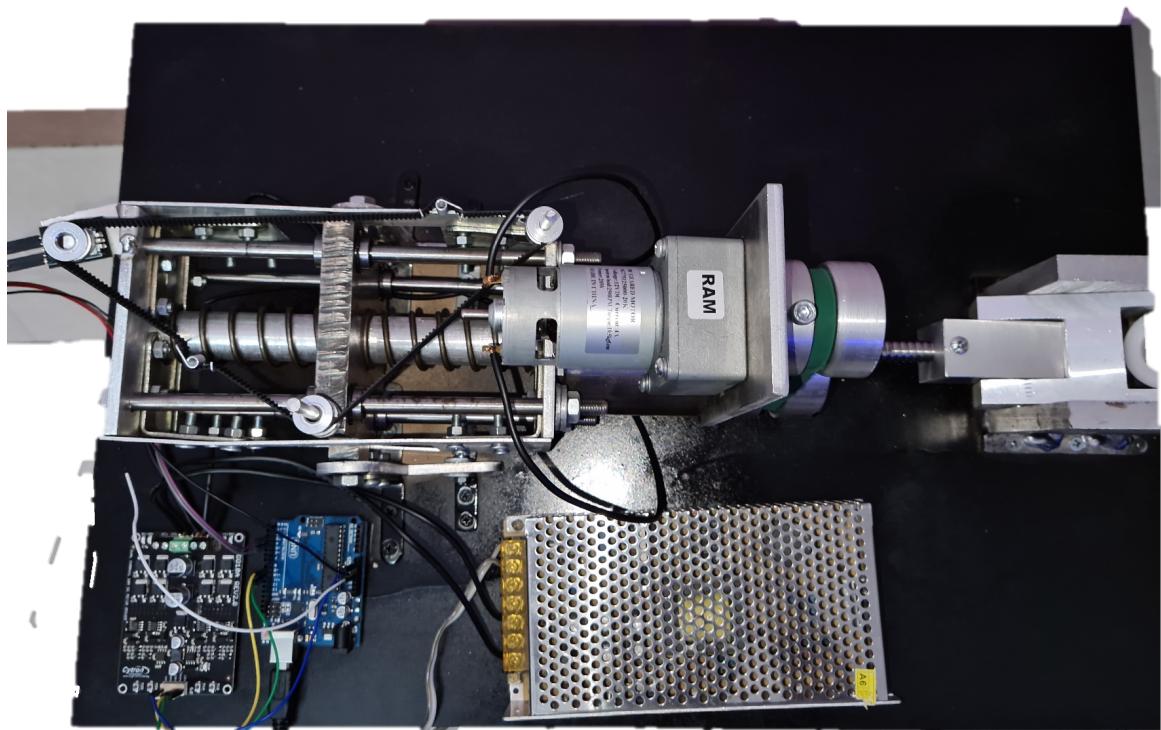


Figure 3.28: Full Control Panel Assembled with actuator

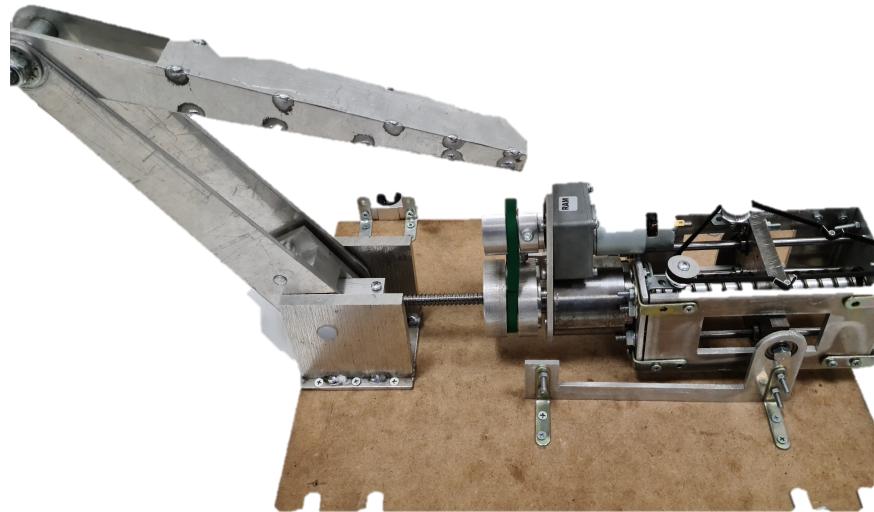


Figure 3.29: Cheetah Robot Leg



Figure 3.30: Kangaroo Robot Leg

Chapter 4

Results

In this section, the simulation results are presented. Simulations are performed on Mathworks software MATLAB and SIMULINK. The mathematical model is added to the software in order to design a controller. The control method used is a PID control and was implemented by a feedback system between the output measured force the the desired motor force. The PID is tuned, setting the PID gains K_p, K_i, K_d to a desirable or acceptable values. These values are decided by checking the time-domian specifications, rise time, peak time, settling time, and overshoot. These parameters are then validated by testing the stability of the closed-loop system. If the system guaranteed stability, then the controller designed is then moved to the hardware for testing. All tuning performed is based on reference tracking as the goal of the feedback system is to track a desired force.

4.1 Spring Simulations

In this section, the results of spring selection were tested in simulations and in hardware. The results shown in Figure 4.1 showed little damping which is not desirable in the motion of the mechanism. The same is shown in the second spring in Figure 4.2 where the oscillations increase. When both springs were tested on hardware, the compression is nearly the same as the spring length, which causes a failure in the mechanism.

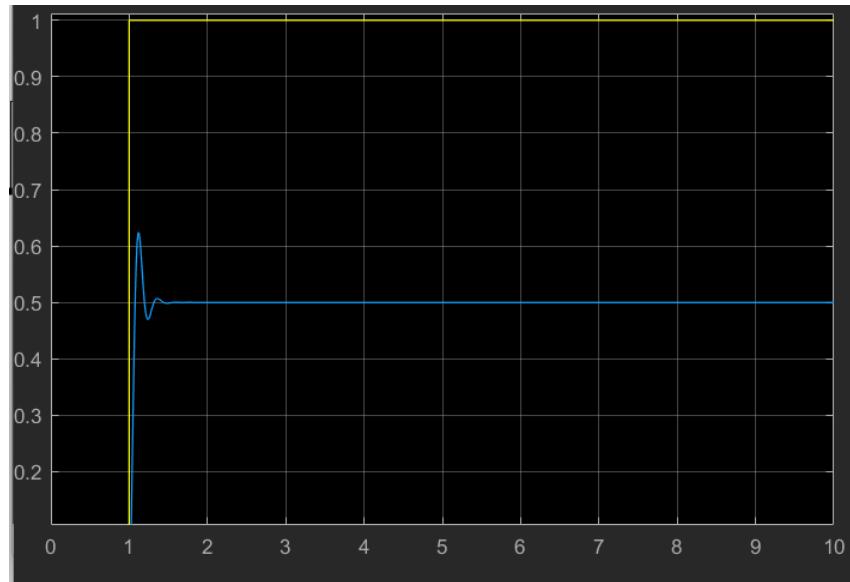


Figure 4.1: Closed-Loop step response for spring with stiffness 3088 N.m

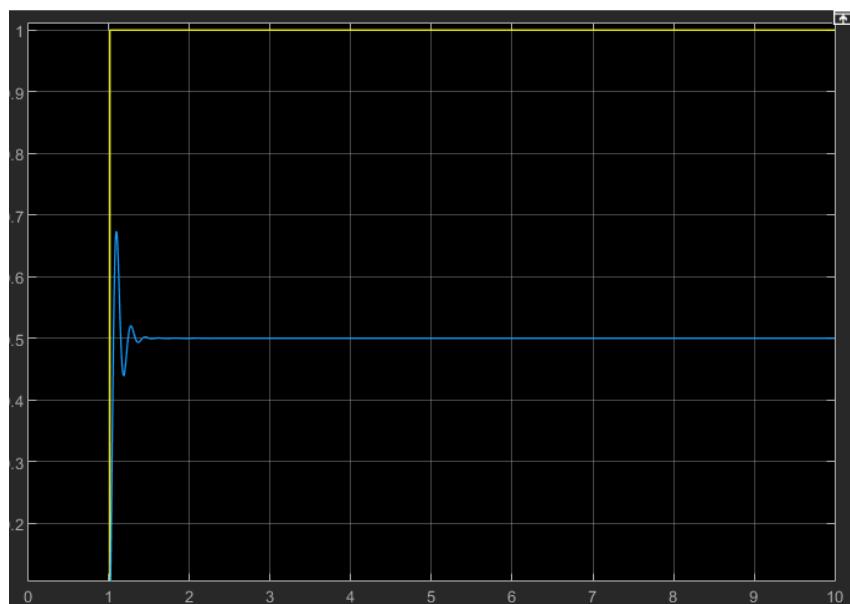


Figure 4.2: Closed-Loop step response for spring with stiffness 4904 N.m

4.2 Testing the Feedback System

The RF-SEA actuator was tested by using a sinusoidal sine signal with a high frequency of 50 Hz applied to the analog input of the motor. The goal is to test the encoder checking its ability to read very fast signals. Figure 4.3 shows the signal generated by Arduino.



Figure 4.3: The waveform signal generated by Arduino

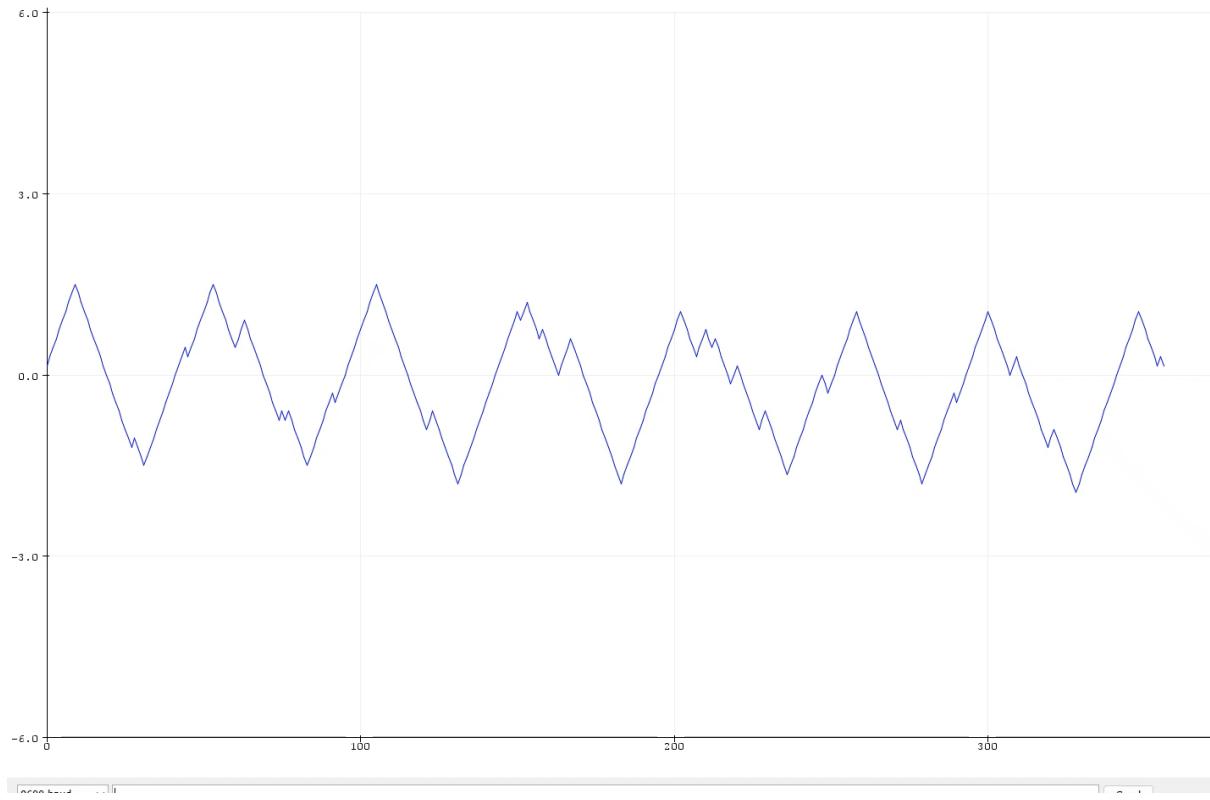


Figure 4.4: The output generated from the encoder output pins reading as spring deflection.

As shown in Figure 4.4, the output is satisfactory as it tracks the sinusoidal input. The output oscillates between 1.5cm and negative 1.5cm, the negative value represents the rotation of the encoder in the opposite direction. The output wave shows non-smooth spikes and that is because the non-smooth operation of the plates of the actuator sliding on the linear bearing guide.

4.3 Time Domain Specifications

1. Rise Time: t is the time required for the response to rise from 0% to 100% of its final value. This is applicable for the under-damped systems. For the over-damped systems, consider the duration from 10% to 90% of the final value. Rise time is denoted by t_r .
2. Peak Time: t is the time required for the response to reach the peak value for the first time. It is denoted by t_p . At $t = t_p$, the first derivative of the response is zero.
3. Peak Overshoot: Peak overshoot M_p is defined as the deviation of the response at peak time from the final value of response. It is also called the maximum overshoot.
4. Settling time: It is the time required for the response to reach the steady state and stay within the specified tolerance bands around the final value. In general, the tolerance bands are 2% and 5%. The settling time is denoted by t_s .

4.4 Cheetah Robot

First, the feedback is implemented without a controller.

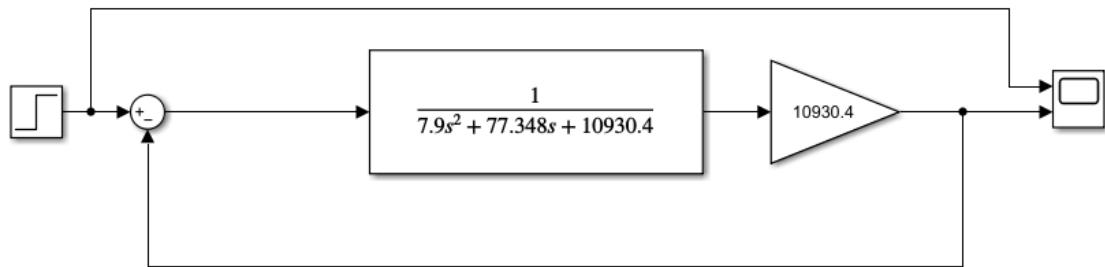


Figure 4.5: Cheetah Robot without Controller Simulink Model

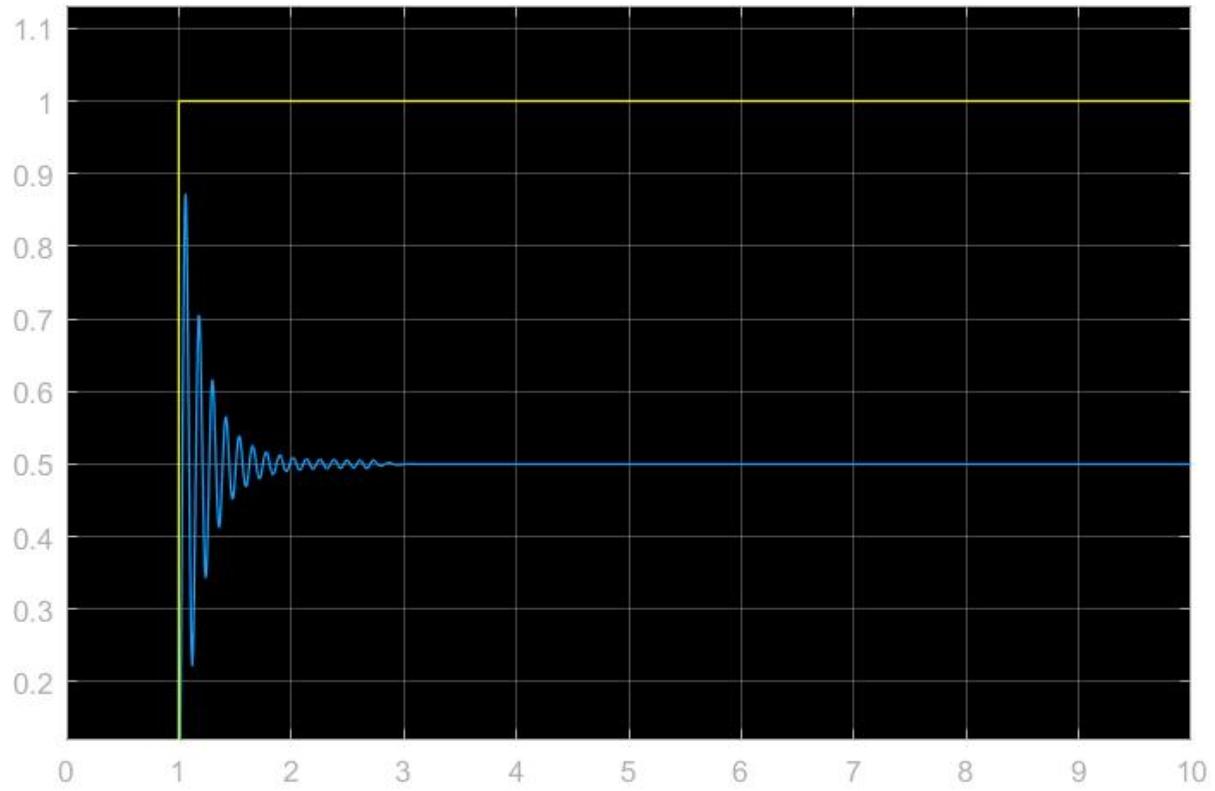


Figure 4.6: Cheetah Robot without Controller Simulink Step plot

In Figure 4.6, the settling time is about 5.3 seconds, the steady state-error is very large, and the percentage overshoot is about 60%. Thus, a controller should be implemented and the closed-loop response alone is not enough.

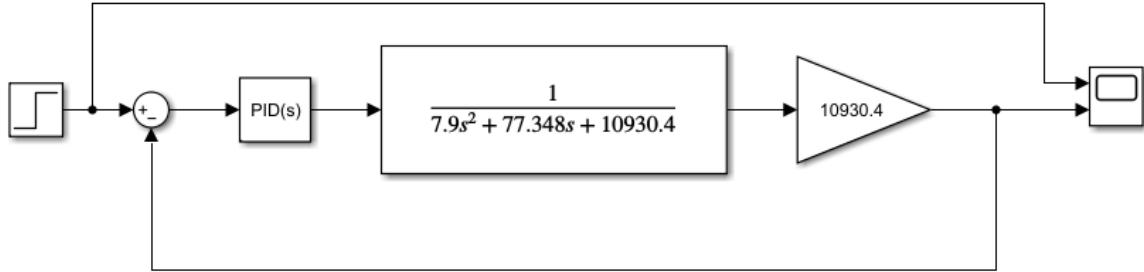


Figure 4.7: Model for Cheetah leg with controller added to be tuned

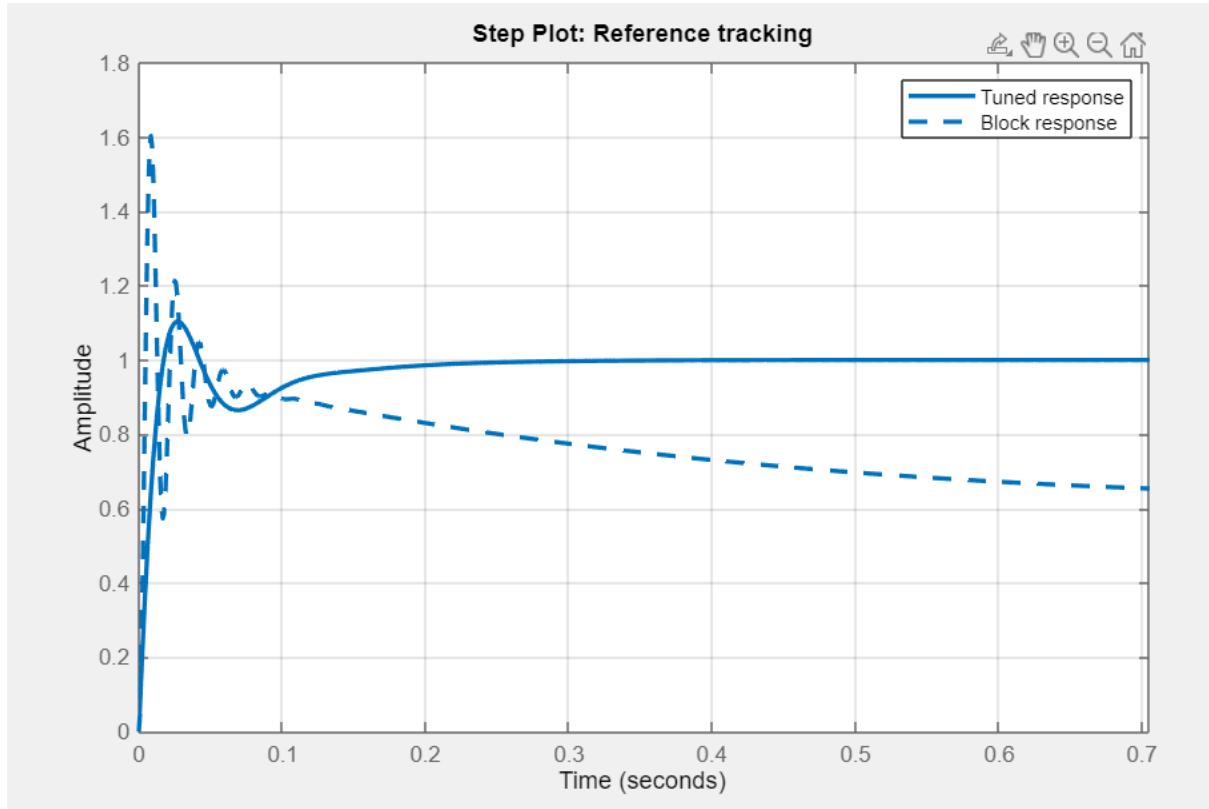


Figure 4.8: Cheetah Robot when tuning Controller on Simulink

In Figure 4.8 2 plots are presented, a block response, and a tuned response. The block response assumes an initial PID gains of $K_p = K_i = K_d = 1$. the tuned response is the response which is chosen adjusting the response speed and the robustness of the response.

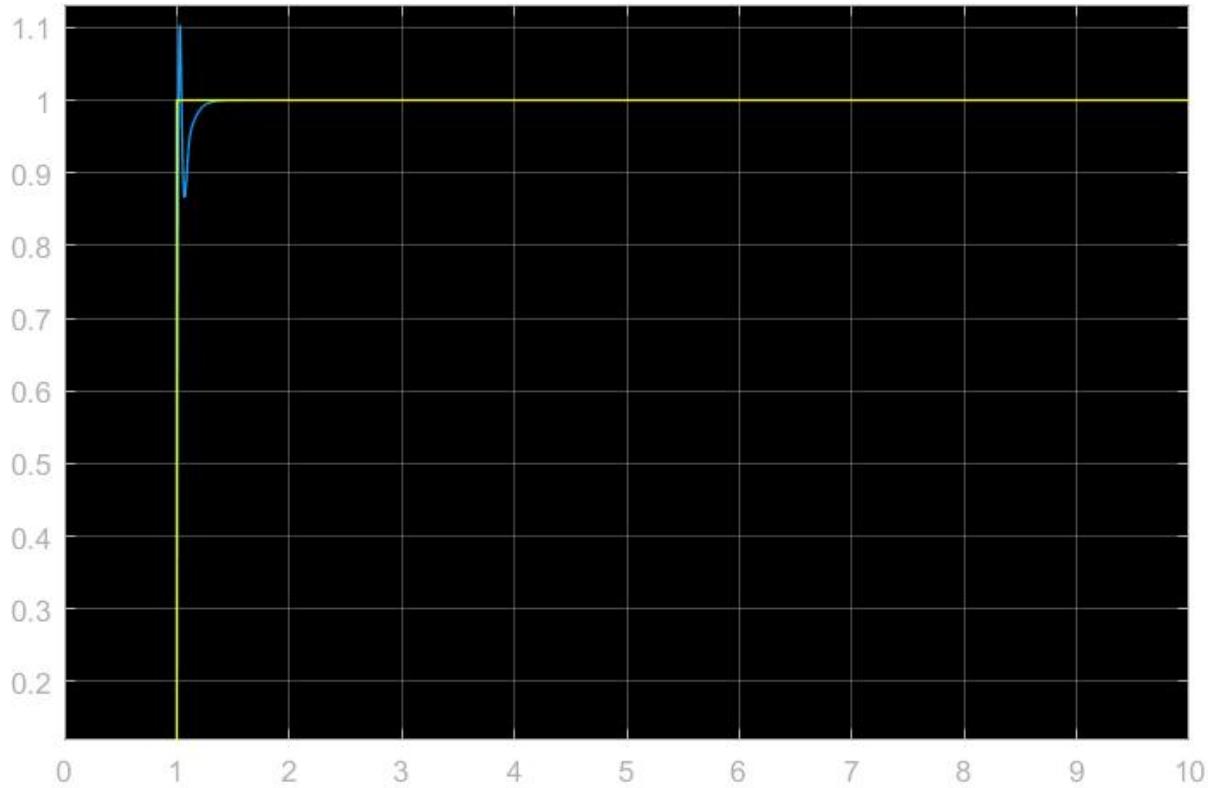


Figure 4.9: Closed-Loop Step plot of the Cheetah Robot

As observed in Figures 4.9 and Tables 4.4 and 4.4, the settling time decreased from 5.63 seconds in the block response to 0.179 seconds in the tuned response. Percentage overshoot also decreased.

	Tuned	Block
P	4.2294	1
I	65.5255	1
D	0.068083	1

	Tuned	Block
Rise Time	0.0128 seconds	0.00314 seconds
Settling Time	0.179 seconds	5.63 seconds
Overshoot	10.4%	60.3%
Peak	1.1	1.6
Phase Margin	64.5 deg at 113 rad/sec	16.5 deg at 369 rad/sec
Closed-Loop Stability	Stable	Stable

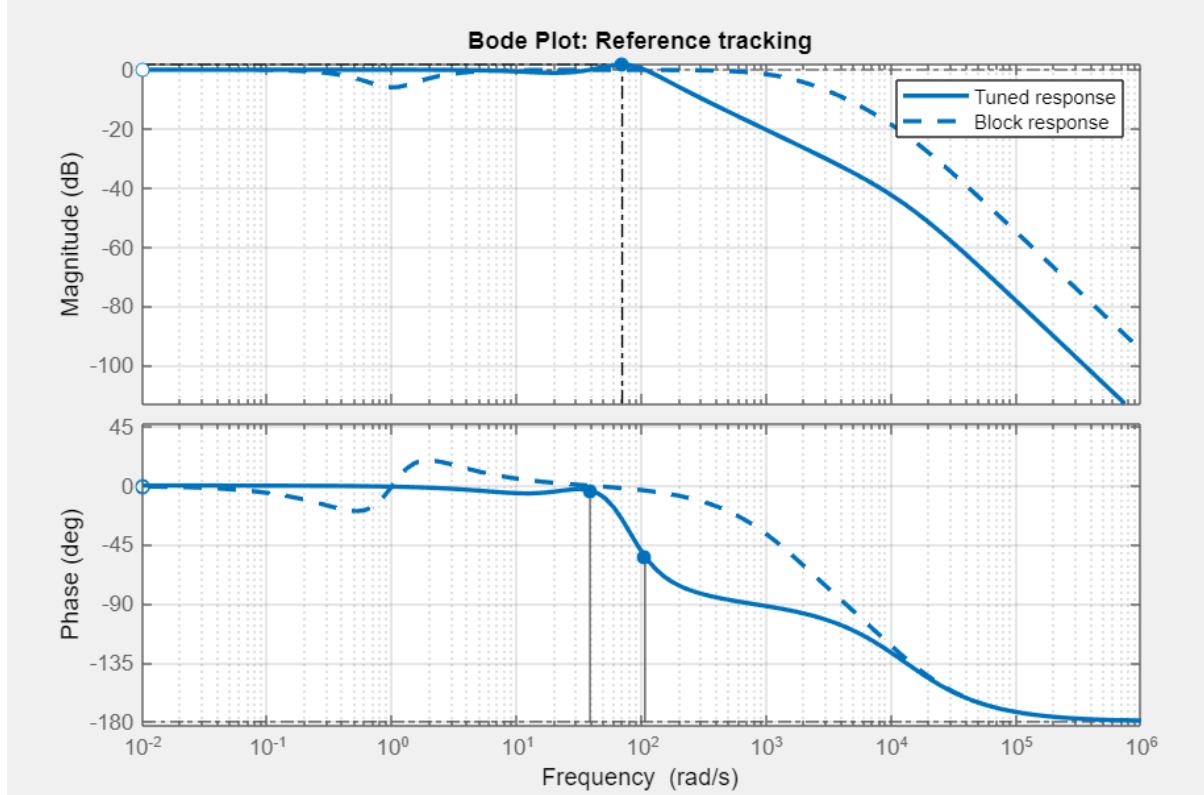


Figure 4.10: Bode Plot for Cheetah Robot Leg, Frequency Response

As shown in Figure 4.10, the gain margin and the phase margin increased while keeping the phase margin greater than the gain margin which guarantees stability.

Cheetah Closed-Loop Transfer Function:

$$Controller = 4.23 + 65.5 * \frac{1}{s} + 0.0681 * \frac{s}{13000s + 1} \quad (4.1)$$

$$CheetahCL = \frac{4.229s^2 + 65.52s + 0.005055}{7.9s^4 + 68s^3 + 10930s^2 + 66.37s + 0.005055} \quad (4.2)$$

CheetahCL in z-domain using a sample time of 0.01 seconds:

$$= \frac{0.00002706z^3 - 0.00002257z^2 - 0.0000282z + 0.0000237}{z^4 - 3.786z^3 + 5.49z^2 - 3.621z + 0.9175} \quad (4.3)$$

4.4.1 Tuning Choice

When tuning the controller, 2 choices were given. Either choosing a faster response time or a more robust transient response. Faster Response means more overshoot, while slower response means zero overshoot. A more robust transient response means a slower response. When the controller is tested on hardware, a faster response was more favorable for the jumping ability as can be shown in 4.20. Therefore the controllers were designed based on a faster response. Figure 4.11 shows a slower step response which has degraded most of the requirements but decreased the overshoot of the closed loop response. Table compares the characteristics.

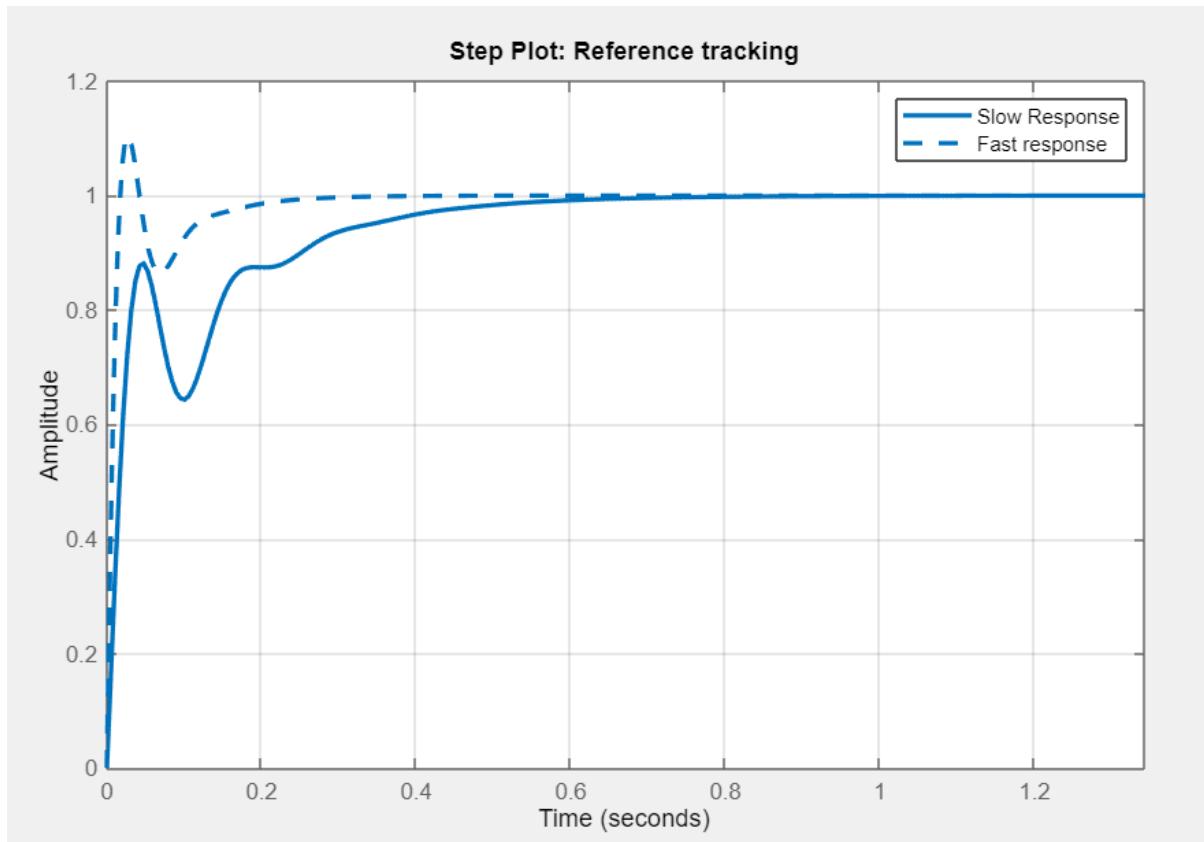


Figure 4.11: Step Plot showing the new slower response vs the faster response

	Faster Response	Slower Response
Rise Time	0.0128 seconds	0.248 seconds
Settling Time	0.179 seconds	0.471 seconds
Overshoot	10.4%	0%
Peak	1.1	1
Gain Margin	20dB at 113 rad/sec	3.78 dB at 37.2 rad/sec
Phase Margin	64.5 deg at 113 rad/sec	60 deg at 59.5 rad/sec
Closed-Loop Stability	Stable	Stable

4.5 Kangaroo Robot

First, the feedback is implemented without a controller.

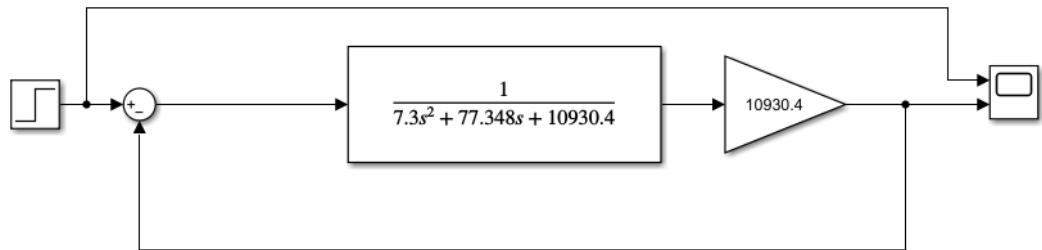


Figure 4.12: Kangaroo Robot without Controller Simulink Model

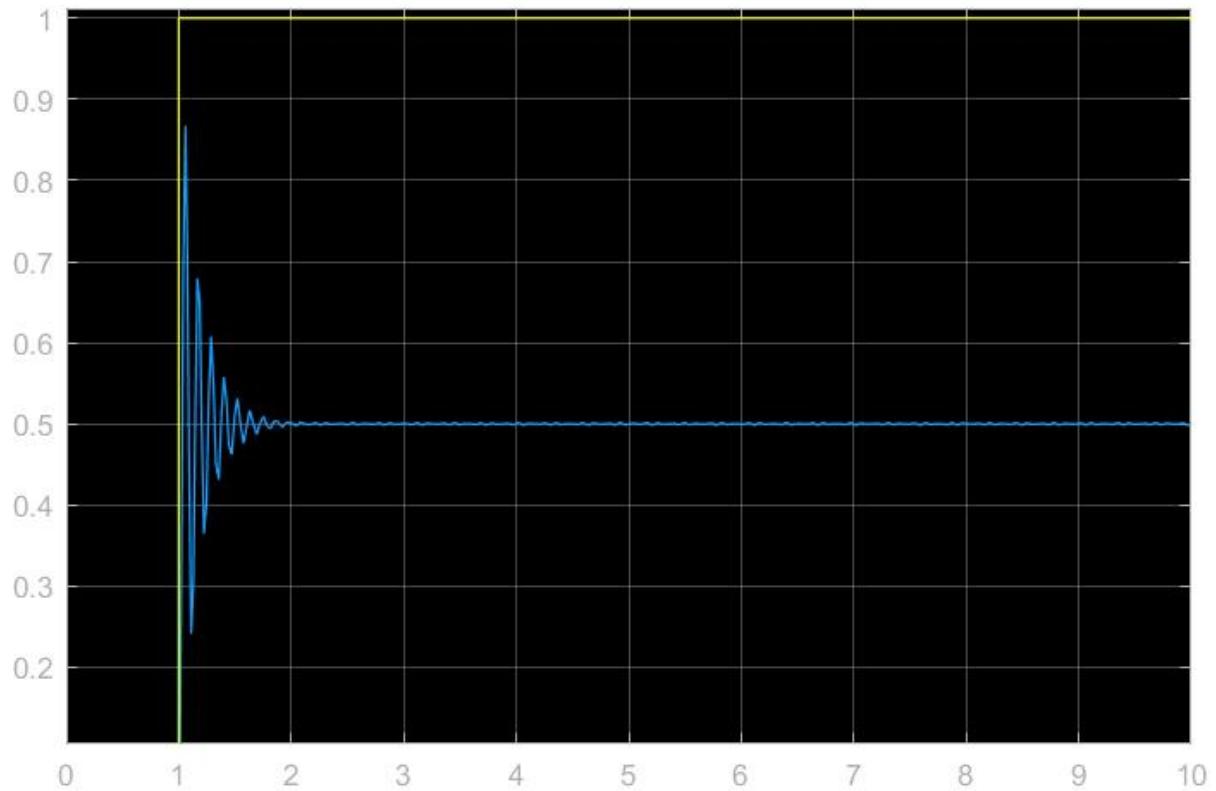


Figure 4.13: Kangaroo Robot without Controller Simulink Step plot

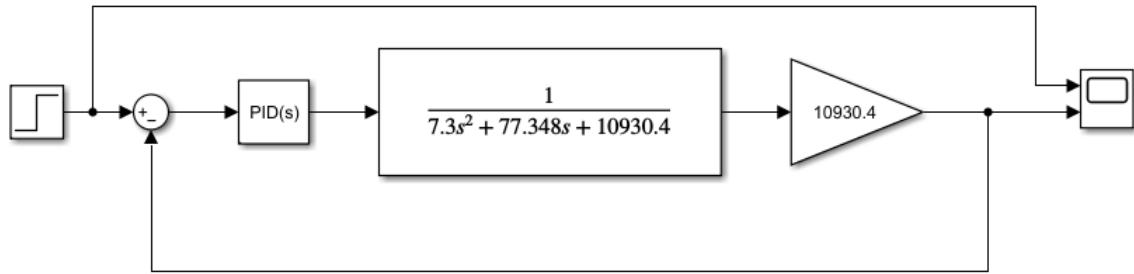


Figure 4.14: Model for Kangaroo leg with controller added to be tuned

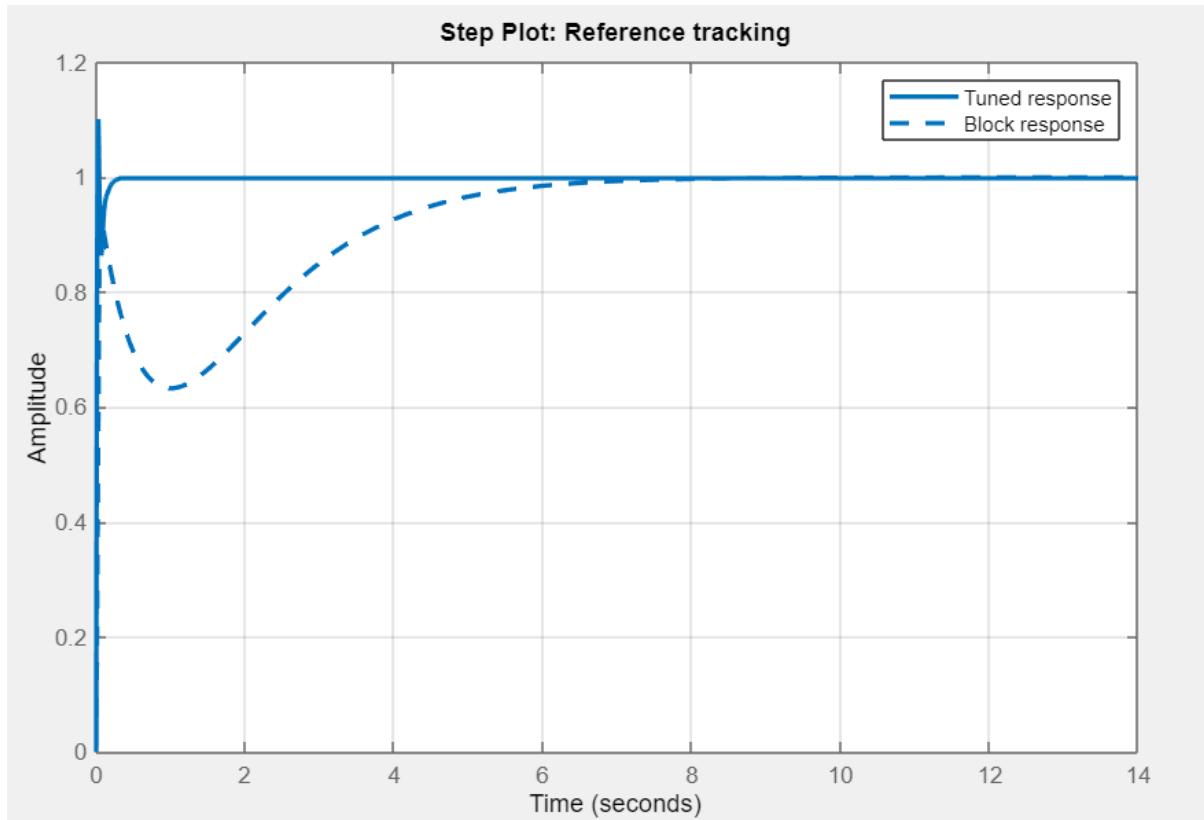


Figure 4.15: Kangaroo Robot when tuning Controller on Simulink

In Figure 4.15 2 plots are presented, a block response, and a tuned response. The block response assumes an initial PID gains of $K_p = K_i = K_d = 1$. the tuned response is the response which is chosen adjusting the response speed and the robustness of the response.

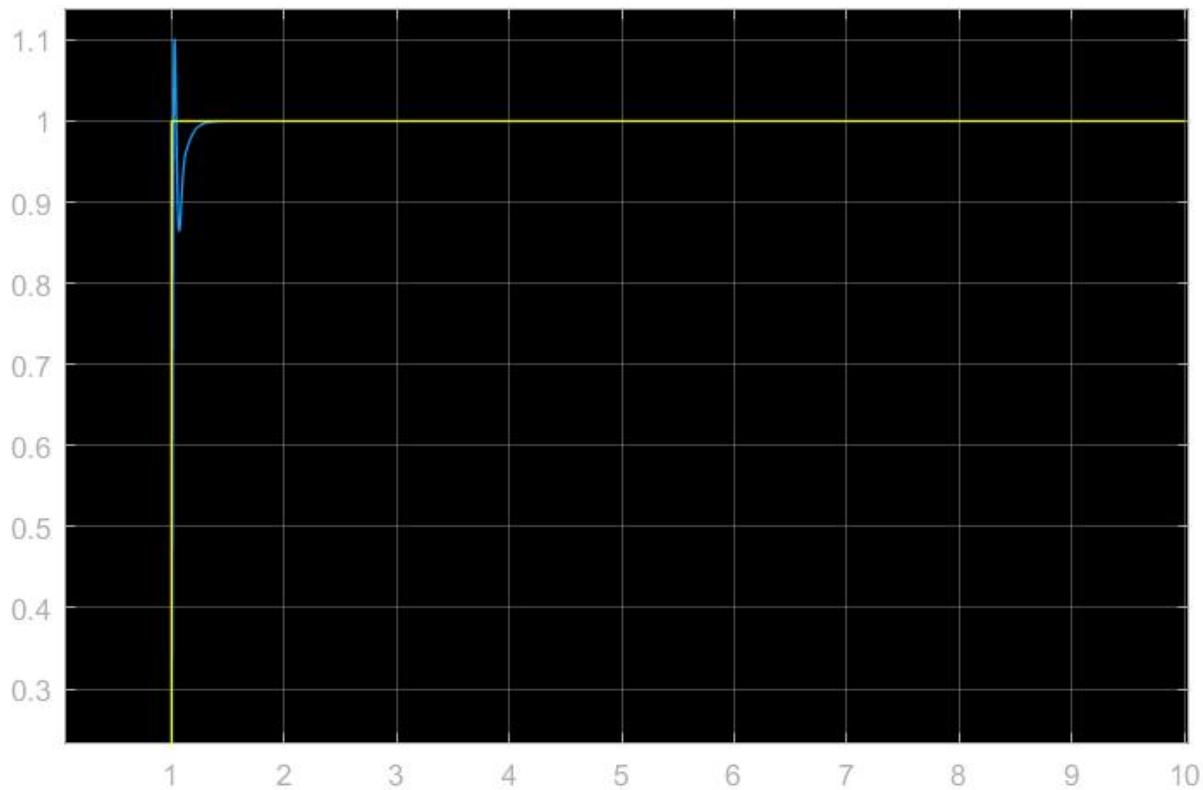


Figure 4.16: Closed-Loop Step plot of the Kangaroo Robot

	Tuned	Block
P	4.1693	1
I	67.1737	1
D	0.064537	1

	Tuned	Block
Rise Time	0.0125 seconds	0.0039 seconds
Settling Time	0.173 seconds	5.63 seconds
Overshoot	10.2%	64%
Peak	1.1	1
Phase Margin	64.5 deg at 117 rad/sec	84 deg at 1490.3 rad/sec
Closed-Loop Stability	Stable	Stable

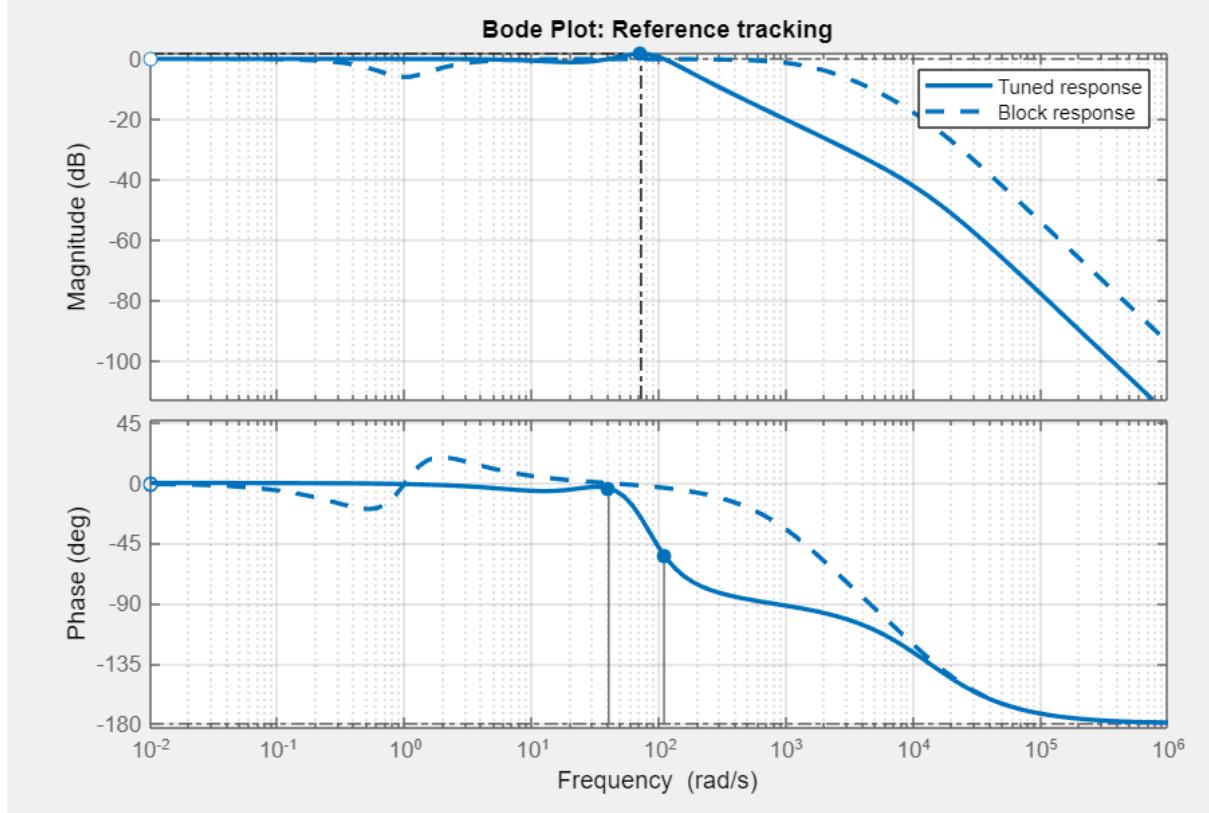


Figure 4.17: Bode Plot for Kangaroo Robot Leg, Frequency Response

As shown in Figure 4.17, the gain margin and the phase margin increased while keeping the phase margin greater than the gain margin which guarantees stability.

Kangaroo Closed-Loop Transfer Function:

$$\text{Controller} = 4.17 + 67.2 * \frac{1}{s} + 0.0645 * \frac{s}{13400s + 1} \quad (4.4)$$

$$\text{KangarooCL} = \frac{4.169s^2 + 67.17s + 0.005015}{7.3s^4 + 68s^3 + 1.093e04s^2 + 67.99s + 0.005015} \quad (4.5)$$

KangarooCL in z-domain using a sample time of 0.01 seconds:

$$= \frac{0.00002884z^3 - 0.00002391z^2 - 0.00003002z + 0.00002509}{z^4 - 3.786z^3 + 5.49z^2 - 3.621z + 0.9175} \quad (4.6)$$

4.6 Force Tracking

In this section, an exponential chirp signal is applied with a frequency range of 0.1Hz to 100 Hz with an amplitude of 50 to test the controller designed for force tracking. Figure 4.18 shows the open loop response for the kangaroo robot leg and Figure 4.19 shows the closed loop response. The results are satisfactory as the force input signal is accurately tracked.

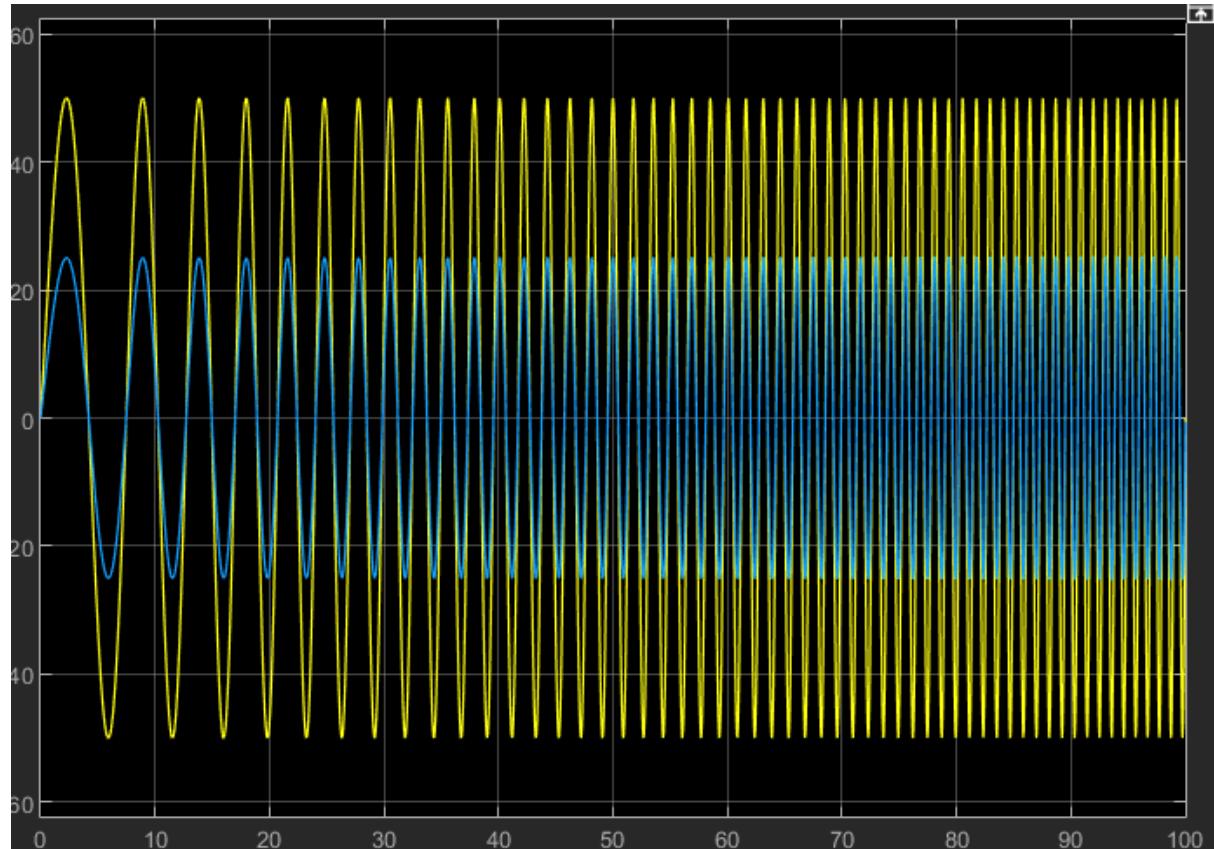


Figure 4.18: Open-Loop Response for exponential chirp signal. The yellow signal is the input signal and the blue is the output signal.

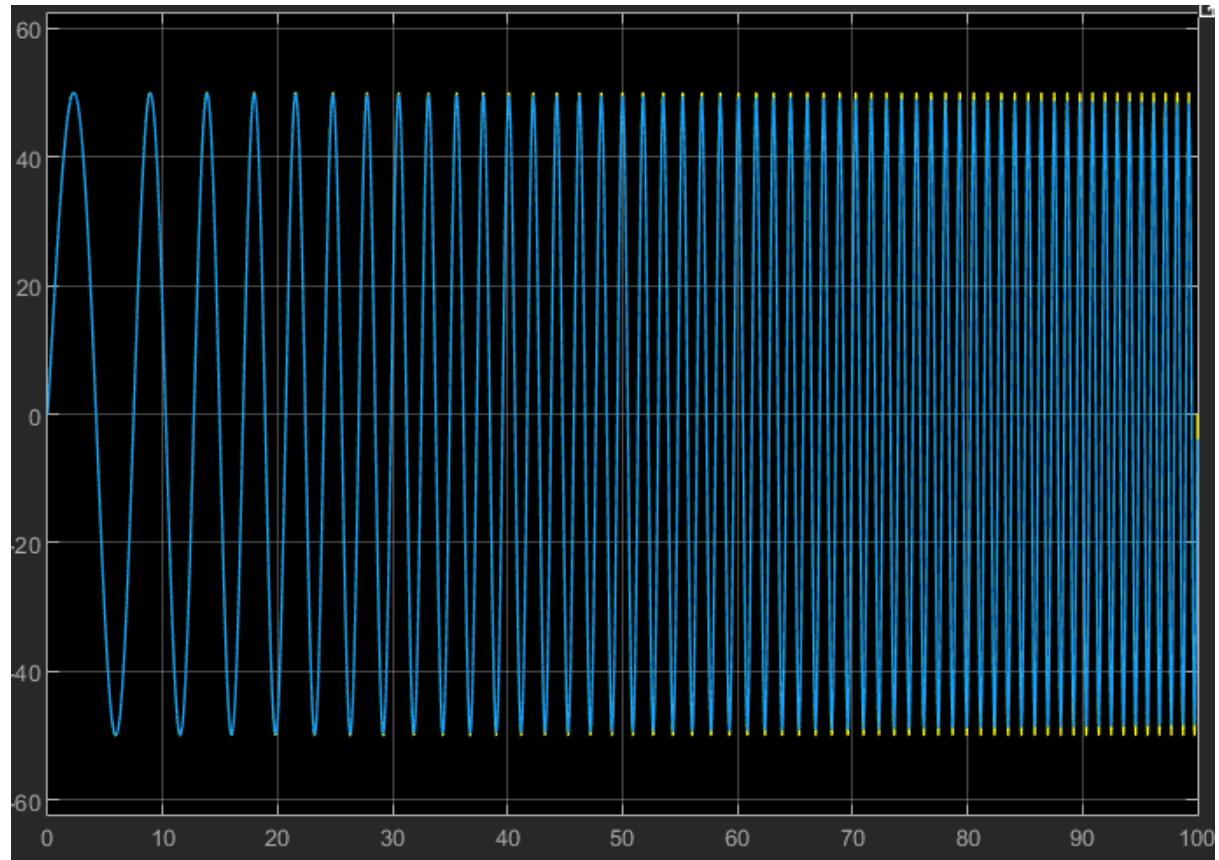


Figure 4.19: Closed-Loop Response for exponential chirp signal. The yellow signal is the input signal and the blue is the output signal.

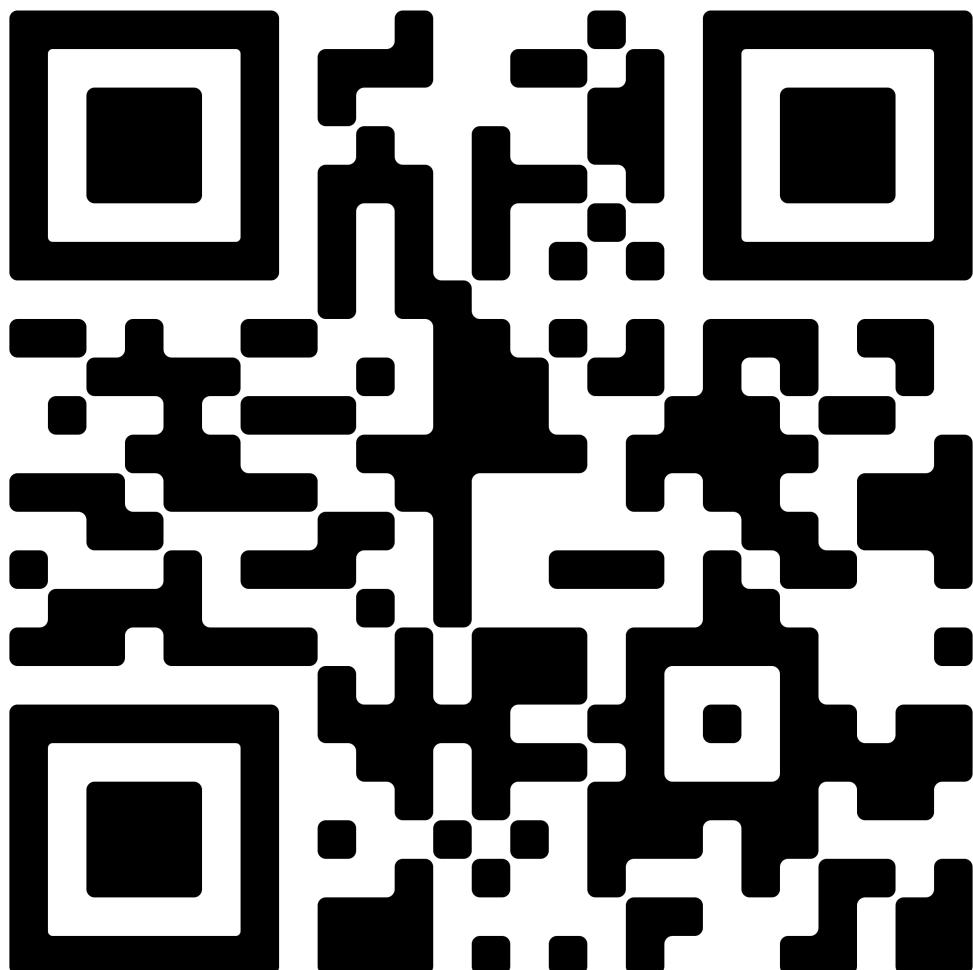


Figure 4.20: Link providing the videos of testing both the feedback system and the PID controllers on hardware.

Chapter 5

Conclusion

Series Elastic Actuators provide many benefits in force control of robots in unconstrained environments. These benefits include high force fidelity, extremely low impedance, low friction, and good force control bandwidth. Series Elastic Actuators employs a novel mechanical design architecture that goes against the common machine design principle of “stiffer is better”. A compliant element is placed between the gear train and the driven load to intentionally reduce the stiffness of the actuator. These characteristics are desirable in many applications including legged robots, exoskeletons for human performance amplification, robotic arms, haptic interfaces, and adaptive suspensions.

One of these characteristics was tested in this paper, the ability to jump. A mechanism is applied to the leg till the leg touches the ground and the spring compresses. When the spring compresses, a position sensor measures the deflection using a pulley system designed and implemented, and the force output is accurately calculated using Hooke’s Law ($F = Kx$). This calculated force is compared to the desired force calculated from the motor. A control loop then servos the actuator to the desired output force. The resulting actuator has inherent shock tolerance, high force fidelity, and extremely low impedance.

Chapter 6

Future Work

The Bachelor's project work throughout the semester was to design and implement a feedback system on a Series Elastic Actuator Robotic Legs, Cheetah, and Kangaroo. Future Work that can be performed :

- Performing a design, model, and control for a Series Elastic Actuated Running and walking leg mechanisms.
- Perform high-level control techniques on legged robotics SEA-based like Adaptive Control, Fuzzy Logic Control, or Linear Quadratic Regulator Control(LQR).
- Implement the Series Elastic Actuator using the F-SEA model.
- Implement a dynamic stability criterion for walking and balancing using control architectures using an IMU.
- Use an incremental Rotary Encoder to avoid the tension of the belt on pulley and mount the encoder better.

Appendix

```

Open-Loop:
const int pwm = 9;
const int dir = 10;
const int inputCLK = 4;
const int inputDT = 3;
int pinAstateCurrent = LOW;
int pinAstateLast = pinAstateCurrent;
String encdir = "";
double deflection = 0;
int counter = 0;

void setup() {
    // put your setup code here, to run once:
    pinMode(pwm,OUTPUT);
    pinMode(dir ,OUTPUT);
    pinMode (inputCLK ,INPUT);
    pinMode (inputDT ,INPUT);
    Serial.begin(9600);
    attachInterrupt(digitalPinToInterrupt(inputDT), update , CHANGE);
}

void update() {

    // ROTATION DIRECTION
    pinAstateCurrent = digitalRead(inputCLK);

    // If there is a minimal movement of 1 step
    if ((pinAstateLast == LOW) && (pinAstateCurrent == HIGH)) {

        if (digitalRead(inputDT) == HIGH) {
            counter++;
            deflection = counter*1.8/12.0;
            Serial.println(deflection);
        } else {
            counter--;
            deflection = counter*1.8/12.0;
            Serial.println(deflection);
        }
    }
}

```

```

pinAStateLast = pinAstateCurrent;

}

void loop() {
    // put your main code here , to run repeatedly:
    for( int x=0;x<63;x++)
    {
        float y = float(x)/10;
        delay(100);
        if(y>0 && y<3.14)
        {
            digitalWrite(dir ,HIGH);
        }
        else
        {
            digitalWrite(dir ,LOW);
        }
        analogWrite(pwm,10* sin(y));
        // Serial. print("Wave: ");
        Serial. println(10* sin(y));
        // Serial. print("Deflection: ");
        // Serial. print(deflection);
    }
}

```

Closed-Loop:

```

const int pwm = 9;
const int dir = 10;
const int inputCLK = 6;
const int inputDT = 7;
int max = 0;
double x=0;
int counter = 0;
int currentStateCLK;
int previousStateCLK;
String encdir ="";
double F_meas = 0;
double tau = 0;

```

```

double rpmo = 0;

double kp = 4.2294;
double ki = 65.5255;
double kd = 0.068083;
unsigned long currentTime, previousTime;
double elapsedTime;
double error;
double lastError;
double input, output, setpoint;
double cumError, rateError;

void setup() {
    pinMode(pwm,OUTPUT);
    pinMode(dir ,OUTPUT);
    pinMode (inputCLK ,INPUT);
    pinMode (inputDT ,INPUT);
    Serial .begin (9600);
    setpoint = 5.15;
    previousStateCLK = digitalRead (inputCLK );

}

double measure()
{
    currentStateCLK = digitalRead (inputCLK );
    if (currentStateCLK != previousStateCLK){
        if (digitalRead (inputDT) != currentStateCLK) {
            counter --;
            if (abs (counter)>max)
            {
                max = abs (counter);
            }
        } else {
            counter++;
            encdir ="CW";
            if (abs (counter)>max)
            {
                max = abs (counter);
            }
        }
    }
}

```

```

}

Serial.print("    ");
Serial.print("counter= ");
Serial.print(counter);
Serial.println();
Serial.print("max= ");
Serial.print(max);
Serial.println();
Serial.print("x= ");
Serial.print(x);
Serial.println();

}

previousStateCLK = currentStateCLK;
return max*1.8/11.0;
}

double computePID(double inp){
    currentTime = millis();
    elapsedTime = (double)(currentTime - previousTime);

    error = setpoint - inp;
    cumError += error * elapsedTime;
    rateError = (error - lastError) / elapsedTime;

    double out = kp*error + ki*cumError + kd*rateError;

    lastError = error;
    previousTime = currentTime;

    return out;
}

void loop() {
    x = measure();
    F_meas = 10930.4*x*0.01;
    output = computePID(F_meas);
    digitalWrite(dir,LOW);
    // analogWrite(pwm,output);
}

```

```
tau = output/1256.64;
rpmo = 944.82*tau - 250;
analogWrite(pwm, output);
delay(1000);
analogWrite(pwm,255);
delay(3000);
digitalWrite(dir ,HIGH);
analogWrite(pwm,200);
max = 0;
}
```

Bibliography

- [1] Marsrover. <https://www.space.com/17963-mars-curiosity.html>.
- [2] Industrial robot. <https://www.atriainnovation.com/en/robotica-industrial-tradicionales-colaborativos-y-adaptativos/>.
- [3] Industrial robot components. <https://www.osha.gov/otm/section-4-safety-hazards/chapter-4>.
- [4] Cobots. <https://www.automate.org/blogs/cobots-empowering-humans-in-manufacturing>.
- [5] Cobot walking. <https://d-nb.info/1009450484/34>.
- [6] Passive compliance. <https://www.mdpi.com/2076-3417/10/1/340/htm>.
- [7] Series elastic actuator. <https://www.frontiersin.org/articles/10.3389/fnbot.2019.00017/full>.
- [8] Gill A Pratt and Matthew M Williamson. Series elastic actuators. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 1, pages 399–406. IEEE, 1995.
- [9] Nicholas Paine and Luis Sentis. A new prismatic series elastic actuator with compact size and high performance. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1759–1766. IEEE, 2012.
- [10] Gordon Wyeth. Control issues for velocity sourced series elastic actuators. In *Proceedings of the 2006 Australasian Conference on Robotics and Automation*, pages 1–6. Australian Robotics and Automation Association, 2006.
- [11] Model for sea. <https://www.semanticscholar.org/paper/Development-and-Simulation-of-the-Series-Elastic-Yun-Bang/535d6368274dfecbe04808016e694c0af5ccddb1>.
- [12] Motor force. <https://www.orientalmotor.com/technology/motor-sizing-calculations.html>.
- [13] Motor graph. <http://img.hisupplier.com/var/userFiles/2008-09/28/130450.711.pdf>.
- [14] Pid. <https://commons.wikimedia.org/wiki/File:PID.svg>.

- [15] Cytron motor driver. <https://www.amazon.eg/-/en/Cytron-Dual-Channel-Motor-Driver/dp/B091GGDTJS>.
- [16] Arduino pinout. [https://www.arduino.cc/.](https://www.arduino.cc/)
- [17] Arduino uno. <https://store.arduino.cc/products/arduino-uno-rev3>.
- [18] Power supply. <https://dealna.com/ar/Article/Post/6630/Computer-Power-Supply>.
- [19] Rotary encoder. <https://howtomechatronics.com/tutorials/arduino/rotary-encoder-works-use-arduino/>.
- [20] Encoder pinout. <https://forum.arduino.cc/t/arduino-2-rotary-encoder-reading/652273>.
- [21] Timing belt. <https://www.rs-online.vn/p/htd-timing-belt-450-5m-15/1755214/>.
- [22] L-holder. <https://www.amazon.com/Mironey-2-Pack-Bracket-Aluminum-Thickness/dp/B08K8DPYJH>.
- [23] Pulley. <https://www.amazon.eg/-/en/Timing-pulley-Teeth-Pulleys-Printer/dp/B0968ZPMWM>.
- [24] Thomas R Kurfess et al. *Robotics and automation handbook*, volume 414. CRC press Boca Raton, FL, 2005.
- [25] Kevin W Hollander, Robert Ilg, Thomas G Sugar, and Donald Herring. An efficient robotic tendon for gait assistance. 2006.
- [26] ZM Bi, Chaomin Luo, Zhonghua Miao, Bing Zhang, WJ Zhang, and Lihui Wang. Safety assurance mechanisms of collaborative robotic systems in manufacturing. *Robotics and Computer-Integrated Manufacturing*, 67:102022, 2021.
- [27] Qiang Huang, Kazuhito Yokoi, Shuuji Kajita, Kenji Kaneko, Hirohiko Arai, Noriho Koyachi, and Kazuo Tanie. Planning walking patterns for a biped robot. *IEEE Transactions on robotics and automation*, 17(3):280–289, 2001.
- [28] Eric L Faulring, J Edward Colgate, and Michael A Peshkin. Cobotic architecture for prosthetics. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5635–5637. IEEE, 2006.
- [29] Chi Zhang, Wei Zou, Liping Ma, and Zhiqing Wang. Biologically inspired jumping robots: A comprehensive review. *Robotics and Autonomous Systems*, 124:103362, 2020.
- [30] Elena Garcia, Juan Carlos Arevalo, Gustavo Muñoz, and Pablo Gonzalez-de Santos. Combining series elastic actuation and magneto-rheological damping for the control of agile locomotion. *Robotics and Autonomous Systems*, 59(10):827–839, 2011.

- [31] Marco Ceccarelli, Daniele Cafolla, Matteo Russo, and Giuseppe Carbone. Design and construction of a demonstrative heritagebot platform. In *International Conference on Robotics in Alpe-Adria Danube Region*, pages 355–362. Springer, 2017.
- [32] Nicholas Paine, Sehoon Oh, and Luis Sentis. Design and control considerations for high-performance series elastic actuators. *IEEE/ASME Transactions on Mechatronics*, 19(3):1080–1091, 2013.
- [33] Chan Lee, Suhui Kwak, Jihoo Kwak, and Sehoon Oh. Generalization of series elastic actuator configurations and dynamic behavior comparison. *Actuators*, 6(3), 2017.
- [34] Stuart Bennett. Development of the pid controller. *IEEE Control Systems Magazine*, 13(6):58–62, 1993.
- [35] Yusuf Abdullahi Badamasi. The working principle of an arduino. In *2014 11th international conference on electronics, computer and computation (ICECCO)*, pages 1–4. IEEE, 2014.