DEVELOPMENT AND BALANCING CONTROL OF A UNICYCLE ROBOT

by

Nabeel Ahmad Khan Jadoon

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Mechatronics and Machine Intelligence

Examination Committee:	Prof. Manukid Parnickhun (Chairperson)
	Dr. Mongkol Ekpanyapong
	Prof. Masaki Yamakita
Nationality:	Pakistani
Previous Degree:	Bachelor in Mechatronics Engineering
	NWFP University of Engineering &
	Technology Peshawar, Pakistan
Scholarship Donor:	His Majesty the King's Scholarships (Thailand)

Asian institute of Technology
School of Engineering and Technology
Thailand
May 2024

AUTHOR'S DECLARATION

I, Nabeel Ahmad Khan Jadoon, declare that the research and experimental work carried out for this thesis was in seemingly accordance with the regulations of the Asian institute of Technology. The work presented in it is my own and has been generated and developed by me as the result of my original research, and if external sources were used, such sources have been cited where needed. It is original and has not been submitted to any other institution to obtain another degree or qualification. This is a true copy of the thesis, including final revisions.

Date: May 2024

Name: NABEEL AHMAD KHAN JADOON

Signature: N@bil

ACKNOWLEDGMENTS

I am very thankful to Professor Manukid Parnickhun from the Asian institute of Technology, Thailand for supervising my thesis and guiding me toward excellence. He has been a great source of Inspiration for my core mechatronics and robotics field. Moreover, a bundle of gratitude to Dr. Mongkol Ekpanyapong for being on my thesis committee and for the better selection of courses due to which I was able to grab core knowledge implemented in the thesis. I couldn't agree more that due to my amazing faculty at AIT, I can go for my research exchange activities under the Young Science and Engineering Program at Tokyo Institute of Technology (Tokyo Tech) Japan. I am very grateful to Professor Masaki Yamakita for encouraging me for this venture in Japan and for guiding me throughout the process and being in my committee examination.

I would like to acknowledge the generous contribution of the H.M. KINGs scholarship to making my dream come true. Additionally, I am grateful to the Government of Japan for offering the JASSO scholarship which made it possible to initiate my research ventures in Japan.

I am grateful to my parents, my advisor, and my friends for always encouraging me to achieve something bigger.

ABSTRACT

This thesis introduces the design, modeling, and control strategies employed in the development of a self-balancing unicycle robot. It investigates both classical (model-based) and reinforcement learning (model-free) methodologies. A compact reaction-wheel-based unicycle is carefully designed using Solidworks and produced using 3D printing and the integration of electronics for assembly for practical experimentation. the design of the proposed unicycle offers two unstable degrees of freedom characterized by pitch and roll, providing ground for nonlinear model control research in robotics. for the simulation study, a continuous control scheme is proposed, featuring two distinct algorithms: Linear Quadratic Regulator (LQR) in the classical domain and Deep Deterministic Policy Gradient (DDPG) in the deep reinforcement learning domain, tailored for balancing and maneuvering tasks. MATLAB is employed for classical control simulations, while Pybullet Physics Engine with Python interface is utilized for DDPG-based reinforcement learning simulations, effectively demonstrating the efficacy of the proposed control strategies. for discrete control, a proof-of-concept model based on a 2D inverted pendulum is proposed to explore self-erecting dynamics for unicycles. The performance of the control algorithms is rigorously assessed through comprehensive testing procedures, focusing on metrics such as settling time, overshoot, and robustness to external stimuli. Analysis of the results indicates that both methods demonstrate potential for balancing unicycles. However, LQR outperforms DDPG across various scenarios, showcasing greater robustness and stability, particularly concerning steady-state performance. Conversely, DDPG shows promising results and exploratory behavior yet the effective policy transfer to hardware is left for future reference. For classification, for systems with non-linear study the DRL method is suggested while the classical control methods are recommended for systems with a known dynamic, owing to their simplicity and robustness, contingent upon applicability.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CAD = Computer-Aided Design
RL = Reinforcement Learning
CL = Classical Control
AMLC = Adaptive Model Linear Control
DOF = Degrees of Freedom
PID = Proportional integral Derivative
SRB = Single Rigid Body
STLs = Stereolithography
MPC = Model Predictive Control
ABS = Acrylonitrile butadiene styrene
PD = Proportional Derivative
LQR = Linear Quadratic Regulator
PI = Proportional integral
COG = Centre of Gravity
IMU = inertial Measurement Units
LPF = Low Pass Filters
KF = Kalman Filter
SE = State Estimation
IP = inverted Pendulum
RBM =Rigid body modeling
TO = Trajectory optimization
DP = Dynamic Programming
RLOC = Reinforcement Learning Optimal Control
SARSA = Tuple of State-Action & Reward-State-Action (Pairs)
MC = Monte-Carlo
DDPG = Deep-Deterministic Policy Gradient (DRL)
PPO = Proximal-Policy-Optimization
A2C = Actor-Critic Network

CHAPTER 1
INTRODUCTION

1.1 Inspiration Behind this Research

In recent years, significant advancements have been made in the realm of control systems, particularly with the emergence of model-free approaches like reinforcement learning, alongside traditional classical control methods. these advancements offer considerable flexibility in controlling inherently unstable systems, thereby facilitating their practical application in various domains.

Concurrently, there has been a notable surge in research interest directed towards actuators, sensors, and embedded controllers, fueling innovation in control and robotics. Furthermore, the proliferation of 3D printing technology has revolutionized the creation of advanced testbeds, further propelling the growth of experimental platforms. These platforms serve as invaluable resources for the control and robotics community, providing a robust foundation for investigating dynamic properties and offering novel avenues for mechatronic solutions. Considering these developments, this study endeavors to explore a mechatronic platform characterized by instability and underactuation, specifically focusing on a unicycle robot.

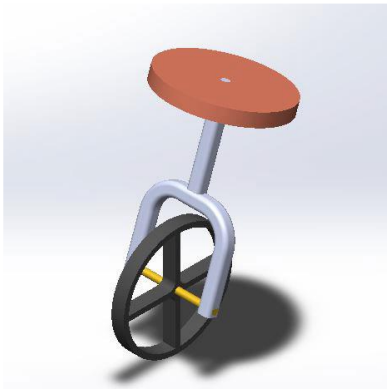
Unicycle robots offer unique challenges and opportunities in the field of robotics. the potential for practical applications, such as transportation search and rescue, and navigating through narrow paths makes them an attractive area of study. Additionally, the development of unicycle robots requires innovative solutions in control and sensing that could have broader implications for other areas of robotics. Finally, the visual impact of a robot balancing on a single wheel makes unicycle robots an exciting and captivating area of research to study both linear and non-linear effects.

The foundational works by A. Schoonwinkle et al., laid the groundwork for understanding the dynamics of unicycles, drawing Inspiration from how humans balance while riding them. As depicted in Figure 1.1, the key components of a unicycle include the wheel, body, and disc. The unicycle is conceptualized as a representation of the lower part of the rider, with the frame serving as the balancing element.

Additionally, a turntable mechanism is employed to model the movements akin to those made by a rider, simulating the twisting motion of the torso and arms for balance.

Figure 1.1

Unicycle Concept (Schoonwinkle et al)



With the introduction of innovative mechatronic solutions, this research endeavors to provide a platform for investigating the dynamic properties of the Wheelbot. Drawing upon various conceptual architectures of robotic testbeds and employing a system-level approach to design, this work scrutinizes critical design decisions, including the determination of the number of (DOFs) and the development of electromechanical components. Unicycle robots, in comparison to other self-balancing counterparts such as biped-wheel, Segway, and ballbot, offer distinct advantages. these benefits include:

1. *High mobility*: the unicycle robot offers the advantage of navigating through narrow spaces, and rough terrain.
2. *Simple design*: the design of a unicycle robot is relatively simple compared to other types of robots. It typically consists of a single wheel, a motor, and a few sensors. This makes it easy to build and maintain.
3. *Low power consumption*: Unicycle robots require less power compared to other types of robots. The single-wheel design allows for efficient use of energy, making it possible for them to operate for longer periods.
4. *Versatile*: Unicycle robots can be used in several applications, including search and rescue, surveillance, and transportation. Their high mobility and simple design make them a great choice for tasks that require agility and flexibility.

5. *Cost-effective*: Because of their simple design and low power consumption, unicycle robots are generally less expensive to build and maintain.
6. *On-board Battery*: this robot could carry its battery onboard unlike other unicycle robots while maintaining the compact size on the go.

1.2 Problem Statement

Testing model-free control and model-based dynamics controls, such as reinforcement learning and classical learning control, on unicycle robots often poses significant challenges due to the substantial hardware development requirements and associated maintenance costs. in today's landscape, hardware development is characterized by its time-consuming nature, mechanical complexity, and high expenses, making it difficult to swiftly produce robots for research activities in university or laboratory settings. Consequently, there arises a pressing need to streamline efforts through compact design solutions and leverage 3D printing technology for the rapid prototyping of robots.

In response to this imperative, this research advocates for the utilization of 3D printing technology alongside electromechanical control, enabling the creation of compact-sized robots. Additionally, the study conducts a comparative analysis of control performances employing both advanced and classical control methodologies, such as reinforcement learning and Linear Quadratic Regulator (LQR), respectively.

1.2.1 For Self-Balancing Part

The Wheelbot, or unicycle, offers a versatile platform for integrating off-the-shelf electric motors, often requiring minimal operating space. Its wheel configuration typically comprises rolling wheels, which exploit friction forces for locomotion, and reaction wheels, capable of applying free torques or moments. Examples of such designs can be found in Ballbots developed by P. Fankhauser and U. Nagarajan in 2010 and 2014, as well as Segway robots like Ascento developed by V. Kemm et al. in 2019.

From a dynamic perspective, achieving balance with a flywheel wheel necessitates swift adjustments in the motor's rotational direction, potentially leading to high motor velocities. Consequently, utilizing an electric motor to actuate a reaction wheel entail addressing rate-dependent control limitations, highlighting the importance of investigating optimization-driven control strategies, as discussed by Xiong et al. in 2021. On the other hand, the rotational movement of rolling wheels typically occurs at

lower speeds, as observed in the Ascento design. Additionally, utilizing rolling wheels may require addressing non-minimum phase linearized closed-loop system dynamics, as evidenced in prior research by P. Fankhauser and U. Nagarajan in 2010 and 2014.

This study concentrates on employing DC motors, known for enhancing the maneuverability of wheeled robots like Cubli, as demonstrated by Muehlebach et al. in 2017, and Ascento, capable of executing rapid and discontinuous dynamic maneuvers. These motors play a pivotal role in enabling the Wheelbot to perform agile movements that are subject to the discontinuity of robot dynamics.

1.2.2 For Reaction Wheel Unicycle

The exploration of single-wheel robotic systems necessitates the creation of compact, underactuated control mechanisms capable of navigating non-holonomic and rapidly changing environments, particularly within constrained laboratory settings. to fulfill these demanding criteria, this investigation confronts design hurdles by integrating a dual-wheel setup affixed to a rigid body, enabling ground traversal while serving as reaction wheels atop the structure. Previous studies have proposed diverse layouts for reaction wheel-equipped unicycle robots, with rotational axes arranged either in alignment or perpendicular to the centers of wheels.

Unicycle robots with coaxial setup cannot control the roll degree of freedom (DOF), instead inducing motion in the tilting while depending on the rolling wheel to avert toppling. Conversely, the orthogonal setup of unicycles facilitates direct manipulation of both the roll and pitch DOFs.

Hence, this study adopts an orthogonal setup to enable direct control of robots in the longitudinal and lateral directions. The coupling dynamics can help to tune both pitch and roll controllers independently. These controllers establish a fundamental framework for addressing more intricate research inquiries robot the yaw dynamics or its navigation along predetermined trajectories. Recent proposals for orthogonal-configuration unicycles, as advanced by G.P. Neves and Angelico in 2021, employ very high-inertia reaction wheels to minimize flywheel wheel acceleration and mitigate the risk of exposing electronics to high-consuming currents at high speeds.

To tackle the research gaps, this investigation endeavors to develop a Wheelbot capable of adeptly managing substantial disturbances in both roll and pitch directions, all while integrating an onboard power source for maneuverability.

1.3 Research Questions

1. Can a proposed robot be produced at the Laboratory level for testing?
2. Can the design of a Wheelbot robot reduce the complexity of 3D-printed parts?
3. Would it be possible for this robot to carry its power supply while maintaining balance and compact size?
4. Can this robot be controlled using the concept of the internet of Things (IoT)?
5. Can classical control and reinforcement learning be implemented on this robot?

1.4 Objectives of Research

This research aims to develop compact-size, self-balancing unicycle robots with orthogonal configurations. the objectives of the research are:

1. To develop a Wheelbot or unicycle robot with one rolling and one reaction wheel for pitch and roll control respectively.
2. To propose the dynamic model of a unicycle robot using rigid body modeling and present simulation results.
3. To investigate state estimation by using the concept of sensor fusion and design estimation curriculum.
4. To design a balancing control scheme for a Wheelbot using a Linear Quadratic Controller (LQR) and reinforcement learning for the robot's self-erection.
5. To implement control algorithms in MATLAB Simulink/Python and compare the performance of RL vs simple LQR-based control.
6. To present an IOT-based (wireless control) mechanism for a real robot using an embedded controller.

1.5 Scope of this Work and Limitations

1.5.1 Scope of the Study

1. The overall robot will weigh a total of 0.8 kg.
2. The robot has a full height of about 120mm and a width of about 100mm.
3. The robot wheel will have a small neighborhood for balancing in an upright position.

4. The robot is meant to maneuver on both smooth acrylic glass and rubber mats.
5. The Wheelbot motor's applied torque will be around 1 Nm to reject the impulse disturbance.

The electronic specifications of the robot are well-mentioned in the appendix section of this report. Other specifications of the Wheelbot as per individual category are mentioned below:

Table 1.1

Specifications of Proposed Robot

Category	Value
Total Weight	0.7 kg
Reaction Wheel weight	0.15 kg
Operating time	Rate of battery charge
Supply Voltage	12 V DC
Battery Capacity	3s 500 25C
Nominal Motor torque	70mN.m (stall)

1.5.2 Limitations of Research

1. The electromagnetic inference (EMI) that comes from brushless motors may add noise/disturbance to the performance control of Wheelbot. (It may then be reduced using shielding – will still be tested after build-up).
2. Increased motor noise may impair reinforcement learning control for performance.
3. Hardware training using RL is difficult because of the unstable nature of the robot.
4. The analysis of the nonlinear yaw dynamics in thesis calculations will exceed the scope of this work.
5. The robot will not perform any other controlling schemes until and unless explicitly mentioned in the scope.

CHAPTER 2
LITERATURE REVIEW

This chapter of the report presents the previous work done by researchers in robotics, especially for the design of unicycle robots, and other mobile underactuated robots and algorithms designed for balancing and jumping control. Moreover, also discusses the control methods (like reinforcement learning and LQR) for unicycle robots.

2.1 Overview of the Literature

Unicycle robots inherently exhibit instability, susceptible to tipping in both the longitudinal and lateral directions while maintaining a heading in the yaw direction. Consequently, when structuring such robots, paramount consideration must be given to achieving equilibrium within the system, a crucial aspect of the stabilization process. Various techniques exist to uphold the robot in an upright orientation.

This section delves into the exploration of balancing mechanisms employed in various research endeavors, elucidating the roles of motors and sensors in achieving stabilization for a typical unicycle robot.

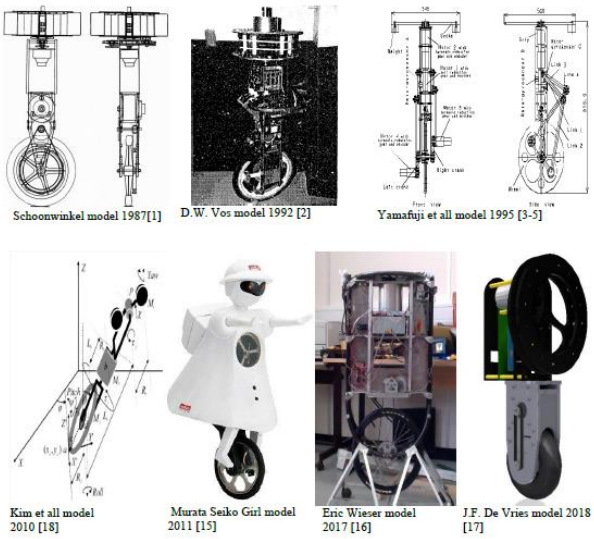
The idea of a unicycle comes from humans riding it and this control structure can be seen in Figure 2.1. this structure encompasses four distinct levels:

1. *Distributed information Stage:* this level represents the distribution of information within the system, with sub-levels further categorizing and organizing the available information.
2. *Logical System Stage:* the logical system utilizes distributed information to formulate logical relationships and rules governing the behavior of the unicycle.
3. *Decision Support System Stage:* this level provides support for decision-making processes based on the logical system's output and external factors. It assists in determining appropriate actions or responses to dynamic changes in the environment or system state.
4. *Dynamic Mechanical System Stage:* At the lowest level of the hierarchy, the dynamic mechanical system represents the physical embodiment of the unicycle, including its mechanical components, sensors, actuators, and interactions with the environment.

Each level of the hierarchy contributes to the overall functioning of the control system, with distributed information serving as the foundation upon which logical reasoning, decision-making, and mechanical actions are built. This hierarchical structure facilitates a systematic approach to understanding and controlling the complex dynamics of the robotic unicycle, akin to the way a human rider navigates and balances on a unicycle.

Figure 2.1

Models of Unicycles from Previous Years (Sergey & Viktor, 2020, INESYS Moscow)



Currently, there is significant global attention directed towards mobile robots, driving extensive research and development endeavors. This encompasses diverse areas such as mechanism design, control algorithms, multi-robot cooperation, and beyond. the operational environments of wheel robots can be broadly classified into 2D and 3D, each presenting distinct challenges and potential avenues for exploration.

In two-dimensional space, mobile robots such as Automated Guided Vehicles (AGVs) and vacuum cleaner robots are prevalent. AGVs are extensively utilized in factory settings, while vacuum cleaner robots find widespread use in households. These robots remain in constant contact with the ground, rendering them physically stable and obviating the need for additional energy for balancing purposes. Research in this domain typically focuses on trajectory tracking, motion control, artificial intelligence (AI), and multi-robot cooperation. Conversely, mobile robots operating in three-

dimensional space present a different set of challenges. these robots, such as bicycle robots, ball bots, unicycle robots, and jumping robots, are inherently unstable.

Figure 2.2 illustrates several types of mobile robots functioning in three-dimensional space. in contrast to their two-dimensional counterparts, these robots must contend with dynamic balance issues and complex motion planning requirements. Research endeavors in this realm often concentrate on developing robust control strategies, dynamic stability mechanisms, and innovative locomotion techniques to navigate challenging terrains and environments.

Figure 2.2

(a) AGV (b) The Cleaner Robot (vacuum) (c) Bicycle Robot (d) Unicycle Robot (e) Jumping Robot (Surachat & Manukid, 2017, AIT Thailand)

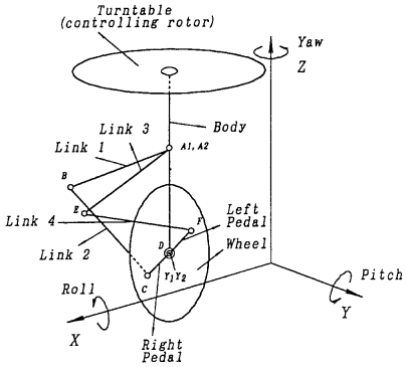
2.2 Mechanical Aspects of Robot Design

2.2.1 Hardware Mechanisms for Unicycle Robot

As previously mentioned, there are two main configurations for reaction wheel unicycle robots: coaxial and orthogonal to the line connecting the centers of both wheels. These configurations have distinct effects on the control and dynamics of the robot. Overall, the choice between coaxial and orthogonal configurations depends on the specific requirements and objectives of the robot's operation, with each configuration offering unique advantages and trade-offs in terms of stability, control complexity, and maneuverability.

Figure 2.3

A Turntable-Based Approach for Balancing Unicycle



The second method employed in stabilizing unicycle robots involves mass balancing, which entails adjusting the robot's center of gravity. The system comprises three primary components: the robot's ground wheel, either pendulum or flywheel at the top, and the robot’s body. The stabilizer, represented by the mass of the pendulum, is inserted into a robot. by swinging the mass of the pendulum sideways, the robot can be adjusted about its pivot point. The advantage of the mass balancing approach lies in its simplicity and straightforward mechanism of operation, allowing for the decoupling of robot movement between roll and pitch motions.

The ultimate approach to stabilize unicycle robots involves leveraging the principle of angular momentum. This technique encompasses two distinct sub-methods: the affixed flywheel and the moving flywheel. in the affixed flywheel setup, the reaction wheel axis aligns with the robot’s roll. by modulating the rotation speed of the flywheel, torque is generated to stabilize the robot. However, this method demands a robust motor to drive the flywheel, as it must contend with the significant inertia associated with its rotation.

2.2.2 Balancing Mechanism of Different Unicycle Robots

Previous studies have classified techniques for balancing unicycle robots into three primary categories: turntable, mass shifting, and angular momentum. The turntable approach involves utilizing a rotating platform akin to a unicycle rider's center. Positioned atop the robot, it can rotate fully, exerting torque to adjust the robot's orientation and stabilize the roll axis. However, symmetric turntables often struggle to

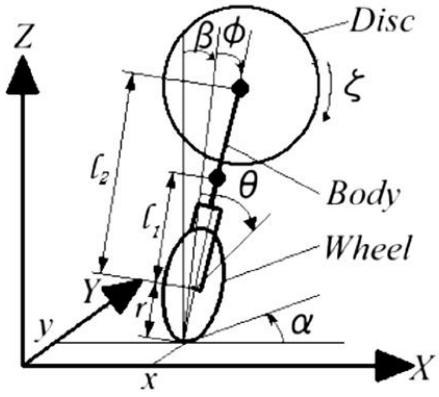
generate sufficient reaction torques for effective rotation, prompting exploration into asymmetric models.

An alternative method proposed by Majima and Kasai (2005) involves utilizing a disk rotating perpendicular to the wheel to apply momentum in the lateral direction, avoiding the need for moving weights to shift the center of gravity. While simplifying the design, this approach introduces additional dynamic and static effects, necessitating separate handling of lateral and longitudinal issues.

Similarly, Chantarachit (2011) suggested managing lateral and longitudinal control independently before combining them to achieve self-balancing. This approach resembles an inverted pendulum with a flywheel, where the inertial of the wheel can balance the lateral direction, as demonstrated by Xiaogang and Yu-Feng in 2010.

Figure 2.4

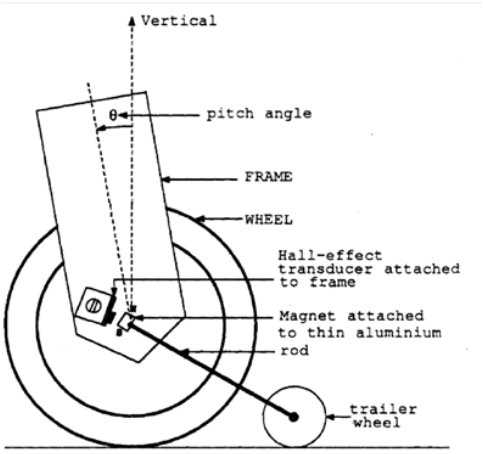
Unicycle Design Proposal by Majima & Kasai in 2005



The two classes of unicycle designs draw Inspiration from the control mechanisms exhibited by human riders. (Schoonwinkel, 1987) employed a comparable setup, modeling a human with three solid parts: the wheel to roll, Body structure, and a rotary turntable simulating the torso and arms of a rider. in the current work, a similar mechanism is proposed, taking Inspiration from Schoonwinkel's approach. This mechanism involves placing a supporting wheel to enable independent testing of pitch control.

Figure 2.5

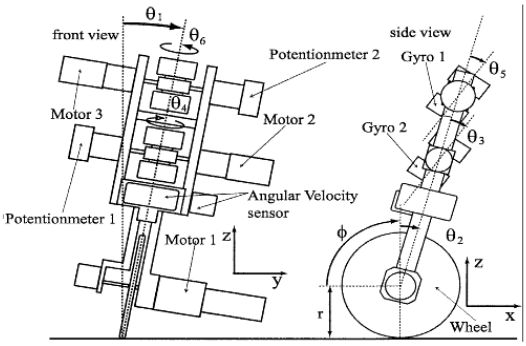
Mechanical Design Proposed by (Schoonwinkel, 1987)



An alternative method for stabilization harnesses the gyroscopic principle. When a disk rotates around the x-axis and is tilted about the y-axis, it produces a torque about the z-axis, a phenomenon known as gyroscopic precession. Here, the spinning wheel's angular momentum causes it to lag by 90 degrees to the applied torque.

Figure 2.6

Design of Unicycle Developed by (Minh-Quan & Kang-Zhi, 2005)

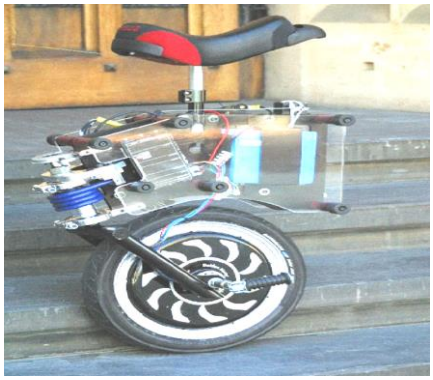


The turntable-based robot design has encountered challenges due to the high-power demand of the turntable actuator, necessary to rotate the unicycle body. This requirement arises from the need for the turntable's torque to arrange a setup where the torque is greater than the torque between ground and the wheel of the robot.

An intriguing alternative to this concept is showcased in the human-ridden unicycle, depicted in Figure 2.7. This design is proposed by A. Kadis et al., which diverges from traditional turntable-based approaches. Instead of relying on the turntable to change the unicycle's heading, the rider adjusts the direction by twisting their torso. This innovative adaptation bypasses the need for a high-power turntable actuator, offering a potentially more efficient and intuitive control mechanism for the unicycle.

Figure 2.7

Unicycle Robot Model by Kadis et al.,2010.



2.3 Developments in Previous Research at Robotics Lab, AIT

2.3.1 Balancing Control of Unicycle

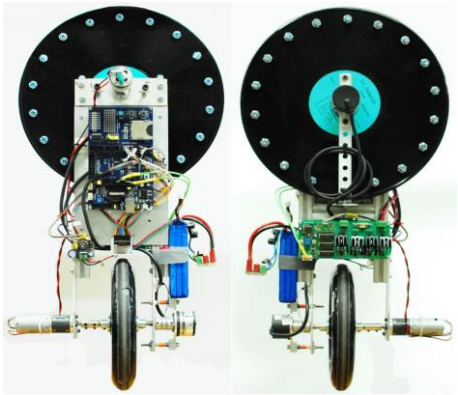
At Robotics Lab, AIT, previous research adopts the disk driver technique to generate torque for lateral balancing and the inverse pendulum technique for longitudinal balancing. Various sections of the thesis, including sensor fusion, mathematical modeling, and control algorithms, are explored. The focus lies on developing control algorithms to govern the robot's behavior.

Both PID and LQR controllers are applied to both simulation models and real plants to observe the robot's response. Simulation results are obtained using Simulink in the Matlab program. However, a significant challenge arises in achieving lateral balancing in the real system due to hardware limitations. for instance, the motor's response is too slow compared to the command input.

To address this challenge, the author suggests modifications in two key areas: mechanical enhancements, such as increasing motor power, and algorithmic improvements to ensure robust control. The proposed research aims to develop a robust algorithm for optimal control of the Wheelbot. Notably, an exploration into reinforcement learning techniques has not been conducted thus far, presenting an opportunity for future investigation (Surachat & Manukid, 2011)

Figure 2.8

Unicycle Robot Developed by Surachat & Manukid, 2011, AIT, Thailand



2.3.2 Self-Balancing Unicycle

At Robotics Lab AIT, a recent research project has successfully utilized control algorithms to achieve precise control actions. in this project, a DC motor was employed to stabilize longitudinal motion, while a single flywheel was used for lateral stabilization. The researchers tackled mathematical modeling for each direction separately, utilizing the Euler-Lagrange equation and state space representation to simplify linear equations.

Through a series of experiments, conducted independently for each direction with one side stabilized, the researchers achieved self-balancing capabilities. These experiments demonstrated that while PID control exhibited robustness, it was prone to oscillations. Conversely, LQR control proved to be robust and offered smoother operation.

Challenges were encountered due to interdependencies among roll, pitch, and yaw angles, as well as the overweight and slow response of the motor, which posed a risk of

the robot falling in any direction. Moreover, the research was limited to classical control schemes, and the investigation into reinforcement learning control was not explored.

The proposed research, which has already commenced, aims to address these challenges. It seeks to develop a control scheme where pitch and roll directions are coupled yet independently tunable. Additionally, the integration of self-erection capabilities in the proposed robot addresses the issue of falling on either side. Furthermore, the investigation into reinforcement learning control, combined with LQR, promises to identify key research gaps for parameter tuning and enhance the robustness of the control scheme.

Figure 2.9

Self-balancing Unicycle by Abhisesh & Manukid., 2012, AIT.

2.3.3 Two Flywheel Mechanism

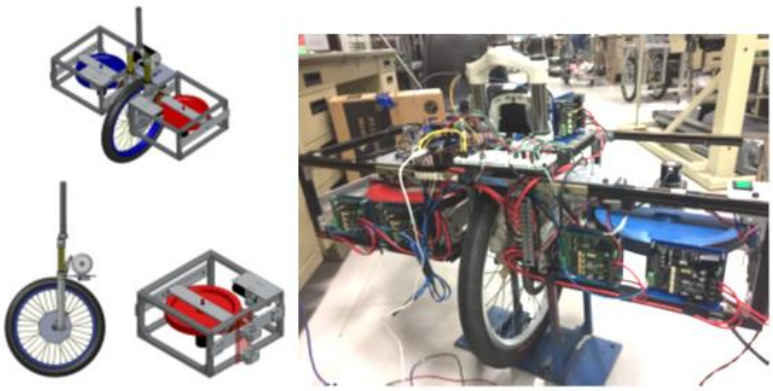
In this project, a flywheel robot achieves self-balancing on the lateral axis through synchronous spinning of flywheels positioned on either side, while balance on the longitudinal axis is maintained using the inverse pendulum effect. While balancing lateral and longitudinal motions concurrently prevent the robot from falling, this project focuses on demonstrating stability in each axis separately.

In this setup, the PID controller is implemented for both lateral and longitudinal motions. However, the design is mechanically intensive and complex, and the author acknowledges that the correct use of electronics was not deployed, resulting in troublesome errands.

In the proposed research, the aim is to enhance the design's simplicity and compactness while ensuring accurate and high-resolution sensing. This will be achieved through the utilization of high-accuracy and resolution sensor IMUs (inertial Measurement Units) and Gyros, thereby addressing the challenges encountered in the previous project.

Figure 2.10

Two Flywheel Robot, Pathamrajah & Manukid., 2018, AIT

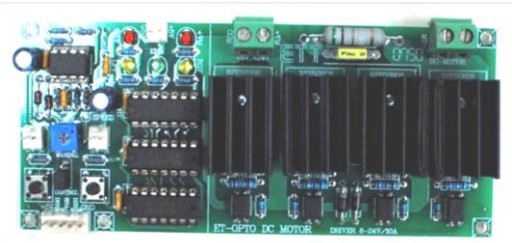


2.4 Electrical Aspects of Robot Design

The research work done by Surachat for unicycle robot balancing has utilized the following scheme for electronics.

Figure 2.11

Unicycle Robot Electronics Used ET-OPTO DC Driver (Surachat & Manukid, 2011)



In the current setup, the board has limited applications for the DC motor, with a maximum draw of 10 amps. However, the proposed research introduces the use of

STM32 controller, which offers enhanced flexibility and robustness in controlling brushless DC motors.

Figure 2.12

Signal Conditioning Circuit Developed by Abhisesh 2012, AIT

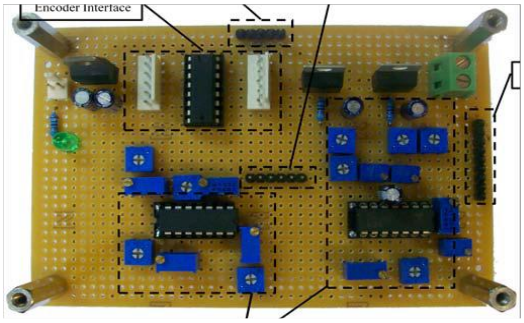


Figure 2.13

Electrical Circuit Used in Pathamrajah, Thesis, 2018, AIT

The proposed electronics for this research will be similar to the board used by Pathamrajah, 2018 in his research work as shown. in the previous research conducted by Abhisesh, a custom signal conditioning board and Gyros interface were designed, requiring significant time and resources to implement. in contrast, the proposed research will utilize off-the-shelf components to develop the robot, with a primary focus on crucial aspects such as balancing and control. This approach streamlines the development process and reduces the resource-intensive nature of electronics implementation.

2.5 Related Work on Other Robots in Literature

2.5.1 Ballbot Robot

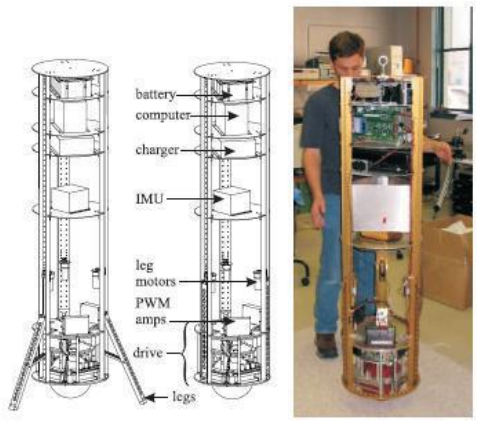
The first ballot was pioneered at Carnegie Mellon University in year 2006 by T.B. Lauwers et al., Resembling an inverted mouse ball, its drive mechanism was designed to mimic this structure. Standing at the height of an average human, it represented a significant development in robotic design.

In 2008, another robot model was proposed at Tohoku Gakuin University, spearheaded by Professor Masaaki. This iteration weighed approximately 7.5 kg and stood at around 0.5 meters in height, making it more compact and smaller than its predecessor. Unlike the first ballot, this robot employed a drive mechanism with wheels positioned 120 degrees apart, minimizing the number of wheels required for locomotion. However, the unique orientation of its wheel position posed challenges in defining its movement and functionality.

Despite these differences, both robots adopted similar concepts, including the use of LQR and inner PI controllers, which resonate with the proposed Wheelbot robot's design principles.

Figure 2.14

Ballbot Developed by Carnegie Mellon University (CMU) 2006



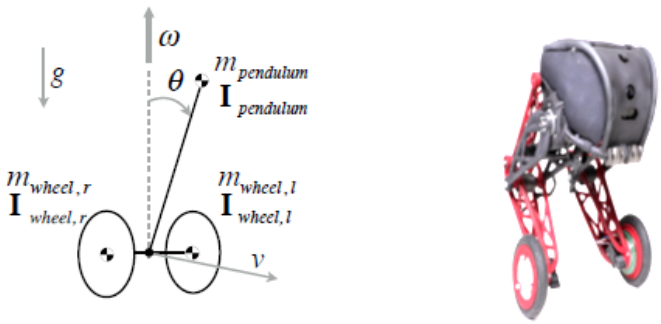
2.5.2 Bipedal Robots – Acento and Four Bar Mechanism

The concept of a compact wheeled bipedal robot capable of swift movement on flat terrain and adeptly overcoming obstacles through jumping was introduced by Victor Klemm et al. in 2019. Their innovation showcased a parallel elastic jumping mechanism constructed from topology-optimized components. This mechanism exhibited remarkable capabilities in balancing and executing locomotion patterns.

Inspired by this pioneering work, the proposed thesis aims to harness the potential of robotics in real-world scenarios. Drawing upon the advancements demonstrated by Klemm et al.'s robot, the thesis seeks to explore and expand upon the applications of robotic technology, to improve agility in complex environments.

Figure 2.5

Ascento Robot by ETH Zurich, Autonomous System Lab, 2019



An innovative design introduces a leg on wheels robot. Extensive analysis was conducted on the kinematics and dynamics of this robot, incorporating an LQR controller for balance and a fuzzy logic-based PD controller for jumping tasks.

In the analysis, torque curves and hip joint angles were obtained through Simulink experiments. This comprehensive investigation sheds light on the intricate mechanics and control strategies necessary for achieving stable locomotion and dynamic jumping capabilities in the bipedal wheel-legged robot.

Figure 2.16

Jumping Wheel-Legged Robot Developed

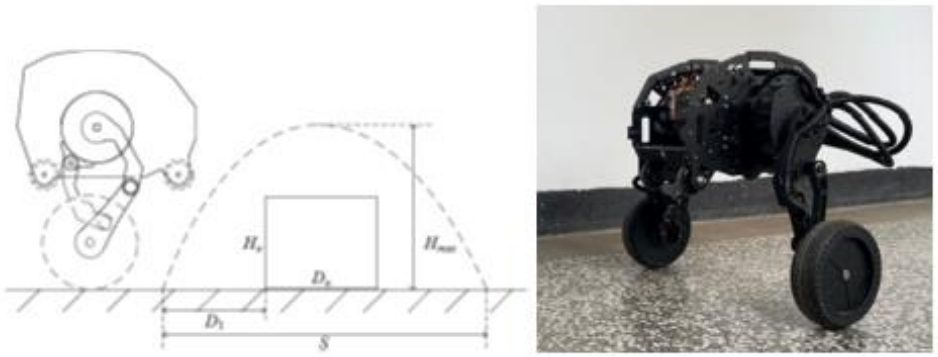
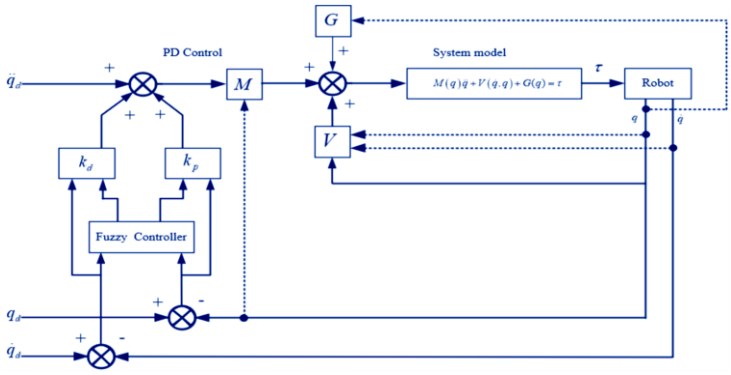


Figure 2.17

Fuzzy Proportion Differentiation Control



2.5.3 Cubli Robots – 3D inverted Pendulum

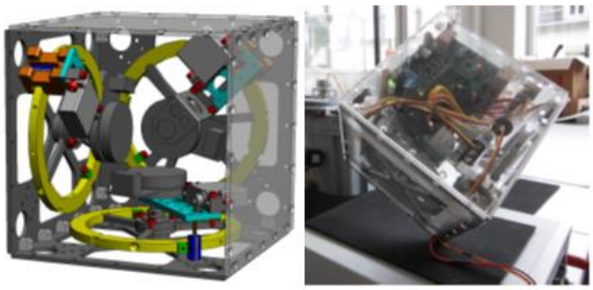
A significant portion of this research work draws Inspiration from the Cubli, a 3D inverted pendulum based on reaction wheels. The Cubli, measuring 15x15x15 cm, is equipped with three momentum-applying reaction wheels. The Cubli demonstrates the ability to balance on its edges.

The nonlinear system governing Cubli's dynamics is computed using a frequency-based approach. Additionally, the Cubli showcases corner-balancing capabilities with a classical controller, and further exploration involves corner-balancing using a linear feedback controller.

Moreover, this research aims to leverage the jumping mechanism exhibited by the Cubli, which can jump from a resting position without external support. This mechanism offers valuable insights into dynamic motion and control strategies that can be applied to enhance the capabilities of bipedal wheel-legged robots.

Figure 2.18

CAD Model and Cubli Robot (ETH Zurich)

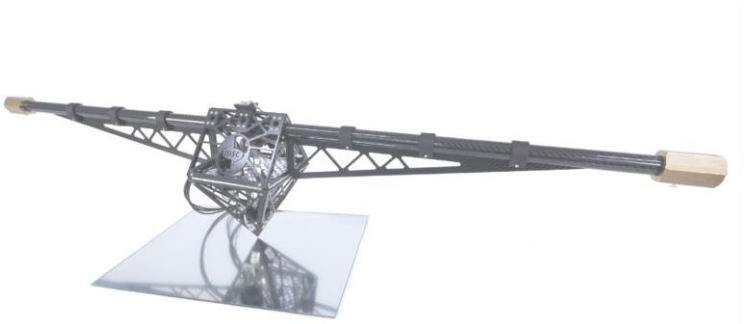


In a recent study conducted by Mathias Hofer et al. at ETH Zurich in 2023, the development of the one-Wheel Cubli was introduced. This innovative design features just one wheel for balancing on the edges. This achievement is made possible through meticulous consideration of the inertia of mass, ensuring a significant disparity between the inertia along the tilt axis of the system.

To maximize controllability, two inertia values ratio is optimized, and the study discusses sensor placement strategies aimed at minimizing variance in tilt estimates, contributing to innovative design. Notably, the lightweight design of the one-Wheel Cubli introduces challenges related to cantilever deflections, necessitating careful modeling and compensation. Figure 2.19 showcases the unique design of this robot, representing a significant advancement in the field of robotics.

Figure 2.19

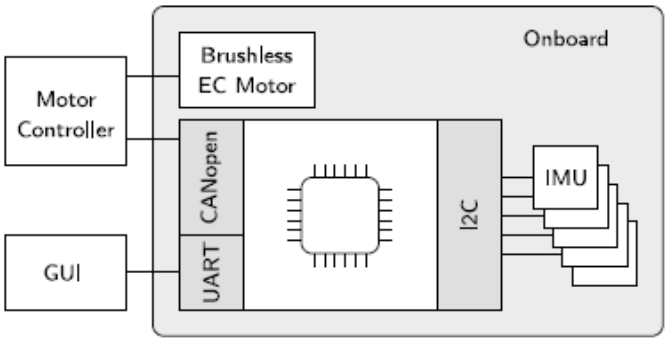
One Wheel Cubli (ETH Zurich, 2023)



The dynamic model of this one-wheel Cubli is amazing in its nature. Thus, this work expects to leverage this part and incorporate the electrical and electronic components accordingly as shown in Figure 2.20.

Figure 2.20

One Wheel Cubli Setup for Electronics (ETH Zurich, 2023)



The microcontroller, IMU, and actuator are the integral components of the system, all playing crucial roles. However, it's important to note that the motor driver is located offboard. for visualization, a laptop computer hosts a graphical user interface (GUI). Communication between the microcontroller and the motor is done using the CAN open protocol, while the IMUs communicate via the I2C protocol. the IMU and motor data are transmitted to the GUI via UART serial communication, allowing for real-time monitoring and control of the system's performance.

2.6 Intelligent Robust Control of Unicycle Robots: Artificial intelligence

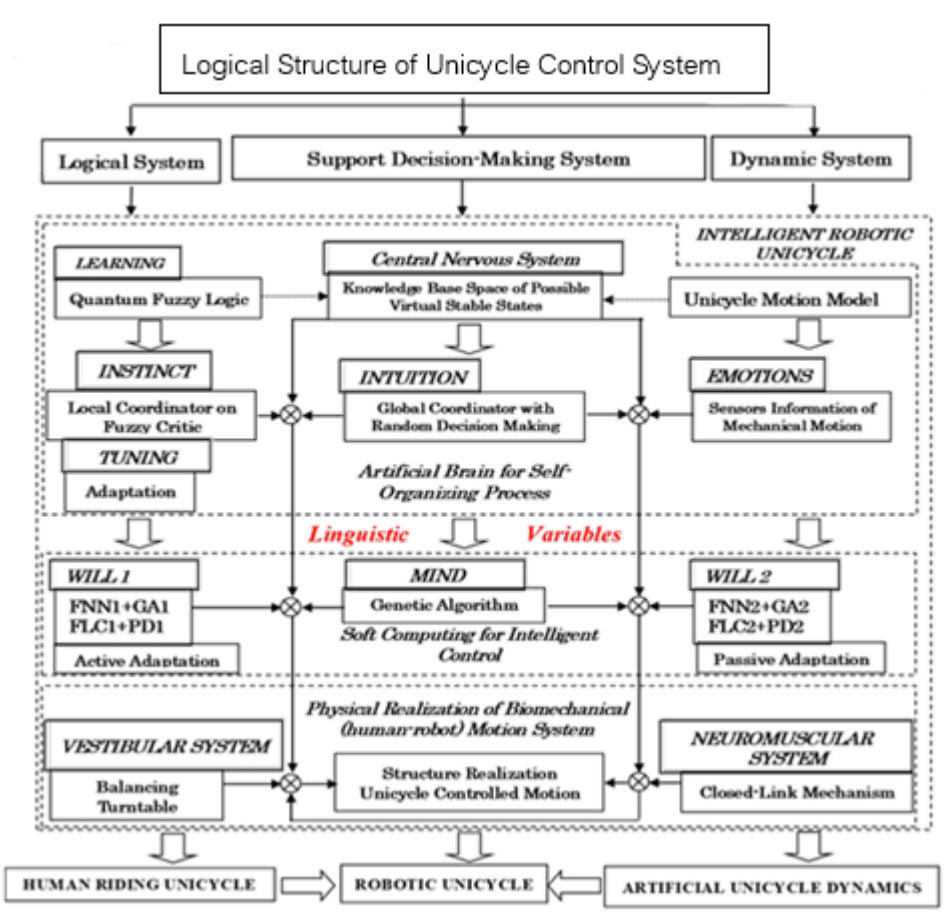
2.6.1 End-to-end Control Using Soft Computing Technology with Biomedical Control

The research conducted by Sergey and Viktor (2020) delves into the utilization of soft computing techniques, specifically fuzzy logic control, for achieving robust control in unicycles. this study focused on exposing the characteristics of unicycles, which are non-holonomic, inherently nonlinear, globally spatially unstable, and highly constrained in terms of linkage.

The proposed model, illustrated in Figure 2.21, addresses the complex dynamics and constraints inherent in robotic unicycles, paving the way for more effective and robust control strategies leveraging fuzzy logic principles.

Figure 2.21

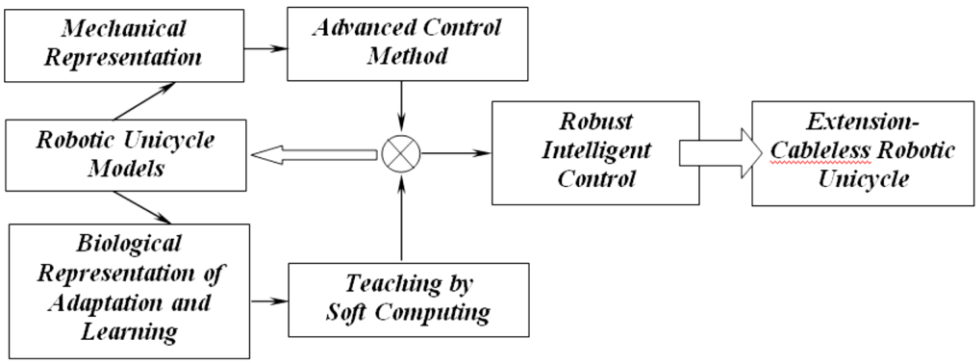
The Logical Structure of Distributed Knowledge Representation (Dubana State University Russia, 2020)



The author has also presented the conceptual scheme for unicycle robot control about human-in-the-loop. The proposed research aims to leverage a biomedical system for controlling the unicycle robot.

Figure 2.22

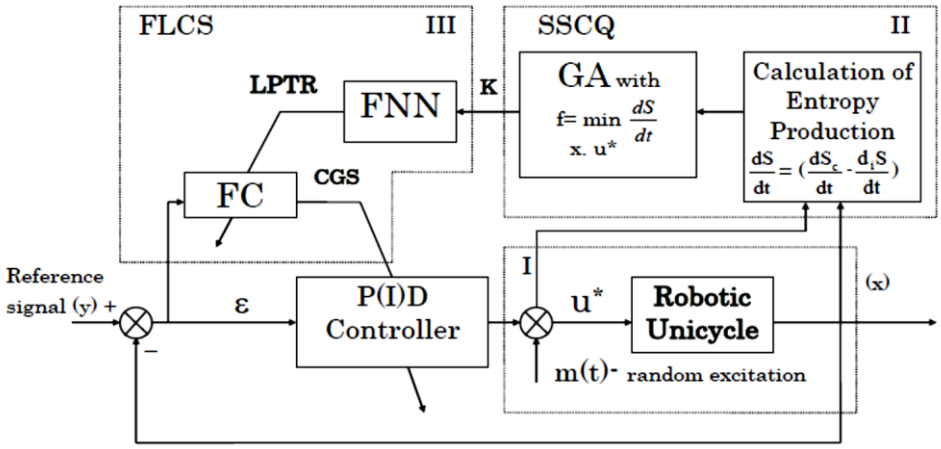
The Conceptual Scheme of the Robotic Unicycle R & D



The heuristic AI techniques have been further explored.

Figure 2.23

Robust Control System Design on AI Structure



2.6.2 Integrating RL and CL for Nonlinear System Management

Imperial College London has investigated thoroughly the optimal control schemes and reinforcement learning for non-linear dynamics systems like cart poles, inverted pendulums, and unicycle robots. the comparison table for RL and OC is given below:

Table 2.1

Reinforcement Learning and Optimal Control Comparison (Doctoral Dissertation, Imperial College London)

Reinforcement Learning	Optimal Control
Machine Learning	Control Engineering
Markov Property and Bellman	Bellman Optimality Principle
Model – Free	Model- Based
Learns from Reward	Require Full Knowledge of Dynamics
Discrete States and Actions	Analytical Solution for Linear Systems
Linear and nonlinear Systems	Suitable for Linear Systems Only

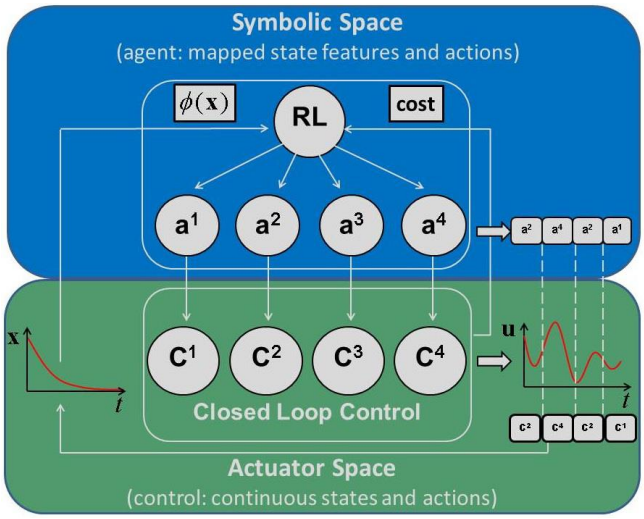
2.6.3 Mapping Optimal Control Problems into Reinforcement Learning (MDP) Framework

The conceptual diagram in Figure 2.24 explains the integration of symbolic AI with closed-loop control in robotics from a third-person perspective. It illustrates how state features and actions, which are mapped in symbolic space, can be translated into continuous states and actions within the actuator space. This integration is crucial for the development of advanced robotic systems capable of executing complex tasks with high precision. The diagram is expected to include components such as Reinforcement Learning (RL), cost calculation, and the flow of information and control signals between the symbolic and actuator spaces.

A cornerstone in the realm of decision-making frameworks, the Markov Decision Process (MDP) embodies a versatile structure for encapsulating the dynamics of stochastic environments and the decision-maker's strategic maneuvers. Defined by its components—states, actions, transition probabilities, and rewards—the MDP framework orchestrates the interplay between random events and deliberate choices. Agents, governed by policies dictating action selection in states, strive to optimize long-term rewards through judicious decision sequences. Leveraging the Markov property's essence, which asserts that future outcomes hinge solely on the present state and action, MDPs navigate intricate decision landscapes with computational efficiency.

Figure 2.24

RLOC Architecture by Ekaterina, Department of Computing, Imperial College London, 2015



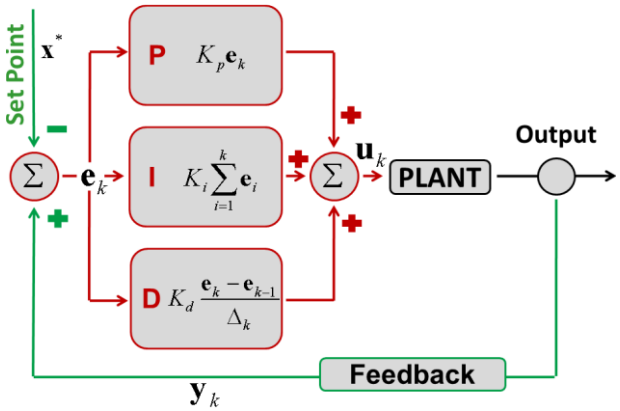
2.6.4 Top-Down Classical vs RL Design Hierarchically

The RL operates within a hierarchical structure, featuring, such as LQRs and PIDs. This hierarchical setup allows the algorithm to function effectively in the Symbolic Space and the Actuator Space. The Symbolic Space encompasses discrete actions and states of the symbolic level Markov Decision Process (MDP), while the Actuator Space includes continuous states and controls modeled over multiple time steps per episode.

In adapting the top-down hierarchical approach of the RLOC algorithm for controlling the unicycle robot, a reinforcement learning agent operates at the high-level, abstract tier. interacting iteratively with its environment formulated as an MDP, the agent selects actions and observes resulting states and rewards, aiming to maximize the reward for the agent. in RL algorithms, Monte Carlo emerges as a natural choice for RLOC due to its capacity to handle semi-Markovian environments and its alignment with the episodic nature of tasks like pole balancing and arm reaching. Moreover, Monte Carlo's non-bootstrapping approach retains all information about visited states without approximation, making it particularly suitable for episodic tasks compared to SARSA or Q-learning. At the low-level, the selection of closed-loop controllers is flexible, with the algorithm prioritizing controllers yielding the highest long-term reward, including LQR, PID, and H∞ controllers.

Figure 2.25

PID Controller in Discrete Time Setting



2.6.5 The LQR Controller

The LQR stands out as a famous controller renowned for offering a solution for linear systems. Its key allure lies in its offline calculability, making it an appealing low-level controller. Notably, the LQR can be intelligently integrated into the domain of RL where applicable as linear adaptive control.

Table 2.2

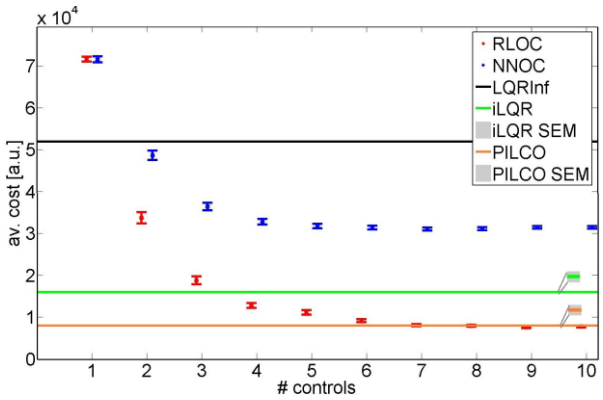
Simulation Parameters - RLOC, iLQR Benchmark (ICL, 2015)

Parameter	Sym.	Robotic Arm	Cart Pole	SI Unit
RLOC num symbolic state	n,	36	49	[a.u.]
RLOC num LQR controllers	na	5	10	[a.u.]
iLQR initial control	ilqr	{-1,0,1}	{-1,0,1}	[N ml. [N]

The best comparison is presented in the graph below where several controllers have been implemented to compare the reinforcement learning optimal control-based performance.

Figure 2.26

RLOC Performance vs Cart Pole System (Number of Controllers Implemented)

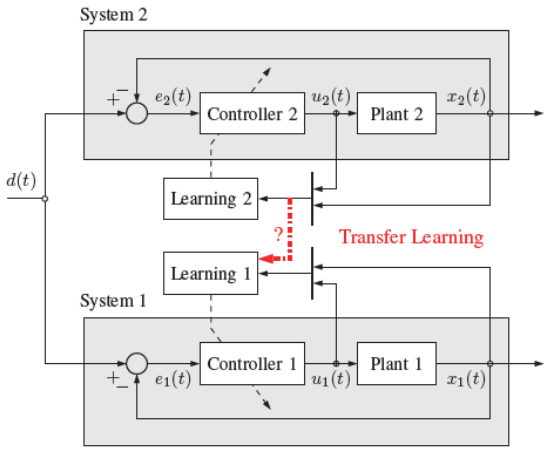


2.6.6 Machine Learning on Unicycle Robots

University of Toronto, Canada has explored the transfer learning between unicycle robots to improve the performance of control systems in cases when accurate models of the system or the environment are not given. It has shown two non-linear, unicycle robots and derived analytical error bounds for the linearized robot models and experiments with the Pioneer 3-AT robot.

Figure 2.27

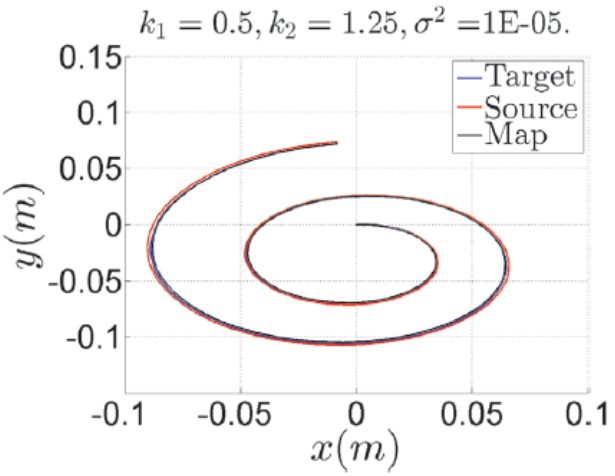
Transfer Learning Framework for Unicycle Robots (University of Toronto, Canada, 2016)



The target comparison can be shown in figure 29 where the extracted model for an unknown system has been implemented on another unicycle robot. The two unicycle robots move in an outward spiral. The TL objective aligns the red trajectory to the blue.

Figure 2.28

RL Framework for Unicycle Robots (University of Toronto, Canada, 2016)



This research shows the quality of alignment-based transfer learning using machine learning concepts for linearized unicycle robot models with proportional feedback control. Two linearized unicycle models, each with a different controller gain, follow the spiral trajectory which shows the advantage of machine learning controller implementation. However, this research aims to leverage the reinforcement learning optimal control as a model-free algorithm design with a low-level controller called LQR and will compare the performance over classical control methods.

CHAPTER 3
METHODOLOGY

In the development of the unicycle robot, the paramount objective is to engineer a design that ensures optimal stability from a mechanical standpoint. However, the reality of real-life scenarios often entails challenges in achieving perfection in design. Therefore, significant emphasis has been placed on refining the design and ensuring its reproducibility, particularly for laboratory applications.

The methodology employed in this research unfolds across three distinct tracks: real robot design, simulation work for the unicycle, and stand-up dynamics concept. The simulation track is further delineated into MATLAB simulation for classical control and PyBullet simulation for reinforcement learning. The primary aim of the simulation work is to showcase the efficacy of control approaches for achieving balance, both in MATLAB and Pybullet environments. These simulation findings are then integrated with real-robot design considerations to validate the feasibility of algorithms and establish a testing platform for the unicycle robot. Additionally, stand-up dynamics experiments are conducted to elucidate the concept of jump-up maneuvers akin to the renowned robot Cubli.

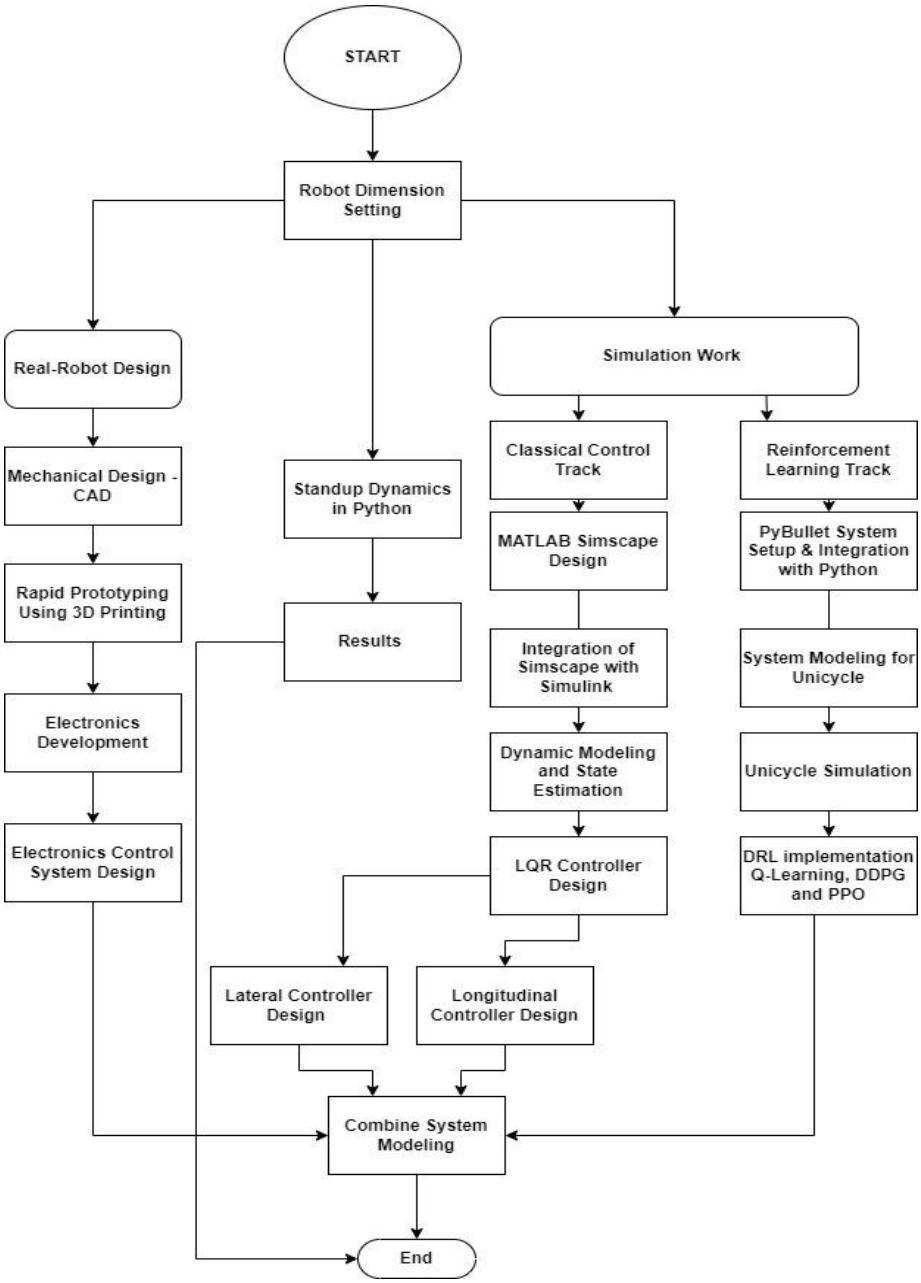
In essence, the amalgamation of these methodologies is meticulously coupled, rigorously tested, and systematically presented within the scope of this thesis to comprehensively address the challenges associated with unicycle robot design and control.

3.1 The Flow Chart of this Methodology

The methodology framework flow for this work is presented below.

Figure 3.1

Comprehensive Workflow of Methodology for Research Work



3.2 Dimension Settings for Unicycle Robot

The dimension parameters of the robot have been meticulously selected to facilitate the development of both simulation and real-robot models. The envisioned robot is designed to be compact and under-actuated, featuring two wheels in its configuration. Specifically, one of the wheels serves as the reaction wheel, while the other functions

as the rolling wheel that maintains contact with the ground. This strategic design allows for optimal balance and maneuverability, aligning with the objectives of the project.

3.2.1 Height of Robot for Wheel Inertia

The height of the robot holds significant importance in ensuring its maneuverability within laboratory settings or on a testbench. This consideration is approached from a technical standpoint, considering factors such as wheel size and its effect on inertia, as well as the torque changes induced by variations in angular velocity (w).

To maximize inertia and minimize the impact on motor torque caused by changes in angular velocity, careful consideration is given to the size and positioning of the wheels relative to the robot's rotation axis. The orthogonal configuration of the robot serves as a reference, highlighting the need for the wheel diameter to be substantial enough to maximize inertia while also being sufficiently distant from the rotation axis.

Through thorough analysis, the optimum height of the robot is determined to be 120mm, with a corresponding wheel diameter of D = 95mm. This configuration strikes a balance between inertia optimization and practical considerations, ensuring effective maneuverability and control of the robot within the intended operational environment.

3.2.2 Geometry of Robot Body and Chassis

For demonstration purposes of stand-up dynamics and real-robot design/MATLAB, two geometries were designed. Considering only stand-up dynamics, the symmetrical configuration of the robot is desired by setting exact parameters of Lengths L1, and L2 around 62mm and the robot’s width at 85mm. on the other hand, for development purposes, the robot structure is chosen as a one-frame body accommodating both actuator motors for ground and reaction wheel where the chassis full height is 95mm and width 57mm providing optimum adjustments for chosen electronics and hardware.

3.3 Modeling of Jump-up Dynamics for Demonstration

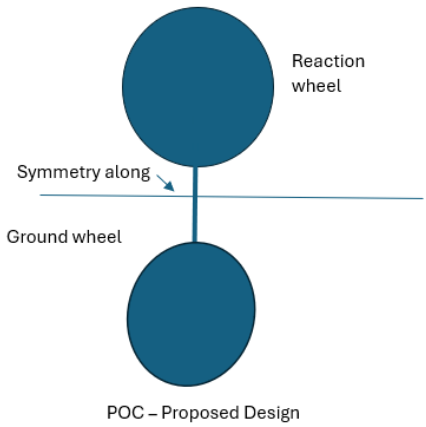
To demonstrate the proof-of-concept (POC), the dynamics of jumping up were developed to illustrate the unicycle robot's ability to self-balance from an initial falling position. this idea draws Inspiration from the renowned robot CUBLI, leveraging its jumping mechanism to autonomously reorient itself from various initial orientations, ultimately stabilizing on its edges.

It's imperative to note that for jump-up dynamics to be effective, the unicycle's design must exhibit symmetry. Consequently, the dimensions of the ground and reaction wheels are identical on both sides of the robot. When the robot experiences a fall, only one wheel is required to contact the ground to generate sufficient acceleration, while the other wheels remain elevated. To facilitate this, the robot's structure is encapsulated within a cube-shaped bracket, ensuring uniformity and stability.

The conceptual modeling of this approach is depicted below, showcasing the strategic arrangement and design considerations implemented to achieve effective jump-up dynamics.

Figure 3.2

Stand-up or Jump-up Dynamic Symmetric Concept Model for POC only



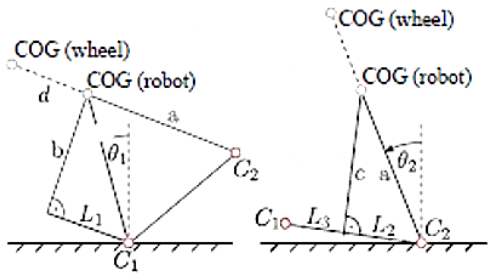
From a technical standpoint, the robot is engineered to function akin to a CUBLI robot, employing a mechanism where one wheel is accelerated until the moment the other wheel, typically the rolling wheel, makes successful contact with the ground. This process is analogous to the action of the CUBLI robot jumping to its edges to achieve balance. in the case of unicycle robots, maintaining an upright equilibrium position is paramount.

To model the robot's behavior, this setup can be conceptualized as a 2D inverse pendulum with the reaction wheel, as illustrated in Figure 3.3. This representation

shows the dynamics of the robot's motion and the design of control strategies aimed at achieving stable equilibrium during the jump-up process. by simulating this configuration, insights can be gained into the effectiveness of the proposed mechanism in facilitating self-balancing and stability of the unicycle robot.

Figure 3.3

Unicycle as 2D inverse Reaction Wheel Pendulum



The assumption was made to show that the robot should be self-positioned by first accelerating around point C_1 and then stand-up at point C_2.

the modeling resorts to Ganzalo and Nahuel's work on a reaction wheel-balanced inverted pendulum. The dynamics of jump-up require the maximum torque of the motors. Starting with the equation of motion of the pendulum as follows:

$$I\ddot{\theta} = kmgL \cdot \sin\theta - b \cdot \dot{\theta} - \tau_c \qquad 1$$

Where motion equation for the reaction wheel can be:

$$I\dot{\omega} = Tc \qquad 2$$

the relative velocity relation between the pendulum arm and reaction wheel is given by:

$$\omega_I = \dot{\vartheta} + \omega_r \qquad 3$$

Where WI represents the inertial properties and Wr denotes the reaction wheel. While providing a detailed description of the inverse pendulum model is beyond the scope of this section, interested readers are encouraged to refer to (Ganzalo & Nuhuel, 2018) for comprehensive insights.

Equating the equations further in terms of the state space model, we can express:

$$\frac{d}{dt}\begin{bmatrix} dot(\theta) \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{Qg(\theta i) - Qw(w)}{Itotal} \\ \frac{Qw(w)}{Iw} \end{bmatrix} \quad 4$$

Where the w is wheel rate, Qg is gravitational torque, Qw is the torque of motors and rotational axis inertia I_w, and total rotational inertia is I_{total} as derived concerning the contact point C. As per figure 3.3, the max torque required for the motor to do the jump-up action is equal to

$$T3 = L3 * mtotal * g \quad 5$$

Before deriving the mathematical modeling, it is important to set the layout for the proposed mechanical design of a robot.

3.4 Mechanical and Electrical Design with 3D Prototyping

The CAD model of the robot is meticulously crafted using Solidworks, a sophisticated CAD design software renowned for its interactive features and versatility. To materialize the design, CURA 3D slicing software is employed for slicing and preparing the parts, utilizing ABS material for robustness and durability.

Two distinct designs of the unicycle robot were initially proposed, each undergoing iterative improvements to refine the final vision. The design of the unicycle encompasses various components, with the majority falling into two primary categories: chassis design and reaction wheel design. these components are intricately integrated to ensure optimal functionality and performance of the robot.AD model of this robot is designed in Solidworks, a professional CAD design software that offers

interactive features and CURA 3D slicing software is used to print the parts with ABS material. Two designs of the unicycle robot were proposed and further improved in the vision. The design of a unicycle consists of various parts, but the majority can be divided into two categories. Chassis design and reaction wheel design.

3.4.1 Initial Design and 3D Prototype.

The initial design of the unicycle was designed and assembled using various parts and the full design is presented in Fig below.

Figure 3.4

Initial 3D CAD of Unicycle Robot with Description of Electronics

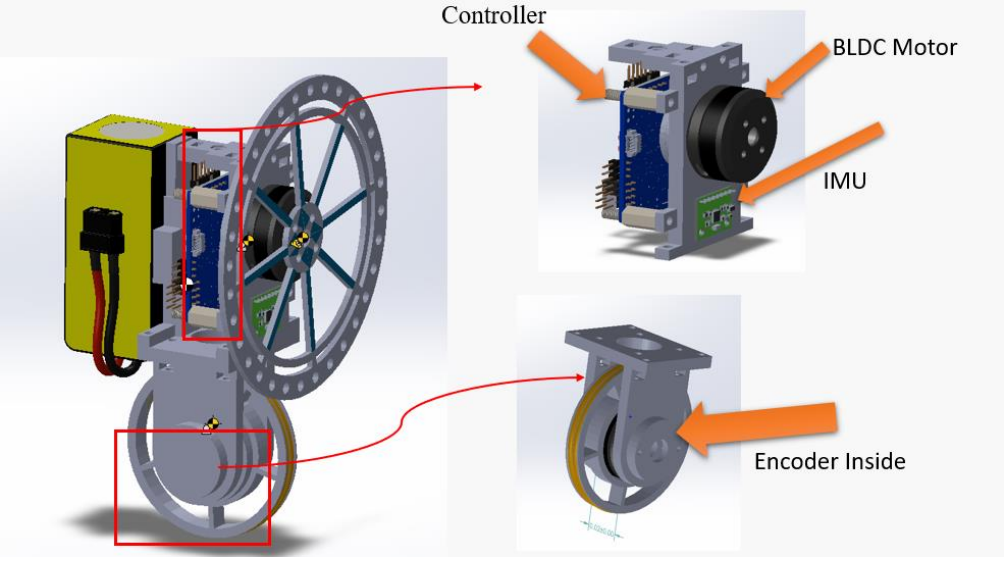


Figure 3.5

3D Printed Parts for Unicycle Robot Captured on 3D Printed Bed.

All the CAD designs of the unicycle robot were 3D printed using PLA material. Special care is recommended to print the 3D models. for this thesis, the CAD model is designed in such a way that it is 3D printable. All the settings for 3D printing are done using the fine option, with a shell thickness of 1.15 of nozzle size of 0.4mm.

Figure 3.6

Assembled Robot with Electronics (Front and Side View)

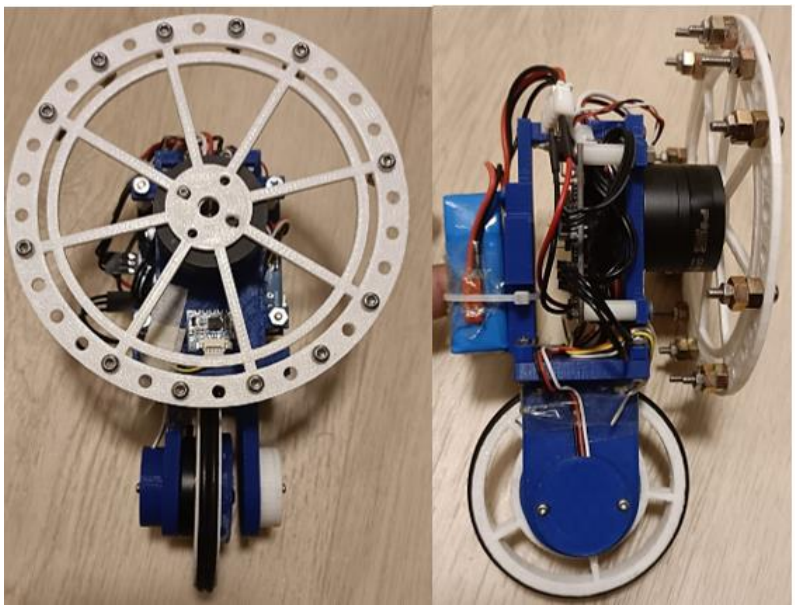
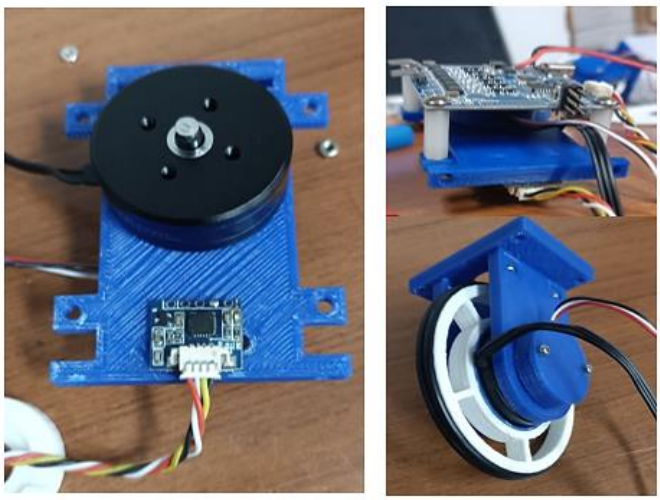


Figure 3.7

Assembled Robot with Electronics.



The Bill of materials for the first version of the robot is as follows:

Table 3.1

Bill of Materials for Unicycle

Item. No	Name	Purpose
1	GM2804H-100T Brushless Motors	Actuator for Pitch & Roll
2	AS5600 Magnetic Encoder	Precise control of motors
3	HC-05 Wireless Module	IoT control
4	BGC 3.1 control Board	Control of hardware
5	MPU-5050 IMU	State estimation
6	M1.6 M2 M2.5 Screws set	Reaction wheel, assembly
7	HJ 3S 11.1V 500mah LiPo battery	Power source
8	Rubber O-ring	Ground wheel damping
9	10/10 M-to-M, M-to-F wires	Connections
10	Bootloader USB	Firmware update

3.4.2 Challenges in Initial Design and Improvement

During the control of the robot, an issue arose when the BLDC (Brushless Direct Current) motor started to heat up excessively, leading to buckling of the PLA (Polylactic Acid) material. To address this challenge, a strategic decision was made to replace the BLDC 2480-100T 2.8-Watt motors with Nidec-24H404H160 motors equipped with built-in encoders. These motors boast higher power output, with 11 watts and a stall torque of 70 mN.m, ensuring improved performance and reliability in terms of robot control. The upgraded motors successfully mitigated the heating issues and provided optimal functionality during operation.

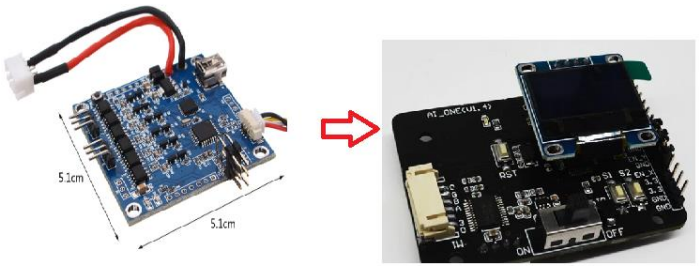
Figure 3.8

Actuator Improvements in the Second Version of Unicycle

Furthermore, to enhance the control system of the robot, the Arduino-based gimbal controller BGC 3.1 was substituted with the STM 32F103 by STM Electronics. This strategic replacement was made to overcome the limitations posed by the Arduino software, which restricted the utilization of advanced programming frameworks. The new STM 32F103 board offered an advanced version of the ARM KEIL software specifically tailored for embedded system applications. Moreover, it featured a complementary LCD screen that displayed crucial parameters during the balancing process, providing real-time feedback for improved control and monitoring of the robot's operations.

Figure 3.9

Electronics Controller Board Improvements in Unicycle Robot



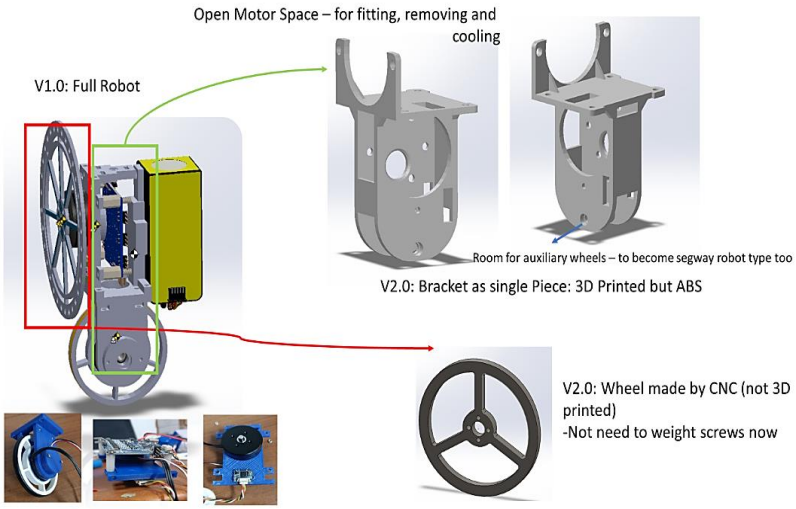
In response to the challenges encountered during the assembly of the robot's initial version, significant improvements were implemented in the subsequent iteration. one of the key enhancements involved redesigning the chassis frame to create a single unified structure, streamlining the assembly process, and minimizing the efforts

required. Additionally, this unified frame was fabricated using ABS material, chosen for its superior heat resistance properties compared to PLA material, ensuring enhanced durability and performance.

Furthermore, to optimize the design and functionality of the reaction wheel assembly, a decision was made to utilize CNC manufacturing. This approach not only reduced the reliance on additional screws but also helped in minimizing the overall weight of the assembly compared to the first version. By integrating these improvements, the assembly process became more efficient, and the resulting robot exhibited improved robustness and reliability.

Figure 3.10

Design Improvements in Unicycle.

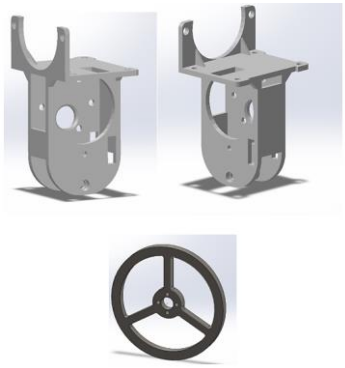


3.5 Second CAD Design and 3D Prototype

in the updated version of the robot, significant efforts were made to simplify the manufacturing process and enhance the structural integrity. one notable improvement was the reduction in the number of 3D printed parts, achieved by redesigning the frame to form a single cohesive body. This approach not only streamlined the assembly process but also contributed to a more robust and durable overall structure.

Figure 3.11

Improved Version 2 Design with one Bracket and CNC Manufactured Reaction Wheel.



Furthermore, a decision was made to manufacture the reaction wheel using CNC turning machinery in the workshop, as depicted in the accompanying figure. this machining technique offered several advantages over 3D printing, including superior precision and material strength. By adopting these manufacturing enhancements, the new version of the robot boasted improved reliability and performance with a reduced number of 3D printed parts as the frame was designed as one part of the body. Moreover, the reaction wheel was manufactured using a CNC turning machine at the workshop instead of 3D printed as shown in Figure 3.12 below.

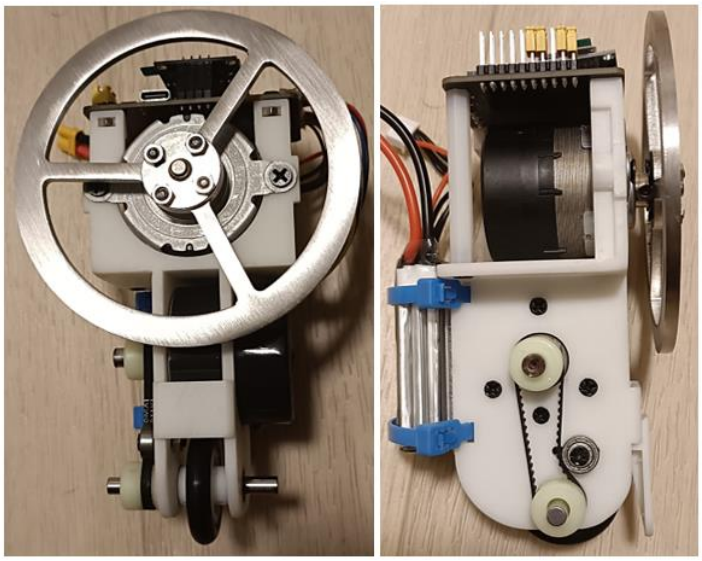
Figure 3.12

CNC Manufacturing of Reaction Wheel (Left) and Final version (Right)

To accommodate the required motor size and ground wheel dimensions, a novel mechanism was devised utilizing a belt-and-pulley system to drive the wheel from the actuator, as illustrated in Figure 3.9. This innovative design allowed for efficient power transmission while maintaining a compact footprint, ensuring optimal performance of the robot.

Figure 3.13

Unicycle Robot Assembled Front and Side View

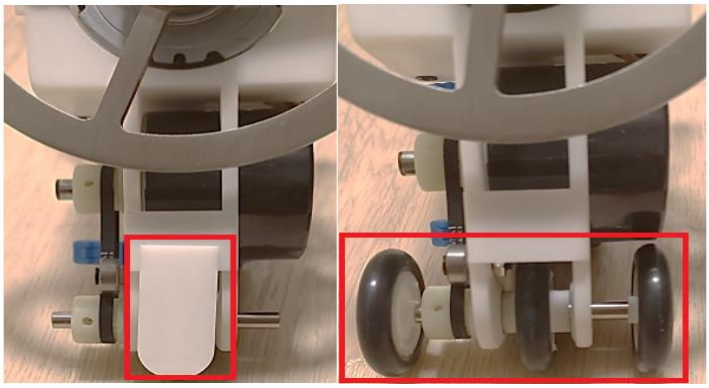


To ensure effective balancing functionality, specific measures were implemented to control pitch and roll movements independently. for pitch control, a small bracket was meticulously designed to prevent the robot from moving forward and backward, enabling precise control over its pitch angle. Additionally, supporting wheels were incorporated to facilitate roll control, thereby preventing lateral movement of the robot while the roll angle was being independently controlled.

These design enhancements significantly contributed to the robot's stability and maneuverability during balancing operations necessary for a robot to perform balancing functions effectively therefore for pitch control a small bracket was designed to prevent the robot from moving forward and backward so that it can be controlled independently. Additionally supporting wheels were also designed for roll control to prevent robots from moving left and right during the independent control of roll angle.

Figure 3.14

Unicycle with Supporting Bracket (Pitch Control) and Supporting Wheel (Roll Control).



In the electronics setup of the unicycle robot, an STM32 board was utilized in conjunction with an LCD. this choice of board conferred numerous advantages over conventional Arduino boards, including enhanced processing power and a wider range of functionalities. Additionally, a LiPo battery was selected to power the robot, providing extended operating time compared to traditional battery options. these electronic components were carefully chosen to optimize the performance and efficiency of the unicycle robot's control system.

Figure 3.15

Electronics of Proposed Unicycle STM32 Electronics Board (Left, Red Bracket) & LCD Display (Green Bracket) and Battery (Right, Blue Bracket) for Unicycle

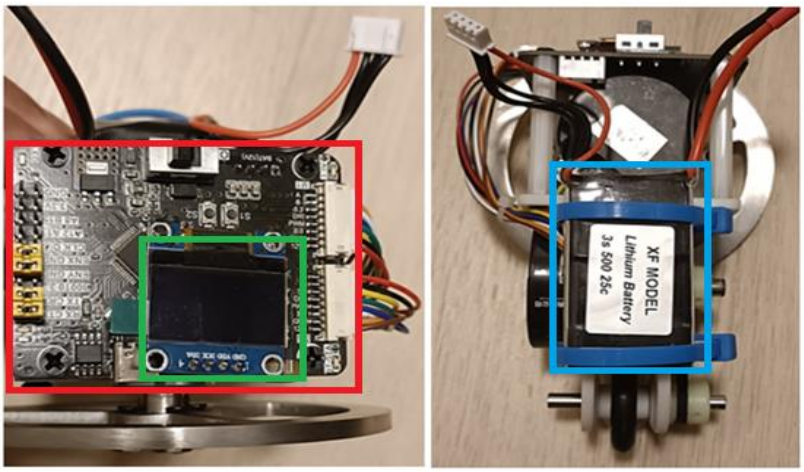


Figure 3.16

LiPo Battery Charger.

Table 3.2

Bill of Materials for Improved Version

Item. No	*Name*	*Purpose*
1	Nidec-24H Brushless Motors	Actuator for Pitch & Roll
2	integrated Wireless Control Module	IoT control
3	STM32 Board with LCD	Control of hardware
4	onboard- IMU Unit	State estimation
5	M1.6 M2 M2.5 Screws set	Assembly
6	3S 500mah LiPo Battery	Power source
7	Small Wheel x 3 units	Ground wheels
8	10/10 M-to-M, M-to-F wires	Connections
9	Belt and Pulley	Ground wheel drive
10	1x 3D Print Frame ABS Material	Hosting hardware

3.6 Unicycle Robot Dynamic Modeling

In deriving the system model for the unicycle robot, various methods are available for dynamic modeling, including the Newton method and the Euler-Lagrange method. for this research, the Euler-Lagrange method was selected due to its advantages over the Newton method. Unlike the Newton method, the Euler-Lagrange method offers a more robust modeling process that incorporates energy. Additionally, the Euler-Lagrange

method removes the need for redundant forces and torques, reducing the likelihood of errors in the model. These factors make it a preferred choice for accurately capturing the dynamics of the unicycle robot.

3.6.1 Assumptions for Modeling

The following assumptions were made during the modeling process:

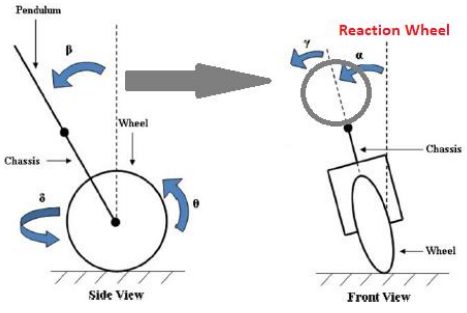
1. The Unicycle robot is considered a three-link robot, wheel, reaction wheel, and chassis of the robot.
2. The unicycle is a rigid body.
3. The unicycle robot is symmetrical in sagittal and coronal planes.
4. The unicycle robot stays in contact with the flat ground surfaces.
5. The slippage between the wheel and the ground surface is zero.

3.6.2 Defined Coordinate System for Robot

The dynamic equations for the proposed unicycle robot can be derived analogously to the pendulum-based unicycle, as discussed in detail by Yohanes Daud and Abdullah (2015). Hence, similar configurations of the unicycle robot were considered, with the lateral pendulum replaced by a reaction wheel, as illustrated in the figure below.

Figure 3.17

Concept Modeling for Reaction-Wheel Unicycle.



the coordinate frame system of this robot is mentioned in Fig 3.12 and is defined as:

- x, y, contact points between the rolling wheel and surface.
- q_4, q_5 : ground wheel angle, and reaction wheel angle respectively.
- q_1, q_2 , q_3: roll, pitch angle, and yaw respectively.

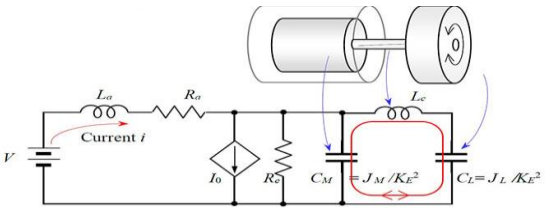
- $Tw\ and\ Tp$ are the torques input by the ground wheel motor and reaction wheel motor respectively.

3.6.3 Dynamic Model of an Actuator

To comprehend the behavior of the chosen Nidec-24H actuators, the dynamic model of the motor is derived to understand startup, acceleration, and vibration phenomena. This is accomplished by employing the concept of kinetic energy along with voltage-current relations for the mechanical and electrical circuits, respectively, as depicted in Fig 3.18 below.

Figure 3.18

Electrical Circuit of Motor Control



The control of motor and angular speed (w) and rotor angle (Θ) are represented in one equation below:

$$T_M T_E \frac{\partial^2 w}{dt^2} + T_M \frac{\partial w}{dt} + \omega = \frac{V}{K_E} \quad 6$$

$$T_M = CR_a = \frac{JR_a}{K_E^2} \quad 7$$

$$T_E = \frac{L_a}{R_a} \quad 8$$

in dynamic equation , $\omega = angular\ speed$, $V = voltage$,

$$La = winding\ inductance, Ra = winding\ resistance, Ke = Back - emf\ constants$$

The full control of this model is represented by the following block diagram.

Figure 3.19

Block Diagram for Control of Nidec Actuator (Courtesy of Nidec Corporation)

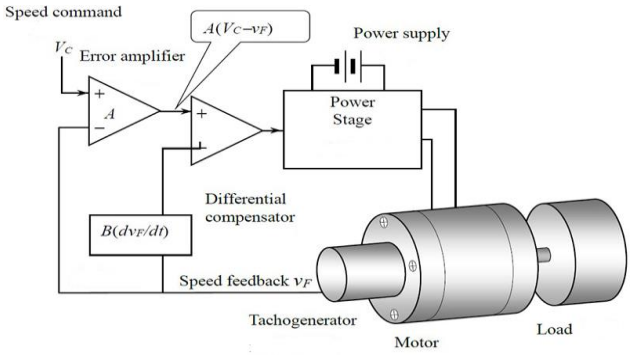
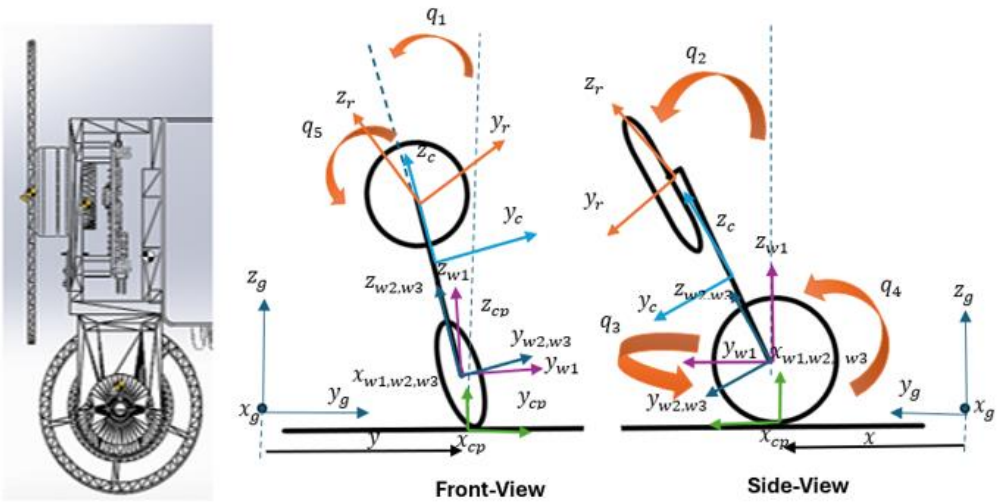


Figure 3.20

Definitions of Frames and Coordinate System for Unicycle Robot.



3.6.4 Constraints for Robot

When there is no slip the velocity of the ground wheel at the contact point concerning the coordinate system (inertial) is zero. This relationship can be shown in the equation below.

$$V(contact\ point) = \begin{bmatrix} x' - rwq4' \cos(q3) \\ y' - rwq4' \sin(q3) \\ z' \end{bmatrix} = [\,0\ 0\ 0\,]^t \qquad 9$$

Hence the motion of the unicycle robot satisfies the following constraints which are non-holonomic.

$$x' = rwq4' \cos(q3) \quad 10$$

$$x' = rwq4' \sin(q3) \quad 11$$

$$z' = 0 \quad 12$$

3.6.5 Lagrange Method for Modeling

To drive the Lagrangian we have difference of energies as follows.

$$L = K.E(total) - P.E(total) \quad 13$$

To give the proper illustration for Lagrange method system can be represented in equation 14.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial q^{.i}}\right) - \frac{\partial L}{\partial qi} = Qi + \sum_{k=1}^{n} \lambda_k a_{ki} \quad 14$$

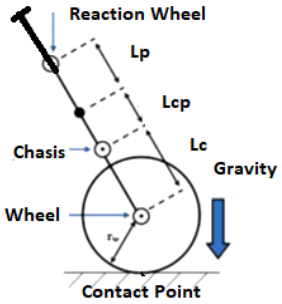
L = Lagrangian of the system a
qi = is the coordinate,
Qi = force,
m = the number of coordinates,
n = is the number of constraints,
$\lambda_k a_{ki}$ = product is a Lagrangian multiplier and kinematic constraints

For final dynamic modeling, it is required to derive the models for the wheel, chassis, and reaction wheel for the position, velocity, and angular velocity respectively.

3.6.5.1 **Position Vectors for Unicycle**. The model for the position vector can be derived Using homogenous transformation matrices with respect to the inertial frame system.
for Ground Wheel:

Figure 3.21

Ground Wheel Contact and Definition.



Using the notations defined in Figure 3.20 set of matrices are as follows:

$$ {}^{cp}_{w2}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(q1) & -\sin(q1) & 0 \\ 0 & \sin(q1) & \cos(q1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & rw \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 15 $$

This resulted in the following.

$$ {}^{cp}_{w2}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(q1) & -\sin(q1) & -rwsin(q1) \\ 0 & \sin(q1) & \cos(q1) & rwcos(q1) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 16 $$

Now from contact point to inertial frame of reference

$$ {}^{g}_{cp}T = \begin{bmatrix} \cos(q3) & -\sin(q3) & 0 & x \\ \sin(q3) & \cos(q3) & 0 & -y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 17 $$

From wheel frame to inertial frame given as

$$ {}^{g}_{w2}T = {}^{g}_{cp}T * {}^{cp}_{w2}T \quad 18 $$

$$ {}^{g}_{w2}T = \begin{bmatrix} \cos(q3) & {}^{g}_{w2}t12 & {}^{g}_{w2}t13 & {}^{g}_{w2}t14 \\ \sin(q3) & {}^{g}_{w2}t22 & {}^{g}_{w2}t23 & {}^{g}_{w2}t24 \\ 0 & \sin(q1) & \cos(q1) & rwcos(q1) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 19 $$

in above matrix where:

$$ {}^{g}_{w2}t12 = -sin(q3)\,cos(q1) \quad 20 $$

$$ {}^{g}_{w2}t13 = sin(q3)\,cos\,q1 \quad 21 $$

$$ {}^{g}_{w2}t14 = x + rw\,sin(q3)\,sin(q1) \quad 22 $$

$$ {}^{g}_{w2}t22 = cos(q3)\,cos(q1) \quad 23 $$

$$ {}^{g}_{w2}t23 = -cos(q3)\,sin(q1) \quad 24 $$

$$ {}^{g}_{w2}t24 = y - rw\,sin(q3)\,sin(q1) \quad 25 $$

Now in the frame of reference, the position of the ground wheel's COG can be represented by

$$ w2\,P_w = [0 \quad 0 \quad 0 \quad 1]^T \quad 26 $$

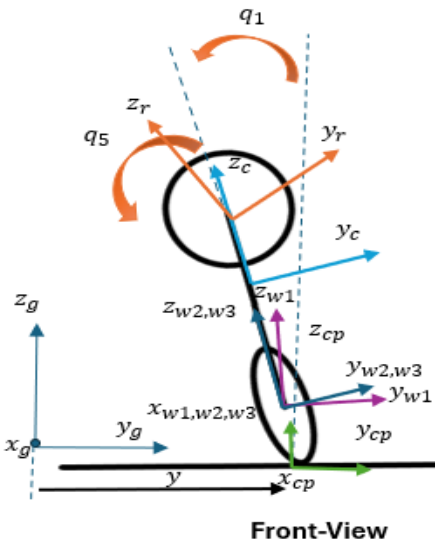
$$ {}^{g}_{w}P = {}^{g}_{w2}T * {}^{w2}_{w}P \quad 27 $$

$$ {}^{g}_{w}P = \begin{bmatrix} x + rw\,sin(q1)\sin(q3) \\ y - rw\sin(q1)\cos(q3) \\ rw\cos(q1) \\ 1 \end{bmatrix} \quad 28 $$

For the chassis of the Unicycle Robot, we have the following configurations.

Figure 3.22

Body/Chassis Frame Definition and Angles.



Using Fig 3.21 and 3.22 the frame transformation using a homogenous matrix is given by:

$$ {}^{p}_{c}T = \begin{bmatrix} \cos(q2) & 0 & \sin(q2) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(q2) & 0 & \cos(q2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & lc \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 29 $$

for chassis COM, and its transformation concerning frame w_2 is as follows.

$$ {}^{w2}_{c}T = \begin{bmatrix} \cos(q2) & 0 & \sin(q2) & lc\sin(q2) \\ 0 & 1 & 0 & 0 \\ -\sin(q2) & 0 & \cos(q2) & lc\cos(q2) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 30 $$

From chassis frame to inertial frame given as

$$ {}^{g}_{c}T = {}^{g}_{w2}T * {}^{w2}_{c}T \quad 31 $$

$$ {}^{g}_{c}T = \begin{bmatrix} {}^{g}_{c}t11 & {}^{g}_{c}t12 & {}^{g}_{c}t13 & {}^{g}_{c}t14 \\ {}^{g}_{c}t21 & {}^{g}_{c}t22 & {}^{g}_{c}t23 & {}^{g}_{c}t24 \\ -\cos(q1)\sin(q2) & \sin(q1) & \cos(q1)\cos(q2) & {}^{g}_{c}t34 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 32 $$

in above matrix where:

$${}^{g}_{c}t11 = \cos(q2)\cos(q3) - \sin(q1)\sin(q2)\sin(q3) \quad (33)$$

$${}^{g}_{c}t12 = -\cos(q1)\sin(q3) \quad (34)$$

$${}^{g}_{c}t13 = \sin(q2)\cos(q3) + \sin(q1)\cos(q2)\sin(q3) \quad (35)$$

$${}^{g}_{c}t14 = x + rw\sin(q3)\sin(q1) + lc\sin(q2)\cos(q3) + lc\sin(q1)\cos(q2)\sin(q3) \quad (36)$$

$${}^{g}_{c}t21 = \cos(q2)\sin(q3) + \sin(q1)\sin(q2)\cos(q3) \quad (37)$$

$${}^{g}_{c}t22 = \cos(q3)\cos(q1) \quad (38)$$

$${}^{g}_{c}t23 = \sin(q2)\sin(q3) - \sin(q1)\cos(q2)\cos(q3) \quad (39)$$

$${}^{g}_{c}t24 = y - rw\cos(q3)\sin(q1) + lc\sin(q2)\sin(q1) - lc\sin(q1)\cos(q2)\cos(q3) \quad (40)$$

$${}^{g}_{c}t34 = rw\cos(q1) + lc\cos(q1)\cos(q2)\,. \quad (41)$$

Now in frame of reference, the position Chasis's COG can be represented in the inertial frame by

$$c\,P_c = [0 \quad 0 \quad 0 \quad 1]^T \quad (42)$$

$${}^{g}_{c}P = {}^{g}_{c}T * {}^{c}_{c}P \quad (43)$$

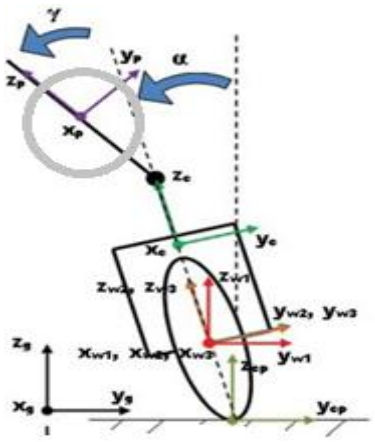
$${}^{g}_{c}P = \begin{bmatrix} x + rw\sin(q1)\sin(q3) + lc\cos(q2)\sin(q1)\sin(q3) + lc\sin(q2)\cos(q3) \\ y - rw\sin(q1)\cos(q3) - lc\cos(q2)\sin(q1)\sin(q3) + lc\sin(q2)\sin(q3) \\ rw\cos(q1) + lc\cos(q2)\cos(q3) \\ 1 \end{bmatrix}$$

44

for Reaction Wheel of Unicycle:

Figure 3.23

Definition of Reaction Wheel Coordinate System (Grey)



$$ {}^{cp}_{w2}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(q5) & -\sin(q5) & 0 \\ 0 & \sin(q5) & \cos(q5) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & lcp \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & lp \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (45) $$

$$ {}^{cp}_{w2}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(q5) & -\sin(q5) & -lp * \sin(q5) \\ 0 & \sin(q5) & \cos(q5) & lcp + lp * \cos(q5) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (46) $$

From wheel frame to inertial frame given as

$$ {}^{g}_{p}T = {}^{g}_{c}T * {}^{c}_{p}T \quad (47) $$

$$ {}^{g}_{w2}T = \begin{bmatrix} \cos(q3) & {}^{g}_{p}t12 & {}^{g}_{p}t13 & {}^{g}_{p}t14 \\ \sin(q3) & {}^{g}_{p}t22 & {}^{g}_{p}t23 & {}^{g}_{p}t24 \\ 0 & {}^{g}_{p}t32 & {}^{g}_{p}t33 & {}^{g}_{p}t34 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

48

in above matrix where:

$$ {}^{g}_{p}t12 = -sin(q3)\,cos(q1)\,cos(q5) + cos(q3)\,sin(q2)\,sin(q5) + sin(q3)\,sin(q1)\,cos(q2)\,sin(q5), \quad (49) $$

$$ {}^{g}_{p}t13 = sin(q3)\,cos(q1)\,sin(q5) + cos(q3)\,sin(q2)\,cos(q5) + sin(q3)\,sin(q1)\,cos(q2)\,cos(q5), \quad (50) $$

$$ {}^{g}_{p}t14 = lp\,sin(q3)\,cos(q1)\,sin(q5) + (lcp + lp\,cos(q5))(cos(q3)\,sin(q2) + sin(q3)\,sin(q1)\,cos(q2)) + lc\,sin(q2)\,cos(q3) + lc\,cos(q2)\,sin(q3)\,sin(q1) + x + rw\,sin(q3)\,sin(q1), \quad (51) $$

$$ {}^{g}_{p}t22 = cos(q3)\,cos(q1)\,cos(q5) + sin(q3)\,sin(q2)\,sin(q5) - cos(q3)\,sin(q1)\,cos(q2)\,sin(q5), \quad (52) $$

$$ {}^{g}_{p}t23 = -cos(q3)\,cos(q1)\,sin(q5) + sin(q3)\,sin(q2)\,cos(q5) - cos(q3)\,sin(q1)\,cos(q2)\,cos(q5), \quad (53) $$

$$ {}^{g}_{p}t24 = -lp\cos(q3)\cos(q1)\sin(\gamma) + (lcp + lp\cos(\gamma))(\sin(q3)\sin(\beta)) - \cos(q3)\sin(q1)\cos(\beta)) + lc\sin(\beta)\sin(q3) - lc\cos(\beta)\cos(q3)\sin(q1) + y - rw\cos(q3)\sin(q1), \quad (54) $$

$$ {}^{g}_{p}t32 = sin(q1)\,cos(q5) + cos(q1)\,cos(q2)\,sin(q5), \quad (55) $$

$$ {}^{g}_{p}t33 = -sin(q1)\,sin(q5) + cos(q1)\,cos(q2)\,cos(q5), \quad (56) $$

$$ {}^{g}_{p}t33 = -lp\,sin(q1)\,sin(q5) + (lcp + lp\,cos(q5))\,cos(q1)\,cos(q2) + lc\,cos(q2)\,cos(q1) + rw\,cos(q1), \quad (57) $$

Now in frame of reference, the position of the reaction wheel's COG can be represented by

$$ {}^{p}P_{p} = [0 \quad 0 \quad 0 \quad 1]^{T} \tag{58} $$

$$ {}^{g}_{p}P = {}^{g}_{p}T * {}^{p}_{p}P \tag{59} $$

$$ {}^{g}_{p}P = \begin{bmatrix} {}^{g}_{p1}P \\ {}^{g}_{p2}P \\ {}^{g}_{p3}P \\ 1 \end{bmatrix} \tag{60} $$

$$ {}^{g}_{p1}P = x + rw\sin(q1) * \sin(q3) + (lc + lcp)\cos(q2) * \sin(q1)\sin(q3) + (lc + lcp)\sin(q2) * \cos(q3) + lp\cos(q5)\cos(q2)\sin(q1)\sin(q3) + lp\cos(q5)\sin(q2)\cos(q3) + lp\sin(q5)\cos(q1)\sin(q3), \tag{61} $$

$$ {}^{g}_{p2}P = y - rw\sin(q1)\cos(q3) - (lc + lcp)\cos(q2) * \sin(q1)\cos(q3) + (lc + lcp)\sin(q2) * \sin(q3) - lp\cos(q5)\cos(q2)\sin(q1)\cos(q3) + lp\cos(q5)\sin(q2)\sin(q3) - lp\sin(q5)\cos(q1) * \cos(q3), \tag{62} $$

$$ {}^{g}_{p3}P = rw\cos(q1) + (lc + lcp)\cos(q2) * \cos(q1) + lp\cos(q5) * \cos(q2)\cos(q1) - lp\sin(q5)\sin(q1). \tag{63} $$

3.6.5.2 **Linear Velocity Vectors for Unicycle.** Now moving forward with same modeling for linear velocity for ground wheel, body, and reaction wheel. This can be derived from equation 25, 45, 57.

for Ground Wheel Velocity Vector:

The position vector for ground wheel we have

$$ {}^{g}_{w}P = \begin{bmatrix} x + rw\,sin(q1)\sin(q3) \\ y - rw\sin(q1)\cos(q3) \\ rw\cos(q1) \\ 1 \end{bmatrix} \tag{64} $$

$$Vw = \frac{\partial Pw}{dt} \quad 65$$

for Chassis Linear Velocity Vector:

Similarly, the position vector for chassis we have

$$ {}^{g}_{c}P = \begin{bmatrix} x + rw\ sin(q1)\sin(q3) + lc\cos(q2)\sin(q1) * \sin(q3) + lc\sin(q2) * \cos(q3) \\ y - rw\sin(q1)\cos(q3) - lc\cos(q2)\sin(q1) * \sin(q3) + lc\sin(q2) * \sin(q3) \\ rw\cos(q1) + lc\cos(q2) * \cos(q1) \\ 1 \end{bmatrix} \quad 66$$

$$Vc = \frac{\partial Pc}{dt} \quad 67$$

for Reaction Wheel Velocity Vector:

for reaction wheel using equation 60 we have:

$$Vw = \frac{\partial Pp}{dt} \quad 68$$

The full derivation for these vectors has been done in MATLAB as the calculations are very complex in this manner and repository is attached with this report.

3.6.5.3 **Angular Velocity Vectors for Unicycle**. From the frame assignment, the angular velocity for this unicycle parts (ground wheel, chassis, and reaction wheel) can be represented in terms of frames $\boldsymbol{w_2}$, C, and P respectively and this turns the matrices for all three parts of unicycle into diagonal matrices.

Ground Wheel Angular Velocity:

this vector consists of wheel velocity, angular velocity (lean), and turning velocity in terms of ω, $q1$, and $q3$.

$$\Omega_w = \begin{bmatrix} 0 \\ q4 \cdot \\ 0 \end{bmatrix} + \begin{bmatrix} q1 \cdot \\ 0 \\ 0 \end{bmatrix} + {}^{w2}_{g}T\ x \begin{bmatrix} 0 \\ 0 \\ q3 \cdot \end{bmatrix} \quad 69$$

$$\Omega_w = \begin{bmatrix} 0 \\ q4 \cdot \\ 0 \end{bmatrix} + \begin{bmatrix} q1 \\ 0 \\ 0 \end{bmatrix} + {}^{g}_{w2}T^{-1}\ x \begin{bmatrix} 0 \\ 0 \\ q3 \cdot \end{bmatrix} \quad 70$$

$$\Omega_w = \begin{bmatrix} q1^{\cdot} \\ q4^{\cdot} + q3^{\cdot}\sin(q1) \\ q3^{\cdot}\cos(q1) \end{bmatrix} \quad 71$$

Angular Velocity of Chassis of Unicycle:

The chassis angular velocity vector consists of two lean angular velocities (time rate q1 β) and turning velocity q3.

$$\Omega_c = \begin{bmatrix} 0 \\ q2^{\cdot} \\ 0 \end{bmatrix} + {}^{c}_{w2}T\ x \begin{bmatrix} q1^{\cdot} \\ 0 \\ 0 \end{bmatrix} + {}^{c}_{g}T\ x \begin{bmatrix} 0 \\ 0 \\ q3^{\cdot} \end{bmatrix} \quad 72$$

$$\Omega_c = \begin{bmatrix} 0 \\ q2^{\cdot} \\ 0 \end{bmatrix} + {}^{w2}_{c}T^{-1}\ x \begin{bmatrix} q1^{\cdot} \\ 0 \\ 0 \end{bmatrix} + {}^{g}_{c}T^{-1}\ x \begin{bmatrix} 0 \\ 0 \\ q3^{\cdot} \end{bmatrix} \quad 73$$

$$\Omega_c = \begin{bmatrix} q1^{\cdot} * \cos(q2) - q3^{\cdot} * \cos(q1) * \sin(q2) \\ q2^{\cdot} + q3^{\cdot}\sin(q1) \\ q1^{\cdot}\sin(q2) + q3^{\cdot}\cos(q1) * \cos(q2) \end{bmatrix} \quad 74$$

Reaction Wheel Angular Velocity:

the reaction wheel velocity vector consists of lean angular (time rate q1), longitudinal velocity (time rate of β) and reaction wheel angular velocity (time rate of ϒ).

$$\Omega_p = \begin{bmatrix} q5^{\cdot} \\ 0 \\ 0 \end{bmatrix} + {}^{p}_{c}T\ x \begin{bmatrix} 0^{\cdot} \\ q2^{\cdot} \\ 0 \end{bmatrix} + {}^{p}_{w2}T\ x \begin{bmatrix} q1^{\cdot} \\ 0 \\ 0 \end{bmatrix} + {}^{p}_{g}T\ x \begin{bmatrix} 0 \\ 0 \\ q3^{\cdot} \end{bmatrix} \quad 75$$

$$\Omega_p = \begin{bmatrix} q5^{\cdot} \\ 0 \\ 0 \end{bmatrix} + {}^{c}_{p}T^{-1}\ x \begin{bmatrix} 0^{\cdot} \\ q2^{\cdot} \\ 0 \end{bmatrix} + {}^{w2}_{p}T^{-1}\ x \begin{bmatrix} q1^{\cdot} \\ 0 \\ 0 \end{bmatrix} + {}^{g}_{p}T^{-1}\ x \begin{bmatrix} 0 \\ 0 \\ q3^{\cdot} \end{bmatrix} \quad 76$$

$$\Omega_p = \begin{bmatrix} q5^{\cdot} + q1^{\cdot}\cos(q2) - q3^{\cdot}\cos(q1)\sin(q2) \\ q2^{\cdot}\cos(q5) + q1^{\cdot}\sin(q2)\sin(q5) + q3^{\cdot}\sin(q1)\cos(q5) + q3^{\cdot}\cos(q1)\cos(q2)\sin(q5) \\ -q2^{\cdot}\cos(q5) + q1^{\cdot}\sin(q2)\sin(q5) - q3^{\cdot}\sin(q1)\cos(q5) + q3^{\cdot}\cos(q1)\cos(q2)\cos(q5) \end{bmatrix} \quad 77$$

3.6.5.4 **Kinetic and Potential Energies for Unicycle.** Using the above calculations for the ground wheel, chassis, and reaction wheel for position, linear, and angular velocities, the energies of the robot parts can be expressed in standard mechanics.

the Kinetic and Potential Energy for Ground Wheel:

the kinetic energy is:

$$T_w = \frac{1}{2} m_w V_w^T V_w + \frac{1}{2} \Omega_w^T I_w \Omega_w \quad 78$$

the potential energy is:

$$P_w = m_w g P_{w^*} \quad 79$$

Pw* is the height of COM of wheel.

Kinetic and Potential Energy for Chassis:

the kinetic energy is:

$$T_c = \frac{1}{2} m_c V_c^T V_c + \frac{1}{2} \Omega_c^T I_c \Omega_c \quad 80$$

the potential energy is:

$$P_c = m_c g P_{c^*} \quad 81$$

Pc* is the height of COM of the chassis.

the Kinetic and Potential Energy for Reaction Wheel:

the kinetic energy is:

$$T_p = \frac{1}{2} m_p V_p^T V_p + \frac{1}{2} \Omega_p^T I_p \Omega_p \quad 82$$

the potential energy is:

$$P_p = m_p g P_{p^*} \quad 83$$

Pp* is the height of COM of the reaction wheel.

3.6.5.5 **Total Energy of the Unicycle Robot.** The total energy is the summation of all three components for unicycle.

$$T_{total} = T_p + T_c + T_w \quad (84)$$

and similarly, the potential energy of the unicycle

$$P_{total} = P_p + P_c + P_w \quad (85)$$

Now using energy values, the Lagrangian equation can be solved. the coordinates that are defined for this unicycle are m = 7, and constraints (kinematic) are n = 2. The computations are solved using MATLAB ODE solver however variables can be seen in the table below.

Table 3.3

Lagrange Equation's Variables

S. No	qi	Qi	q1i	q2i
1	x	0	1	0
2	y	0	0	1
3	q4	Tw	-rwcos(q3)	-rwsin(q3)
4	q1	0	0	0
5	q2	-Tw	0	0
6	q5	Tp	0	0
7	q3	0	0	0

the equation for Lagrange is given by:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial y\cdot}\right) - \frac{\partial L}{\partial y} = \lambda_2 \quad (86)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \theta\cdot}\right) - \frac{\partial L}{\partial \theta} = T_w - r_w\, cos(q3)\lambda_1 - r_w\, sin(q3)\,\lambda_2 \quad (87)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial q2\cdot}\right) - \frac{\partial L}{\partial q2} = -T_w \quad (88)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial q1\cdot}\right) - \frac{\partial L}{\partial q1} = 0 \quad (89)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial q5\cdot}\right) - \frac{\partial L}{\partial q5} = T_p \quad (90)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial q3\cdot}\right) - \frac{\partial L}{\partial q3} = 0 \quad (91)$$

To calculate the dynamics, all the values were fed into MATLAB ODE solver for the four equations from 82-87.

the Ground Wheel Dynamics:

$$m_{11}q1^{\cdot\cdot} + m_{12}q5^{\cdot\cdot} + m_{13}q3^{\cdot\cdot} \neq m_{14}q4^{\cdot\cdot} + c_{11}q2^{\cdot 2} + c_{12}q5^{\cdot 2} + c_{13}q3^{\cdot 2} + c_{14}q1^{\cdot 2}\,q3^{\cdot} + c_{15}q5^{\cdot}q2^{\cdot} + c_{16}q3^{\cdot}q2^{\cdot} + c_{17}q5^{\cdot}q3^{\cdot} = T_w \quad (92)$$

Chassis Dynamics in Longitudinal Direction:

$$m_{21}q1^{\cdot\cdot} + m_{22}q2^{\cdot\cdot} + m_{23}q3^{\cdot\cdot} \neq m_{24}q4^{\cdot\cdot} + c_{21}q1^{\cdot 2} + c_{22}q3^{\cdot 2} + c_{23}q1^{\cdot}q5 + c_{24}q1^{\cdot}q3^{\cdot} + + c_{25}q5^{\cdot}q2^{\cdot} + c_{26}q5^{\cdot}q3^{\cdot} + c_{27}q5^{\cdot}q4^{\cdot} + u21 = -T_w \quad (93)$$

Chassis Dynamics in Lateral Direction:

$$m_{31}q1^{\cdot\cdot} + m_{32}q2^{\cdot\cdot} + m_{33}q5^{\cdot\cdot} \neq m_{34}q3^{\cdot\cdot} + c_{31}q2^{\cdot 2} + c_{32}q5^{\cdot 2} + c_{33}q3^{\cdot^2} + c_{34}q1^{\cdot}q2^{\cdot} + + c_{35}q5^{\cdot}q1^{\cdot} + c_{36}q2^{\cdot}q3^{\cdot} + c_{37}q2^{\cdot}q3^{\cdot} + c_{38}q5^{\cdot}q3^{\cdot} + c_{39}q4^{\cdot}q3^{\cdot} = 0 \quad (94)$$

Reaction Wheel Dynamics:

$$m_{41}q1^{\cdot\cdot} + m_{42}q5^{\cdot\cdot} \neq m_{43}q3^{\cdot\cdot} + m_{44}q4^{\cdot\cdot} + c_{41}q1^{\cdot 2} + c_{42}q2^{\cdot 2} + c_{43}q3^{\cdot^2} + c_{44}q1^{\cdot}q2^{\cdot} + + c_{45}q3^{\cdot}q1^{\cdot} + c_{46}q2^{\cdot}q3^{\cdot} + c_{47}q4^{\cdot}q3^{\cdot} + u41 = Tp \quad (95)$$

Turning Dynamic, the interesting findings through this model can be seen in turning dynamics where dynamics of the yaw are coupled. this yaw direction is controlled by coordinated control of ground wheel and reaction wheel indirectly as the direct control of yaw is not possible in this orthogonal configuration of the unicycle.

$$m_{51}q1^{\cdot\cdot} + m_{52}q2^{\cdot\cdot} + \ m_{53}q5^{\cdot\cdot} + m_{54}q3^{\cdot\cdot} + m_{55}q4^{\cdot\cdot} + \ c_{51}q1^{\cdot 2} + \ c_{52}q2^{\cdot 2} + \ c_{53}q5^{\cdot 2} + c_{54}q1^{\cdot}q2^{\cdot} + \ c_{55}q5^{\cdot}q1^{\cdot} + c_{56}q1^{\cdot}q3^{\cdot} + c_{57}q4^{\cdot}q1^{\cdot} + c_{58}q2^{\cdot}q5^{\cdot} + \ c_{59}q2^{\cdot}q3^{\cdot} + c_{510}q5^{\cdot}q3^{\cdot} + c_{511}q4^{\cdot}q3^{\cdot} \ = \ 0 \qquad 96$$

the coefficient in equations 88-92 m, c is mentioned in the MATLAB file attached. The dynamic model is seen as a nonlinear model for the unicycle robot. Therefore, model simplification is done.

3.6.6 Simplification of the Unicycle Model

The Unicycle robot model is simplified using pseudo linearization around the upright equilibrium position of the unicycle robot. All the dynamics were solved in MATLAB and the model equations are presented here. for linearization, the following conditions are met:

$$q1 = \ q2 = q5 = 0^0 \qquad 97$$

$$q1^{\cdot} = \ q2^{\cdot} = q5^{\cdot} = 0^0/sec \qquad 98$$

Procedure,

1. The $q1$ $(roll\ angle)$, $q2$ $(pitch\ angle)$ and $q5$ $(fly - wheel\ angle)$ sines angles are approximated.
   for example,

$$Sin(q1) = \ Sin(q2) = Sin(q5) \cong \ q1, q2, \ q5\ respectively \qquad 99$$

Similarly, all the cosines' angles are approximated to 1.

$$Cos(q1) = \ Cos(q2) = Cos(q5) \cong \ 1 \qquad 100$$

2. By neglecting the higher order terms of the dynamic model of unicycles along with their coefficient parameters m, c, and u.

After simplification, the resulting dynamics are as follows

Linearized Ground Wheel Dynamics:

$$\epsilon_1 r_w q2^{\cdot\cdot} + (I_{w2} + \epsilon_3 r_w + \epsilon_1 r_w \ )q1q3^{\cdot\cdot} + \ \epsilon_2 r_w \ q5q3^{\cdot\cdot} + (I_{w2} + \epsilon_3 r_w)q4^{\cdot\cdot} - \epsilon_1 r_w \ q2q3^{\cdot 2} + (\ I_{w2} + 2\epsilon_3 r_w + \ 2\epsilon_1 r_w)q1^{\cdot}q3^{\cdot} + 2\epsilon_2 r_w \ q5^{\cdot}q3^{\cdot} = \tau_w \qquad 101$$

Linearized Chassis Longitudinal Dynamics:

$$q2^{\cdot\cdot}\left(I_{c2}+I_{p2}+\ \rho_1\right)+\left(I_{c2}+I_{p2}+\ \rho_1+\ \epsilon_1 r_w\ \right)q1q3^{\cdot\cdot}\Big)+(\sigma_3+\ \ \rho_3)q5q3^{\cdot\cdot}+$$
$$\epsilon_1 r_w\ q4^{\cdot\cdot}-(\ \sigma_4+\ \ \rho_1)q2q3^{\cdot 2}+\left(I_{c1}+\sigma_2+\ I_{p1}+\sigma_3+\ 2\rho_1+2\rho_1+2\epsilon_1 r_w\right)q1^{\cdot}q3^{\cdot}+$$
$$\left(I_{p1}+\sigma_3+\ 2\rho_1\right)q5^{\cdot}q3^{\cdot}-\epsilon_1 g q2=\ -\tau_w \qquad 102$$

Linearized Chassis Lateral Dynamics:

$$q1^{\cdot\cdot}\left(I_{w1}+I_{p1}+I_{c1}+\ \rho_1+\epsilon_3 r_w+\ 2\epsilon_1 r_w\right)+\left(I_{p1}+\ \rho_3+\ \epsilon_2 r_w\ \right)q5^{\cdot\cdot}\Big)-$$
$$(\sigma_4+\ \epsilon_1 r_w+\ \rho_1)q2q3^{\cdot\cdot}-(\sigma_1+\sigma_3+\ \sigma_2+\epsilon_3 r_w+\rho_1+\epsilon_1 r_w)q1q3^{\cdot 2}\ -$$
$$(\ \sigma_3+\ \ \rho_3+\ \epsilon_2 r_w)q5q3^{\cdot 2}-q2^{\cdot}q3^{\cdot}\left(I_{c1}+\sigma_2+\ I_{p1}+\sigma_3+\ 2\rho_1+2\epsilon_1 r_w\right)-q1^{\cdot}q3^{\cdot}+$$
$$(I_{w2}+\epsilon_3 r_w+\ \epsilon_1 r_w\ )\ q3^{\cdot}q4^{\cdot})-(\epsilon_1+\epsilon_3)g q1-\ \epsilon_2 g q5=\ 0 \qquad 103$$

Linearized Reaction Wheel Dynamics:

$$\left(I_{p1}+\ \rho_3+\ \epsilon_2 r_w\ \right)q1^{\cdot\cdot}\Big)+\left(I_{p1}+\epsilon_2 I_p\right)q5^{\cdot\cdot}-\left(I_{p1}+\ \rho_3\right)q2q3^{\cdot\cdot}-(\sigma_3+\ \epsilon_2 r_w+$$
$$\rho_3)q1q3^{\cdot 2}-\left(I_{p1}+\ 2\rho_3+\ \sigma_3\right)q2^{\cdot}q3^{\cdot}-\epsilon_2 r_w q3^{\cdot}q4^{\cdot}-\epsilon_2 g q5-\ \epsilon_2 g q1=\ \ \tau_p$$
$$104$$

Linearized Turning Dynamics:

$$-(\ \sigma_4+\ \ \rho_1+\ \epsilon_1 r_w)q2q1^{\cdot\cdot}+\left(I_{c2}+\ I_{p2}+\ \rho_1+\epsilon_1 r_w\right)q1q2^{\cdot\cdot}+(\sigma_3+\ \ \rho_3)q5q2^{\cdot\cdot}-$$
$$\left(I_{p1}+\ \rho_3\right)q2q5^{\cdot\cdot}+(I_{w3}+\epsilon_1 r_w+\epsilon_3 r_w)q1q4^{\cdot\cdot}+\epsilon_2 r_w q5q4^{\cdot\cdot}+\ I_{w2}q1^{\cdot}q4^{\cdot}+$$
$$\epsilon_1 r_w q3^{\cdot}q4^{\cdot}\ q2=\ 0 \qquad 105$$

The values for variables used in equations are mentioned below.

$$\epsilon_1=m_c l_c+m_p\left(l_c+l_{cp}+l_p\right) \qquad 106$$

$$\epsilon_2=\ m_p l_p \qquad 107$$

$$\epsilon_3=\left(m_w+\ m_c+m_p\right)r_w \qquad 108$$

$$\rho_1=\ m_c l_c^2+m_p\left(l_c+l_{cp}+l_p\right)^2 \qquad 109$$

$$\rho_2=\ m_p l_p^2 \qquad 110$$

$$\rho_3=m_p\left(l_c+l_{cp}+l_p\right)l_p \qquad 111$$

$$\sigma_1=I_{w2}-\ I_{w3} \qquad 112$$

$$\sigma_2=I_{c2}-\ I_{c3} \qquad 113$$

$$\sigma_3=I_{p2}-\ I_{p3} \qquad 114$$

$$\sigma_4=I_{c1}-\ I_{c3}+I_{p1}-\ I_{p3} \qquad 115$$

The simplified model has more non-linear characteristics concerning q3, $q4$, and their first and second derivatives.

3.7 State Estimation Model

In this robot, a gyroscope is used in the inertial measurement unit (IMU) whose location is known as can be seen in Fig 3.4. Therefore, tilt estimation and rate estimation are done using an accelerometer and gyroscope respectively.

3.7.1 Definition of Model Parameters

$q_1: Roll\ angle$

$q_2: Pitch\ angle$

$q_3: yaw\ angle$

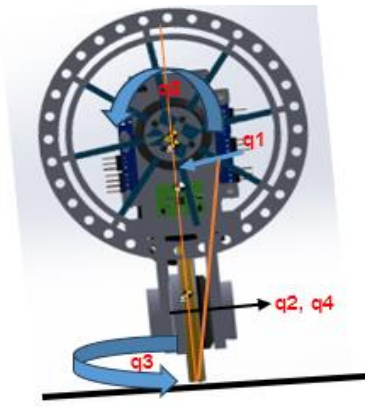
$q_4: ground\ wheel\ angle$

$q_5: reaction\ wheel\ angle$

$\tilde{q_n} = is\ used\ for\ representing\ estimation$

Figure 3.24

Unicycle Pose and Variable Assignments



The Euler rate estimates for roll, pitch and yaw angles are transformed into body frames concerning inertial frame. Here the body frame is represented by {B} and inertial frame by {i}.

3.7.2 Using Gyroscope

$${}^{B}_{i}\omega = R_{Bi} * {}^{i}_{i}W \qquad 116$$

From the sensor data previous tilt estimates are extracted mathematically as follows

$$q_1^{\sim}(k-1), q_2^{\sim}(k-1)\ into\ inertial\ frame\ \{I\}$$

The detailed implementation can be found in MATLAB code. However, after transformation the final version from average body rates to Euler rates is mentioned here.

$$\begin{bmatrix} q_{1,g}^{\cdot} \\ q_{2,g}^{\cdot} \\ q_{3,g}^{\cdot} \end{bmatrix} = [R_2^T e1 \quad e2 \quad R_2^T R_1^T e3]^{-1} \sum_{k=1}^{4} \frac{{}_i^B\omega(k)}{4} \qquad 117$$

Here

$$[R_2^T e1 \quad e2 \quad R_2^T R_1^T e3]^{-1} = \begin{bmatrix} \cos(q_2) & 0 & \sin(q_2) \\ 0 & 1 & 0 \\ -\cos(q_1) * \sin(q_2) & \sin(q_1) & \cos(q_1) * \cos(q_2) \end{bmatrix} \qquad 118$$

3.7.3 Tilt Estimation Using Accelerometer

For tilt estimation, two main approaches were explored: Kalman filters and extended Kalman filters, as mentioned by V. Klemm (2019) and L. Hertig (2013), and the non-accelerated pivot point method, as expressed by D'andrea (2018). in Kalman filtering, the approximation model relies on local approximations and is susceptible to errors in dynamic and noise parameters. Conversely, the pivot point method has been widely employed in various robots, as documented by M. Gajamohan (2013) and M. Muelhlebah (2017). Therefore, this approach was chosen as it provides an optimal solution when the kinematic model is known. The actuators used for this robot are equipped with encoders, enabling estimation via encoder readings.

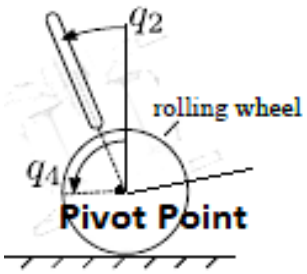
in an inertial Measurement Unit (IMU), the accelerometer measurement at location \(P_{acc}, for example, of the i^{th} sensor with respect to the body frame {B}, is represented by:

$${}_i^B m = {}_i^B \ddot{p} - B_g + {}_i^B n \qquad 119$$

Here for this research work, the center of wheel is considered as pivot point as shown figure 3.25.

Figure 3.25

Unicycle with Pivot Point



The estimation routine implemented in MATLAB resulted in the following final expression and methodology for tilt-estimates is extrapolated from Cubli robot to Unicycle robot as mentioned in research paper by (D'andrea and Muehlebach, 2018).

$$q1 = \arctan\left(\frac{g\tilde{}^{B}_{2}}{\sqrt{g\tilde{}^{2\,B}_{1} + g\tilde{}^{2\,B}_{3}}} \right) \quad 120$$

$$q2 = \arctan\left(-\frac{g\tilde{}^{B}_{1}}{g\tilde{}^{B}_{3}} \right) \quad 121$$

Acceleration estimation is performed through encoder measurements in MATLAB, followed by sensor fusion techniques using complementary filtering for final data implementation. The fusion process combines gyroscope rate estimates and accelerometer tilt estimates according to a predetermined scheme, with a fusion parameter q1 set to 0.02.

$$\begin{bmatrix} \tilde{q_1} \\ \tilde{q_2} \end{bmatrix} = \alpha \begin{bmatrix} q_{1A} \\ q_{2A} \end{bmatrix} + (1 - \alpha) \begin{bmatrix} q_{1g} \\ q_{2g} \end{bmatrix} \quad 122$$

The complementary filter research work implemented in this robot is Inspired by (R.G. Brown and Hwang.,2014) work where the discrete filter combines low-frequent and high-frequent parts from the accelerometer and gyroscope for tilt estimate respectively.

3.8 Controller Design for Unicycle

3.8.1 LQR Control

In the classical control domain, a linear quadratic regulator (LQR) is designed since it is known to be the well-established controller for unstable robots (P. Fankhauser et al.,2010 and J. Lee & S. Han.,2013 and V.Klemm et al., 2019). Above linearized dynamics for unicycle robots are used and the state-space model of the bicycle is represented below. The definition of variables is mentioned in the state estimation section of this chapter.

Substituting system matrices for multiplication as follows

$$x^{\cdot} = Ax + Bu \tag{123}$$

Control law here:

$$u = -K * x \tag{124}$$

to obtain the optimal gain:

$$K = R^{-1}B'P \tag{125}$$

and the Ricotti equation is used to get the value of P matrix.

$$A'P + PA - PBR^{-1}B'P + Q = 0 \tag{126}$$

Here {Q, R} are chosen in MATLAB with system response time 10msec and obtained gains K1, K2.

$$\left\{\begin{bmatrix} q_1 \\ \dot{q_1} \\ q_5 \\ \dot{q_5} \end{bmatrix}\right\} = [Q1\,] \tag{127}$$

$$\left\{\begin{bmatrix} q_2 \\ \dot{q_2} \\ q_4 \\ \dot{q_4} \end{bmatrix}\right\} = [Q2\,] \tag{128}$$

$$\frac{d}{dt}\begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}\begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} + \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix}\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \tag{129}$$

For the unicycle robot, the estimator values for U1 and U2 are coupled with the controller through the IMU calibration routine before the controller starts operating. It

is widely acknowledged that yaw dynamics, represented by q3 here, are not controllable using LQR, which is why they are omitted from consideration.

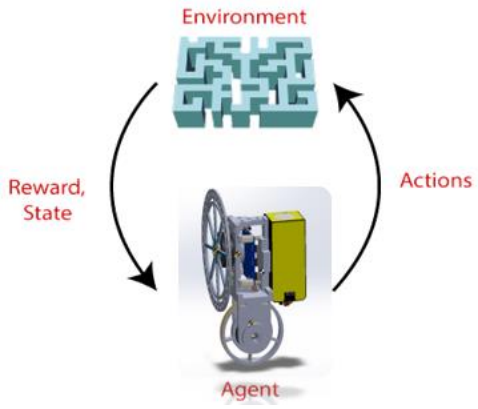
Upon analyzing the system's response in MATLAB, it was observed that the coordinated control of both the ground and reaction wheels resulted in turning behavior for the unicycle. in dynamic system modeling, numerous coupling terms are calculated, which are then mapped with other states. As mentioned earlier in this section, diagonal matrices for the two controllers for this unicycle were obtained as diagonal matrices using MATLAB. The results from these computations are discussed in Chapter 4.

3.9 Reinforcement Learning Controller and Selection Criteria

In the model-free domain, the reinforcement learning controller is designed in PyBullet using Python programming. The typical flow of control consists of an agent (robot), the environment, and the action the agent takes. It is mentioned below:

Figure 3.26

Unicycle Robot in Reinforcement Learning Framework.



for unicycle robots, various other algorithms were tried like Q-learning, Double Q-learning (DQN), Proximal Policy Optimization (PPO) and Deep Deterministic Policy Gradient (DDPG). they were several reasons to choose the DDPG algorithm out of other start-of-the-art like PPO:

1. DDPG utilizes experience replay, a technique that stores past experiences in a replay buffer and samples from it during training. This mechanism helps break correlations between consecutive samples and stabilizes the learning process for unicycles. Experience replay can be particularly advantageous in environments

with complex dynamics or sparse rewards, facilitating more efficient learning compared to PPO (Mnih et al., 2015).

2. DDPG learns the deterministic policy as compared to stochastic policy of PPO. the deterministic policy can be beneficial in situations where deterministic behavior is desired. for example, maintaining specific poses in the case of unicycle robots. Tasks that require precise and predictable actions can benefit from DDPG's deterministic nature, leading to stability and reliability in decision-making for balancing controls (T.P. et al., 2016).
3. DDPG is widely used and proven effective for continuous control tasks, such as robotics applications while the same case exists for PPO at the expense of carefully tuning hyperparameters for optimal performance.

Therefore, considering all the facts, DDPG emerged as natural choice for this work and has been tailored for this unicycle robot.

3.9.1 Deep Deterministic Policy Gradient (DDPG) Algorithm

The algorithm chosen for this unicycle is the DDPG, which simultaneously learns the Q-function and the policy. It is assumed that the reader possesses a basic understanding of reinforcement learning processes to comprehend this section of the report. the approach of the DDPG algorithm is somewhat akin to Q-learning, where the optimal action-value function Q*(s,a) is determined for any given state and action using the following expression derived from the Bellman equation.

$$a*(s) = \arg max_a \, Q^*(s,a) \qquad 130$$

The unicycle control is a continuous action space problem therefore, function Q*(s,a) should be differentiable with regard to action. So, the gradient-based learning rule is developed. The algorithm development process is divided into two parts. Learning the Q-function and learning the policy for the unicycle.

3.9.2 Q-Learning for DDPG

Q-learning starts with Bellman equation and can be written for the optimal action-value function (OAVF), Q*(s,a) like below (OpenAI:Spinningup)

$$Q^*(s,a) = E_{s\sim P}\,[\,r\,(s,a) + \gamma \max a' Q^*(s',a')] \qquad 131$$

In this context, the Bellman equation should be satisfied by the optimal Q-value function, meaning that the transition of tuple (state-action pairs) does not significantly affect it. This is due to the inclusion of a replay buffer, which retains past experiences, and the adjustment of actions taken by the unicycle as it transitions from one state to another through the Q-function with mean-squared Bellman error (MSBE). Additionally, Q-learning incorporates target networks using a neural network approach.

$$r + \gamma\,(1-d)\max a' Q(s', a') \qquad 132$$

When MSBE loss is minimized, Q-network can be more likely to represent as target network. DDPG overcomes the problem of computing the maximum over actions in the target network by utilizing the target policy network which can maximize Q(target). Hence the approach used in DDPG with MSBE, and stochastic gradient descent is as follows:

$$L(\phi, D) = Expt_{(s,,a,r,s',d)\sim D}[\left(Q_\phi(s,a) - \left(r + \gamma(1-d)Q_{\phi targ}\left(s', \mu\theta_{targ}(s')\right)\right)^2\right] \qquad 133$$

$$\mu\theta_{targ}\ is\ the\ target\ policy\ of\ the\ network$$

3.9.3 Policy Learning of DDPG

The deterministic policy is learned which maximizes the $Q_\phi(s,a)$. for continuous action space it is written:

$$max_\theta E_{s\sim D}\left[Q_\emptyset\left(s, \mu_\theta(s)\right)\right] \qquad 134$$

For implementation, the author resorted to deterministic policy gradient algorithm approach implemented by (Silver et al.2014) and pseudo code is presented below.

Figure 3.27

Pseudo Code for DDPG

Algorithm 1 DDPG for Unicycle Control

Initialize: randomly assign actor network parameters θ^{μ} and critic network parameters θ^{Q}.
Initialize: set target network $\theta^{\mu'} \leftarrow \theta^{\mu}$ and $\theta^{Q'} \leftarrow \theta^{Q}$.
Initialize: create replay buffer R.
Initialize: set exploration noise parameters μ, θ.
For each episode from 1 to M
  Initialize a random process N for action exploration.
  Observe initial state s.
  For each time step from 1 to T
    Choose action $a = \mu(s|\theta^{\mu}) + N(t)$.
    Execute action a, observe reward r and new state s'.
    Store transition (s, a, r, s') in R.
    Sample minibatch of transitions (s_i, a_i, r_i, s_i') from R.
    Set target value $y_i = r_i + \gamma \cdot Q'(s_i', \mu'(s_i'|\theta^{\mu'})|\theta^{Q'})$.
    Update critic by minimizing loss:
      $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^{Q}))^2$.
    Update actor policy using policy gradient:
      $\nabla_{\theta^{\mu}} J \approx \frac{1}{N}\sum \nabla_a Q(s, a|\theta^{Q})|_{(s,a)=(s_i, \mu(s_i|\theta^{\mu}))}$.
    Update target networks:
      $\theta^{Q'} \leftarrow \tau \cdot \theta^{Q} + (1-\tau) \cdot \theta^{Q'}$
      $\theta^{\mu'} \leftarrow \tau \cdot \theta^{\mu} + (1-\tau) \cdot \theta^{\mu'}$.
    Update current state: $s = s'$.

CHAPTER 4

RESULTS AND DISCUSSION

This chapter encompasses the discussion of results and their interpretation, covering both simulation and experimental studies. Initially, the results were validated through simulations conducted for classical control in MATLAB and for reinforcement learning using the Bullet Physics engine. Subsequently, real-time control was achieved using custom-designed hardware.

4.1 Classical Control in MATLAB

4.1.1 The Dynamics Modeling & Lagrangian

In the initial phase, the implementation of these dynamics in MATLAB necessitated the utilization of data extracted from the CAD model of the unicycle. Consequently, crucial physical dimension parameters of the unicycle were derived from the CAD software Solidworks, offering essential data for accurate calculations.

4.1.2 MATLAB Coding Procedure

Table 4.1

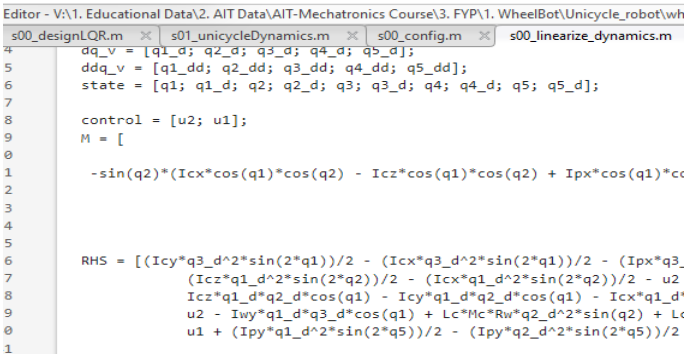
Unicycle Parameters Extracted from CAD Model.

Unicycle's Symbol	Description	Value & Unit
m_w	Ground wheel mass	0.015 kg
m_c	Mass of chassis	0.150 kg
m_r	Mass of reaction wheel	0.13 kg
r_w	Radius of the wheel	0.095m
l_c	Distance between COM of ground wheel and chassis	0.052 m
l_{cp}	Distance between COM of reaction wheel and chassis	0.052 m
l_p	Distance between COM of reaction wheel and joint of reaction wheel	0.00 m
g	Gravity acceleration	9.81 m/sec^2

In MATLAB one may find the file Unicycleconfig.m. The configuration is followed by the non-linear model definition and pseudo-linearization around the upright equilibrium position. Utilizing the MATLAB symbolic toolbox all the defined modeling was written in MATLAB code and linearized. the snippet from MATLAB code is attached (for full detail please refer to file unicycle_dynamics.m).

Figure 4.1

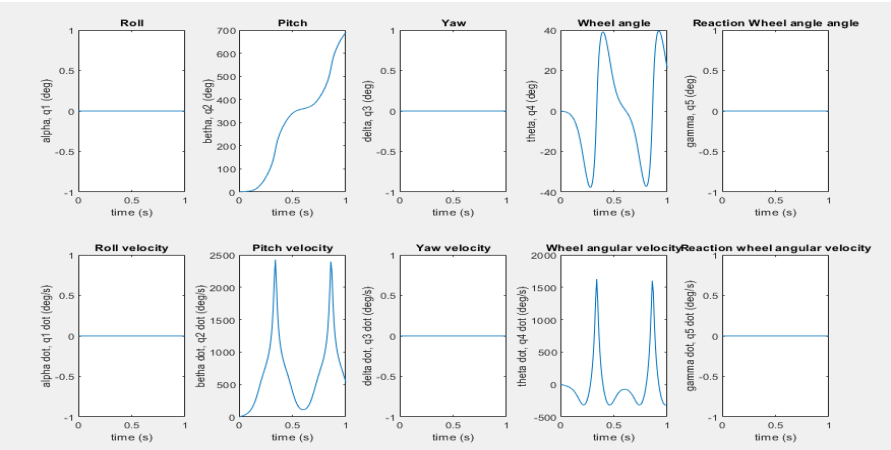
MATLAB Code Snippet for Dynamics



The implementation of the Lagrangian was done concerning energy and states. to assess the states and trajectory, the roll, pitch, and yaw states were collected and depicted in the results, as illustrated in Figure 4.2. The varying states of the robot corresponded to the desired trajectory, indicating satisfactory performance.

Figure 4.2

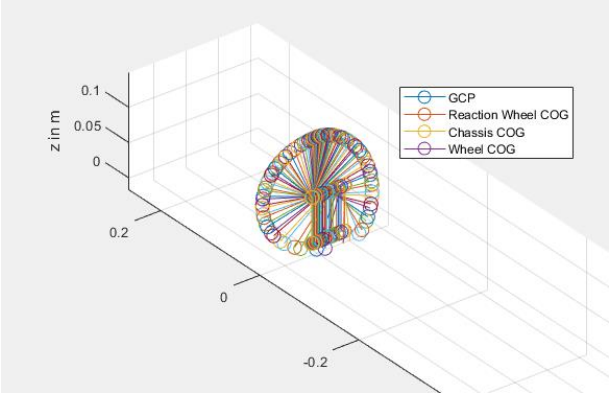
Unicycle States and Trajectory from the Dynamics



The center of gravity for joints over time is checked for ground wheel, chassis and reaction wheel.

Figure 4.3

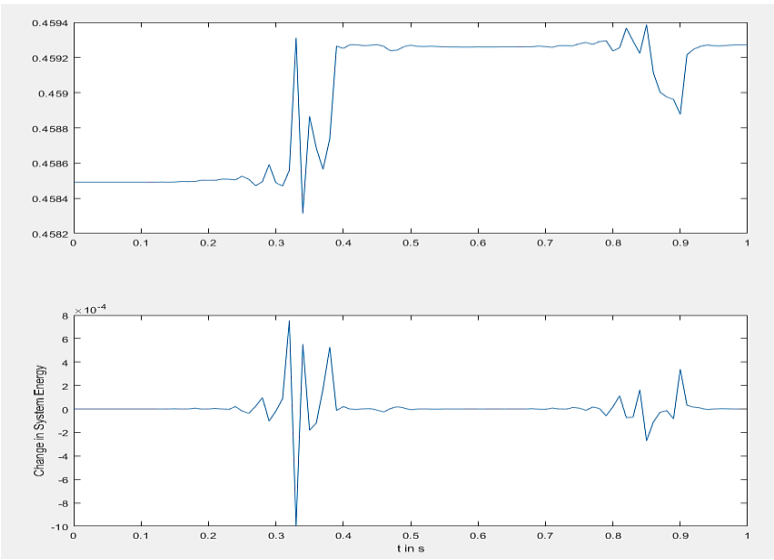
Joint Positions of Unicycle Over Time



The results for Lagrangian energy for this study prove that the derived model is correct, and the system conserves the energy. for example, this unicycle robot system exhibits smooth and bounded energy fluctuations over time. Some fluctuations are seen due to factors like friction drag and control inputs for this model.

Figure 4.4

Change in System Energy for Unicycle



The analytical results from the MATLAB validated the modeling and then to show the actual robot control in MATLAB, these results were further extended into the Simulink for controller design and state estimation.

4.2 Robot Stand-up Dynamics Proof-of-Concept.

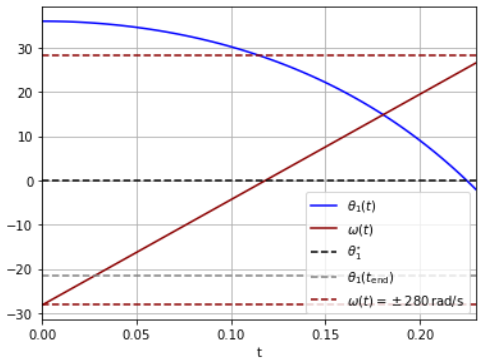
Considering the unicycle as symmetric, where both wheels can function as either rolling or reaction wheels, the mathematical model adopts a 2D inverse pendulum approach. Inspired by the renowned robot CUBLI, known for self-erecting onto its edges, the possibility of extending this capability to a unicycle robot is explored. However, transforming this concept into tangible hardware requires substantial resources, custom-designed software, and electronics to achieve simulated results.

The dynamics of the robot's self-erection maneuvers are depicted in Figure 4.5. During self-erection, one wheel accelerates until it contacts the ground. the dimensions of the robot are selected to minimize motor torque during stand-up. the cube-shaped center assembly facilitates self-erection by rotating around two points (C1 and C2).

The dynamics of stand-up mimic those of a 2D inverse reaction wheel pendulum, considering gravitational torque, motor torque, rotational inertia, and system inertia. Moreover, the image illustrates how sliding contact points reduce the required torque for self-erection. Overall, it highlights the intricate process of the robot balancing torques and optimizing motor performance during self-erection.

Figure 4.5

2D Reaction Wheel Concept Development



4.3 MATLAB Simulink Procedure

4.3.1 SIMSCAPE Model of Unicycle in MATLAB

The information provided in Table 4.1 served as the foundation for constructing the model in Simscape within MATLAB. Simscape is a powerful tool within the MATLAB environment, enabling the creation of models for physical systems that seamlessly integrate with the Simulink environment. Utilizing block diagrams for actuators, body frame, and wheel, a Simscape model was developed. While the complete details of the model design are available in the accompanying file.

Figure 4.6

Simscape Model of Unicycle

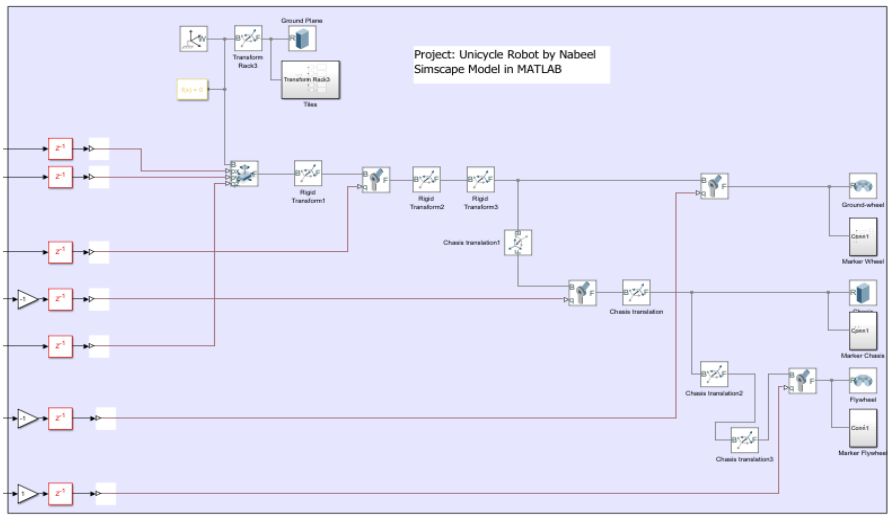
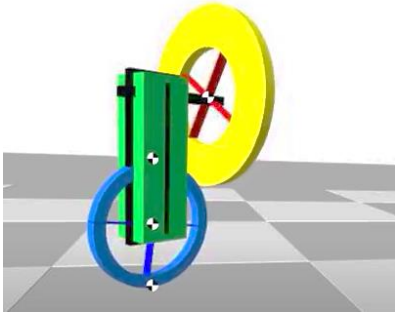


Figure 4.7

Physical Mode Shown in MATLAB



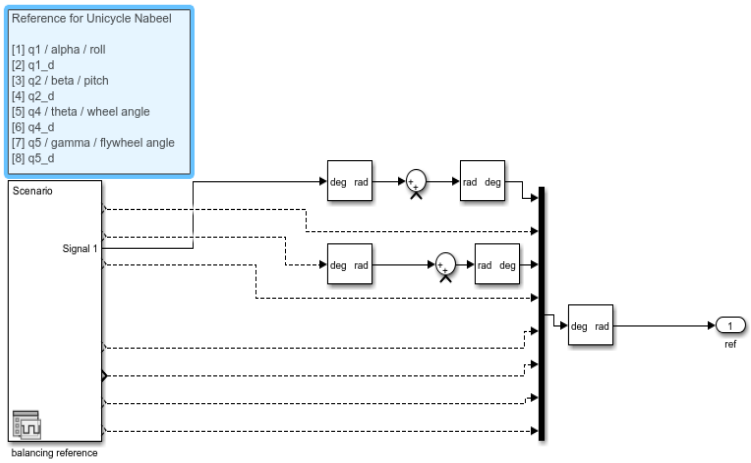
4.3.2 SIMULINK Configurations in MATLAB

4.3.2.1 **Reference Scenario for Unicycle.** To facilitate the operation of the Simscape model within MATLAB, several components are essential, including reference inputs, state estimation routines, and controller designs. As a preliminary step, a zero-state balancing reference is generated. The input comprises an 8x1 vector encompassing roll, pitch, wheel angle, reaction wheel angle, and their respective derivatives. initially, this reference corresponds to vectors of zeros for all states.

for the Simscape model in MATLAB, the need for reference, state estimation routines, and controller design is required. Therefore, first, a zero-state balancing reference is created, and the input is of an 8x1 vector consisting of roll, pitch, wheel angle, reaction wheel angle, and their derivatives respectively. initially, this reference is nothing but states of all zero vectors.

Figure 4.8

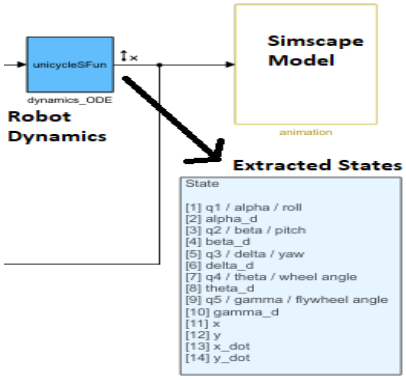
Reference Balancing Scenario



4.3.2.2 **State Estimation Routine.** The theory behind the state estimation routine for this unicycle has been discussed in Chapter 3 and the same implementation is done in Simulink. The body rates are converted into Euler coordinates and used for tilt and ray roll estimation using accelerometer and gyroscope respectively. All the states are real-time states coming from the dynamics of the unicycle to the state-estimation block these states are 14 in number:

Figure 4.9

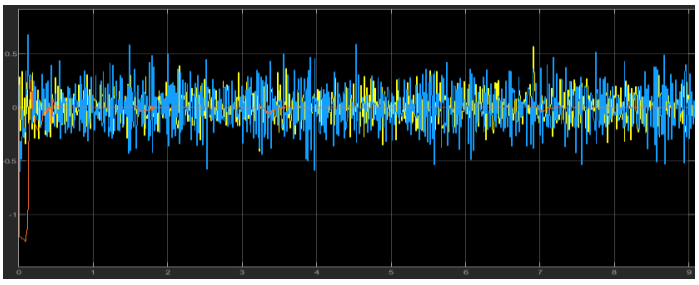
States From Robot Dynamics for Estimation Routine



These states from the model are fed into custom-designed estimation blocks in Simulink. Firstly, the random IMU data's perturbation is done with gaussian noise to show the IMU simulation as if it is the real system. This simulated IMU rates were combined with pivot point acceleration in inertial frame.

Figure 4.10

IMU-based and Estimated Acceleration of Pivot Point (Center of Ground wheel)

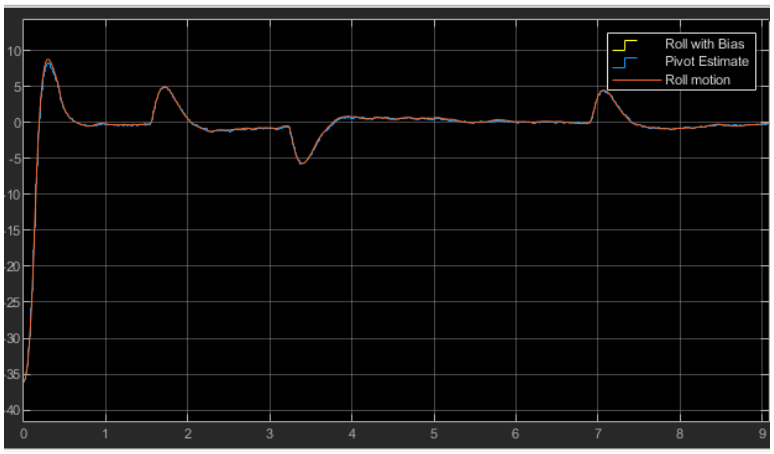


The final estimates for roll and pitch angle referenced from inertial to body frames are represented below.

For Roll angle, we have the following results.

Figure 4.11

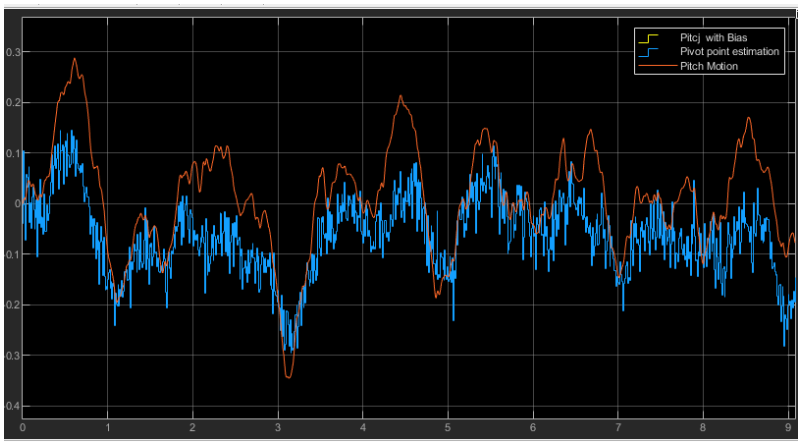
Estimate Routine for Roll angle.



For pitch angle, we have the following results.

Figure 4.12

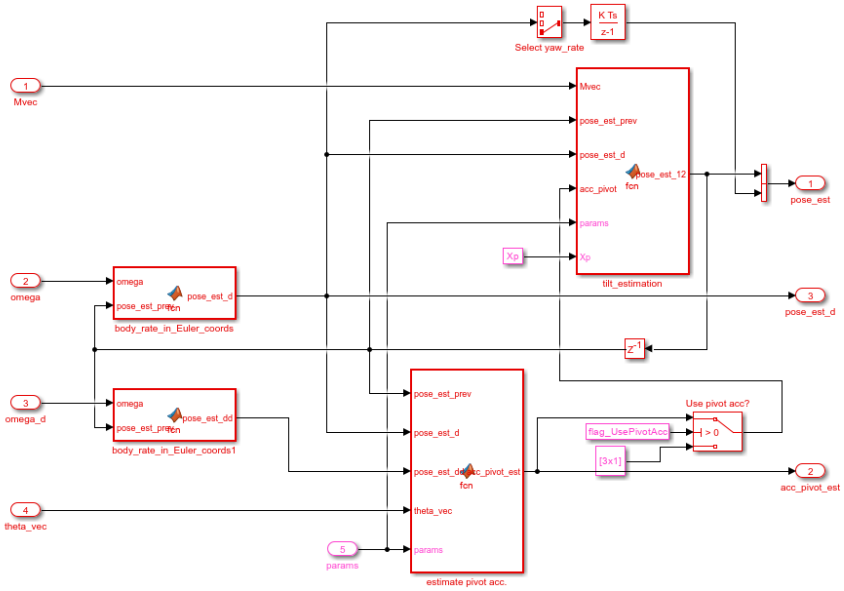
Estimate Routine for Pitch angle.



After the robot has enough information about its states. Only the real-time observable states were fed into the LQR controller. these states are the same as reference scenarios states.

Figure 4.13

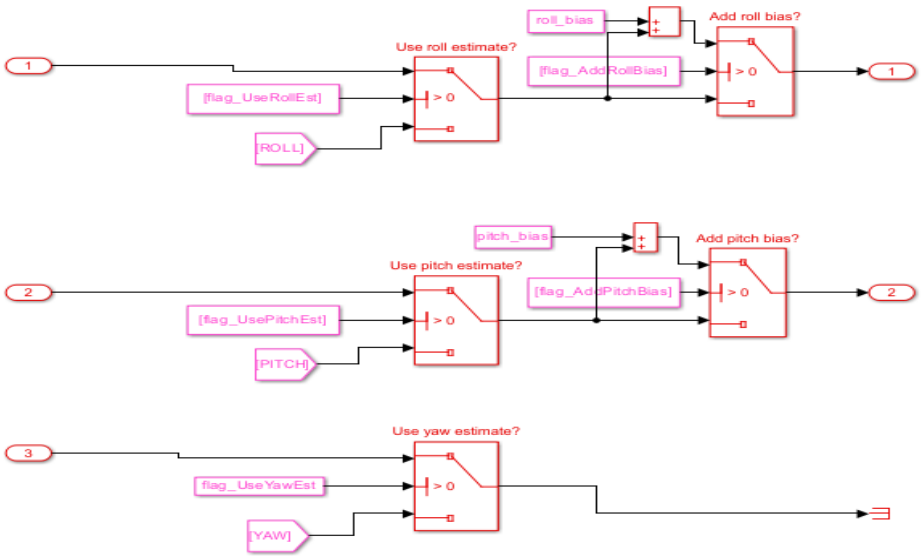
State Estimation Routine in MATLAB



The state estimation information is important for role and pitch control. Here the yaw dynamics are not implemented since as per the configuration of the unicycle, Yaw control is out of the scope of this project.

Figure 4.14

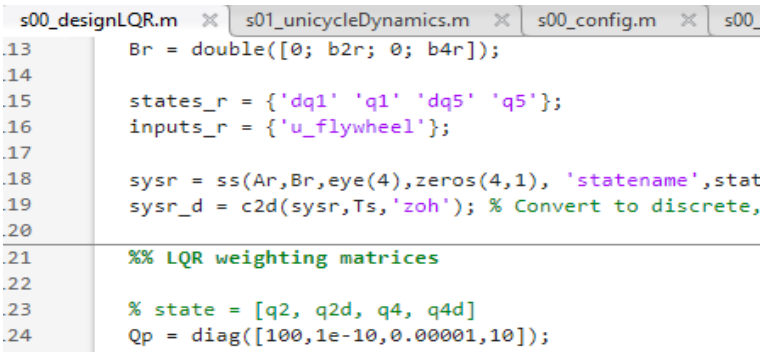
State Estimation Values input to Pitch and Roll



4.3.2.3 **Linear Quadratic Controller (LQR) Design.** To regulate the system, a linear quadratic control (LQR) is formulated, and the corresponding gains are computed. The outcomes obtained from MATLAB are illustrated in Figure 4.7. Through iterative adjustments, the semi-definite weighting matrices for the pitch and roll controllers in the LQR are meticulously selected. This process yields a diagonal matrix, as depicted below.

Figure 4.15

MATLAB code snippet for LQR Controller



For Pitch Control:

$$Q_p = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 1X10^{-10} & 0 & 0 \\ 0 & 0 & 0.00001 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \quad 135$$

$$Rp = [1000] \quad 136$$

the final gains for pitch controller are:

$$Kp = [-1.1779 \quad -0.1177 \quad -2.5929 * 10^{-5} \quad -0.0259\,] \quad 137$$

Feedforward term for pitch controller:

$$Mp = [-2.0965 * 10^{-4} \quad 2.0965 * 10^{-4} \quad -0.0183 \quad -0.0183\,] \quad 138$$

for Roll Control:

$$Q_r = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \quad 139$$

$$Rr = [1000] \quad 140$$

the final gains for roll controller found:

$$Kr = [-9.2484 \quad -1.0910 \quad -0.0083 \quad -0.0107\,] \quad 141$$

$$Mr = [-3.9414 * 10^{-6} \quad -3.9414 * 10^{-6} \quad -0.0111 \quad -0.0111\,] \quad 142$$

Linear quadratic controller is implemented in Simulink as discussed in the controller section of this chapter. It is designed to control the pitch and roll angle of the unicycle report which can be seen in Fig 4.16.

Figure 4.16

LQR Model in Simulink

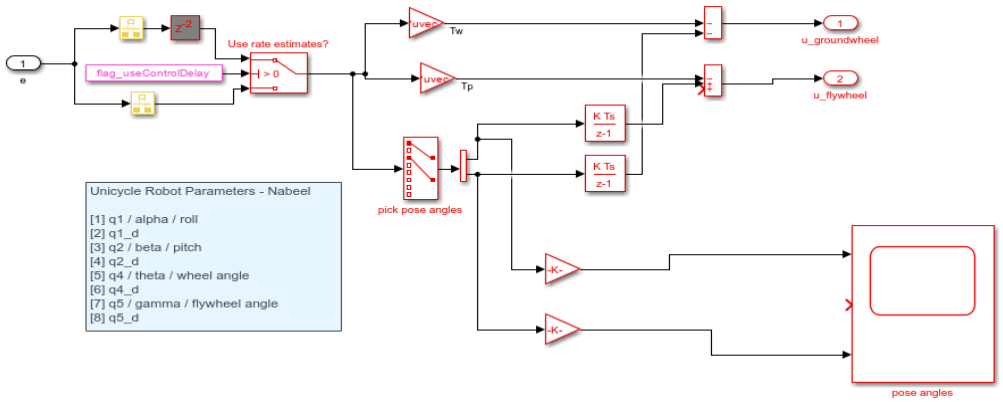


Figure 4.17 shows that unicycles can handle sufficient disturbances in the lateral direction.

Figure 4.17

Testing of Roll Control and with Disturbances

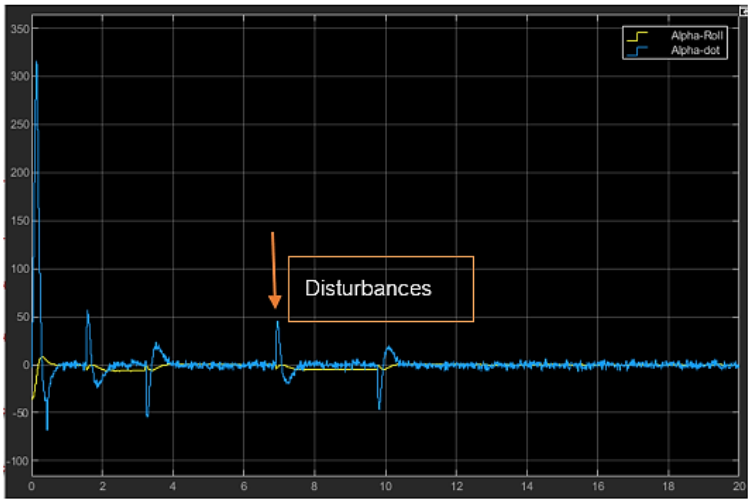
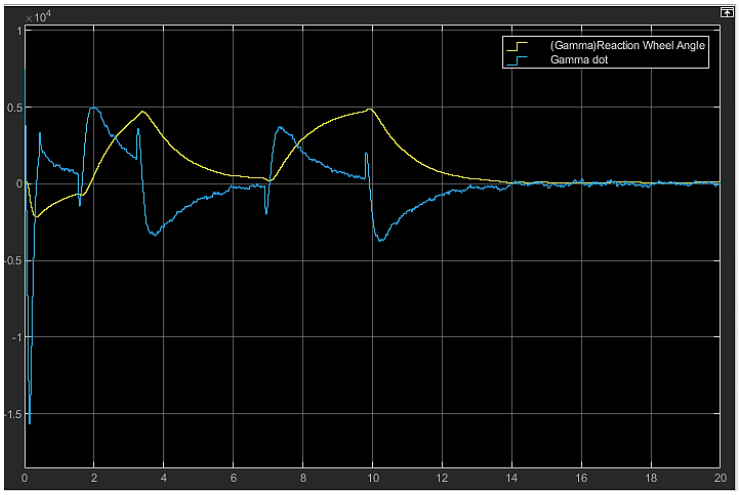


Figure 4.18

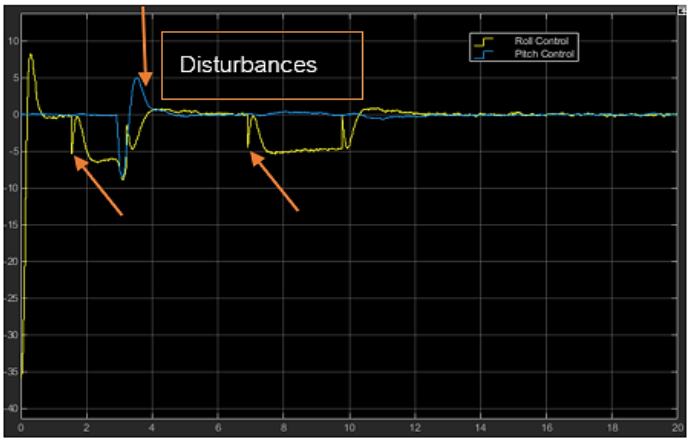
Reaction Wheel Control and Rate



The coordinated control of roll and pitch is presented in Figure 4.19 where along with disturbances unicycle gets in equilibrium under 12 sec settling time.

Figure 4.19

Roll and Pitch Control with Disturbances Before Integration.



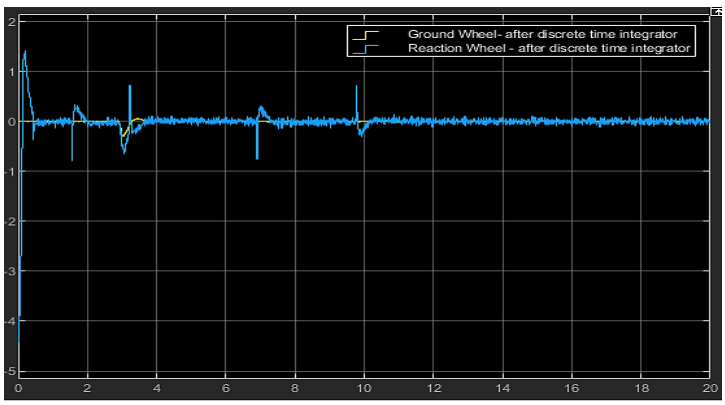
The discrete-time integrator block in Simulink can create a purely discrete model and state-space realization of the system in terms of output equations. for a given step of samples, the Simulink integration mode updates the system states as per the following scheme.

$y(n)\ and\ x\ (n+1)$

Where y is output, x is input, and n is the number of given steps for the LQR controller. The results are presented in Figure 4.19 along with the MATLAB video.

Figure 4.20

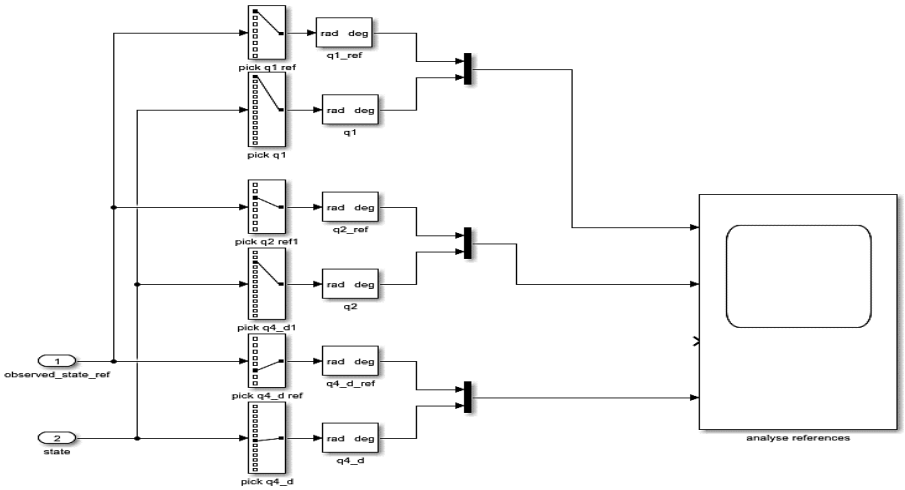
Roll and Pitch Control with Disturbances After Discrete-Time Integrator. (https://www.youtube.com/watch?v=fFfvmddeXDI)



***4.3.2.4*Tracking for Controller**. The real-time dynamics of the robot are tracked using the controller tracker designed by feeding the observed reference states and states from the dynamics of the unicycle robot.

Figure 4.21

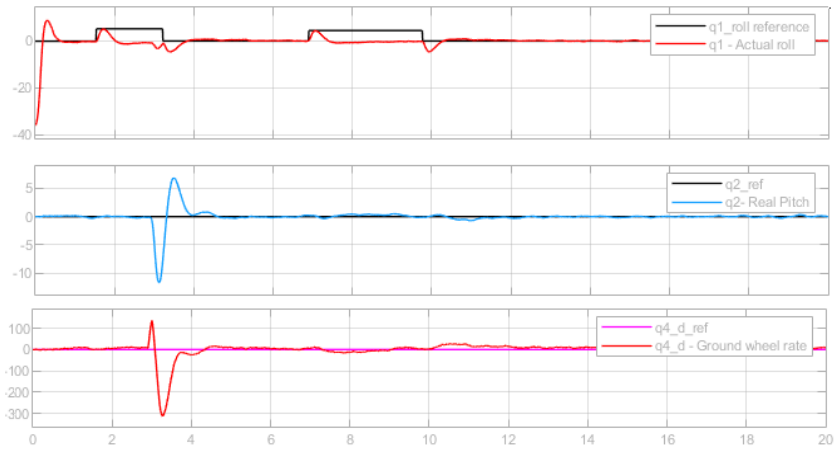
Controller Tracking Scheme for Unicycle in MATLAB Simulink



To check the error, a feedback system is implemented for controller tracking. the real states of the unicycle are observed with respective to reference states explained above and results are presented in Figure 4.22.

Figure 4.22

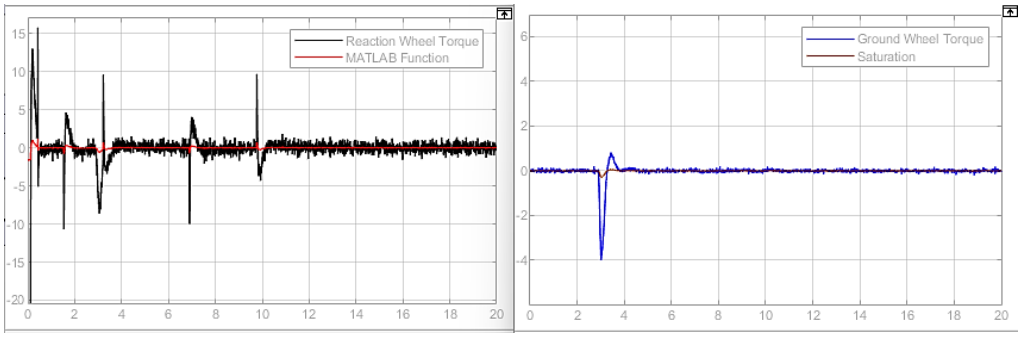
Real-States VS Reference States Controller Tracking



From system dynamics, reaction wheel and rolling wheel rates are extracted and as per selected actuator torque is presented in Figure 4.23.

Figure 4.23

Reaction and Ground Wheel Actuator Torque with Disturbance.



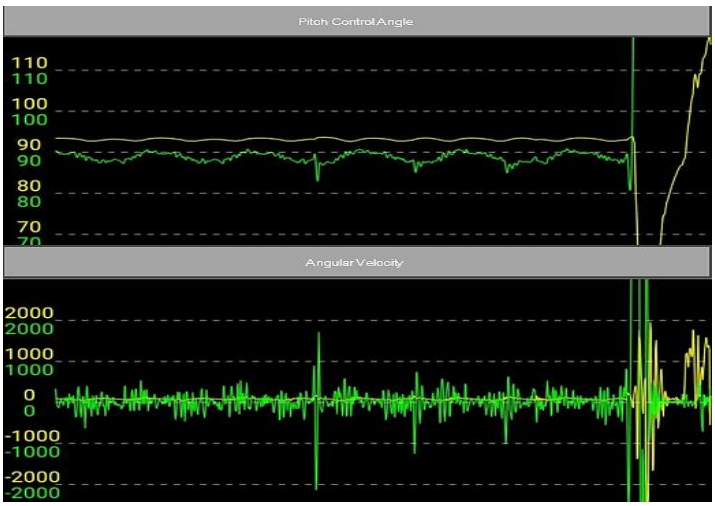
4.4 Testing on Real-Hardware Unicycle

The findings were confirmed through MATLAB simulations and subsequently validated via real-time control on the custom-designed unicycle robot employing the LQR controller. This involved the initial assembly of the hardware as detailed in Chapter 3, with the outcomes presented below.

Figure 4.24

Unicycle Hardware Pitch Control Real-Time Results from App

(https://youtube.com/shorts/SgrMcSsnuIk?feature=share)



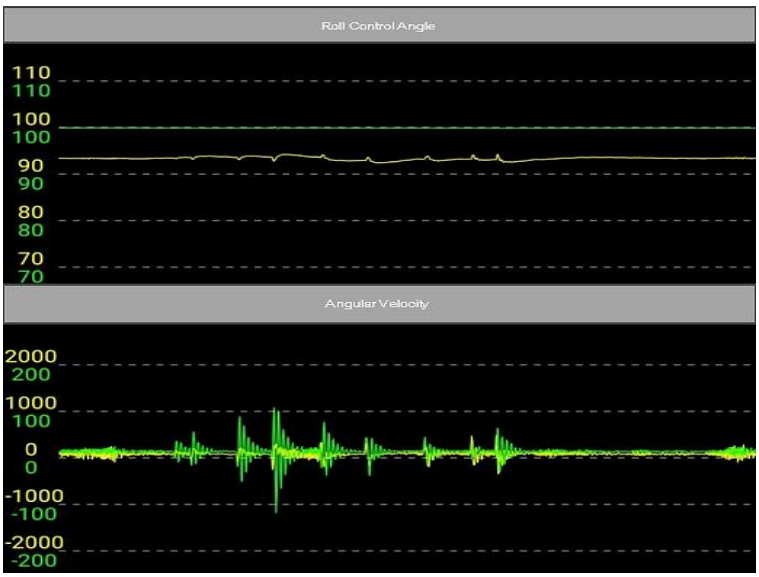
For Pitch Control, the real-time implementation on hardware (unicycle robot) is done to validate the required modeling and control. The unicycle is balanced and maneuverable independent of pitch control. The peaks shown in Figure 4.24 are the disturbances introduced in the video link. The supporting wheel is attached to balance the robot's pitch angle.

For roll control, it is successfully balanced and supporting bracket was placed to prevent unicycle moving forward and backward. Peaks shown in Figure 4.25 are the disturbances introduced shown in video link.

Figure 4.25

Unicycle Hardware Roll Control

(https://youtube.com/shorts/58sFqxdgJ6k?feature=share)



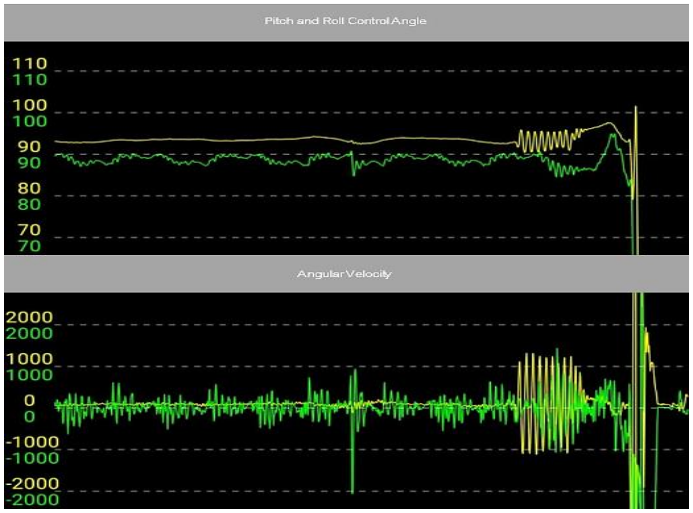
4.4.1 Combined Pitch and Roll Control

Firstly, the LQR was independently designed for pitch and roll control and then combined to check the coordinated control of both lateral and longitudinal motion of the unicycle. Sufficient disturbances are also introduced as can be seen in the video.

Figure 4.26

Unicycle Hardware Coordinated Control of Roll & Pitch Control

(https://youtube.com/shorts/vjdEADPw-SM?feature=share)



4.5 Unicycle Control Using Deep Reinforcement Learning

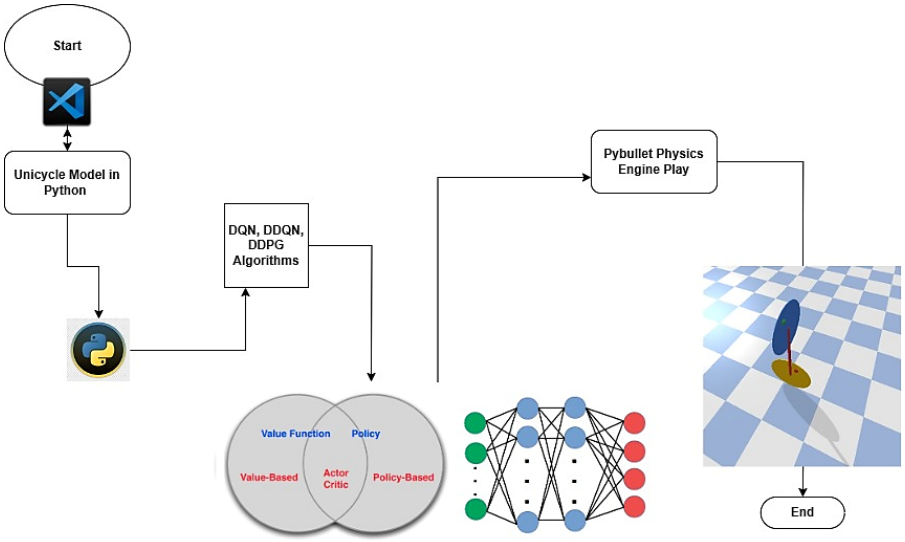
In Chapter 3, the methodology details the development of the final algorithm employed, namely DDPG (Deep Deterministic Policy Gradient), specifically tailored for use on unicycle robots. Additionally, other algorithms such as (DQN) and DDQN were implemented and evaluated. However, it was determined that DDPG exhibited the best performance for balancing the unicycle robot effectively.

4.5.1 Overview for the System:

As previously discussed, unicycles comprise three rigid bodies: the reaction wheel, chassis, and ground wheel. Like classical control, the same assumptions were applied for RL control, eliminating the need for system modeling. Instead, the system was simulated using a Physics Engine called Pybullet. the program's explanation is outlined as follows:

Figure 4.27

DRL Implementation Scheme in Python Environment



4.5.2 Unicycle Robot Model Creation in Python

Expanding upon the detailed description provided, the `unicycle_robot.py` file, accessible via the provided GitHub repository, offers comprehensive insights into the code structure and functionality of the unicycle class. This file contains meticulously annotated sections, guiding users through the implementation details and logic behind the unicycle model within the Pybullet simulation environment. Users can leverage this resource to gain a deeper understanding of the underlying mechanics and behaviors of the simulated robot.

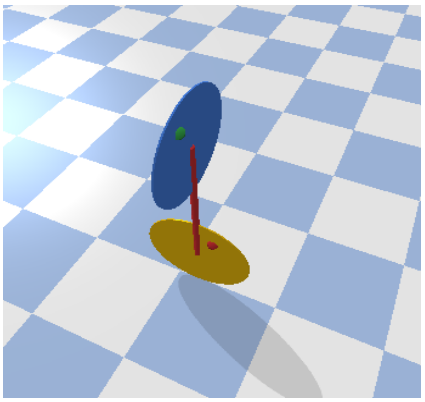
Furthermore, Figure 4.26 serves as a visual aid, showcasing the rendered representation of the unicycle model within the Pybullet simulation environment. This image provides a concrete visualization of the simulated robot, offering researchers and enthusiasts alike a glimpse into its design and characteristics.

Additionally, the development of specialized indicators for monitoring the ground wheel and reaction wheel movements adds a layer of practicality to the simulation setup. By visualizing the effects of applied torque on these components, researchers can better analyze the system's response to different control inputs and environmental conditions. Overall, the combination of detailed code explanations, visual

representations, and monitoring capabilities enhances the accessibility and utility of the simulation platform for studying unicycle robot dynamics and control strategies.

Figure 4.28

Unicycle Robot Model Created in Pybullet.



The key parameters for implemented mode l and scheme are presented in this table.

Table 4.2

Deep Learning Algorithm Parameters

Hyperparameters and Implementation Scheme	Descriptions
Environment	Pybullet unicycle balancing
State Space	Coordinates + quaternion
Action Space	torques
Actor-Network	2-Layer FC, tanh, linear
Critic Network	2-Layer FC, tanh, ReLU
Target Network Update	τ=0.001
Actor Learning Rate	$1e-4$
Critic Learning Rate	$1e-3$
the Discount Factor (γ)	0.99

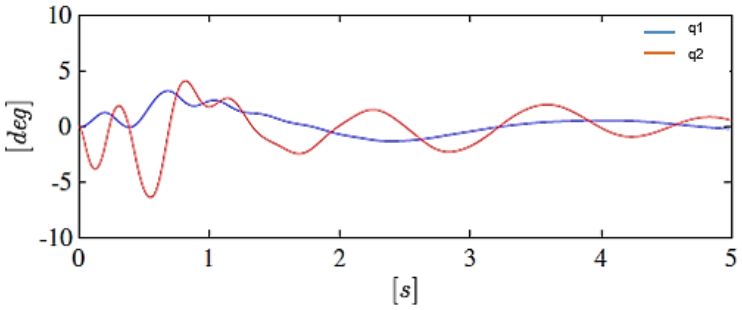
4.5.3 Results from Balancing Control

The policy gradient method used in reinforcement learning control is to optimize the policies concerning reward function. When the unicycle interacts with the environment designed in Pybullet, this creates the optimal policy. The controller performance for pitch and roll control is presented in Fig 4.28 along with the simulation video.

Figure 4.29

DRL Controller Performance for Pitch (q1) and Roll (q2) Control

(https://www.youtube.com/watch?v=C4y_fMJTPMQ)



4.6 Controller Performance Comparison (CC vs DRL)

It has been observed that both controller methods effectively balanced the unicycle in the upright equilibrium position, even when subjected to impulse disturbances in lateral and longitudinal directions. to comprehensively evaluate the performance of these two methods, this work proposes two testing techniques aimed at assessing the robustness of the controllers:

1. *Performance Matrix Test for Overshoot and Stabilization Time*: this test aims to measure the maximum overshoot and the time required for the unicycle to stabilize from its zero state. By quantifying these parameters, we can assess the effectiveness of each controller in stabilizing the unicycle.

2. *Test for Response to External Stimulus:* this test is designed to evaluate the robustness of the unicycle controller in response to external impulses. By subjecting the unicycle to various external stimuli, we can determine its maximum threshold

for sustaining such disturbances. This test provides insights into the resilience of the controller under real-world conditions.

By conducting these tests, we can gain a comprehensive understanding of the performance of both controller methods and make informed comparisons between them.

4.6.1 Test 1: Performance Matrix Test – Overshoot and Stabilization Time

To evaluate the performance of classical control (LQR) and DRL control (DDPG), the initial posture test is performed on a non-linear model with minimum disturbances to get the steady-state position for robot $[q1_0, q2_0]$ in simulation. This scenario is as close to as the unicycle in practice is fixed from two sides with the help of threads or strings before turning ON the controller simulation. When robots start falling once the strings are cut, the controller gets activated with a velocity set at 3(deg/sec). So, the initial test state is:

$$S_0 = [0 \quad 0 \quad q_{\dot{1}} \quad q_{\dot{2}} \quad q_1 \quad q_1] \qquad 143$$

$$q_{\dot{1}} = [0, \quad W_i + N1]$$
$$q_{\dot{2}} = [0, \quad W_i + N2]$$
$$q_1 = [q1_0 - N3, \quad W_i + N3]$$
$$q_2 = [q2_0 - N4, \quad W_i + N4]$$

robot $[q1_0, q2_0]$ = [roll, pitch initial position]

N1 – N4 is small threshold value in angle kept at 0.3 [deg].

The tested DOF is done as per equation 119 and resulting angular velocities are recorded. Due to large stabilization interval of roll angle the performance is evaluated at roll angle = 6 deg.

Figure 4.30

LQR Vs DDPG Stabilization From initial Pitch = 12 Deg

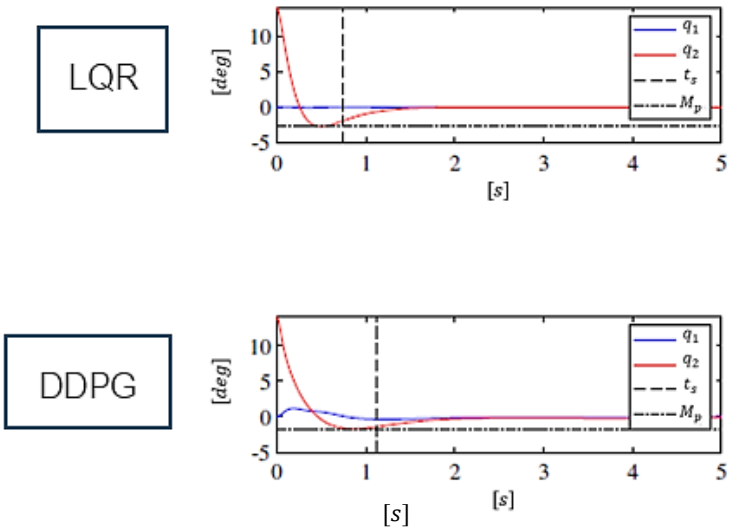


Table 4.3

Tests Result for Stabilization and Overshoot of LQR Vs DDPG

The conclusion for the time of stabilization and overshoot is presented in the table.

initial angle [deg]	Controller	Stabilization time [ts in sec]	Overshoot Mp [deg]
$q1_0 = 7$	LQR	1.01	4.70
$q1_0 = 7$	DDPG	1.83	8.20
$q2_0 = 12$	LQR	0.74	2.74
$q2_0 = 12$	DDPG	1.12	1.70

Using LQR from initial posture when learning sideward roll angle is:

$$q1_0 = 7\ deg$$

Table 4.4

Stabilization of Unicycle from Initial Posture When Roll Angle Is 7 Deg with LQR

Test No	Unicycle Stabilized	q_1 [deg]	$q_{\dot{1}}$ [deg/sec]	q_2 [deg]	$q_{\dot{2}}$[deg/sec]
1	Yes	7.12	3.186	0.0435	1.041
2	Yes	7.11	2.063	0.0624	0.775
3	Yes	7.07	2.740	-0.150	-0.962
4	No	7.01	3.211	-0.253	-0.260
5	Yes	7.05	2.34	-0.125	0.746

Using DDPG from initial posture when learning sideward roll angle is:

$$q1_0 = 7\ deg$$

Table 4.5

Stabilization of Unicycle from Initial Posture When Roll Angle Is 7 Deg with DDPG

Test No	Unicycle Stabilized	q_1 [deg]	$q_{\dot{1}}$ [deg/sec]	q_2 [deg]	$q_{\dot{2}}$[deg/sec]
1	Yes	7.11	3.340	-0.265	-0.358
2	No	7.02	3.839	-0.408	-1.663
3	Yes	7.15	3.012	-0.091	0.365
4	Yes	7.10	3.438	-0.046	0.716
5	No	7.01	3.285	-0.049	0.264

Using LQR from initial posture when learning forward Pitch angle is:

$$q2_0 = 12\ deg$$

Table 4.6

Stabilization of Unicycle from Initial Posture When Pitch Angle Is 12 Deg for LQR

Test No	Unicycle Stabilized	q_1 [deg]	$q_{\dot{1}}$ [deg/sec]	q_2 [deg]	$q_{\dot{2}}$[deg/sec]
1	Yes	12.22	3.067	-0.005	1.314
2	No	12.96	3.191	-0.392	-0.662
3	No	12.16	4.062	-0.064	-1.743
4	Yes	12.24	4.109	-0.575	-4.089
5	No	12.91	4.325	-0.037	-1.039

Using DDPG from initial posture when learning forward Pitch angle is:

$$q2_0 = 12\ deg$$

Table 4.7

Stabilization of Unicycle from Initial Posture at Pitch Angle Is 12 Deg for DDPG

Test No	Unicycle Stabilized	q_1 [deg]	$q_{\dot{1}}$ [deg/sec]	q_2 [deg]	$q_{\dot{2}}$[deg/sec]
1	Yes	12.96	3.844	0.303	-2.43
2	Yes	12.29	4.813	-0.091	-1.12
3	No	12.023	3.512	-0.123	0.272
4	Yes	12.10	3.224	-0.123	3.224
5	No	12.27	3.887	-0.275	-2.142

4.6.2 Test 2: Test for Response to External Stimuli

This test can be applied by subjecting the unicycle to quantified impulse responses both through simulation and in real-time operation. one method involves simulating the impulse response in a controlled environment, while the other entails conducting physical experiments using a mechanism such as a ball hanging from a ceiling.

in the practical setup, the ball serves as the impulse source, which can be conceptualized as a rate of momentum transfer (Reza N. Jazar, 2011). The test setup for the practical case is illustrated in Fig 4.31. When the ball strikes the unicycle robot at a specific time, it imparts an impulse to the system, which can be quantified using the impulse function. This impulse function can be used to evaluate the unicycle's response to external stimuli and determine its ability to withstand such disturbances. Through careful analysis of the system's response to these impulses, valuable insights into the robustness and resilience of the controller can be obtained.

$$J_m = \int_{t1}^{t2} F dt \approx p = m_b v_2 \ = m_b v_1 \qquad 144$$

The impulse (J_m) resulting from the impact of the ball can be calculated based on the momentum (p), where (m_b) represents the mass of the ball hitting the unicycle at time (t_1), with velocities (v_1) and (v_2) from initial to final positions, respectively.

The typical velocity at impact can be determined using the law of conservation of energy. According to this law, when the ball possesses no velocity before dropping from a height (h), the velocity at impact can be calculated as:

$$[\, v \ = \ \sqrt{\{2gh\}}] \qquad 145$$

Where, (g) is the acceleration due to gravity.
This calculation enables the determination of the velocity of the ball at the moment of impact, which is crucial for assessing the magnitude of the impulse imparted to the unicycle.:

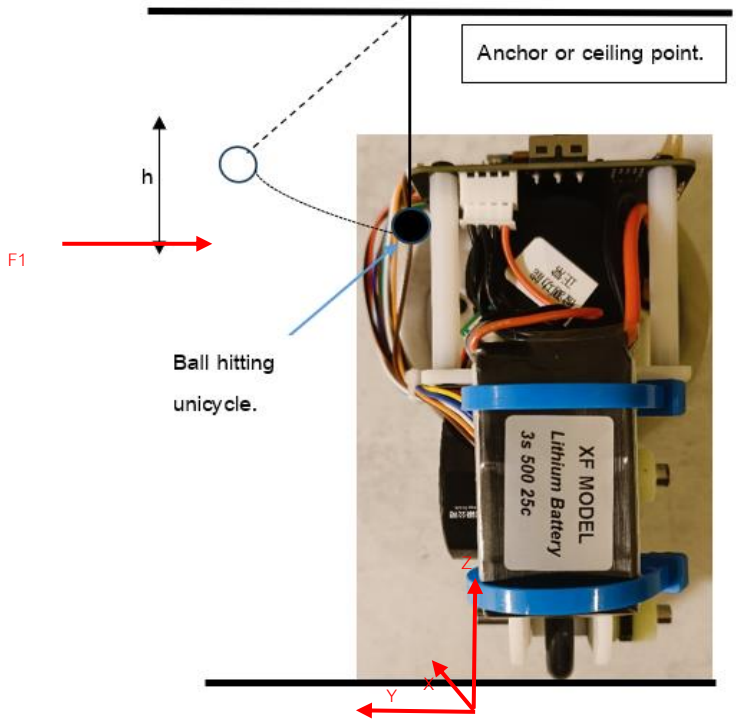
$$m_b g h \ = \frac{1}{2}(m_b v_1^2) \qquad 146$$

The test is executed following a procedure akin to the initial posture setup, whereby the control algorithm is triggered upon the ball's impact on the robot, accompanied by a set angular velocity of 3 degrees per second.

In the simulation study, additional inputs in the form of external stimuli are introduced to the system to enhance its versatility and robustness. this incorporation of supplementary inputs is seamlessly integrated into MATLAB's Simulink environment, as illustrated in Figure 4.30. Through this configuration, researchers can simulate a wide range of scenarios, thereby facilitating a comprehensive assessment of the unicycle robot's behavior and performance under diverse external conditions. Such simulations enable researchers to study the system's response to various stimuli, providing valuable insights into its dynamics and control strategies.

Figure 4.31

Test Setup on Proposed Unicycle for External Stimuli.

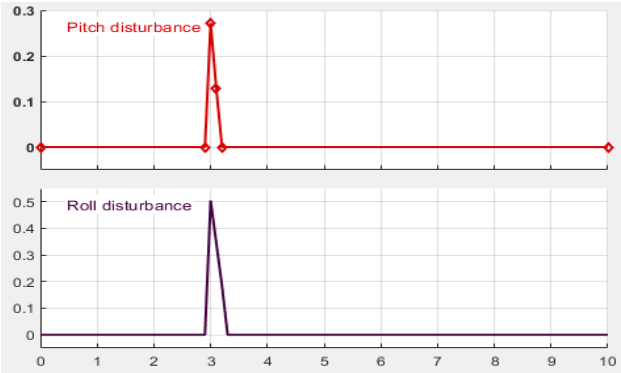


for reinforcement study, the linear force is considered at time t, and the impulse is calculated as follows.

$$J_{impluse} = F_i * \mathrm{Q}3t \qquad 147$$

Figure 4.32

Impulse Stimuli in MATLAB Simulink



The combined results from MATLAB and Python indicate that the LQR controller exhibits greater resilience to impulse forces in the pitch direction compared to the DDPG algorithm. However, in the roll direction, both controllers demonstrate a similar capability in handling impulse forces, with comparable magnitudes. the findings are summarized in the table below: the results are shown in Table below.

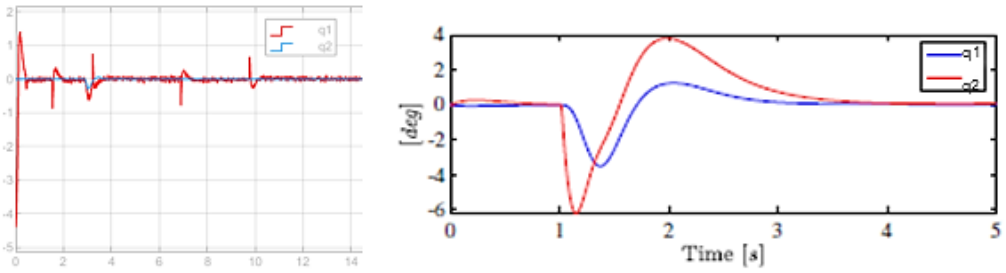
Table 4.8

Test on External Stimuli Response for LQR Vs DDPG

Controller	Sustained Pitch Impulse [Ns]	Sustained Roll Impulse [Ns]
LQR	0.48	0.37
DDPG	0.19	0.2

Figure 4.33

Matlab (LQR – Left Image) vs Python (DDPG – Right Image) Response to Impulses



4.7 Discussion for Results and Comparative Analysis of RL Vs CC

Firstly, it's crucial to recognize that the performance of these controllers is intricately tied to the fine-tuning of their parameters and weight matrices. Even slight variations in these configurations can lead to significantly different outcomes for each controller. Addressing the challenges of implementation, it becomes evident that applying the DDPG controller on hardware presents notable difficulties, primarily due to the inherent instability of the unicycle robot. This instability poses a significant hurdle, leading to unexpected behaviors, especially in the longitudinal direction, where the DDPG controller may converge to local optima, resulting in unforeseen responses.

However, there are noteworthy observations regarding DDPG's behavior. While it may exhibit unpredictability in certain scenarios, particularly longitudinal movements, it tends to adopt a more conservative approach in lateral stabilization. This cautious stance results in the stabilization of the system with smaller maximum pitch angles, offering a level of stability in lateral movements.

In terms of performance metrics, it's important to consider both the speed of stabilization and the degree of overshoot. LQR typically achieves faster stabilization, whereas DDPG displays less overshot. However, the longer settling time observed with DDPG can be attributed to the inherent exploration-exploitation trade-off characteristic of reinforcement learning algorithms. DDPG's need to explore the action space to learn optimal policies often results in suboptimal control actions during the learning phase, prolonging the stabilization process.

Successful training with DDPG hinges on the robust definition of state spaces. Without clearly defined boundaries for these spaces, the performance of the controller cannot be guaranteed. Nevertheless, DDPG offers notable advantages, such as utilizing experience buffers and off-policy learning, setting it apart from other reinforcement learning methods like PPO.

In practical terms, the choice between LQR and DDPG depends on various factors, including the desired performance metrics and the constraints imposed by the hardware architecture. While DDPG may demand higher computational resources, its ability to learn directly from sensor data without the need for explicit system modeling can be advantageous. Ultimately, the decision between classical control algorithms and reinforcement learning approaches must be carefully considered based on specific requirements and system constraints.

CHAPTER 5

CONCLUSION AND FUTURE RECOMMENDATION

5.1 Thesis Conclusion

In conclusion, this thesis has made significant contributions to the field of robotics, particularly in the design and control of unicycle robots. The research presented two distinct versions of the unicycle robot design, emphasizing optimization and component selection to achieve a practical laboratory-scale robot.

The control schemes devised for this project were divided into continuous and discrete control categories. for continuous control tasks such as balancing and maneuvering, two approaches were explored: the classical control-based Linear Quadratic Regulator (LQR) and the reinforcement learning-based (DDPG). Both controllers were tested extensively in MATLAB and Pybullet simulations, with successful transfer of the simulated training scheme to physical hardware, demonstrating effective balancing and control capabilities.

In addition to continuous control, the thesis also proposed a proof-of-concept model for discrete control tasks, such as stand-up or jump-up dynamics, using a 2D inverted pendulum approach. This model offers unique features akin to the Cubli robot, further expanding the test platform for both linear and non-linear control strategies.
Furthermore, two comprehensive testing procedures were developed to evaluate the performance of the controllers: the performance matrix and response to external stimuli. the performance matrix analysis provided insights into settling time and overshoot for both LQR and DDPG controllers, while the response to external stimuli demonstrated the robustness of the unicycle robot against impulse disturbances in simulation and real-world scenarios.

Overall, the results indicate that the LQR controller outperformed DDPG in most dynamic scenarios where hardware limitations were not a factor, exhibiting greater robustness to external stimuli. These findings offer valuable guidance for future research and serve as a reference point for comparing the capabilities of classical and reinforcement learning control methods in similar robotic systems.

5.2 Future Recommendations

Looking ahead, while the proposed unicycle robot demonstrated successful balancing and maneuvering capabilities in both pitch and roll directions using classical and reinforcement learning (RL) control schemes, there remains room for improvement, particularly in the RL domain. one key area for future development lies in enhancing the process of transferring learned policies from simulation to real-world hardware.

To address this challenge, a two-step approach can be adopted. Firstly, the reinforcement learning algorithm can be trained extensively in simulation environments, allowing for rapid exploration, and learning without the constraints of physical hardware. Once a satisfactory policy is obtained in simulation, the second step involves fine-tuning and adaptation of the learned policy on the actual hardware platform. This iterative process of simulation-based training followed by real-world fine-tuning can help bridge the gap between virtual and physical environments, enabling more seamless deployment of RL-based control strategies on real robotic systems.

Moreover, future research endeavors could explore the integration of both continuous and discrete control hardware designs within the same robotic platform. by combining the advantages of continuous control for tasks such as balancing and maneuvering with the capabilities of discrete control for specialized tasks like stand-up or jump-up dynamics, a more versatile and adaptable robotic system can be realized. this hybrid approach could offer enhanced performance and flexibility across a wider range of applications and operating conditions.

Overall, the proposed recommendations aim to further advance the capabilities of unicycle robots, paving the way for more efficient and robust control strategies that can be seamlessly deployed on real-world hardware platforms. By leveraging the strengths of simulation-based training and integrating both continuous and discrete control designs, future developments in this field hold great promise for achieving even greater levels of autonomy and functionality in robotic systems.

In the current orthogonal setup of a reaction wheel unicycle robot, the focus has predominantly centered on the independent control of pitch (longitudinal) and roll (lateral) directions, as evidenced by simulation results. Consequently, the treatment of yaw dynamics has been deferred for future consideration. This decision is sensible for

several reasons. Firstly, addressing yaw dynamics would introduce additional complexity beyond the current scope of research and development efforts. The introduction of yaw control would necessitate the development of sophisticated control strategies to ensure stability, potentially adding further layers of complexity to the system. It's essential to ensure the robot remains within the linear region (if linear controller is concerned) when leaning for particularly yaw dynamics. By prioritizing lateral dynamics control initially, researchers can lay a solid foundation for future investigations into yaw dynamics by introducing curvature control along with leaning angle of the robot to control. Future research could explore further possibilities to maximize the robot's potential as an advancement of this work, which already provides comprehensive reinforcement learning implementation.

REFERENCES

A. Schoonwinkel, "*Design and Test of a Computer Stabilized Unicycle*," Ph.D. dissertation, Stanford University, Stanford, CA, USA, 1987.

P. Fankhauser and C. Gwerder, "*Modeling and control of a ballbot*," ETH Z¨urich, Tech. Rep., 06 2010.

U. Nagarajan, G. Kantor, and R. Hollis, "*The ballbot: An omnidirectional balancing mobile robot*," the International Journal of Robotics Research, vol. 33, no. 6, pp. 917–930, May 2014.

V. Klemm, A. Morra, C. Salzmann, F. Tschopp, K. Bodie, L. Gulich, N. Kung, D. Mannhart, C. Pfister, M. Vierneisel, F. Weber, R. Deuber, and R. Siegwart, "*Ascento: A two-wheeled jumping robot*," in International Conference on Robotics and Automation. IEEE, May 2019, pp. 7515–7521.

G. Belascuen and N. Aguilar, "*Design, modeling and control of a reaction wheel balanced inverted pendulum*," in 2018 IEEE Biennial Congress of Argentina (ARGENCON). IEEE, 2018, pp. 1–9.

M. Gajamohan, M. Muehlebach, T. Widmer, and R. D'andrea, "*The cubli: A reaction wheel based 3d inverted pendulum*," European Control Conference, pp. 268–274, 2013.

X. Xiong and A. Ames, "*Slip walking over rough terrain via h-lip stepping and backstepping-barrier function Inspired quadratic program*," IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 2122–2129, 2021.

M. Muehlebach and R. D'andrea, "*Nonlinear analysis and control of a reaction-wheel-based 3-d inverted pendulum*," Transactions on Control Systems Technology, vol. 25, no. 1, pp. 235–246, Jan 2017.

D. W. Vos, "*Nonlinear control of an autonomous unicycle robot: practical issues*," Ph.D. dissertation, MIT, 1992.

G. P. Neves and B. A. Ang´elico, "*A discrete lqr applied to a self-balancing reaction wheel unicycle: Modeling, construction, and control*," in 2021 American Control Conference, 2021, pp. 777–782.

Sheng, Z., & Yamafuji, K. (1995b). Study on the Stability and Motion Control of a Unicycle: Part I: Dynamics of a Human Riding a Unicycle and Its Modeling by

Link Mechanisms. JSME international journal. Ser. C, Dynamics, control, robotics, design, and manufacturing, 38(2), 249-259.

Majima, S., & Kasai, T. (2005). A controller for changing the yaw direction of an underactuated unicycle robot. From http://icta05.teithe.gr/papers/27.pdf.

Schoonwinkel, A. (1987). Design and test of a computer-stabilized unicycle. Stanford University, Stanford, California U.S.A.

Minh-Quan, D., & Kang-Zhi, L. (2005). Gain-scheduled stabilization control of a unicycle robot. JSME International Journal, 48(No. 4), 649-656.

Caldecott et al., Modelling, simulation, and control of an electric unicycle. in Conference on Robotics and Automation, Brisbane, Australia., vol. 13; 2010.

Surachat Chantarachit, Balancing Control of a Unicycle, bachelor's Dissertation, 2011, AIT, Thailand.

Surachat Chantarachit, the development and Control of a Unicycle Robot, master's Thesis, 2017, AIT, Thailand.

Abhisesh Silwal, Design of a self-balancing unicycle, master's thesis, 2012, AIT, Thailand.

Buddhika Lakmal Aththanayaka, Dynamic modeling of a multi-body unicycle in three-dimensional space, Master Thesis, Kansas State University, Manhattan, 2022.

Guo et al., (2022) Design and dynamic analysis of jumping wheel-legged robots in complex terrain environment. Doi: 10.3389/fnbot.2022.1066714

Mathias Hofer, M. Muehlebach, and R. D'andrea, "The one-wheel Cubli: A 3D inverted pendulum that can balance with a single reaction wheel," Elsevier: Journal of Mechatronics, 2023.

M. Giftthaler, M. Neunert, M. St¨auble, J. Buchli, and M. Diehl, "A family of iterative gauss-newton shooting methods for nonlinear optimal control,"(IROS). IEEE, 2018, pp. 1–9.

Ekaterina Abranova, "Combining Reinforcement Learning and Optimal Control for the Control of Nonlinear Dynamical Systems" Doctoral Dissertation, Imperial College London, May 2015.

R. Batke, F. Yu, J. Dao, J. Hurst, R. L. Hatton, A. Fern, and K. Green, "Optimizing bipedal maneuvers of single rigid-body models for reinforcement learning," in 2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids), 2022, pp. 714–721.

Kwok, C., Fox, D. (2004). Reinforcement learning for sensing strategies. Reference Journal Volum 4:3158–3163.

Kaizad V. Raimalwala, Bruce A. Francis and Angela P. Schoellig "A Preliminary Study of Transfer Learning between Unicycle Robots" University of Toronto, 2015.

G. Belascuen and N. Aguilar, "Design, modeling and control of a reaction wheel balanced inverted pendulum," in 2018 IEEE Biennial Congress of Argentina (ARGENCON). IEEE, 2018, pp. 1–9.

V. Klemm et al., "Ascento: A two-wheeled jumping robot," in International Conference on Robotics and Automation. IEEE, May 2019, pp. 7515 7521.

S. Trimpe and R. D'andrea, "The balancing cube: A dynamic sculpture as a test bed for distributed estimation and control," IEEE Control Systems, vol. 32, no. 6, pp. 48–75, Dec 2012.

M. Gajamohan, M. Muehlebach, T. Widmer, and R. D'andrea, "The cubli: A reaction wheel based 3d inverted pendulum," European Control Conference, pp. 268–274, 2013.

P. Fankhauser and C. Gwerder, "Modeling and control of a ballbot," ETH Z¨urich, Tech. Rep., 06 2010.

J. Lee, S. Han, and J. Lee, "Decoupled dynamic control for pitch and roll axes of the unicycle robot," Transactions on Industrial Electronics, vol. 60, no. 9, pp. 3814–3822, Sep 2013.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... & Wierstra, D. (2016). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen, S. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540),

APPENDIX

PROJECT FILES AND CODES

1. For the Solidworks CAD File – the link is:
   (https://github.com/nabeeljadoon/Unicycle_DesignCAD/tree/main)
2. For LQR: All the MATLAB files and Codes are on GitHub
(https://github.com/nabeeljadoon/Unicycle-Balancing-Control/tree/main)
3. For DDPG: Python Code is also uploaded on GitHub
(https://github.com/nabeeljadoon/Unicycle-Balancing-Using-DRL)

VITA

Nabeel Ahmad Khan Jadoon is currently pursuing a master's degree in the Mechatronics and Machine intelligence (MMI) program at the Asian institute of Technology, situated within the School of Engineering and Technology. His academic journey is further enriched by his participation as a JASSO laureate in the Young Science and Engineering Program (YSEP) at the prestigious Tokyo institute of Technology (TokyoTech). within TokyoTech's Department of Control System and Engineering, he is actively engaged in exploring cutting-edge research and innovations.

Driven by a passion for engineering, Nabeel Ahmad Khan Jadoon embraces a philosophy of striving for excellence and impactful achievements. His academic pursuits and professional endeavors reflect a commitment to pushing boundaries and making meaningful contributions to the field of industrial system engineering. With a steadfast determination to excel, he embodies the ethos of being an achiever rather than merely a doer, continuously seeking new opportunities for growth and advancement in his chosen field.