Adaptive Torque Control of a Novel 3D-Printed Humanoid Leg

Philip J. Hancock

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Mechanical Engineering

Alexander Leonessa, Chair
Kaveh Akbari Hamed
Alan T. Asbeck

July 6, 2020
Blacksburg, Virginia

Keywords: Adaptive Control, Impedance Control, Robotics, Force Control, Humanoid, Compliance, 3D Printed

Adaptive Torque Control of a Novel 3D-Printed Humanoid Leg

Philip J. Hancock

(ABSTRACT)

In order to function safely in a dynamic environment with humans and obstacles, robots require active compliance control with force feedback. In these applications the control law typically includes full dynamics compensation to decouple the joints and cancel out nonlinearities, for which a high-fidelity model of the robot is required. In the case of a 3D-printed robot, components cannot be easily modeled due non-uniform densities, inconsistencies among the 3D printers used in manufacturing, and the use of different plastics with mechanical properties that are not widely known. To address this issue, this thesis presents an adaptive control framework which modifies the model parameters online in order to achieve satisfactory tracking performance. The inertial properties are estimated by adapting with respect to functions of the unknown parameters. This is achieved by rewriting the robot dynamics equations as the product of a matrix of known nonlinear functions of the joint states and a vector of constant unknowns. The result is a nonlinear system linearly parameterized in terms of the of the unknowns, which can be estimated using adaptation laws derived from Lyapunov stability theory. The proposed control system consists of an outer-loop impedance controller to regulate deviations from the nominal trajectory in the presence of disturbances, and an inner loop force controller to track the joint torques commanded by the outer-loop. The proposed system is evaluated on an early prototype consisting of a 3DOF leg, and two actuator test setups for the low-level controller.

Adaptive Torque Control of a Novel 3D-Printed Humanoid Leg

Philip J. Hancock

(GENERAL AUDIENCE ABSTRACT)

In order to function safely in a dynamic environment with humans and obstacles, a robot must be able to actively control its interaction forces with the outside environment. In these applications a high-fidelity model of the robot is required. In the case of a 3D-printed robot, the components in the robot cannot be easily modeled due non-uniform densities, inconsistencies among the 3D printers used in manufacturing, and the use of different plastics with mechanical properties that are not widely known. To address this issue, this thesis presents an adaptive control framework which actively modifies the model parameters in order to achieve satisfactory tracking performance. In this work, the equations of motion of the robot are manipulated in such a way that the unknown quantities are separated from the known quantities. The unknowns are updated in real time using adaptive laws derived from Lyapunov stability theory. The proposed control system consists of a high-level torque controller to regulate deviations from the nominal trajectory, and a low-level force controller to track the joint torques commanded at the high-level. The proposed system is evaluated on an early prototype of the robot consisting of a 3 degree of freedom leg, and two actuator test setups for the low-level controller.

Acknowledgments

I would like to thank my advisor, Dr. Alexander Leonessa, for his enthusiasm, patience, and invaluable advice. I am extremely grateful to Dr. Leonessa for giving me the opportunity to work in the TREC lab and for taking the time to provide feedback and guidance whenever needed, and for ensuring that I could still finish my work while the lab was shut down due to the COVID-19 pandemic. I would also like to thank my lab mates for all their help. I extend my special thanks to Bo Pang and Connor Herron for helping me with the electronics, to Alex Fuge for his work on the robot, and to Ben Beiter for his advice regarding dynamics. I would also like to thank Dr. Garret Burks for helping me with Overleaf and giving me general advice for graduate school.

I want to thank my family for their endless love and support. I thank my Mom and Dad for providing me the means to attend engineering school, for their constant encouragement and patience through the years, and for being so actively involved in my life. I also thank my two brothers for their encouragement.

Lastly, I want to thank my dear Soyoung for always being by my side during my time in graduate school and for having the utmost confidence in me. I could not have gotten through this without her support and positive influence along the way.

Contents

List of Figures

List of Tables

List of Abbreviations

ADP-PI Adaptive Augmented Proportional Integral

AIVSDE Adaptive Integral Variable Structure Derivative Estimator

DOF Degree of Freedom

MRAC Model Reference Adaptive Control

THOR Tactical Hazardous Operations Robot

Chapter 1

Introduction

1.1 Motivation

Robotics systems are characterized by highly nonlinear dynamics. This is due to the nature of multibody dynamics systems, in which multiple rigid bodies are interconnected by joints that constrain their motion. Nonlinearities arise from the Coriolis and centripetal accelerations which vary with the square of velocity, and joints often rotate about a wide range of angles, invalidating the small angles approximation typically used in linearization. In order to achieve fast, precise control across a robot's full operating range, nonlinear model-based control strategies are required. One of the most common approaches is to use the method of inverse dynamics, which uses full dynamics compensation terms to cancel out the nonlinearities in the system [25],[5]. The full dynamics compensation allows for decoupling of the robot's joints and cancellation of the Coriolis, centripetal, and gravitational accelerations. This in turn enables each joint to be controlled individually using linear control techniques. In the case of rigid robots, the inverse dynamics method is equivalent to feedback linearization [6].

While under ideal circumstances this approach provides many advantages, there are also practical limitations to this control strategy. Inverse dynamics control typically requires a high-fidelity model of the robot in order to achieve precise tracking. Developing an accurate mathematical model of a robot requires extensive knowledge of the mechanical properties,

including but not limited to three-dimensional center of mass locations, mass moment of inertia tensors, and precise geometric dimensions for each link. Even if all of these properties are known, there are still many practical issues such as joint friction, backlash, link flexion, and transmission losses that result in unmodeled dynamics in the final control system. In the case of a 3D-printed robot, the components cannot be easily modeled due to their non-uniform densities, the inconsistencies among the 3D printers used to manufacture the parts, the use of different plastics with mechanical properties that are not widely known, and so on. Therefore the problem that arises is how to achieve the desired performance of a model-based control system when the robot model is not explicitly known. In particular, the desired performance is that of an active compliance control system in which the robot is allowed to deviate from its trajectory when obstructed by humans or obstacles in the environment.

To address this issue, this thesis presents an adaptive control framework which modifies the model parameters online in order to achieve satisfactory tracking performance. The inertial properties are estimated by adapting with respect to functions of the unknown parameters. This is achieved by rewriting the robot dynamics equations as the product of a matrix of known nonlinear functions of the state variables and a vector of constant unknowns as done in [5]. The result is a nonlinear system linearly parameterized in terms of the unknowns, which can be estimated using adaptation laws derived from Lyapunov stability theory. The proposed control system consists of an outer-loop impedance controller, referred to as the high-level torque controller, and an inner loop force tracking controller, referred to as the low-level force controller. The high-level controller is used to regulate deviations from the nominal trajectory in the presence of disturbances or during interactions with humans and the environment, while the low-level controller commands the necessary forces in the linear actuators in order to achieve the joint torques commanded by the high-level controller. The proposed system is evaluated on an early prototype of the robot consisting of a 3DOF leg,

as well as two actuator testing configurations for the low-level controller.

1.2 Background

1.2.1 Adaptive Control in Robotics

Model Reference Adaptive Control (MRAC) is a well-known adaptive control strategy that can be leveraged when dealing with model uncertainties in robotics systems [28]. In an MRAC immplementation, a plant with known structure and unknown parameters is driven by a feedback control law to follow a reference model specified by the designer. The reference model represents the ideal tracking performance of the system. The feedback control law contains adaptive gains, which are modified online in order to ensure tracking of the reference model. Design of the feedback control law requires the existence of ideal gains which guarantee convergence of the plant to the reference model, and the adaptive update laws are typically determined such that stability is guaranteed under Lyapunov's direct method [14]. In Horowitz and Tomizuka [11], the authors presented a control system with an inner loop MRAC algorithm for cancellation of nonlinearities and decoupling of the joint dynamics and an outer-loop PID controller for tracking. In this implementation the effects of gravity were neglected, and the mass-inertia matrix and Coriolis and centripetal terms were estimated by the MRAC algorithm. The controller was then validated through simulation of a three-link planar manipulator. A notable result of the simulations in [11] was that the tracking performance of the robot was not significantly affected when the vector of Coriolis and centripetal torques was left out of the adaptation algorithm and when only the diagonal elements of the mass-inertia matrix were modified. This result could potentially be useful in the design of MRAC algorithms for higher degree of freedom robots, as this will significantly simplify the

adaptation laws. In [5], Craig et al. proposed a parameter adaptive control system to take advantage of known robot model parameters while estimating the unknowns online. This allowed for nonlinear model-based control using adaptive estimates of the model parameters, and thus full dynamics compensation was achieved. In 1987, Slotine and Li [24] presented a globally stable adaptive control algorithm using the same concept of reparameterization and achieved asymptotic convergence of the tracking errors to zero. This was done by implementing a reference trajectory that constrained the tracking errors to a sliding surface with stable first order dynamics, then designing the controller to guarantee convergence of the system trajectories to the sliding surface. Simulations were then conducted for a 2DOF planar manipulator with known properties and an unknown payload, and showed favorable results. Sadegh and Horowitz presented an improvement over their previous works in [23], in which two adaptive control algorithms, Exact Compensation Adaptation Law (ECAL) and Desired Compensation Adaptation Law (DCAL) were discussed. Global asymptotic stability was proven for the generalized dynamics of a robot manipulator, and the controllers were validated with simulations of a 2DOF SCARA robot. In a more recent example, Khan et al. [15] presented a model reference adaptive compliance controller for a 4DOF arm on the BERT II humanoid robot. This implementation bypassed the need to evaluate the complex nonlinear functions in the robot dynamics by only considering the rough structure of the mass-inertia matrix, Coriolis and centripetal terms, and gravity terms. The adaptive update laws for these parameters were simple linear differential equations which were functions of the filtered tracking error. The algorithm was further simplified by formulating the control laws directly in the task space. Each of the adaptive control schemes mentioned in this section with the exception of Horowitz and Tomizuka [11] utilizes Lyapunov's direct method [14] in the analysis of stability with the control and adaptation laws. In this thesis, Lyapunov's direct method will be heavily used in the derivations of the adaptive control systems presented.

1.2.2 Previous Iterations of THOR

SAFFIR, a predecessor to THOR, utilized force-controllable custom linear series elastic actuators in parallel configurations. In SAFFIR, torque control was not used throughout the robot's legs. Instead, a hybrid position control and force control approach was implemented, in which the force controlled ankle joint was used to to guide the trajectory of the robot center of gravity and the knee and hip joints were position controlled [18]. The force controlled ankle also utilized feedback from a 6-axis force torque transducer in each foot to to maintain the center of pressure (COP) inside the support polygon, which is the smallest convex shape containing all of points that are touching the ground [27], [18]. The resulting control system employed a combination of two popular bipedal locomotion strategies: passive dynamic walking which enforces tracking of predetermined stable limit cycles, and an active ZMP tracking scheme which regulates the location of the center of gravity in order to maintain balance during walking. The use of position control in SAFFIR had the negative result of limiting the robot's ability to exhibit compliant behavior when subjected to disturbances, as the force output in each actuator was not directly regulated. As with all high gain position control robots, the ability to interact safely with humans and the environment is not guaranteed, as injury or even death can occur from a collision with the robot. Therefore THOR was introduced with special emphasis on compliant locomotion, which is achieved by torque control rather than position control.

The original THOR platform is a 1.8 m, 60 kg torque-controlled humanoid robot with 34 total degrees of freedom, with 6DOF in each leg [10]. THOR's legs also utilized custom linear series elastic actuators in parallel configurations to control 2DOF hip and ankle mechanisms, and used dedicated microcontrollers for each 2DOF mechanism to enable distributed control. Compliant tracking was achieved with a cascaded joint impedance controller with an inner-

loop force controller. The inner-loop force controllers utilize of a combination of feedforward control and PID control with a disturbance observer based on an empirically determined open-loop plant. Dynamic locomotion was achieved by tracking a time-varying extension of the divergent component of motion, which is a transformation of the robot center of mass dynamics. This transformation decomposes the unstable 2^{nd} order dynamics into two first order systems, of which one is the stable center of mass dynamics and the other is the unstable divergent component of motion dynamics [10]. In order to track the DCM trajectories, optimized joint torques are calculated using the whole-body controller, which uses a linearly constrained quadratic cost function to minimize tracking errors while satisfying dynamic constraints [10].

With regards to the future implementation of the new 3D-printed humanoid robot, the control scheme in this thesis is designed with compliant joint torque control in mind so as to be compatible with the locomotion strategies of the original THOR.

Chapter 2

Description of Prototype

2.1 3D-Printed Humanoid

The original THOR platform was found to have difficulty supporting the weight of the upper body when additional hardware was mounted in the torso. This prompted the design of ESCHER, a new humanoid with double actuators in the knee joints for increased torque output [16]. The 3D-printed humanoid discussed in this work is a new iteration of the THOR platform, where the intention is to improve performance by lightening the chassis rather than increasing the number of actuators. Figure 2.1 shows a side-by-side comparison of the 3D-printed humanoid and the original THOR, and Figure 2.2 shows the degrees of freedom in the legs. Since the torso-mounted hardware such as computers, batteries, heat sinks, and additional sensors have not been fully specified at the time of writing this thesis, the final weight of the 3D printed robot is not known. However, early estimates from the chassis alone show a reduction of weight of at least 50% from the original robot. Though the final weight of the robot is still to be determined, the use of adaptive control in this system is intended to ensure adequate tracking performance regardless of the inertial properties of the robot. An additional motivation for the 3D-printed robot design is to reduce the number of components required in the robot chassis. The original THOR chassis consisted of hundreds of small aluminum parts produced in house on 3-axis CNC milling machines. The proposed 3D-printed design reduces this to fewer than one hundred 3D printed parts for the frame,

with some additional hardware such as bearings, nuts, bolts, and spacers. Not only does this reduce complexity of the design and improve ease of assembly, but in the event of a part breaking during testing, it can be easily replicated on a large enough 3D printer. Provided the printed part is sturdy enough to withstand the loads applied during operation, the robot should still function even if characteristics of the new part such as material, density, and infill are different due to the adaptive controller's robustness to model uncertainties.

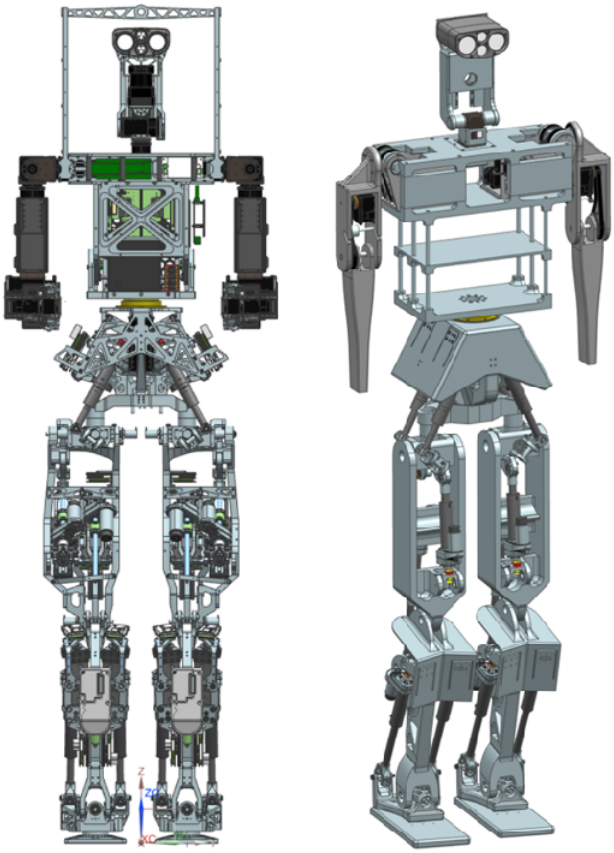

Figure 2.1: CAD renderings showing a side-by-side comparison of the of the original THOR and the 3D-printed version.

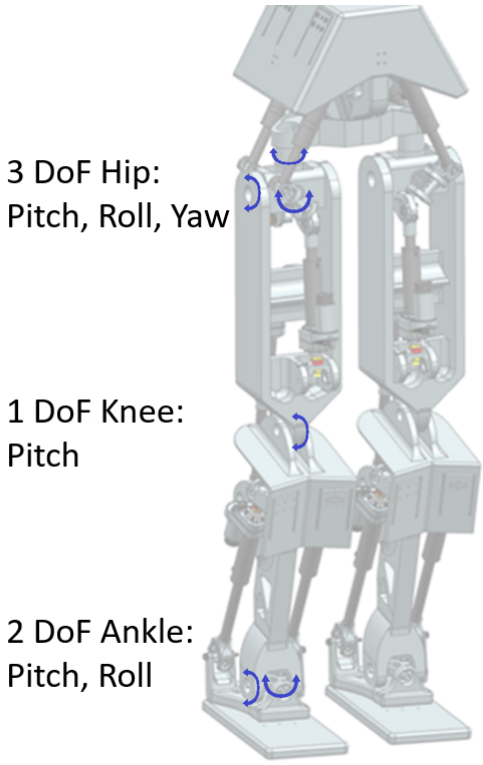

Figure 2.2: Diagram of the 6 degrees of freedom of each leg in the proposed design of the new 3D-printed humanoid.

2.2 3DOF Leg Test Bench

Experimental testing and validation of the controllers developed in this thesis was conducted using an early prototype of the 3D-printed robot. This prototype consists of a single leg with ankle pitch, ankle roll, and knee pitch degrees of freedom. The leg has three links: the foot, shin, and thigh links. The foot is bolted down to a steel plate to effectively fix it to the ground, and the thigh link is fitted with a steel rod through the hip pitch axis to enable the robot to carry different payloads. The 3D-printed components are made out of PLA and ABS plastics. The 3DOF prototype used in the experiments is shown in Figure 2.3.

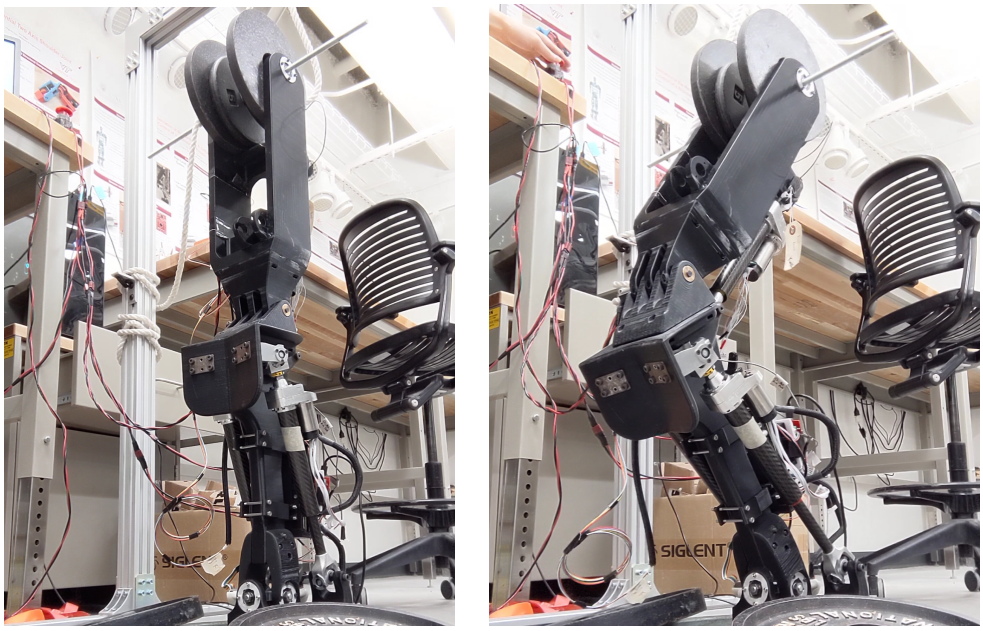

Figure 2.3: Image of the 3DOF leg performing a squat, showing the full range of motion of the knee joint for this particular prototype.

2.2.1 Sensing

Each joint on the robot is equipped with an optical encoder to measure the joint angle. The encoders used are the Gurley Precision A19 absolute rotary encoders. The A19 features a 15-bit output resolution for a single turn with a Synchronous Serial Interface (SSI) format [9]. This is equivalent to a resolution of about 0.011°. Each encoder is powered by 5V supplied directly from the microcontroller. The use of absolute encoders on the joints simplifies operation as knowledge of the joint position is not lost when the robot is powered off, however an initial calibration must be performed when the encoders are mounted onto the robot.

Each actuator is fitted with a Futek LCM200 load cell for force sensing. These load cells utilize metal foil strain gauge technology and are rated for tension and compression loads of up to 500 lbs [8]. CSG110 strain gauge amplifiers from Futek are used in order to amplify the

analog voltage output from the load cells before they are sent to the microcontroller ADCs [7]. Since the output of the CSG110 is ±10V, additional conditioning is required to bring the output down to 0-5V to be used in the microcontroller ADC. This was accomplished using a custom PCB that be stacked on top of the microcontroller.

Figure 2.4: Futek LCM200 load cell [8].

2.2.2 Actuation

Each custom linear actuator uses a Maxon EC-4Pole 200W, 48V brushless DC motor to supply torque to a ballscrew mechanism [19]. The motors are each driven by an off-the-shelf AZBDC12A8 PWM servo drive from Advanced Motion Controls [1]. The servo drive receives a 20kHz PWM signal from the microcontroller and regulates the motor current to be proportional to the duty cycle. A direction input on the servo drive receives a digital high or low from the microcontroller to control the direction of rotation. An external PSU supplies 48V DC to power the drive.

Figure 2.5: AMC AZBDC12A8 servo drive [1].

2.2.3 Microcontrollers

The knee joint and 2DOF ankle joint are each equipped with dedicated microcontrollers for interfacing with the sensors, calculating the control inputs, and data logging. The microcontroller used was the Tiva C Series TM4C123G LaunchPad from Texas Instruments, referred to from here on as the Tiva launchpad. The Tiva launchpad houses an 80MHz 32-bit ARM Cortex M4 processor, has dual 12-bit ADCs, and has UART, I2C, SPI, and CAN interfaces built in [26]. The two Tiva launchpads are connected to a host computer running Matlab which sends commands simultaneously to each of the boards over UART to start and stop the controller. The host computer also receives joint states, control input history, and ADC readings for data logging. Communication between the two Tiva launchpads also occurs through Matlab: position feedback is sent to Matlab over UART and then Matlab sends the position data to each of the other Tivas, also through UART.

Chapter 3

Adaptive Torque Control Algorithm

3.1 Description of Sagittal Plane Model

In order to simplify the analysis and controller design, only the sagittal plane dynamics will be considered in this chapter. The model developed in this section will function under the assumption that the ankle roll torque is zero at all times. In other words, the two ankle actuators will always supply the same force, resulting in zero net torque about the ankle roll axis. Practical issues with this assumption will be addressed later in this chapter as well as in Chapter 5 when the experimental implementation is introduced. A schematic of the leg in the sagittal plane with only knee pitch and ankle pitch degrees of freedom is given in Figure 3.1. The angle of the shin link, q_1, is measured counter clockwise from the y-axis of the world frame, and the angle of the thigh link, q_2 is measured counter clockwise from the shin link axis. The origin of the world frame is considered to be coincident with the ankle joint. Point H is the location of the hip pitch axis, which is where the thigh link would connect to the pelvis if this were the full humanoid robot. The shin and thigh links have masses m_1 and m_2 respectively, and the center of mass locations are given by $l_{\mathrm{cm}_1 x}$, $l_{\mathrm{cm}_1 y}$ and $l_{\mathrm{cm}_2 x}$, $l_{\mathrm{cm}_2 y}$. The shin and thigh links have moments of inertia J_1 and J_2 about the ankle and knee joints respectively. Due to the lack of accurate modeling of the internal structure of the 3D-printed parts in the robot, the inertial properties of the two links along with their center of mass locations are considered unknown.

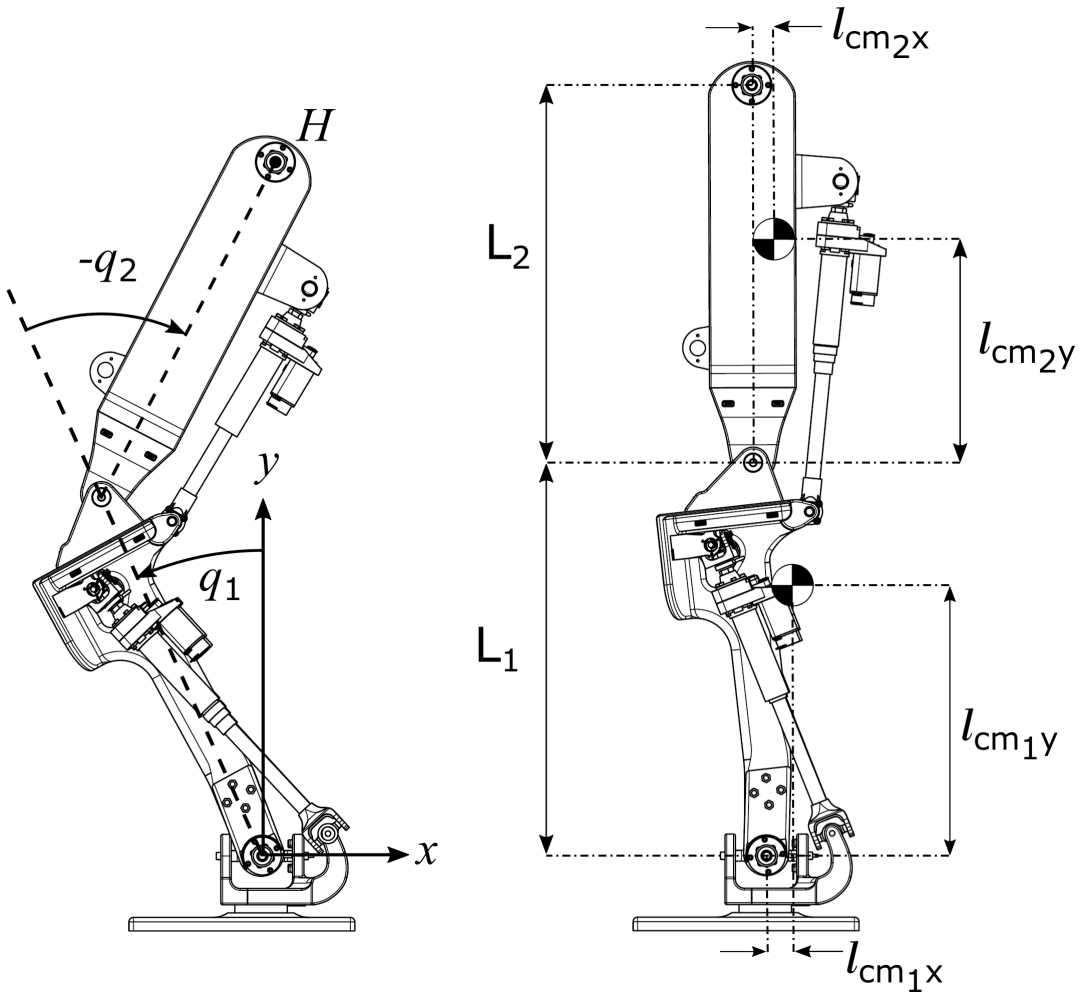

Figure 3.1: Schematic of the leg in the sagittal plane showing generalized coordinates (left) and geometric properties (right).

3.1.1 Mapping Linear Actuator Force to Joint Torque

Due to the use of linear actuators, it is necessary to determine a conversion between the forces produced by the actuators and the resulting torques acting on each joint. The knee joint contains a single linear actuator connecting the thigh and shin links, while the ankle joint contains two linear actuators that are symmetric across the frontal plane and sagittal plane of the robot. Symmetry in the configuration of the ankle actuators allows for simplification of the analysis through the assumption of the two forces being identical at all times, resulting in

zero roll torque. The relationship between the actuator force and joint torque is determined via the Principle of Virtual Work [25], which results in the following equation,

$$\tau = J^{\mathrm{T}}(q)F, \tag{3.1}$$

where for a single degree of freedom mechanism in two-dimensional space, $\tau \in \mathbb{R}$ is the joint torque, $J(q) \in \mathbb{R}^2$ is the mechanism Jacobian, q is the joint angle, and $F \in \mathbb{R}^2$ is the actuator force vector. Note that (3.1) assumes the system is in static equilibrium, and consequently the dynamics of the actuators will be neglected in this analysis. In order to use this relationship in the control algorithm, we must determine analytical expressions for the mechanism Jacobian and the direction of the actuator force vector in terms of the mechanism joint angles. Considering the knee joint, we define the parameters x_1, y_1, x_2, and y_2 in accordance with Figure 3.2 to solve the forward kinematics of the mechanism. Numerical values are determined from a CAD model of the robot. The position vector from the knee joint to the actuator attachment on the thigh link, r_{A} is given by

$$r_{\mathrm{A}} = R(q_2) \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \tag{3.2}$$

where $R(q_2) \in \mathbb{R}^2$ is a rotation matrix of the form $R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$. The actuator moment arm is the vector from the knee joint to the actuator attachment point on the shin link, r_{B}, given by

$$r_{\mathrm{B}} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}. \tag{3.3}$$

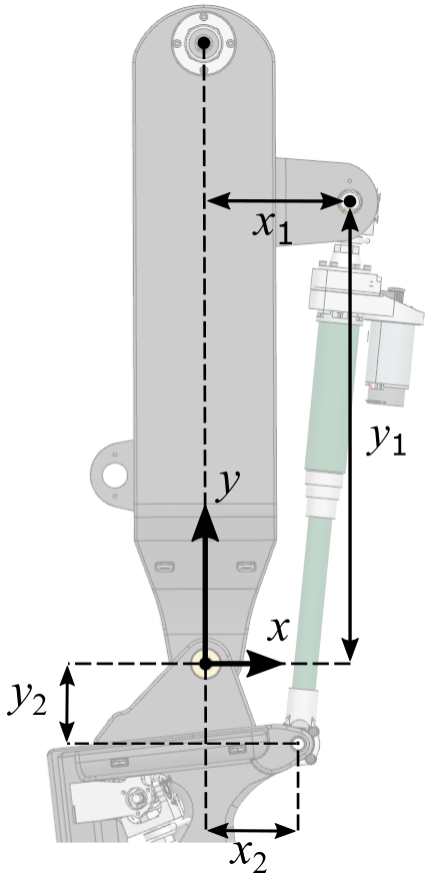

Figure 3.2: Knee joint schematic showing geometric properties used in the knee actuator forward kinematics.

The vector representing the length of the actuator, l_{A}, is calculated as

$$l_{\mathrm{A}} = r_{\mathrm{A}} - r_{\mathrm{B}}. \tag{3.4}$$

The actuator force, F_{A} can be written as a product of the force magnitude and the unit vector pointing in the direction of the actuator as follows,

$$F_{\mathrm{A}} = f_{\mathrm{A}} \frac{l_{\mathrm{A}}}{||l_{\mathrm{A}}||}, \tag{3.5}$$

where f_{A} is the magnitude of the actuator force. A diagram showing the vectors r_{A}, r_{B}, l_{A}, and f_{A} is given in Figure 3.3 for clarity. Using this notation, the knee torque is given by

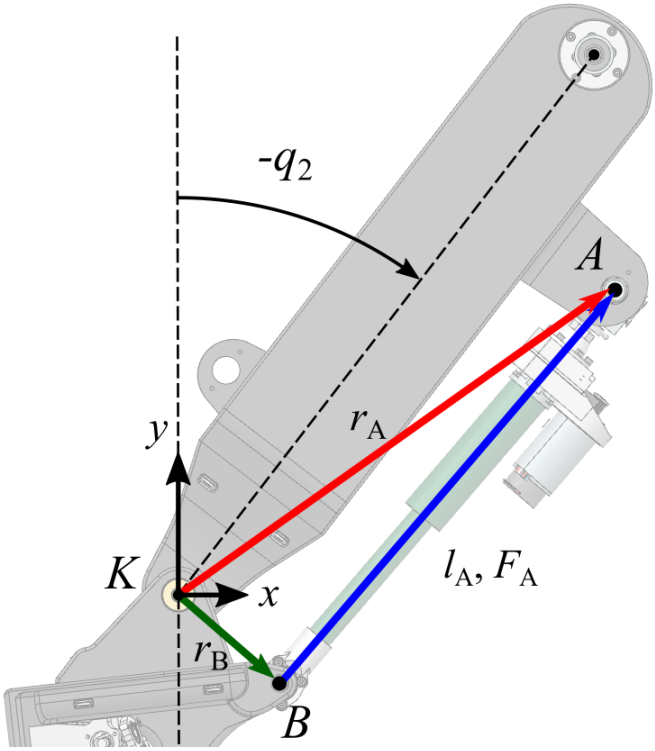

Figure 3.3: Knee joint schematic showing vectors used in the knee actuator forward kinematics. All position vectors shown are expressed in the shin link coordinate frame.

$$\tau_{\mathrm{knee}} = J^{\mathrm{T}}(q_2) F_{\mathrm{A}}, \tag{3.6}$$

where the mechanism Jacobian is $J(q_2) = \frac{\partial l_{\mathrm{A}}}{\partial q_2} \in \mathbb{R}^{2\times 2}$. In order to minimize the need to perform vector operations in the control algorithm, (3.6) is rewritten as a scalar function as follows,

$$\tau_{\mathrm{knee}} = f_{\mathrm{A}} f_{\mathrm{m,k}}(q_2), \tag{3.7}$$

where $f_{\mathrm{m,k}}(q_2) = J^{\mathrm{T}}(q_2)\frac{l_{\mathrm{A}}}{||l_{\mathrm{A}}||}$ is a nonlinear function mapping the actuator force to the resulting joint torque for a given joint angle. The analytical expression for $f_{\mathrm{m,k}}(q_2)$ is

$$f_{\mathrm{m,k}}(q_2) = \frac{(x_2y_1 - x_1y_2)\cos q_2 + (x_1x_2 + y_1y_2)\sin q_2}{\sqrt{(x_2 - x_1\cos q_2 + y_1\sin q_2)^2 + (y_1\cos q_2 - y_2 + x_1\sin q_2)^2}}. \tag{3.8}$$

Note that in the event of the actuator moment arm being equal to zero, (3.8) would also be equal to zero. This is equivalent to a singularity in the actuated mechanism. The robot was specifically designed to avoid singularities in the ankle and knee joints over their range of motion, and therefore given the geometric properties of the ankle and knee joints, (3.8) and the analogous function for the ankle joint are always greater than zero. Due to the highly nonlinear form of (3.8), the function is not calculated directly in the hardware implementation. Instead, numerical values were substituted into (3.8) across the range of motion of the joint in 0.1 degree intervals and a 3^{rd} order polynomial curve fit was applied to the data. The numerical value of $f_{\mathrm{m}}(q_2)$ is calculated online in real time via the equation of the curve fit. This results in a significantly faster calculation with no approximation error induced, as the coefficient of determination for the fit was found to be $R^2 = 1$.

When considering the sagittal plane dynamics, the moment arm for actuation of the ankle pitch can be derived in the same exact manner as the knee pitch due to symmetry in the configuration of the ankle actuators. For the case of the ankle actuator moment arm with respect to torque about the ankle pitch axis, the equation maintains the same exact structure as (3.8) but the geometric properties x_1, y_1, x_2, y_2 are taken from the shin link, shown in Figure 3.4, and the ankle pitch angle q_1 is used instead of q_2. The distance values for the knee and ankle actuator forward kinematics are tabulated in Table 3.1.

While the assumption of zero ankle roll torque due to the two ankle actuator forces being equal is valid for analysis of the sagittal plane dynamics, it is not practical for the purposes

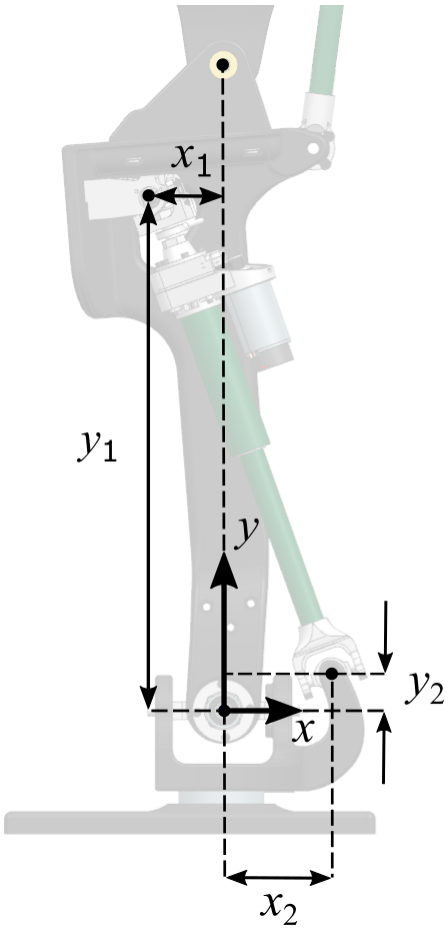

Figure 3.4: Ankle joint schematic showing geometric properties used in the simplified ankle actuator forward kinematics. Due to symmetry in the placement of the two ankle actuators, they can be considered as a single actuator affecting only ankle pitch when the actuator forces are assumed to be equal.

of the actual experiments. In reality, sending the same exact control input to both ankle actuators will not result in the same forces as each actuator is unique. Even if the two actuators converge to the same steady state force values commanded by the controller, a roll torque will inevitably be induced during the transient phase in which the two forces are converging. Therefore a closed-loop controller of the ankle roll is required for control of this prototype to be feasible. This will be further explained in Chapter 5, where the experimental implementation is discussed in detail. For a derivation of the actuator kinematics in 3D for

Table 3.1: Numerical values of the geometric properties used in the knee and ankle actuator forward kinematics, taken from the robot CAD model. All distances are given in millimeters.

	Knee	Ankle
x_1	98.101	-51.015
y_1	311.072	347.759
x_2	60.438	72.428
y_2	-56.243	24.797

the 2DOF ankle joint, refer to Appendix B.

3.2 Robot Dynamics

The equations of motion of the 2DOF sagittal plane model are derived applying the Lagrange method. This is achieved by determining the total kinetic and potential energy of the two-link system, writing the Lagrangian, and rearranging the Euler-Lagrange equation into its matrix form [25]. Therefore the dynamic model is given by

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = B(q)u, \tag{3.9}$$

where $q = [q_1 \quad q_2]^{\mathrm{T}}$ is the vector of generalized coordinates, $\dot{q} = [\dot{q}_1 \quad \dot{q}_2]^{\mathrm{T}}$ are the joint velocities, $D(q) \in \mathbb{R}^{2\times2}$ is the positive definite mass-inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{2\times2}$ is the matrix of Coriolis and centripetal terms, $G(q) \in \mathbb{R}^2$ is the vector of torques induced by gravity, $B(q) \in \mathbb{R}^{2\times2}$ is the input matrix, and $u = [f_1 \quad f_2]^{\mathrm{T}}$ is the vector of actuator forces, with f_2 being the knee actuator force and f_1 being the sum of the two ankle actuator forces, with the two ankle actuator forces assumed equal at all times. While (3.9) is linearly paramaterized in terms of the joint states, the matrices $D(q)$, $C(q,\dot{q})$, $G(q)$, and $B(q)$ are

highly nonlinear. The sum of the kinetic energy of the links $K(q, \dot{q})$ in quadratic form is

$$K(q, \dot{q}) = \frac{1}{2}\dot{q}^{\mathrm{T}} D(q)\dot{q}. \tag{3.10}$$

Given the scalar equation obtained by summing the kinetic energy of each link, the mass-inertia matrix is the second order Jacobian of the kinetic energy with respect to joint velocity, $D(q) = \frac{\partial^2 K}{\partial \dot{q}^2}$. The matrix of Coriolis and centripetal terms is defined in the same manner as [25] and [27], where element (k, j) of $C(q, \dot{q})$ is

$$C_{kj} = \sum_{i=1}^{n} \frac{1}{2}\Big(\frac{\partial D_{kj}}{\partial q_i} + \frac{\partial D_{ki}}{\partial q_j} - \frac{\partial D_{ij}}{\partial q_k}\Big)\dot{q}_i, \tag{3.11}$$

where n is the number of degrees of freedom of the system. The vector of gravity torques $G(q)$ is simply the Jacobian of the total potential energy of the system with respect to the generalized coordinates [27], and is therefore given by

$$G(q) = \frac{\partial V(q)}{\partial q}, \tag{3.12}$$

where $V(q)$ is the total potential energy of the system. Lastly, the input matrix $B(q)$ relates the linear force in each actuator to the applied torque in each joint, and is given by

$$B(q) = \begin{bmatrix} f_{\mathrm{m,a}}(q_1) & 0 \\ 0 & f_{\mathrm{m,k}}(q_2) \end{bmatrix}, \tag{3.13}$$

where $f_{\mathrm{m,a}}(q_1)$ is a function of the same form as (3.8) that uses the geometric properties of the ankle joint as defined in Table 3.1 and has input argument q_1. Note that since $f_{\mathrm{m,a}}(q_1), f_{\mathrm{m,k}}(q_2) > 0$ over the range of joint angles, $B(q)$ is positive definite and therefore invertible. The robot model can be written as a nonlinear system with the state vector

$x(t) = [q \quad \dot{q}]^{\mathrm{T}} \in \mathbb{R}^4$ and corresponding dynamics given by

$$\dot{x}(t) = f(x) + g(x)u, \tag{3.14}$$

where

$$f(x) = \begin{bmatrix} \dot{q} \\ -D^{-1}(q)(C(q,\dot{q})\dot{q} + G(q)) \end{bmatrix}, \tag{3.15}$$

$$g(x) = \begin{bmatrix} 0 \\ D^{-1}(q)B(q) \end{bmatrix}. \tag{3.16}$$

Note that the matrix inverse $D^{-1}(q)$ is guaranteed to exist since $D(q)$ is positive definite. Using (3.10)-(3.12) and the geometric and inertial properties of the robot, equations for the mass-inertia matrix, Coriolis and centripetal terms, and gravity torques are given as follows,

$$D(q) = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}, \tag{3.17}$$

$$\begin{aligned} D_{11} &= J_1 + J_2 + m_1(l_{\mathrm{cm}_1 x}^2 + l_{\mathrm{cm}_1 y}^2) + m_2(L_1^2 + l_{\mathrm{cm}_2 x}^2 + l_{\mathrm{cm}_2 y}^2) + \sin(q_2)(2L_1 l_{\mathrm{cm}_2 x} m_2) \\ &\quad + \cos(q_2)(2L_1 l_{\mathrm{cm}_2 y} m_2), \end{aligned} \tag{3.18}$$

$$D_{12} = D_{21} = m_2(l_{\mathrm{cm}_2 x}^2 + l_{\mathrm{cm}_2 y}^2) + J_2 + \sin(q_2)(L_1 l_{\mathrm{cm}_2 x} m_2) + \cos(q_2)(L_1 l_{\mathrm{cm}_2 y} m_2), \tag{3.19}$$

$$D_{22} = m_2(l_{\mathrm{cm}_2 x}^2 + l_{\mathrm{cm}_2 y}^2) + J_2, \tag{3.20}$$

$$C(q,\dot{q}) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}, \tag{3.21}$$

$$C_{11} = \dot{q}_2 \cos(q_2)(L_1 l_{\mathrm{cm}_2 x} m_2) - \dot{q}_2 \sin(q_2)(L_1 l_{\mathrm{cm}_2 y} m_2), \tag{3.22}$$

$$C_{12} = \dot{q}_1 \cos(q_2)(L_1 l_{\mathrm{cm}_2 x} m_2) - \dot{q}_1 \sin(q_2)(L_1 l_{\mathrm{cm}_2 y} m_2) + \dot{q}_2 \cos(q_2)(L_1 l_{\mathrm{cm}_2 x} m_2) - \dot{q}_2 \sin(q_2)(L_1 l_{\mathrm{cm}_2 y} m_2), \tag{3.23}$$

$$C_{21} = -\dot{q}_1 \cos(q_2)(L_1 l_{\mathrm{cm}_2 x} m_2) + \dot{q}_1 \sin(q_2)(L_1 l_{\mathrm{cm}_2 y} m_2), \tag{3.24}$$

$$C_{22} = 0, \tag{3.25}$$

$$G(q) = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}, \tag{3.26}$$

$$G_1 = \cos(q_1)(m_1 g\, l_{\mathrm{cm}_1 x}) - \sin(q_1)(m_2 g L_1 + m_1 g\, l_{\mathrm{cm}_1 y}) - \sin(q_1 + q_2)(m_2 g\, l_{\mathrm{cm}_2 y}) + \cos(q_1 + q_2)(m_2 g\, l_{\mathrm{cm}_2 x}), \tag{3.27}$$

$$G_2 = \cos(q_1 + q_2)(m_2 g\, l_{\mathrm{cm}_2 x}) - \sin(q_1 + q_2)(m_2 g\, l_{\mathrm{cm}_2 y}), \tag{3.28}$$

where g is the acceleration due to gravity. The functions contained in $D(q)$, $C(q, \dot{q})$ and $G(q)$ are intentionally written with nonlinear functions of the joint variables separated out from the constant robot parameters in order to facilitate the design of the adaptive controller.

3.3 Controller Design

The objective of this controller is to use the actuator forces to regulate the net torque at each joint such that compliant tracking of each time-varying desired joint trajectory is achieved. This control algorithm is derived in three steps. First we must specify a reference system that converges to the desired trajectories and defines the ideal behavior of the system. Next, an ideal control law is designed to drive the system trajectories to the reference model. The

ideal controller assumes that the ideal parameters to drive the plant to the reference system are known, and is not implementable in practice. Therefore the final step is to design an adaptive control law that is a function of time-varying adaptive gains which estimate the ideal values. Update laws for the adaptive gains are derived using Lyapunov's Direct Method [14].

3.3.1 Reference Model

The proposed control framework will utilize both a position reference system and a velocity reference system. The position reference model contains standard second order systems for each joint, while the velocity reference model is a modified version of the velocity states in the position reference model. The position reference system, $q_{\mathrm{r}}(t) \in \mathbb{R}^4$ has the state vector $q_{\mathrm{r}}(t) = [q_{\mathrm{r}_1} \;\; q_{\mathrm{r}_2} \;\; q_{\mathrm{r}_3} \;\; q_{\mathrm{r}_4}]^{\mathrm{T}}$ and the dynamics

$$\dot{q}_{\mathrm{r}}(t) = A_{\mathrm{r}} q_{\mathrm{r}}(t) + B_{\mathrm{r}} r(t), \tag{3.29}$$

where A_{r} and B_{r} are defined as

$$A_{\mathrm{r}} \triangleq \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_{\mathrm{n}_1}^2 & -2\zeta_1\omega_{\mathrm{n}_1} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_{\mathrm{n}_2}^2 & -2\zeta_2\omega_{\mathrm{n}_2} \end{bmatrix}, \quad B_{\mathrm{r}} \triangleq \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \tag{3.30}$$

and $\omega_{\mathrm{n}_i} > 0$, $\zeta_i > 0$ are chosen based on a desired settling time and overshoot, and $r(t)$ is given by

$$r(t) = [r_1(t) \;\; r_2(t)]^{\mathrm{T}}, \tag{3.31}$$

$$r_i(t) = \omega_{\mathrm{n}_i}^2 q_{i,\mathrm{d}}(t) + 2\zeta_i \omega_{\mathrm{n}_i} \dot{q}_{i,\mathrm{d}}(t) + \ddot{q}_{i,\mathrm{d}}(t), \tag{3.32}$$

where $q_{i,\mathrm{d}}(t)$ is the desired position trajectory for the i^{th} joint. Let the vector of desired joint positions and velocities be denoted $q_{\mathrm{d}}(t) = [q_{1,\mathrm{d}}(t) \;\; \dot{q}_{1,\mathrm{d}}(t) \;\; q_{2,\mathrm{d}}(t) \;\; \dot{q}_{2,\mathrm{d}}(t)]^{\mathrm{T}}$. Each element in $q_{\mathrm{d}}(t)$ is a continuously differentiable function with respect to time. Since A_{r} is Hurwitz, convergence of the reference model to the desired trajectories can be proven by examining the error dynamics. Let $e_{\mathrm{d}}(t)$ be the reference model error,

$$e_{\mathrm{d}}(t) = q_{\mathrm{r}}(t) - q_{\mathrm{d}}(t). \tag{3.33}$$

Differentiating (3.33) with respect to time and substituting in (3.29), (3.31), and (3.33) yields

$$\begin{aligned} \dot{e}_{\mathrm{d}}(t) &= A_{\mathrm{r}} q_{\mathrm{r}}(t) + B_{\mathrm{r}} r(t) - \dot{q}_{\mathrm{d}}(t) \\ &= A_{\mathrm{r}} e_{\mathrm{d}}(t). \end{aligned} \tag{3.34}$$

Therefore the reference model error $e_{\mathrm{d}}(t)$ converges exponentially to zero. In order to ensure the robot joints track the position reference model, we define the following position tracking error to be used in the feedback control law:

$$e(t) \triangleq q(t) - \begin{bmatrix} q_{\mathrm{r}_1}(t) \\ q_{\mathrm{r}_3}(t) \end{bmatrix}, \tag{3.35}$$

The velocity reference model $q_{\mathrm{rv}}(t)$ is defined as

$$q_{\mathrm{rv}}(t) \triangleq \begin{bmatrix} q_{\mathrm{r}_2}(t) \\ q_{\mathrm{r}_4}(t) \end{bmatrix} - \Lambda_{\mathrm{p}} e(t), \tag{3.36}$$

where $\Lambda_{\mathrm{p}} \in \mathbb{R}^{2\times 2}$ is a positive definite diagonal matrix of velocity reference model gains. The velocity error is defined as

$$e_{\mathrm{v}}(t) \triangleq \dot{q}(t) - q_{\mathrm{rv}}(t). \tag{3.37}$$

Note the similarity of (3.36) to the velocity reference system used by Sadegh and Horowitz in [23]. The velocity reference model converges to $[q_{\mathrm{r}_2} \;\; q_{\mathrm{r}_4}]^{\mathrm{T}}$ as $e(t) \to 0$ by definition. Since $[q_{\mathrm{r}_2}(t) \;\; q_{\mathrm{r}_4}(t)]^{\mathrm{T}}$ converges exponentially to $[\dot{q}_{1,\mathrm{d}}(t) \;\; \dot{q}_{2,\mathrm{d}}(t)]^{\mathrm{T}}$ as $t \to \infty$, $q_{\mathrm{rv}}(t)$ will converge to the desired velocities as long as $e(t)$ is guaranteed to converge to 0. This is achieved by using Lyapunov's Direct Method to find a feedback control law to drive $e(t)$ and $e_{\mathrm{v}}(t)$ asymptotically to zero.

3.3.2 Ideal Controller and Reparameterization

In order to linearize and decouple the dynamics of each joint, the ideal control law will contain a dynamics compensation component to cancel out the nonlinearities in the system. This is known as inverse dynamics control [25], and for the case of rigid robots this is analogous to feedback linearization [6]. In order to facilitate the use of adaptive control later on, the dynamics compensation component of the ideal control law will be written as the product of a matrix of nonlinear functions of the state variables and a vector of constants derived from the robot parameters. The method of reparameterization of the robot dynamics was utilized by Sadegh and Horowitz in [23] and was originally proposed by Craig et al. in [5]. In order for the control law to work in this application, the inverse dynamics component will need to

be

$$W(q,\dot{q},q_{\text{rv}},\dot{q}_{\text{rv}})\theta^{*} = D(q)\dot{q}_{\text{rv}} + C(q,\dot{q})q_{\text{rv}} + G(q), \tag{3.38}$$

where $W(q,\dot{q},q_{\text{rv}},\dot{q}_{\text{rv}}) \in \mathbb{R}^{2\times r}$ is a matrix of nonlinear functions of known state variables and $\theta^{*} \in \mathbb{R}^{r}$ is a vector of constants. In order to achieve this result, each term in the right hand side of (3.38) must also be reparameterized as a product of a matrix of nonlinear functions of the state variables and a vector of constants. Starting with $D(q)\dot{q}_{\text{rv}}$ we have,

$$D(q)\dot{q}_{\text{rv}} = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_{\text{rv}_1} \\ \dot{q}_{\text{rv}_2} \end{bmatrix} = \begin{bmatrix} D_{\text{rv}_1} \\ D_{\text{rv}_2} \end{bmatrix}, \tag{3.39}$$

where

$$\begin{aligned} D_{\text{rv}_1} &= \dot{q}_{\text{rv}_1}\big(J_1 + J_2 + m_1(l_{\text{cm}_1 x}^2 + l_{\text{cm}_1 y}^2) + m_2(L_1^2 + l_{\text{cm}_2 x}^2 + l_{\text{cm}_2 y}^2)\big) \\ &\quad + \big(\sin(q_2)\dot{q}_{\text{rv}_1} + \frac{1}{2}\sin(q_2)\dot{q}_{\text{rv}_2}\big)(2L_1 l_{\text{cm}_2 x} m_2) \\ &\quad + \big(\cos(q_2)\dot{q}_{\text{rv}_1} + \frac{1}{2}\cos(q_2)\dot{q}_{\text{rv}_2}\big)(2L_1 l_{\text{cm}_2 y} m_2) + \dot{q}_{\text{rv}_2}(J_2 + m_2(l_{\text{cm}_2 x}^2 + l_{\text{cm}_2 y}^2), \end{aligned} \tag{3.40}$$

$$\begin{aligned} D_{\text{rv}_2} &= \frac{1}{2}\dot{q}_{\text{rv}_1}\sin(q_2)\big(2L_1 l_{\text{cm}_2 x} m_2\big) + \frac{1}{2}\dot{q}_{\text{rv}_1}\cos(q_2)\big(2L_1 l_{\text{cm}_2 y} m_2\big) \\ &\quad + (\dot{q}_{\text{rv}_1} + \dot{q}_{\text{rv}_2})(J_2 + m_2(l_{\text{cm}_2 x}^2 + l_{\text{cm}_2 y}^2). \end{aligned} \tag{3.41}$$

Equations (3.39)-(3.41) can be written as the product of $W_{\text{D}_{\text{rv}}}(q,\dot{q}_{\text{rv}}) \in \mathbb{R}^{2\times 4}$ and $\theta^{*}_{\text{D}_{\text{rv}}} \in \mathbb{R}^{4}$ as follows,

$$D(q)\dot{q}_{\text{rv}} = W_{\text{D}_{\text{rv}}}(q,\dot{q}_{\text{rv}})\theta^{*}_{\text{D}_{\text{rv}}}, \tag{3.42}$$

where

$$W_{\text{D}_{\text{rv}}}(q,\dot{q}_{\text{rv}}) = \begin{bmatrix} \dot{q}_{\text{rv}_1} & \sin(q_2)(\dot{q}_{\text{rv}_1} + \frac{1}{2}\dot{q}_{\text{rv}_2}) & \cos(q_2)(\dot{q}_{\text{rv}_1} + \frac{1}{2}\dot{q}_{\text{rv}_2}) & \dot{q}_{\text{rv}_2} \\ 0 & \frac{1}{2}\dot{q}_{\text{rv}_1}\sin(q_2) & \frac{1}{2}\dot{q}_{\text{rv}_1}\cos(q_2) & \dot{q}_{\text{rv}_1} + \dot{q}_{\text{rv}_2} \end{bmatrix}, \tag{3.43}$$

$$\theta^*_{\mathrm{D_{rv}}} = \begin{bmatrix} J_1 + J_2 + m_1(l^2_{\mathrm{cm_1}x} + l^2_{\mathrm{cm_1}y}) + m_2(L_1^2 + l^2_{\mathrm{cm_2}x} + l^2_{\mathrm{cm_2}y}) \\ 2L_1 l_{\mathrm{cm_2}x} m_2 \\ 2L_1 l_{\mathrm{cm_2}y} m_2 \\ J_2 + m_2(l^2_{\mathrm{cm_2}x} + l^2_{\mathrm{cm_2}y}) \end{bmatrix}. \tag{3.44}$$

Moving on to the next term, $C(q,\dot{q})q_{\mathrm{rv}}$, we have

$$C(q,\dot{q})q_{\mathrm{rv}} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} q_{\mathrm{rv_1}} \\ q_{\mathrm{rv_2}} \end{bmatrix} = \begin{bmatrix} C_{\mathrm{rv_1}} \\ C_{\mathrm{rv_2}} \end{bmatrix}, \tag{3.45}$$

where

$$\begin{aligned} C_{\mathrm{rv_1}} &= \big(q_{\mathrm{rv_1}}\dot{q}_2\cos(q_2) + q_{\mathrm{rv_2}}\cos(q_2)(\dot{q}_1+\dot{q}_2)\big)(L_1 l_{\mathrm{cm_2}x} m_2) \\ &\quad - \big(q_{\mathrm{rv_1}}\dot{q}_2\sin(q_2) + q_{\mathrm{rv_2}}\sin(q_2)(\dot{q}_1+\dot{q}_2)\big)(L_1 l_{\mathrm{cm_2}y} m_2), \qquad (3.46) \\ C_{\mathrm{rv_2}} &= -\dot{q}_1 q_{\mathrm{rv_1}}\cos(q_2)(L_1 l_{\mathrm{cm_2}x} m_2) + \dot{q}_1 q_{\mathrm{rv_1}}\sin(q_2)(L_1 l_{\mathrm{cm_2}y} m_2). \qquad (3.47) \end{aligned}$$

Equations (3.45)-(3.47) can be written as the product of $W_{\mathrm{C_{rv}}}(q,\dot{q},q_{\mathrm{rv}}) \in \mathbb{R}^{2\times 2}$ and $\theta^*_{\mathrm{C_{rv}}} \in \mathbb{R}^2$ as follows,

$$C(q,\dot{q})q_{\mathrm{rv}} = W_{\mathrm{C_{rv}}}(q,\dot{q},q_{\mathrm{rv}})\theta^*_{\mathrm{C_{rv}}}, \tag{3.48}$$

where

$$W_{\mathrm{C_{rv}}}(q,\dot{q},q_{\mathrm{rv}}) = \begin{bmatrix} q_{\mathrm{rv_1}}\dot{q}_2\cos(q_2) + q_{\mathrm{rv_2}}\cos(q_2)(\dot{q}_1+\dot{q}_2) & -\big(q_{\mathrm{rv_1}}\dot{q}_2\sin(q_2) + q_{\mathrm{rv_2}}\sin(q_2)(\dot{q}_1+\dot{q}_2)\big) \\ -\dot{q}_1 q_{\mathrm{rv_1}}\cos(q_2) & \dot{q}_1 q_{\mathrm{rv_1}}\sin(q_2) \end{bmatrix} \tag{3.49}$$

$$\theta^*_{\mathrm{C_{rv}}} = \begin{bmatrix} L_1 l_{\mathrm{cm_2}x} m_2 \\ L_1 l_{\mathrm{cm_2}y} m_2 \end{bmatrix} \tag{3.50}$$

Finally the vector of torques due to gravity defined in (3.26) can be reparameterized as

$$G(q) = W_{\mathrm{G}}(q)\theta^*_{\mathrm{G}}, \tag{3.51}$$

where

$$W_{\mathrm{G}}(q) = \begin{bmatrix} \cos(q_1) & \sin(q_1) & -\sin(q_1+q_2) & \cos(q_1+q_2) \\ 0 & 0 & -\sin(q_1+q_2) & \cos(q_1+q_2) \end{bmatrix}, \tag{3.52}$$

$$\theta^*_{\mathrm{G}} = \begin{bmatrix} m_1 g l_{\mathrm{cm}_1 x} \\ m_2 g L_1 - m_1 g l_{\mathrm{cm}_1 y} \\ m_2 g l_{\mathrm{cm}_2 y} \\ m_2 g l_{\mathrm{cm}_2 x} \end{bmatrix}. \tag{3.53}$$

Therefore the full dynamics reparameterization required in the inverse dynamics control (3.38) is achieved with the following,

$$W(q, \dot{q}, q_{\mathrm{rv}}, \dot{q}_{\mathrm{rv}}) = \begin{bmatrix} W_{\mathrm{D}_{\mathrm{rv}}}(q, \dot{q}_{\mathrm{rv}}) & W_{\mathrm{C}_{\mathrm{rv}}}(q, \dot{q}, q_{\mathrm{rv}}) & W_{\mathrm{G}}(q) \end{bmatrix}, \tag{3.54}$$

$$\theta^* = \begin{bmatrix} \theta^{*\mathrm{T}}_{\mathrm{D}_{\mathrm{rv}}} & \theta^{*\mathrm{T}}_{\mathrm{C}_{\mathrm{rv}}} & \theta^{*\mathrm{T}}_{\mathrm{G}} \end{bmatrix}^{\mathrm{T}}. \tag{3.55}$$

For brevity, the arguments of $W(q, \dot{q}, q_{\mathrm{rv}}, \dot{q}_{\mathrm{rv}})$ will be dropped for the rest of this chapter. The parameter vector used to reparameterize the dynamics in this manner is not unique for any given robot, and therefore finding the minimum dimension of $\theta \in \mathbb{R}^r$ is difficult [25]. In this implementation, there are ten unique parameters required to reparameterize the dynamics, i.e. $r = 10$.

Let the ideal control law be defined as

$$u^*(t) = B^{-1}(q)\big(W\theta^* - K_{\mathrm{s}}e(t) - K_{\mathrm{d}}e_{\mathrm{v}}(t)\big), \tag{3.56}$$

where $K_{\mathrm{s}} \in \mathbb{R}^{2\times 2}$ and $K_{\mathrm{d}} \in \mathbb{R}^{2\times 2}$ are diagonal and positive definite stiffness and damping matrices respectively. The error dynamics are obtained by first differentiating (3.35) with respect to time and substituting in (3.36),

$$\begin{aligned}\dot{e}(t) &= \dot{q}(t) - \begin{bmatrix} q_{\mathrm{r}_2} \\ q_{\mathrm{r}_4} \end{bmatrix} \\ &= \dot{q}(t) - (q_{\mathrm{rv}} + \Lambda_{\mathrm{p}} e(t)) \\ &= e_{\mathrm{v}}(t) - \Lambda_{\mathrm{p}} e(t), \end{aligned} \tag{3.57}$$

then differentiating (3.37) with respect to time and substituting in (3.9) and (3.56),

$$\begin{aligned}\dot{e}_{\mathrm{v}}(t) &= \ddot{q}(t) - \dot{q}_{\mathrm{rv}}(t) \\ &= D^{-1}(q)\big(B(q)u^*(t) - C(q,\dot{q}) - G(q)\big) - \dot{q}_{\mathrm{rv}}(t) \\ &= D^{-1}(q)\big(W\theta^* - K_{\mathrm{s}}e(t) - K_{\mathrm{d}}e_{\mathrm{v}}(t) - C(q,\dot{q})\dot{q}(t) - G(q)\big) - \dot{q}_{\mathrm{rv}}(t) \\ &= D^{-1}(q)\big(D(q)\dot{q}_{\mathrm{rv}}(t) + C(q,\dot{q})(q_{\mathrm{rv}}(t) - \dot{q}(t)) - K_{\mathrm{s}}e(t) - K_{\mathrm{d}}e_{\mathrm{v}}(t)\big) - \dot{q}_{\mathrm{rv}}(t) \\ &= D^{-1}(q)\big(-C(q,\dot{q})e_{\mathrm{v}}(t) - K_{\mathrm{s}}e(t) - K_{\mathrm{d}}e_{\mathrm{v}}(t)\big). \end{aligned} \tag{3.58}$$

Note that (3.58) represents the velocity error dynamics only when the ideal control law is used. To verify that the closed loop system tracks the position and velocity reference systems properly, Lyapunov's Direct Method is utilized. Consider the following Lyapunov function candidate [14],

$$V(e, e_{\mathrm{v}}) = \frac{1}{2}e_{\mathrm{v}}^{\mathrm{T}}(t)D(q)e_{\mathrm{v}}(t) + \frac{1}{2}e^{\mathrm{T}}(t)K_{\mathrm{s}}e(t). \tag{3.59}$$

Taking the time derivative of (3.59) along the trajectories of (3.35) and (3.37) results in

$$\dot{V}(e, e_{\mathrm{v}}) = e_{\mathrm{v}}^{\mathrm{T}}(t)D(q)\dot{e}_{\mathrm{v}}(t) + \frac{1}{2}e_{\mathrm{v}}^{\mathrm{T}}(t)\dot{D}(q)e_{\mathrm{v}}(t) + e^{\mathrm{T}}(t)K_{\mathrm{s}}\dot{e}(t). \tag{3.60}$$

Substituting in (3.57) and (3.58) yields

$$\begin{aligned}\dot{V}(e,e_{\mathrm{v}}) &= e_{\mathrm{v}}^{\mathrm{T}}(t)D(q)D^{-1}(q)\big(-C(q,\dot{q})e_{\mathrm{v}}(t)-K_{\mathrm{s}}e(t)-K_{\mathrm{d}}e_{\mathrm{v}}(t)\big)\\ &\quad+\frac{1}{2}e_{\mathrm{v}}^{\mathrm{T}}(t)\dot{D}(q)e_{\mathrm{v}}(t)+e^{\mathrm{T}}(t)K_{\mathrm{s}}(e_{\mathrm{v}}(t)-\Lambda_{\mathrm{p}}e(t))\\ &= e_{\mathrm{v}}^{\mathrm{T}}(t)\big(\frac{1}{2}\dot{D}(q)-C(q,\dot{q})\big)e_{\mathrm{v}}(t)-e_{\mathrm{v}}^{\mathrm{T}}(t)K_{\mathrm{d}}e_{\mathrm{v}}(t)-e^{\mathrm{T}}(t)K_{\mathrm{s}}e(t).\end{aligned}\tag{3.61}$$

Since the quantity $\frac{1}{2}\dot{D}(q)-C(q,\dot{q})=\dot{D}(q)-2C(q,\dot{q})$ is skew-symmetric [25], (3.61) can be simplified to

$$\dot{V}(e,e_{\mathrm{v}}) = -e_{\mathrm{v}}^{\mathrm{T}}(t)K_{\mathrm{d}}e_{\mathrm{v}}(t)-e^{\mathrm{T}}(t)K_{\mathrm{s}}e(t).\tag{3.62}$$

Since the Lyapunov function (3.59) is radially unbounded and positive definite for $(e,e_{\mathrm{v}})\neq 0$ and its derivative is negative definite for $(e,e_{\mathrm{v}})\neq 0$, the equilibrium point $(e,e_{\mathrm{v}})=0$ is Globally Asymptotically Stable by Lyapunov's Direct Method [14]. Therefore the position and velocity errors are guaranteed to converge asymptotically to zero as $t\to\infty$ for the closed-loop system with the ideal control law.

Since the ideal control law is a function of the unknown parameters contained in θ^{*}, it cannot be implemented in practice. In order to make this control framework feasible without identifying the inertial properties of the robot, the control law will be modified to be a function of adaptive gains which are estimates of the ideal values contained in θ^{*}.

3.3.3 Adaptive Controller

Let $\theta(t)\in\mathbb{R}^{10}$ be the vector of adaptive gains. The elements of $\theta(t)$ are estimates of the parameters contained in θ^{*}. The parameter estimation error is given by

$$\tilde{\theta}(t)=\theta(t)-\theta^{*}.\tag{3.63}$$

Since the ideal parameters contained in θ^* do not change over time, the time derivative of the parameter estimation error is simply

$$\dot{\tilde{\theta}}(t) = \dot{\theta}(t). \tag{3.64}$$

The control law $u(t)$ is defined as a function of the adaptive parameter vector $\theta(t)$, resulting in the following,

$$u(t) = B^{-1}(q)\big(W\theta(t) - K_{\mathrm{s}}e(t) - K_{\mathrm{d}}e_{\mathrm{v}}(t)\big). \tag{3.65}$$

With the adaptive control law (3.65), the velocity error dynamics become

$$\dot{e}_{\mathrm{v}}(t) = D^{-1}(q)\big(W\theta(t) - C(q,\dot{q})\dot{q}(t) - G(q) - K_{\mathrm{s}}e(t) - K_{\mathrm{d}}e_{\mathrm{v}}(t)\big) - \dot{q}_{\mathrm{rv}}(t). \tag{3.66}$$

In order to guarantee stability of the adaptive controller, we must ensure stability of the parameter error dynamics (3.64) in addition to the position and velocity error dynamics. Therefore we define the following update law for the adaptive parameters,

$$\dot{\theta}(t) = -\Gamma W^{\mathrm{T}} e_{\mathrm{v}}(t), \tag{3.67}$$

where $\Gamma \in \mathbb{R}^{10\times 10}$ is a positive definite diagonal matrix of adaptation gains, and the Lyapunov function (3.59) is modified to contain the parameter errors. Consider the new Lyapunov function candidate,

$$V(e, e_{\mathrm{v}}, \tilde{\theta}) = \frac{1}{2}e_{\mathrm{v}}^{\mathrm{T}}(t)D(q)e_{\mathrm{v}}(t) + \frac{1}{2}e^{\mathrm{T}}(t)K_{\mathrm{s}}e(t) + \frac{1}{2}\tilde{\theta}^{\mathrm{T}}(t)\Gamma^{-1}\tilde{\theta}(t). \tag{3.68}$$

Differentiating (3.68) with respect to time along the trajectories of (3.35), (3.37) and (3.63) yields

$$\dot{V}(e, e_{\mathrm{v}}, \tilde{\theta}) = e_{\mathrm{v}}^{\mathrm{T}}(t) D(q) \dot{e}_{\mathrm{v}}(t) + \frac{1}{2} e_{\mathrm{v}}^{\mathrm{T}}(t) \dot{D}(q) e_{\mathrm{v}}(t) + e^{\mathrm{T}}(t) K_{\mathrm{s}} \dot{e}(t) + \tilde{\theta}^{\mathrm{T}}(t) \Gamma^{-1} \dot{\tilde{\theta}}(t). \tag{3.69}$$

Substituting in (3.37), (3.57), (3.64) and (3.66) and rearranging terms results in

$$\begin{aligned}
\dot{V}(e, e_{\mathrm{v}}, \tilde{\theta}) &= e_{\mathrm{v}}^{\mathrm{T}}(t) D(q) \Big(D^{-1}(q) \big(W\theta(t) - C(q, \dot{q}) \dot{q}(t) - G(q) - K_{\mathrm{s}} e(t) - K_{\mathrm{d}} e_{\mathrm{v}}(t) \big) - \dot{q}_{\mathrm{rv}}(t) \Big) \\
&\quad + \frac{1}{2} e_{\mathrm{v}}^{\mathrm{T}}(t) \dot{D}(q) e_{\mathrm{v}}(t) + e^{\mathrm{T}}(t) K_{\mathrm{s}} \big(e_{\mathrm{v}}(t) - \Lambda_{\mathrm{p}} e(t) \big) + \tilde{\theta}^{\mathrm{T}}(t) \Gamma^{-1} \dot{\theta}(t) \\
&= e_{\mathrm{v}}^{\mathrm{T}}(t) \big(W\theta(t) - D(q) \dot{q}_{\mathrm{rv}}(t) - C(q, \dot{q}) \dot{q}(t) - G(q) - K_{\mathrm{s}} e(t) - K_{\mathrm{d}} e_{\mathrm{v}}(t) + \frac{1}{2} \dot{D}(q) e_{\mathrm{v}}(t) \big) \\
&\quad + e^{\mathrm{T}}(t) K_{\mathrm{s}} \big(e_{\mathrm{v}}(t) - \Lambda_{\mathrm{p}} e(t) \big) + \tilde{\theta}^{\mathrm{T}}(t) \Gamma^{-1} \dot{\theta}(t) \\
&= e_{\mathrm{v}}^{\mathrm{T}}(t) \big(W\theta(t) - D(q) \dot{q}_{\mathrm{rv}}(t) - C(q, \dot{q}) q_{\mathrm{rv}}(t) - G(q) - K_{\mathrm{s}} e(t) - K_{\mathrm{d}} e_{\mathrm{v}}(t) \big) \\
&\quad + e_{\mathrm{v}}^{\mathrm{T}}(t) \big(\frac{1}{2} \dot{D}(q) - C(q, \dot{q}) \big) e_{\mathrm{v}}(t) + e^{\mathrm{T}}(t) K_{\mathrm{s}} \big(e_{\mathrm{v}}(t) - \Lambda_{\mathrm{p}} e(t) \big) + \tilde{\theta}^{\mathrm{T}}(t) \Gamma^{-1} \dot{\theta}(t).
\end{aligned} \tag{3.70}$$

Recognizing the skew-symmetry of $\frac{1}{2} \dot{D}(q) - C(q, \dot{q})$ and rearranging terms gives

$$\begin{aligned}
\dot{V}(e, e_{\mathrm{v}}, \tilde{\theta}) &= e_{\mathrm{v}}^{\mathrm{T}}(t) \big(W\theta(t) - D(q) \dot{q}_{\mathrm{rv}}(t) - C(q, \dot{q}) q_{\mathrm{rv}}(t) - G(q) \big) - e_{\mathrm{v}}^{\mathrm{T}}(t) K_{\mathrm{d}} e_{\mathrm{v}}(t) \\
&\quad - e^{\mathrm{T}}(t) K_{\mathrm{s}} \Lambda_{\mathrm{p}} e(t) + \tilde{\theta}^{\mathrm{T}}(t) \Gamma^{-1} \dot{\theta}(t).
\end{aligned} \tag{3.71}$$

Substituting in (3.38) and (3.63) and simplifying results in

$$\begin{aligned}
\dot{V}(e, e_{\mathrm{v}}, \tilde{\theta}) &= e_{\mathrm{v}}^{\mathrm{T}}(t) W \tilde{\theta}(t) - e_{\mathrm{v}}^{\mathrm{T}}(t) K_{\mathrm{d}} e_{\mathrm{v}}(t) - e^{\mathrm{T}}(t) K_{\mathrm{s}} e(t) + \tilde{\theta}^{\mathrm{T}}(t) \Gamma^{-1} \dot{\theta}(t) \\
&= -e_{\mathrm{v}}^{\mathrm{T}}(t) K_{\mathrm{d}} e_{\mathrm{v}}(t) - e^{\mathrm{T}}(t) K_{\mathrm{s}} e(t) + \tilde{\theta}^{\mathrm{T}}(t) \big(\Gamma^{-1} \dot{\theta}(t) + W^{\mathrm{T}} e_{\mathrm{v}}(t) \big).
\end{aligned} \tag{3.72}$$

Finally, substitute in the adaptation law (3.67) to obtain

$$\dot{V}(e, e_{\mathrm{v}}, \tilde{\theta}) = -e_{\mathrm{v}}^{\mathrm{T}}(t) K_{\mathrm{d}} e_{\mathrm{v}}(t) - e^{\mathrm{T}}(t) K_{\mathrm{s}} e(t) \leq 0. \tag{3.73}$$

Since (3.68) is positive definite for all $(e, e_{\mathrm{v}}, \tilde{\theta}) \neq 0$ and (3.73) is negative semidefinite, the error dynamics of the closed-loop system are Lyapunov Stable by Lyapunov's Direct Method [14]. Furthermore, the position and velocity tracking errors are guaranteed to converge to zero as $t \to \infty$ by the LaSalle-Yoshizawa Theorem [17].

3.3.4 Projection Algorithm

The previous stability analysis of the system has shown stable error dynamics for the case of no disturbances. However, if the system experiences bounded disturbances or state measurements are corrupted by noise, a phenomenon known as parameter drift can occur [12]. In this situation, instability arises due to the adaptive parameters going to infinity with time. Instability due to parameter drift can be avoided by modifying the adaptive laws such that the adaptive gains are bounded. The projection algorithm introduced in [22] restricts the trajectories of the adaptive gains to remain bounded inside a convex set. As long as the ideal parameter values are contained within the convex set, the system is guaranteed to be Lyapunov stable. While this is a weaker conclusion than the asymptotic stability of the tracking errors achieved in the case of no disturbances, this result is not undesirable in the context of compliant trajectory tracking. The robot should not be able to reach its exact target destination if a human or other obstacle is obstructing it, as this would compromise the safety of any human interaction with the robot.

Consider the following convex set:

$$\Omega_{\mathrm{c}} = \{\theta \in \mathbb{R}^{\mathrm{r}} \mid f(\theta) \leq c\}, \tag{3.74}$$

where $0 \leq c \leq 1$ and $f(\theta)$ is a continuously differentiable convex function given by

$$f(\theta) = \frac{\theta^{\mathrm{T}}\theta - \theta_{\max}^2}{\epsilon_\theta \theta_{\max}^2}, \tag{3.75}$$

where $\theta_{\max}$ is a nominal bound imposed on the norm of θ, and ϵ_θ is a tuning parameter that determines the boundary of the convex set Ω_{c} where $c = 1$. The projection operator is defined as follows,

$$\mathrm{Proj}(\theta, y) \triangleq \begin{cases} y & \text{if } f(\theta) < 0, \\ y & \text{if } f(\theta) \geq 0 \text{ and } \nabla f^{\mathrm{T}} y \leq 0, \\ y - \frac{\nabla f}{||\nabla f||}\left\langle \frac{\nabla f^{\mathrm{T}}}{||\nabla f||}, y \right\rangle f(\theta) & \text{if } f(\theta) \geq 0 \text{ and } \nabla f^{\mathrm{T}} y > 0 \end{cases} \tag{3.76}$$

When the adaptive law is changed to $\dot{\theta}(t) = \mathrm{Proj}(\theta, y)$ and θ^* is contained in $\Omega_1 = \{\theta \in \mathbb{R}^{\mathrm{r}} \mid f(\theta) \leq 0\}$, the algorithm guarantees $\theta(t)$ is constrained to the set $\Omega_1 = \{\theta \in \mathbb{R}^{\mathrm{r}} \mid f(\theta) \leq 1\}$.

The following equation describes the robot dynamics when disturbances are applied,

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = B(q)u + d(t), \tag{3.77}$$

where $d(t) \in \mathbb{R}^2$ is the vector of disturbances acting on each joint. The disturbance is bounded by d^* such that $||d(t)|| \leq d^*$. Using the Lyapunov function (3.68) and the robot

dynamics equation with disturbances, the Lyapunov function derivative becomes

$$\begin{aligned}\dot{V}(e, e_{\mathrm{v}}, \tilde{\theta}) &= -e_{\mathrm{v}}^{\mathrm{T}}(t) K_{\mathrm{d}} e_{\mathrm{v}}(t) - e^{\mathrm{T}}(t) K_{\mathrm{s}} e(t) + e_{\mathrm{v}}^{\mathrm{T}}(t) d(t) \\ &= -e_{\mathrm{v}}^{\mathrm{T}}(t)\Big(K_{\mathrm{d}} e_{\mathrm{v}}(t) - d(t)\Big) - e^{\mathrm{T}}(t) K_{\mathrm{s}} e(t). \end{aligned} \tag{3.78}$$

Considering the norm bound on $d(t)$ and the norm of the velocity error vector, we obtain the following inequality,

$$\dot{V}(e, e_{\mathrm{v}}, \tilde{\theta}) \leq -||e_{\mathrm{v}}^{\mathrm{T}}(t)||\Big(\lambda_{\min}(K_{\mathrm{d}})||e_{\mathrm{v}}(t)|| - d^{*}\Big) - e^{\mathrm{T}}(t) K_{\mathrm{s}} e(t), \tag{3.79}$$

where $\lambda_{\min}(K_{\mathrm{d}})$ is the minimum eigenvalue of K_{d}. From the inequality (3.79) we can conclude

$$\dot{V}(e, e_{\mathrm{v}}, \tilde{\theta}) \leq 0 \quad \text{if} \quad ||e_{\mathrm{v}}(t)|| \geq \frac{d^{*}}{\lambda_{\min}(K_{\mathrm{d}})}. \tag{3.80}$$

Note that this result indicates that we no longer have asymptotic stability of the origin when the disturbance $d(t) \neq 0$. Instead, (3.80) and the use of the projection algorithm guarantee ultimate boundedness of the system in the presence of bounded disturbances.

3.4 Simulation Results

For initial validation of the controller, a simulation model was developed in Matlab with the open-loop plant given by (3.14), the reference models (3.29), (3.36), tracking error (3.35), (3.37) and the control input and adaptation law (3.65) and (3.67). Figure 3.5 shows a simple block diagram of the controller, which follows the same basic structure as a conventional MRAC system. Rough estimates of the inertial properties were taken from the CAD model and used in the simulations. Numerical integration of the dynamic

model was performed using the ODE45 solver in Matlab. For all of the simulations, the following gains were used: $K_s = \text{diag}([150\ 50])$, $K_d = \text{diag}([15\ 5])$, $\Lambda_p = I$, and $\Gamma = \text{diag}([0.1\ 0.1\ 0.1\ 0.1\ 1\ 1\ 0.5\ 0.5\ 0.5\ 0.5])$. The reference model tuning parameters were as follows, $\omega_{n_1} = \omega_{n_2} = 3$, $\zeta_1 = \zeta_2 = 1$. The robot parameters were $L_1 = 0.4354$ m, $L_2 = 0.418$ m, $l_{cm_1x} = 0.0073$ m, $l_{cm_1y} = 0.2630$ m, $l_{cm_2x} = -0.0477$ m, $l_{cm_2y} = 0.1952$ m, $m_1 = 2.25$ kg, $m_2 = 1.6458$ kg, $J_1 = 0.0280$ kgm^2, $J_2 = 0.0417$ kgm^2, and $g = 9.81\ \frac{\text{m}}{\text{s}^2}$. The initial conditions were $q(0) = [-4\ \ -6]^{\text{T}}$, $\dot{q}(0) = [0\ 0]^{\text{T}}$, and $q_r(0) = q(0)$, where q is given in degrees.

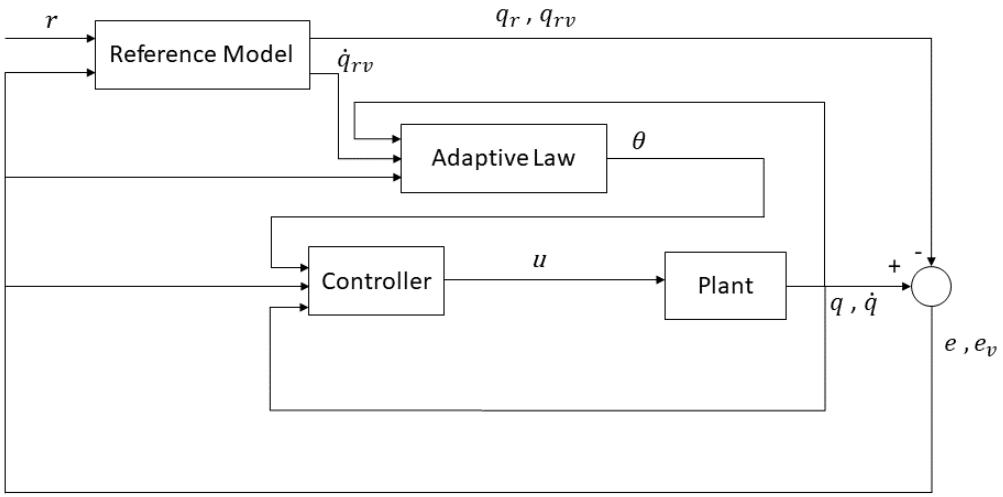

Figure 3.5: Block diagram of the control system used in the simulation of the 2DOF sagittal plane dynamics.

To examine the tracking performance, sinusoidal desired trajectories were used for both the knee and ankle pitch in order to roughly mimic a squatting motion of the robot leg. The initial conditions of the knee and ankle joint were chosen to be offset from the desired trajectories in order to demonstrate the use of the reference model to determine the rate of convergence to the desired joint angles. Simulation results for a sinusoidal trajectory are given in Figures 3.6-3.9. Note that the adaptive parameters do not converge to the ideal

values as this was not guaranteed in the design of the controller nor in the stability analysis. While parameter convergence is possible to achieve by having the desired trajectory fulfil certain requirements for persistent excitation, identification of the parameters is not the goal of this control system and is ultimately not needed.

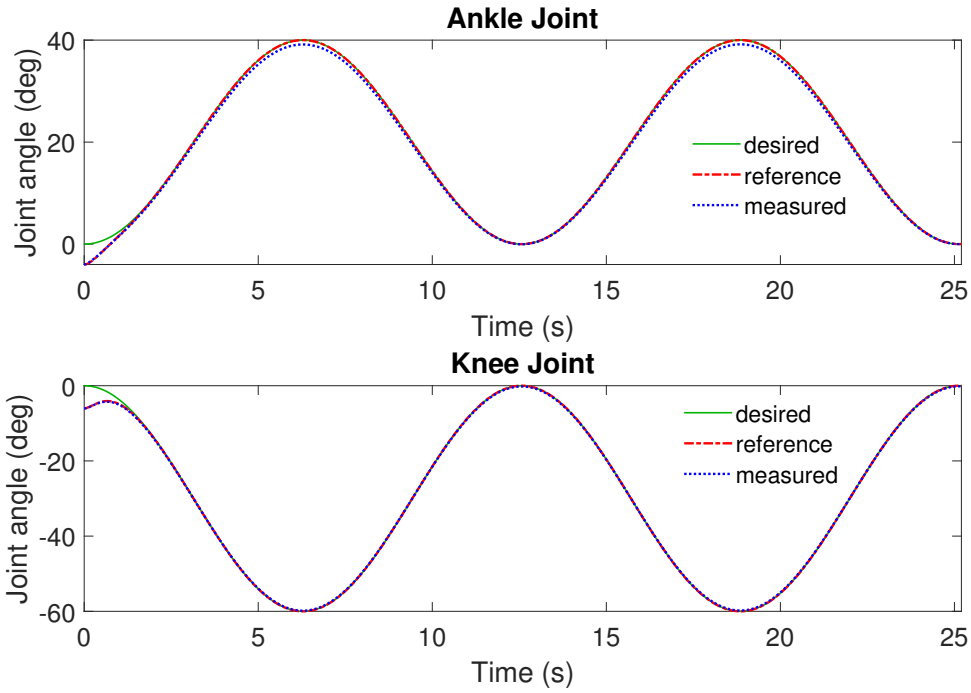

Figure 3.6: Position tracking performance for the knee and ankle joints.

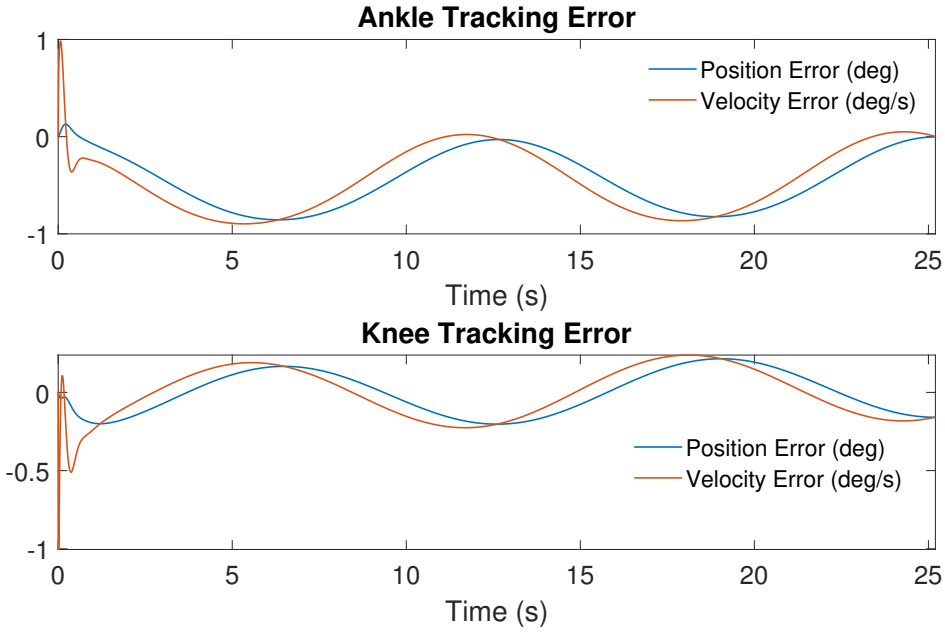

Figure 3.7: Position and velocity tracking errors vs. time for the knee and ankle joints for a sinusoidal trajectory.

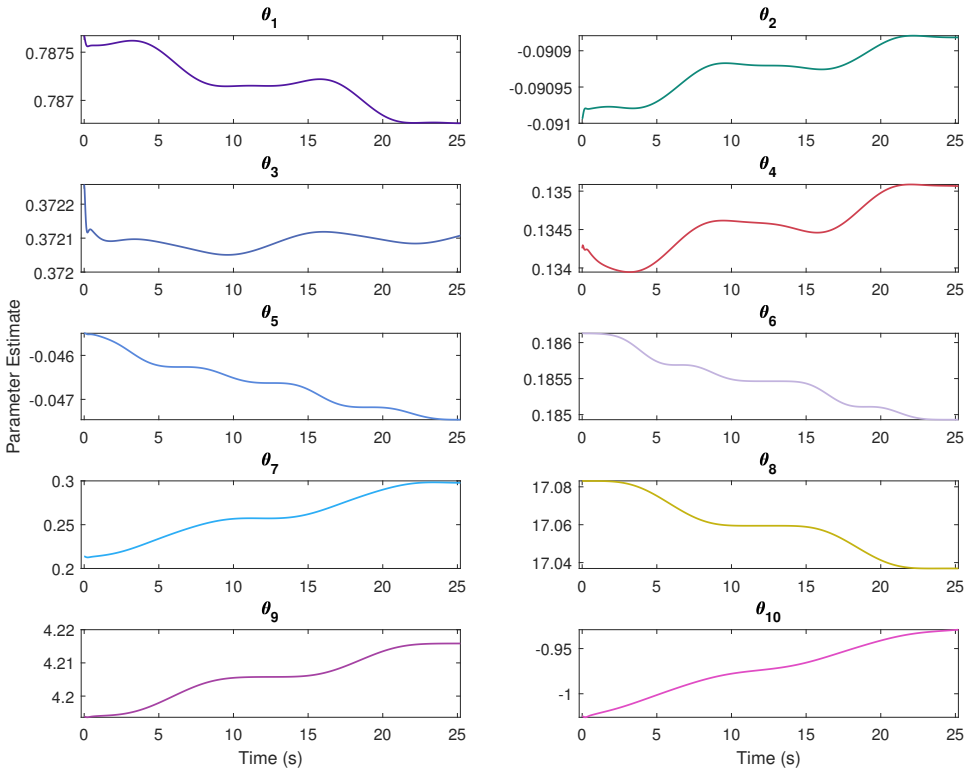

Figure 3.8: Change in adaptive parameters over time.

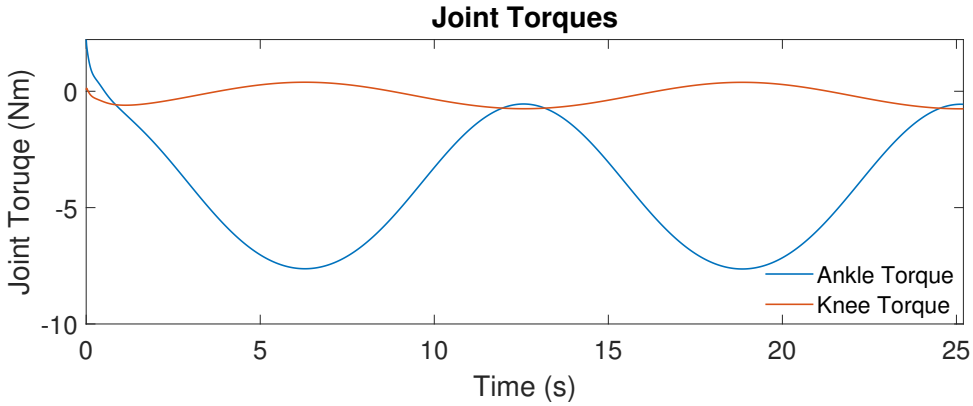

Figure 3.9: Joint torques for a sinusoidal trajectory.

The robot's compliant behavior when acted upon by external forces was also examined through simulations. Disturbances were modeled as a time-varying force input acting upon the hip (point H in Figure 3.1). The robot dynamics equation is therefore modified to include the disturbance force, resulting in

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = B(q)u + J_{\mathrm{H}}^{\mathrm{T}}(q)F_{\mathrm{d}}, \tag{3.81}$$

where $J_{\mathrm{H}}(q) \in \mathbb{R}^{2\times 2}$ is the hip Jacobian and $F_{\mathrm{d}}(t) \in \mathbb{R}^2$ is the force vector expressed in Cartesian coordinates. The applied forces were 100 N in magnitude and 0.25 s in duration, acting parallel to either the x or y-axis of the world coordinate frame. Simulation results for position regulation and trajectory tracking with disturbances are given in Figures 3.10-3.15.

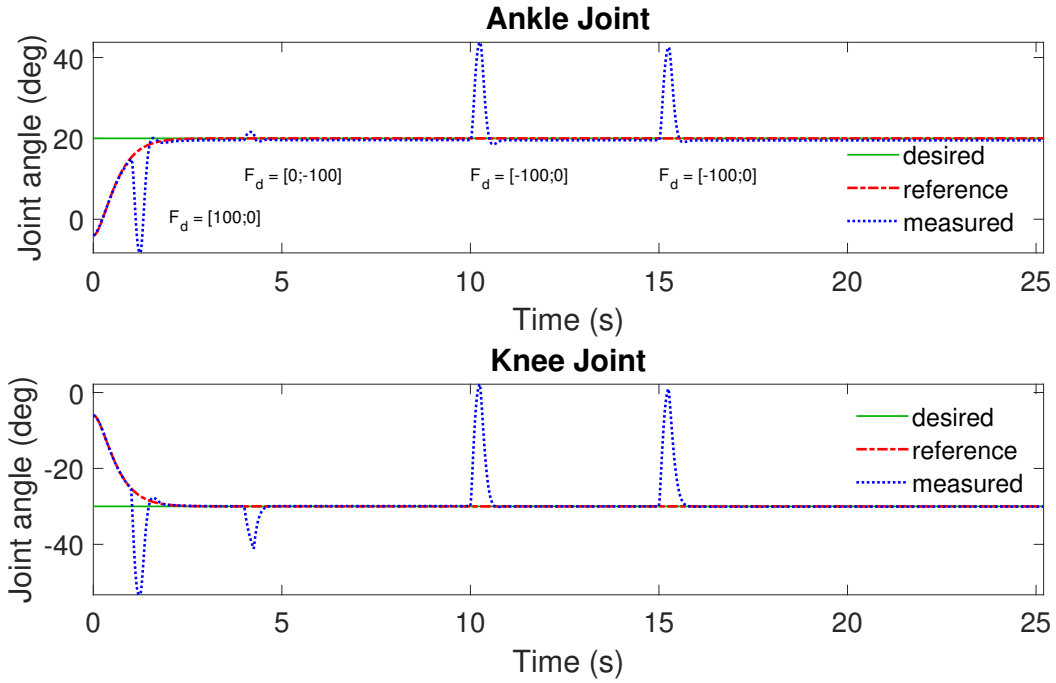

Figure 3.10: Position regulation with disturbances. Disturbances were applied at the hip pitch axis, and are labeled in the ankle pitch plot for clarity.

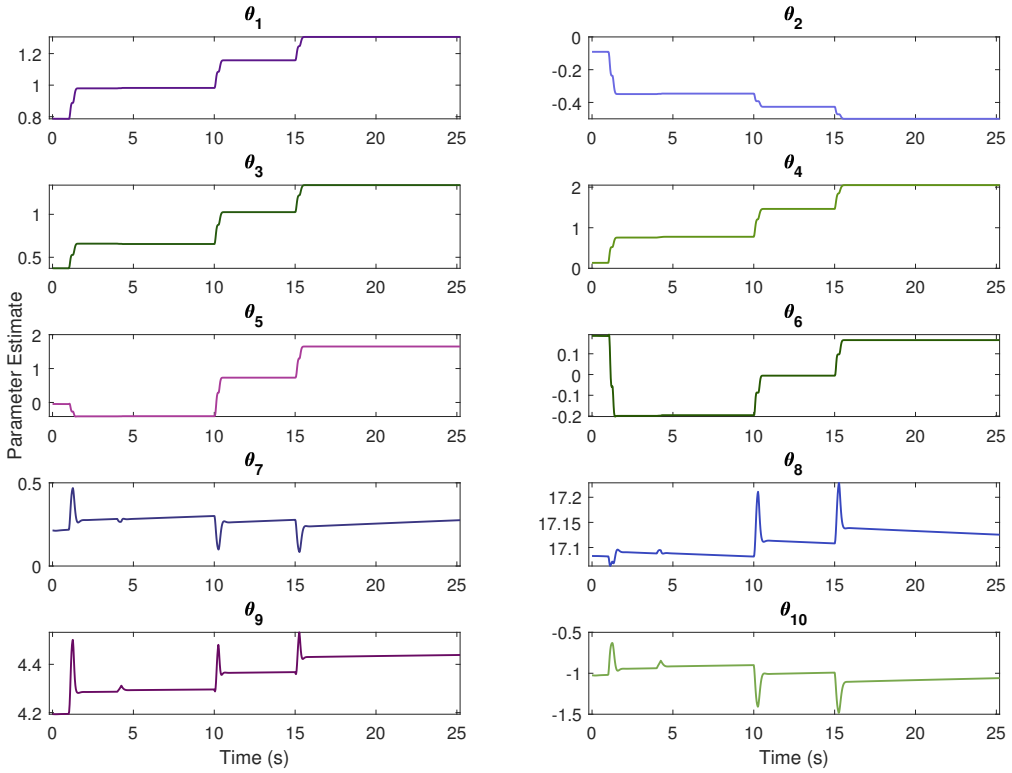

Figure 3.11: Evolution of the adaptive parameters during position regulation with disturbances.

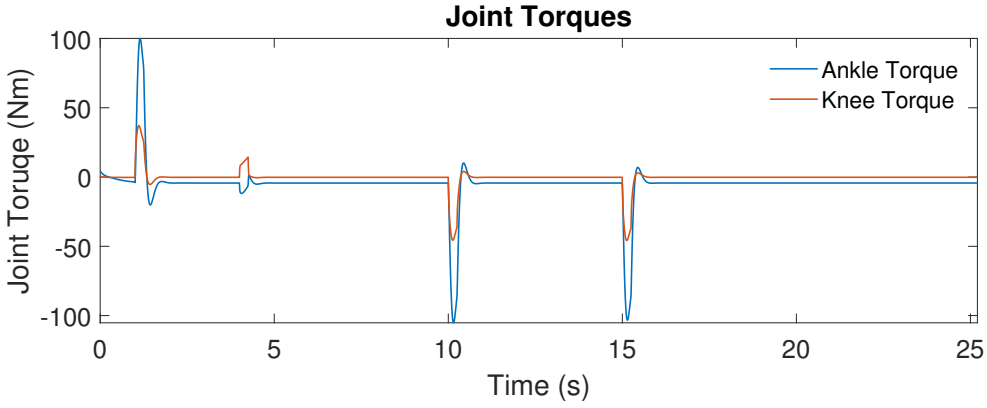

Figure 3.12: Joint torques during position regulation with disturbances.

The simulations show excellent tracking performance in the case of no disturbances, with the position tracking errors not exceeding $\pm 1^\circ$ for sinusoidal trajectories with amplitudes

of $-20°$ and $30°$ at the ankle and knee joints respectively. The robot is shown to give way when acted upon by an external force, then move back to position following the behavior of a mass-spring-damper system. This is observed for both position regulation and trajectory tracking. Therefore it can be concluded that in the ideal case of simulations, this controller achieves accurate and compliant trajectory tracking even when the parameters of the robot are unknown, and may be feasible to implement in an experimental solution.

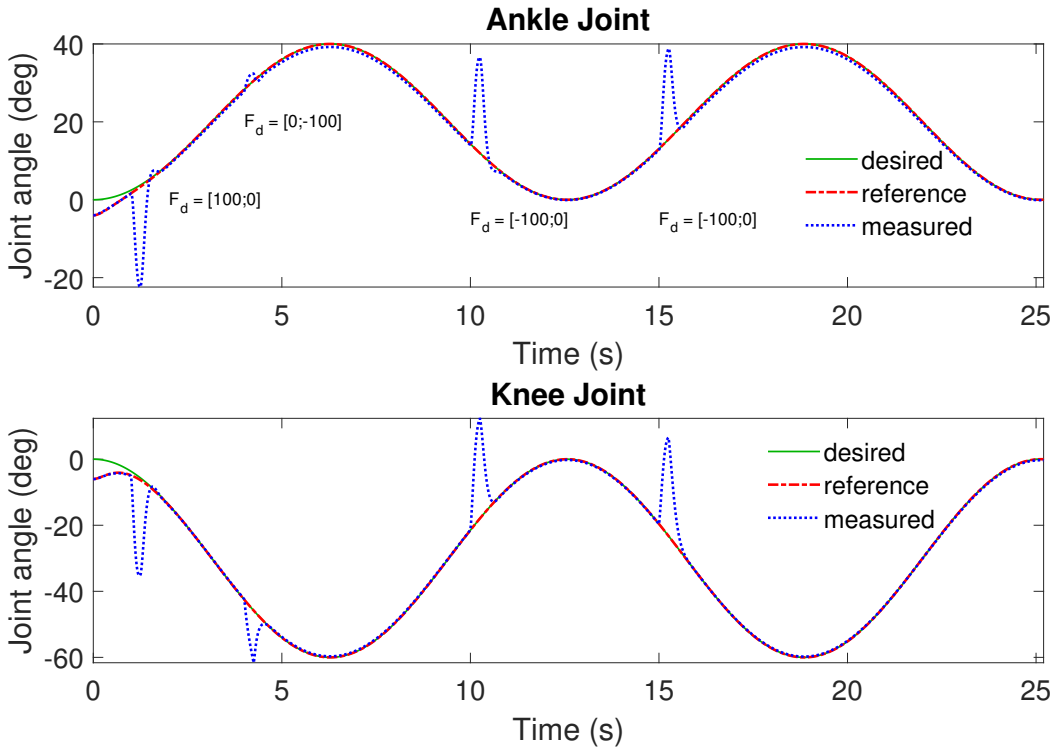

Figure 3.13: Trajectory tracking with disturbances. Disturbances are labeled in the ankle pitch plot for clarity.

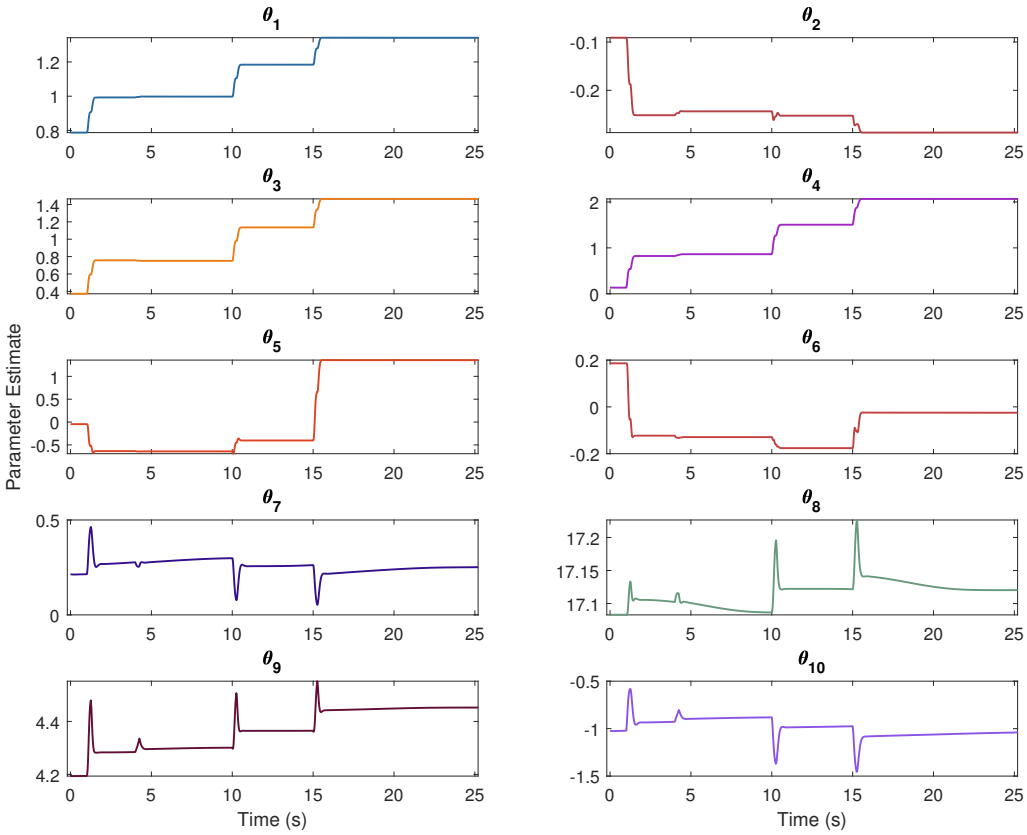

Figure 3.14: Evolution of the adaptive parameters during trajectory tracking with disturbances.

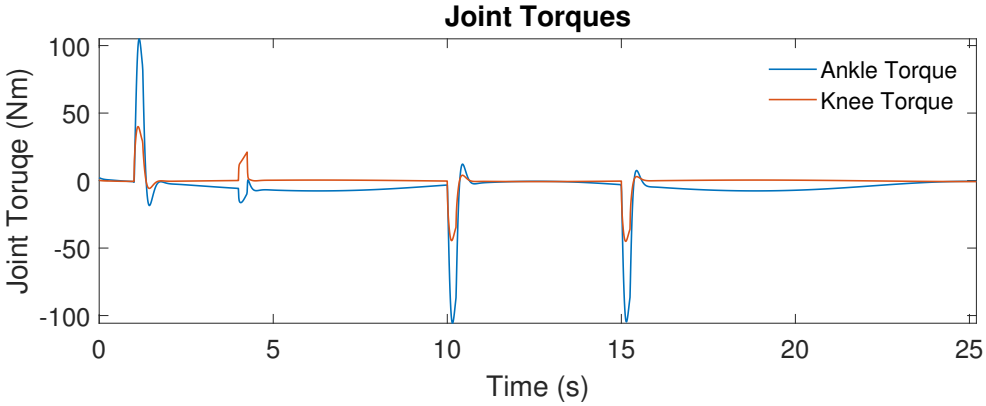

Figure 3.15: Joint torques during trajectory tracking with disturbances.

Chapter 4

Low-Level Force Control

In order to ensure that the actuator forces commanded by the control law derived in Chapter 3 are achieved, an actuator force control loop is utilized. This system will be referred to as the low-level force controller. The low-level controller is designed to drive the motor current such that the measured actuator force converges to the commanded force in between time steps of the joint torque controller. Each actuator will have its own independent force control loop. The low-level control system works as follows: the force command is received as an input, an error signal is determined by comparing the desired force to the measured force, and the controller determines a commanded current to be sent to the actuator. The commanded current is received by the servo driver, which applies a voltage to the motor in order to achieve the desired output current. This current drives the motor to exert a torque, which is translated to a linear force through a ball screw mechanism, which is measured directly by an inline load cell. For the purpose of this analysis, convergence of the servo drive to the commanded current is considered instantaneous. Figure 4.1 gives a block diagram of the actuator and force controller system.

The focus of this chapter will be on the design and evaluation of three different low-level force control algorithms: PI control, Direct Model Reference Adaptive Control (MRAC) and Adaptive Augmented PI (ADP-PI) control.

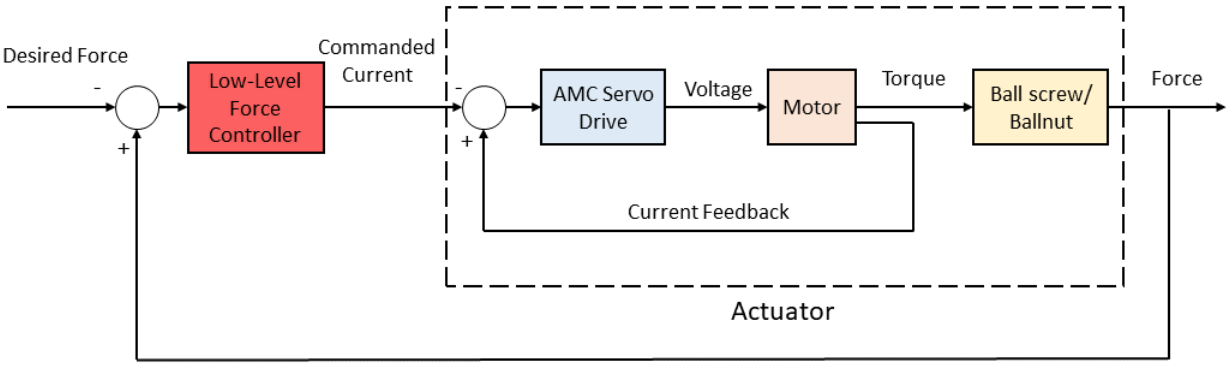

Figure 4.1: Block diagram of the actuator and low-level force controller system.

4.1 PI Control

The most basic of the controllers considered for low-level force tracking is a common PI controller. This was favored over a traditional PID controller due to the significant presence of noise in the loadcell measurements as well as the time delay resulting from numerical differentiation. Minimal processing power is required to compute the control input as neither an actuator model nor adaptation are necessary. Any controller that uses integrator action is vulnerable to integrator windup, a situation that arises from the integral action becoming abnormally large when the control input is saturated, resulting in undesirable oscillations in the system [2]. Since the actuators are operating far below their saturation limits in this application, anti-windup measures were not implemented. The force tracking error is defined as

$$e_{\mathrm{f}}(t) \triangleq x(t) - x_{\mathrm{d}}(t), \tag{4.1}$$

where $x(t) \in \mathbb{R}$ is the measured actuator force, and $x_{\mathrm{d}}(t) \in \mathbb{R}$ is the desired force. The control law is then given by

$$u(t) = k_{\mathrm{p}} e_{\mathrm{f}}(t) + k_{\mathrm{I}} \int_0^t e_{\mathrm{f}}(t)\mathrm{d}t, \tag{4.2}$$

where k_{p} and k_{I} are positive and constant proportional and integral gains. A block diagram of the control system is given in Figure 4.2.

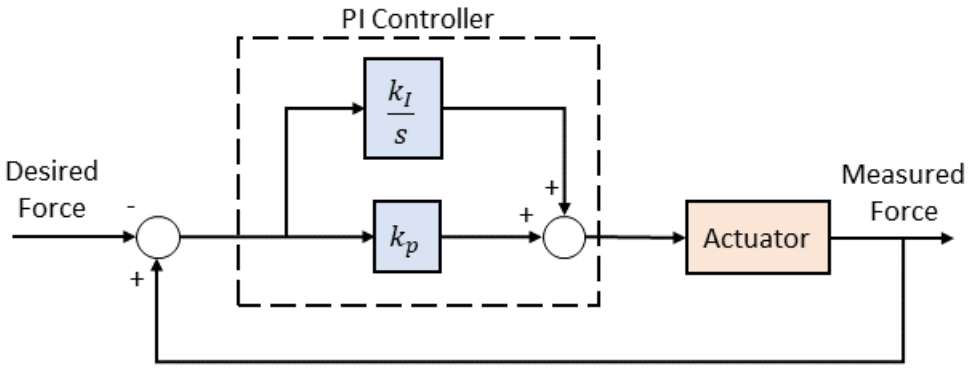

Figure 4.2: Block diagram of the PI controller used for low-level force control.

4.2 MRAC

The Direct MRAC scheme presented in [13] and based on the scalar adaptive tracking example presented in [12] was also considered. The Direct MRAC used in this application assumes first order actuator dynamics, and drives the measured state variable to a first order reference model. The reference model is defined as follows,

$$\dot{x}_{\mathrm{r}}(t) = a_{\mathrm{m}} x_{\mathrm{r}}(t) + b_{\mathrm{m}} r(t), \tag{4.3}$$

where $x_{\mathrm{r}}(t)$ is the reference model state variable, $a_{\mathrm{m}} < 0$ and b_{m} are known tuning parameters and $r(t)$ is the reference input, which is chosen such that the reference model achieves the desired tracking performance of the system. Assume $x(t)$ has the following dynamics,

$$\dot{x}(t) = ax(t) + bu(t), \tag{4.4}$$

where the model parameters a and b are unknown, but the sign of b is known and $b \neq 0$. Consider the ideal control law given by

$$u^*(t) = \theta_x^* x(t) + \theta_{\mathrm{r}}^* r(t), \tag{4.5}$$

where the ideal gains θ_x^* and θ_{r}^* are

$$\theta_x^* = \frac{a_{\mathrm{m}} - a}{b}, \tag{4.6}$$

$$\theta_{\mathrm{r}}^* = \frac{b_{\mathrm{m}}}{b}. \tag{4.7}$$

In order to verify that the ideal control law drives the system to the reference model, we consider the error dynamics. Define the tracking error as follows,

$$e(t) \triangleq x(t) - x_{\mathrm{r}}(t). \tag{4.8}$$

Differentiating (4.8) with respect to time yields

$$\dot{e}(t) = \dot{x}(t) - \dot{x}_{\mathrm{r}}(t) \tag{4.9}$$

$$= ax(t) + bu(t) - a_{\mathrm{m}} x_{\mathrm{r}}(t) - b_{\mathrm{m}} r(t). \tag{4.10}$$

Substituting the ideal control law (4.5) as the control input yields the following,

$$\begin{aligned} \dot{e}(t) &= ax(t) + b\Big(\big(\frac{a_{\mathrm{m}} - a}{b}\big)x(t) + \frac{b_{\mathrm{m}}}{b} r(t)\Big) - a_{\mathrm{m}} x_{\mathrm{r}}(t) - b_{\mathrm{m}} r(t) \\ &= a_{\mathrm{m}}(x(t) - x_{\mathrm{r}}(t)) \\ &= a_{\mathrm{m}} e(t). \end{aligned} \tag{4.11}$$

Since $a_{\mathrm{m}} < 0$, the ideal control law drives the system to the reference model exponentially as $t \to \infty$. Since the model parameters a and b are not explicitly known the ideal control law cannot be implemented. Therefore the ideal control law is modified, resulting in

$$u(t) = \theta_x(t)x(t) + \theta_{\mathrm{r}}(t)r(t), \tag{4.12}$$

where $\theta_x(t)$ and $\theta_{\mathrm{r}}(t)$ are the adaptive gains. Next, define the parameter errors $\tilde{\theta}_x(t) = \theta_x(t) - \theta_x^*$ and $\tilde{\theta}_{\mathrm{r}}(t) = \theta_{\mathrm{r}}(t) - \theta_{\mathrm{r}}^*$. With the control law (4.12) the error dynamics become

$$\dot{e}(t) = ax(t) + b(\theta_x(t)x(t) + \theta_{\mathrm{r}}(t)r(t)) - a_{\mathrm{m}}x_{\mathrm{r}}(t) - b_{\mathrm{m}}r(t). \tag{4.13}$$

Substituting in (4.6), (4.7) and the definitions of the parameter errors yields

$$\begin{aligned}
\dot{e}(t) &= ax(t) + b(\theta_x(t)x(t) + \theta_{\mathrm{r}}(t)r(t)) - a_{\mathrm{m}}x_{\mathrm{r}}(t) - b_{\mathrm{m}}r(t) \\
&= (a + b\theta_x^*)x(t) + b\theta_{\mathrm{r}}^*r(t) + b\tilde{\theta}_x(t)x(t) + b\tilde{\theta}_{\mathrm{r}}(t) - a_{\mathrm{m}}x_{\mathrm{r}}(t) - b_{\mathrm{m}}r(t) \\
&= a_{\mathrm{m}}x(t) + b_{\mathrm{m}}r(t) + b\tilde{\theta}_x(t)x(t) + b\tilde{\theta}_{\mathrm{r}}(t)r(t) - a_{\mathrm{m}}x_{\mathrm{r}}(t) - b_{\mathrm{m}}r(t) \\
&= a_{\mathrm{m}}e(t) + b\tilde{\theta}_x(t)x(t) + b\tilde{\theta}_{\mathrm{r}}(t)r(t).
\end{aligned} \tag{4.14}$$

Update laws for the adaptive gains are chosen to guarantee stability of the closed-loop system. This is done by constructing a Lyapunov function candidate, finding its time derivative, and choosing adaptive update laws to ensure the Lyapunov function derivative is negative semi-definite. Consider the following Lyapunov function candidate,

$$V(e, \tilde{\theta}_x, \tilde{\theta}_{\mathrm{r}}) = \frac{1}{2}e(t)^2 + \frac{|b|}{2\gamma_x}\tilde{\theta}_x(t)^2 + \frac{|b|}{2\gamma_{\mathrm{r}}}\tilde{\theta}_{\mathrm{r}}(t)^2, \tag{4.15}$$

where $\gamma_x, \gamma_r > 0$ are tuning parameters. Differentiating with respect to time yields

$$\dot{V}(e, \tilde{\theta}_x, \tilde{\theta}_\mathrm{r}) = e(t)\dot{e}(t) + \frac{|b|}{\gamma_x}\tilde{\theta}_x(t)\dot{\tilde{\theta}}_x(t) + \frac{|b|}{\gamma_\mathrm{r}}\tilde{\theta}_\mathrm{r}(t)\dot{\tilde{\theta}}_\mathrm{r}(t). \tag{4.16}$$

Recognizing that $\dot{\tilde{\theta}}_x(t) = \tilde{\theta}_x(t)$ and $\dot{\tilde{\theta}}_\mathrm{r}(t) = \tilde{\theta}_\mathrm{r}(t)$ and substituting in (4.14) results in the following,

$$\begin{aligned}\dot{V}(e, \tilde{\theta}_x, \tilde{\theta}_\mathrm{r}) &= e(t)(a_\mathrm{m}e(t) + b\tilde{\theta}_x(t)x(t) + b\tilde{\theta}_\mathrm{r}(t)r(t)) + \frac{|b|}{\gamma_x}\tilde{\theta}_x(t)\dot{\theta}_x(t) + \frac{|b|}{\gamma_\mathrm{r}}\tilde{\theta}_\mathrm{r}(t)\dot{\theta}_\mathrm{r}(t) \\ &= a_\mathrm{m}e(t)^2 + \tilde{\theta}_x(t)\big(bx(t)e(t) + \frac{|b|}{\gamma_x}\dot{\theta}_x(t)\big) + \tilde{\theta}_\mathrm{r}(t)\big(br(t)e(t) + \frac{|b|}{\gamma_\mathrm{r}}\dot{\theta}_\mathrm{r}(t)\big).\end{aligned} \tag{4.17}$$

Choosing the adaptive laws $\dot{\theta}_x(t) = -\gamma_x \mathrm{sign}(b)x(t)e(t)$ and $\dot{\theta}_\mathrm{r}(t) = -\gamma_\mathrm{r}\mathrm{sign}(b)r(t)e(t)$ results in

$$\dot{V}(e, \tilde{\theta}_x, \tilde{\theta}_\mathrm{r}) = a_\mathrm{m}e(t)^2 \leq 0, \tag{4.18}$$

where $a_\mathrm{m} < 0$ by definition. Therefore the system is Lyapunov stable by Lyapunov's Direct Method [14] and the tracking error converges to zero as $t \to \infty$ by the LaSalle-Yoshizawa theorem [17]. The MRAC algorithm presented in this section is summarized in Figure 4.3.

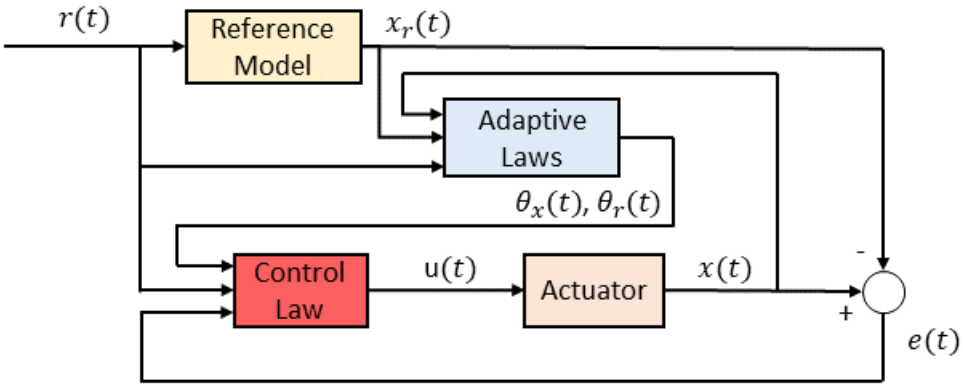

Figure 4.3: Block diagram showing the MRAC scheme used for the low-level force controller.

4.3 ADP-PI Control

The ADP-PI algorithm presented in [13], [4] was also used in the experiments. The ADP-PI controller is essentially a combination of the PI and MRAC algorithms described previously in this chapter. In addition, the ADP-PI controller takes into account potential nonlinearities in the system by adapting with respect to matched uncertainties in the state equation. The proportional and integral gains are also modified in the adaptation algorithm in order to achieve convergence to the reference model. A block diagram of the ADP-PI algorithm is given in Figure 4.4. The actuator is assumed to have the following first order dynamics,

$$\dot{x}(t) = ax(t) + b\lambda^*(u(t) + W^{*\mathrm{T}}\phi(x(t)), \tag{4.19}$$

where $a, b \in \mathbb{R}$ are known constants, λ^* is of unknown magnitude but the sign is known, $W^* \in \mathbb{R}^{\mathrm{p}}$ is a vector of unknown constants, $\phi(x(t)) \in \mathbb{R}^{\mathrm{p}}$ is a vector of known functions of the state variable, and $u(t) \in \mathbb{R}$ is the control input. The state vector is then augmented

with an additional state representing the integral error, resulting in

$$x_a(t) = \begin{bmatrix} x(t) \\ x_{\mathrm{I}}(t) \end{bmatrix}, \tag{4.20}$$

where $x_a(t)$ is the augmented state vector and $x_{\mathrm{I}}(t) = \int_0^t (x(\tau) - x_d(\tau))d\tau$, where $x_d(t)$ is the desired force. The dynamics of the augmented system are given by

$$\dot{x}_{\mathrm{a}}(t) = \begin{bmatrix} a & 0 \\ 1 & 0 \end{bmatrix} x_{\mathrm{a}}(t) + \begin{bmatrix} b \\ 0 \end{bmatrix} \lambda^*(u(t) + W^{*\mathrm{T}}\phi(x(t))) + \begin{bmatrix} 0 \\ -1 \end{bmatrix} x_{\mathrm{d}}(t). \tag{4.21}$$

The reference model used for tracking control consists of the ideal linear component of the dynamics in (4.19) with a PI controller. Therefore the reference model closed-loop dynamics are as follows,

$$\dot{x}_{\mathrm{L}}(t) = A_{\mathrm{r}} x_{\mathrm{L}}(t) + \begin{bmatrix} bk_{\mathrm{p}} \\ -1 \end{bmatrix} x_{\mathrm{d}}(t), \tag{4.22}$$

$$A_{\mathrm{r}} = \begin{bmatrix} a & 0 \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} b \\ 0 \end{bmatrix} K^{*\mathrm{T}}, \tag{4.23}$$

where $K^* = [k_{\mathrm{p}} \;\; k_{\mathrm{I}}]^{\mathrm{T}}$ is the vector of ideal gains used in the PI component of the algorithm. Next we define the following tracking error,

$$e_{\mathrm{a}}(t) = x_{\mathrm{a}}(t) - x_{\mathrm{L}}(t). \tag{4.24}$$

Differentiating (4.24) with respect to time yields the following error dynamics,

$$\begin{aligned}
\dot{e}_{\mathrm{a}}(t) &= \dot{x}_{\mathrm{a}}(t) - \dot{x}_{\mathrm{L}}(t) \\
&= \begin{bmatrix} a & 0 \\ 1 & 0 \end{bmatrix} x_{\mathrm{a}}(t) + \begin{bmatrix} b \\ 0 \end{bmatrix} \lambda^{*}(u(t) + W^{*\mathrm{T}}\phi(x(t))) + \begin{bmatrix} 0 \\ -1 \end{bmatrix} x_{\mathrm{d}}(t) - A_{\mathrm{r}} x_{\mathrm{L}}(t) - \begin{bmatrix} bk_{\mathrm{p}} \\ -1 \end{bmatrix} x_{\mathrm{d}}(t) \\
&= A_{\mathrm{r}} e_{\mathrm{a}}(t) + \left(\begin{bmatrix} a & 0 \\ 1 & 0 \end{bmatrix} - A_{\mathrm{r}} \right) x_{\mathrm{a}}(t) + \begin{bmatrix} b \\ 0 \end{bmatrix} \lambda^{*}(u(t) + W^{*\mathrm{T}}\phi(x(t)) + \begin{bmatrix} -bk_{\mathrm{p}} \\ 0 \end{bmatrix} x_{\mathrm{d}}(t) \\
&= A_{\mathrm{r}} e_{\mathrm{a}}(t) + \begin{bmatrix} b \\ 0 \end{bmatrix} \lambda^{*} \Big(u(t) + \frac{1}{\lambda^{*}} K^{*\mathrm{T}} \Big(x_{\mathrm{a}}(t) - \begin{bmatrix} x_{\mathrm{d}}(t) \\ 0 \end{bmatrix} \Big) + W^{*\mathrm{T}}\phi(x(t)) \Big). \qquad (4.25)
\end{aligned}$$

The ideal control input is

$$u^{*}(t) = u_{\mathrm{L}}(t) + u_{\mathrm{ad}}^{*}(t) \qquad (4.26)$$

Where $u_{\mathrm{L}}(t)$ and $u_{\mathrm{ad}}^{*}(t)$ are the linear and adaptive components of the ideal controller given by

$$u_{\mathrm{L}}(t) = -K^{*\mathrm{T}} \Big(x_{\mathrm{L}}(t) - \begin{bmatrix} x_{\mathrm{d}}(t) \\ 0 \end{bmatrix} \Big), \qquad (4.27)$$

$$u_{\mathrm{ad}}^{*}(t) = -W^{*\mathrm{T}}\phi(x(t)) - \frac{1 - \lambda^{*}}{\lambda^{*}} K^{*\mathrm{T}} \Big(x_{\mathrm{a}}(t) - \begin{bmatrix} x_{\mathrm{d}}(t) \\ 0 \end{bmatrix} \Big). \qquad (4.28)$$

Substituting (4.26) into (4.25) results in the simplified error dynamics

$$\dot{e}_{\mathrm{a}}(t) = A_{\mathrm{r}} e_{\mathrm{a}}(t), \qquad (4.29)$$

where A_{r} is Hurwitz by design. Since W^* and $\frac{1-\lambda^*}{\lambda^*}K^{*\mathrm{T}}$ are unknown, the ideal control law (4.26) is modified to be adaptive. The adaptive control law is

$$u_{\mathrm{ad}}(t) = -\hat{W}^{\mathrm{T}}(t)\phi(x(t)) - \hat{K}^{\mathrm{T}}(t)\Big((x_{\mathrm{a}}(t) - \begin{bmatrix} x_{\mathrm{d}}(t) \\ 0 \end{bmatrix}\Big), \tag{4.30}$$

where $\hat{W}(t) \in \mathbb{R}^{\mathrm{p}}$ and $\hat{K}(t) \in \mathbb{R}^2$ are the adaptive gains, and the parameter error is given by

$$\tilde{W}(t) = \hat{W}(t) - W^*, \tag{4.31}$$

$$\tilde{K}(t) = \hat{K}(t) - \frac{1-\lambda^*}{\lambda^*}K^{*\mathrm{T}}. \tag{4.32}$$

The adaptive update laws are as follows,

$$\dot{\hat{W}}(t) = \mathrm{sign}(\lambda^*)\Gamma_W \phi(x(t)) e_{\mathrm{a}}^{\mathrm{T}}(t) P \begin{bmatrix} b \\ 0 \end{bmatrix}, \tag{4.33}$$

$$\dot{\hat{K}}(t) = \mathrm{sign}(\lambda^*)\Gamma_K \Big(x_{\mathrm{a}}(t) - \begin{bmatrix} x_{\mathrm{d}}(t) \\ 0 \end{bmatrix}\Big) e_{\mathrm{a}}^{\mathrm{T}}(t) P \begin{bmatrix} b \\ 0 \end{bmatrix}, \tag{4.34}$$

where $\Gamma_W \in \mathbb{R}$ and $\Gamma_K \in \mathbb{R}$ are the adaptive gains, and P is the solution to the Lyapunov Equation $A_{\mathrm{r}}^{\mathrm{T}}P + PA_{\mathrm{r}} + Q = 0$, where Q is a positive definite matrix. Substituting $u(t) = u_{\mathrm{L}}(t) + u_{\mathrm{ad}}(t)$ as the control input and the definitions of $\tilde{W}(t)$ and $\tilde{K}(t)$ into (4.25), the error dynamics equation is simplified to

$$\dot{e}_{\mathrm{a}}(t) = A_{\mathrm{r}} e_{\mathrm{a}}(t) + \begin{bmatrix} b \\ 0 \end{bmatrix} \lambda^* \Bigg(- \tilde{W}^{\mathrm{T}}(t)\phi(x(t)) - \tilde{K}^{\mathrm{T}}(t)\Big(x_{\mathrm{a}}(t) - \begin{bmatrix} x_{\mathrm{d}}(t) \\ 0 \end{bmatrix}\Big)\Bigg). \tag{4.35}$$

In order to prove stability of the closed-loop system, Lyapunov's Direct Method [14] is utilized. Consider the Lyapunov function candidate,

$$V(e_{\mathrm{a}}, \tilde{W}, \tilde{K}) = e_{\mathrm{a}}^{\mathrm{T}}(t) P e_{\mathrm{a}}(t) + |\lambda^*| \tilde{K}^{\mathrm{T}}(t) \Gamma_K^{-1} \tilde{K}(t) + |\lambda^*| \tilde{W}^{\mathrm{T}}(t) \Gamma_W^{-1} \tilde{W}(t). \tag{4.36}$$

Differentiating with respect to time and substituting in (4.35) yields

$$\begin{aligned} \dot{V}(e_{\mathrm{a}}, \tilde{W}, \tilde{K}) = {} & e_{\mathrm{a}}^{\mathrm{T}}(t)(A_{\mathrm{r}}^{\mathrm{T}} P + P A_{\mathrm{r}}) e_{\mathrm{a}}(t) \\ & - 2 e_{\mathrm{a}}^{\mathrm{T}}(t) P \begin{bmatrix} b \\ 0 \end{bmatrix} \lambda^* \Big(\tilde{W}^{\mathrm{T}}(t) \phi(x(t)) + \tilde{K}^{\mathrm{T}}(t) \Big(x_{\mathrm{a}}(t) - \begin{bmatrix} x_{\mathrm{d}}(t) \\ 0 \end{bmatrix} \Big) \Big) \\ & + 2 |\lambda^*| \tilde{K}^{\mathrm{T}}(t) \Gamma_K^{-1} \dot{\hat{K}}(t) + 2 |\lambda^*| \tilde{W}^{\mathrm{T}}(t) \Gamma_W^{-1} \dot{\hat{W}}(t). \end{aligned} \tag{4.37}$$

After rearranging terms and substituting in (4.33) and (4.34), the Lyapunov function derivative is simplified to

$$\dot{V}(e_{\mathrm{a}}, \tilde{W}, \tilde{K}) = -e_{\mathrm{a}}^{\mathrm{T}}(t) Q e_{\mathrm{a}} \leq 0. \tag{4.38}$$

Therefore the closed-loop system is Lyapunov Stable and the tracking error $e_{\mathrm{a}}(t)$ converges to zero as $t \to \infty$ by the LaSalle-Yoshizawa theorem [17].

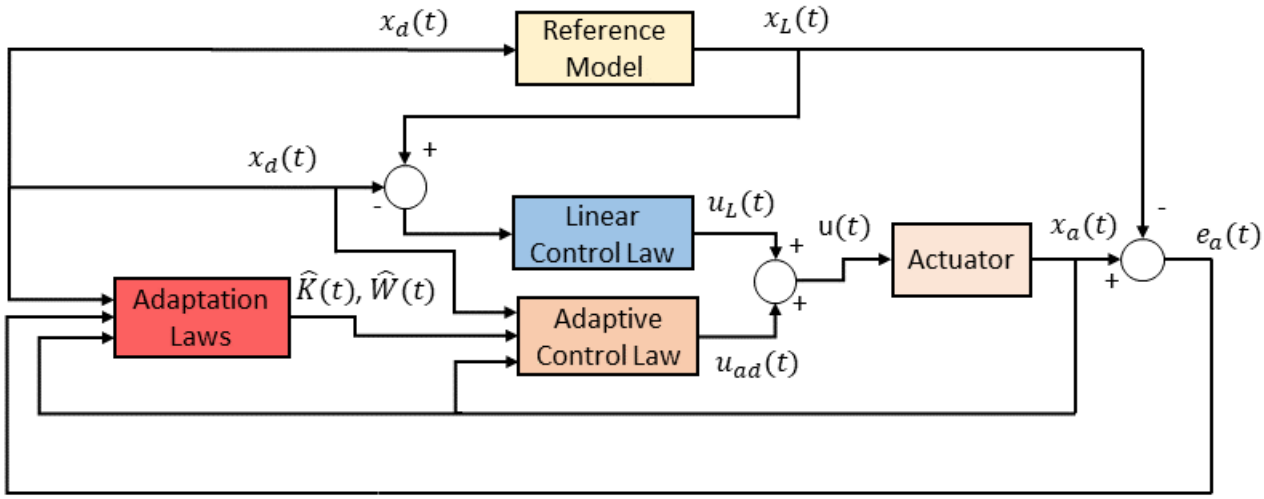

Figure 4.4: Block diagram of the ADP-PI control scheme.

4.4 Force Control Experiments

In order to test the feasibility of the different low-level force control algorithms derived in this chapter, two experimental setups were utilized. The first setup consists of an actuator test stand which holds the actuator with the output fixed. The fixed output condition is similar to how an actuator would operate on the leg that is in contact with the ground and supporting the full weight of the robot, known as the stance leg. An image of the actuator test stand is given in Figure 4.5. In the second set of experiments, the leg was inverted and suspended from a gantry. The shin link was clamped to the gantry as rigidly as possible and was additionally supported by ratcheting straps. This setup is shown in Figure 4.14. The inverted leg setup was intended to mimic the conditions of a moving output, which is similar to how the actuator will operate on the swing leg, i.e. the leg that is swinging forward during walking and has not yet made contact with the ground. The main advantage to having the leg be inverted rather than upright is that the system is open-loop stable. An upright leg can be thought of as an inverted pendulum, which has an unstable equilibrium point in the

upright position. This system would require careful choice of the actuator force trajectories as the leg can easily go unstable and fall over. Using the upside down leg allowed us to work with the stable equilibrium point, as the zero angle corresponds to the bottom of a pendulum's range of motion.

4.4.1 Actuator Test Stand Results

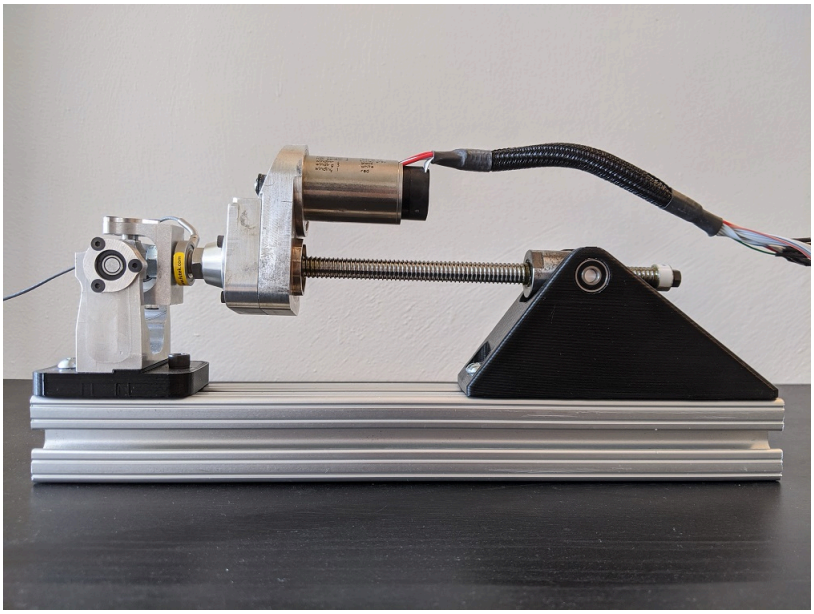

Figure 4.5: Actuator test stand with a fixed output as used in the experiments. In this setup, the actuator is bolted to an aluminum extrusion using 3D-printed adapters. This is meant to emulate the case of the actuator output being approximately fixed during the stance phase of walking.

PI Control

The force tracking performance of the PI controller was tested with a series of trajectories involving sine waves of varying frequencies and amplitudes, as well as step inputs of varying sizes. In this section, results from two notable experiments are presented. Figure 4.6 shows

the experimental results to a step input with the following gains: $k_{\mathrm{p}} = 0.2$, $k_{\mathrm{I}} = 0.4$. Figure 4.7 shows results from tracking a sinusoidal trajectory with the following gains: $k_{\mathrm{p}} = 0.6$, $k_{\mathrm{I}} = 0.5$.

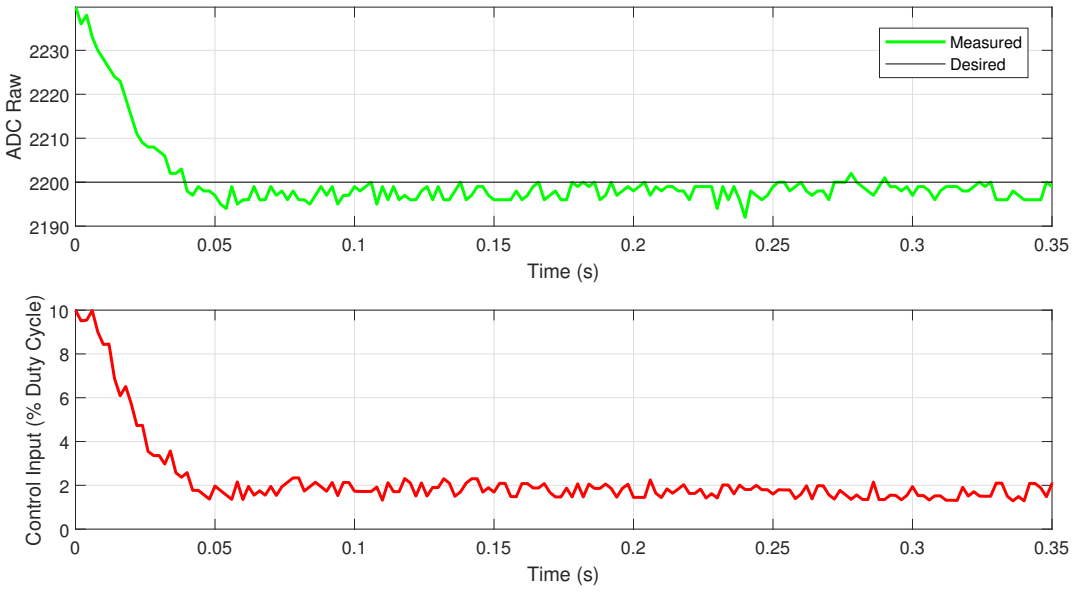

Figure 4.6: Step response with PI controller. Note that the measured ADC values include software lowpass filtering. The difference between the starting value (2240) and the commanded value (2200) corresponds to approximately 56.5 N of force based on the loadcell calibration curve.

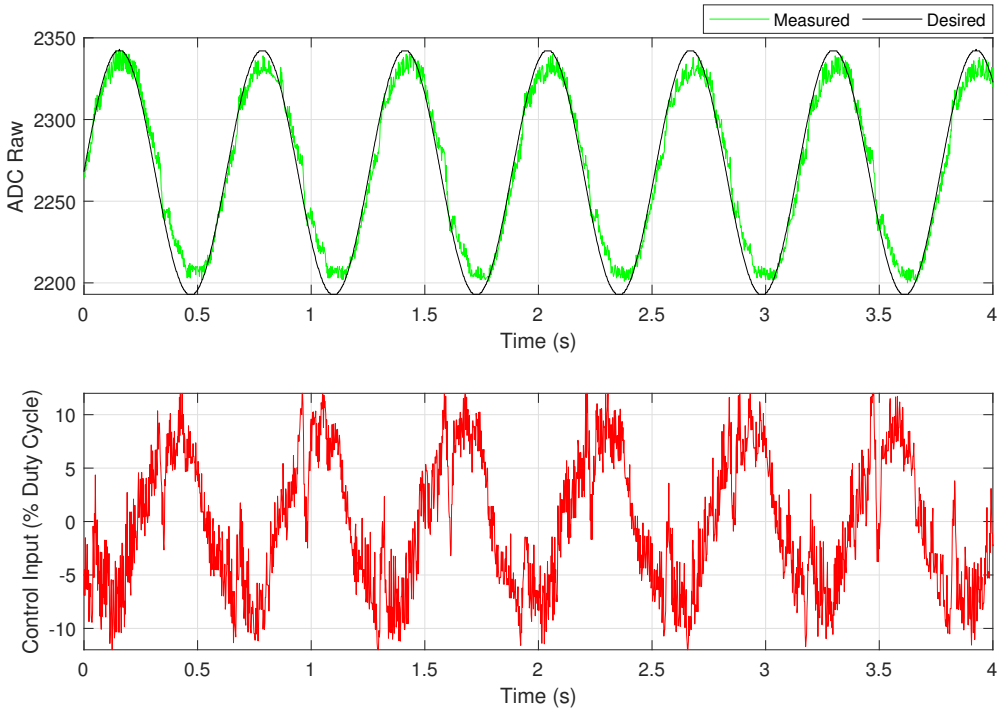

Figure 4.7: Tracking a 10 $\frac{\text{rad}}{\text{s}}$ sine wave with the PI controller. The peak-to-peak amplitude of the sine wave corresponds to approximately 212 N based on the loadcell calibration curve.

MRAC

For the MRAC experiments presented in this section, the following parameters were used: $\gamma_{\text{x}} = \gamma_{\text{r}} = 5 \times 10^{-7}$, $a_{\text{m}} = b_{\text{m}} = 5$. The initial conditions of the adaptive gains were $\theta_{\text{x}}(0) = \theta_{\text{r}}(0) = 0$. Figure 4.8 shows the tracking performance of a cosine trajectory with a frequency of 1 $\frac{\text{rad}}{\text{s}}$ and a peak-to-peak amplitude of approx 83 N. Figure 4.9 shows tracking of a much faster and larger cosine wave with 3 $\frac{\text{rad}}{\text{s}}$ frequency and 157 N peak-to-peak amplitude.

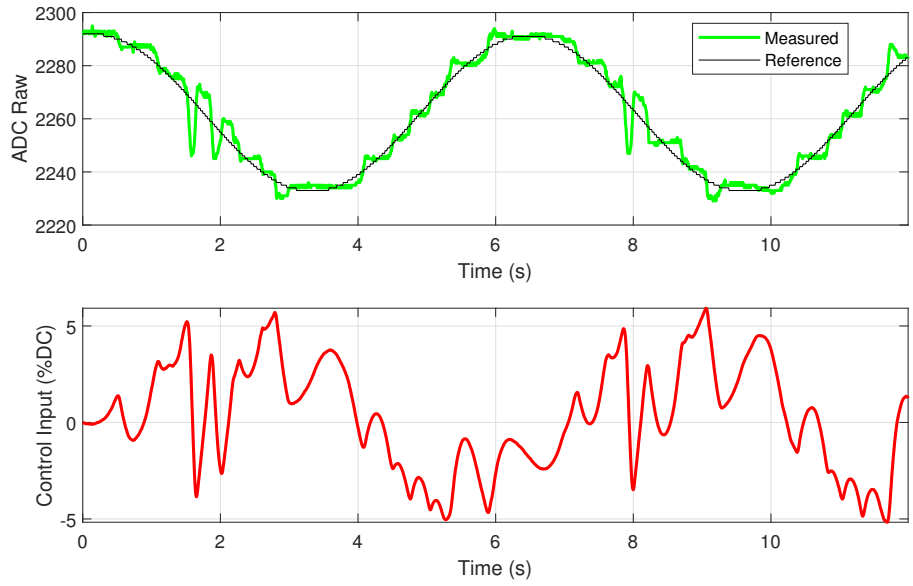

Figure 4.8: Tracking of a 1 $\frac{\text{rad}}{\text{s}}$ cosine wave with peak-to-peak amplitude of approximately 83 N using the MRAC algorithm. This controller is shown to have undesirable oscillations.

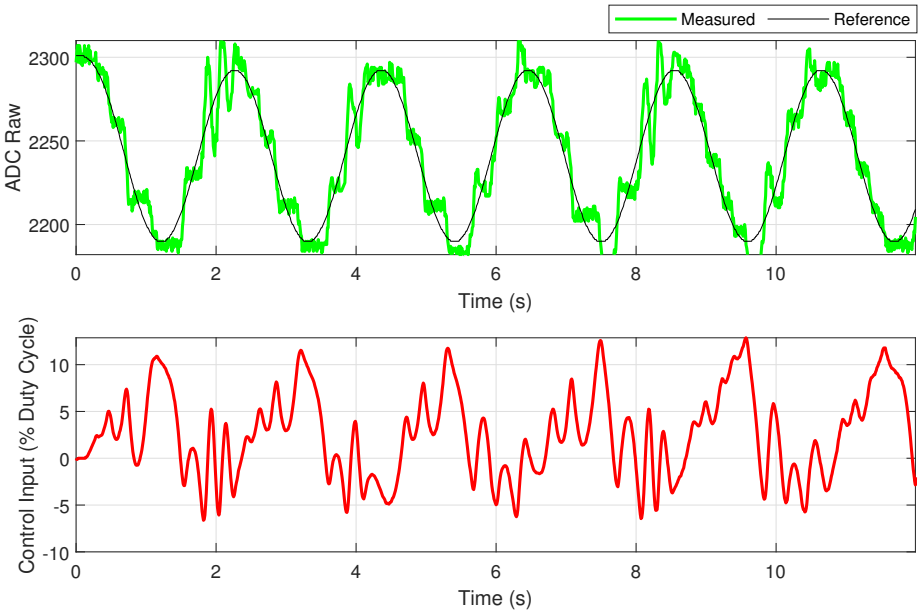

Figure 4.9: Tracking a 3 rad/s cosine wave with a peak-to-peak amplitude of approximately 157 N with the MRAC algorithm.

ADP-PI Control

In the ADP-PI experiments, sinusoidal trajectories were used in a similar fashion to the MRAC and PI experiments. Several different nonlinearities were used in the $\phi(x(t))$ vector including but not limited to polynomial functions, exponentials, and sine and cosine functions. Ultimately the nonlinear functions tested were found to have minimal impact on the control input, and therefore the following vector was used to collect the data presented in this section: $\phi(x(t)) = 0.012x(t)$. The following gains were used in the ADP-PI experiments: $\Gamma_K = \left[\begin{smallmatrix} .001 & 0 \\ 0 & .01 \end{smallmatrix}\right]$, $\Gamma_w = 1 \times 10^{-6}$, $k_\mathrm{p} = 0.6$, $k_\mathrm{i} = 0.5$, $a = -0.01$, and $b = 100$. All initial conditions were set to zero. Results for a cosine trajectory with approximately 127 N peak-to-peak amplitude and 1 $\frac{\mathrm{rad}}{\mathrm{s}}$ frequency are presented in Figures 4.10-4.11, and results for tracking a cosine wave with 170 N peak-to-peak amplitude and 13 $\frac{\mathrm{rad}}{\mathrm{s}}$ frequency are given in Figures 4.12-4.13.

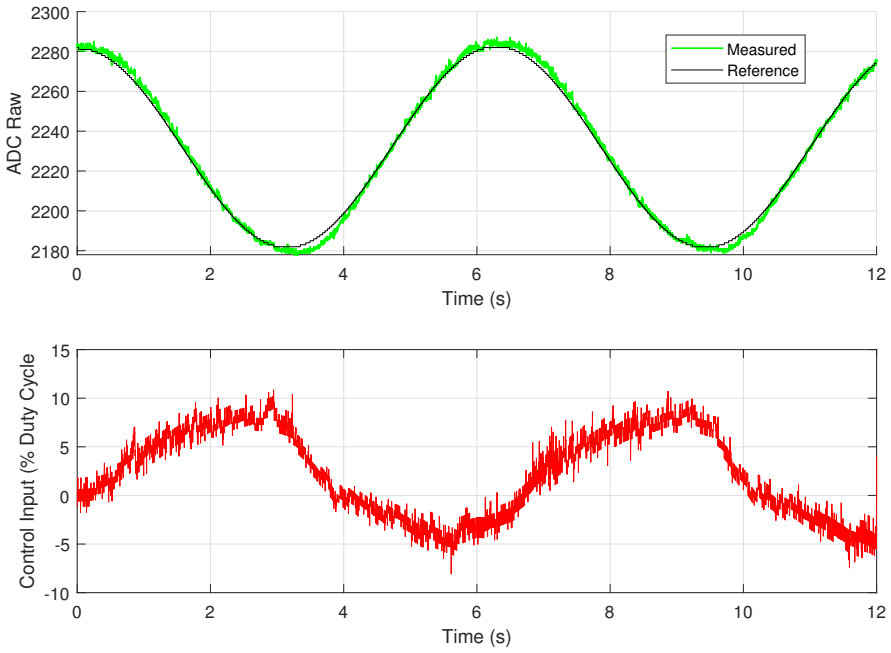

Figure 4.10: Force tracking of a 1 rad/s, 127 N peak-to-peak cosine wave using the ADP-PI algorithm.

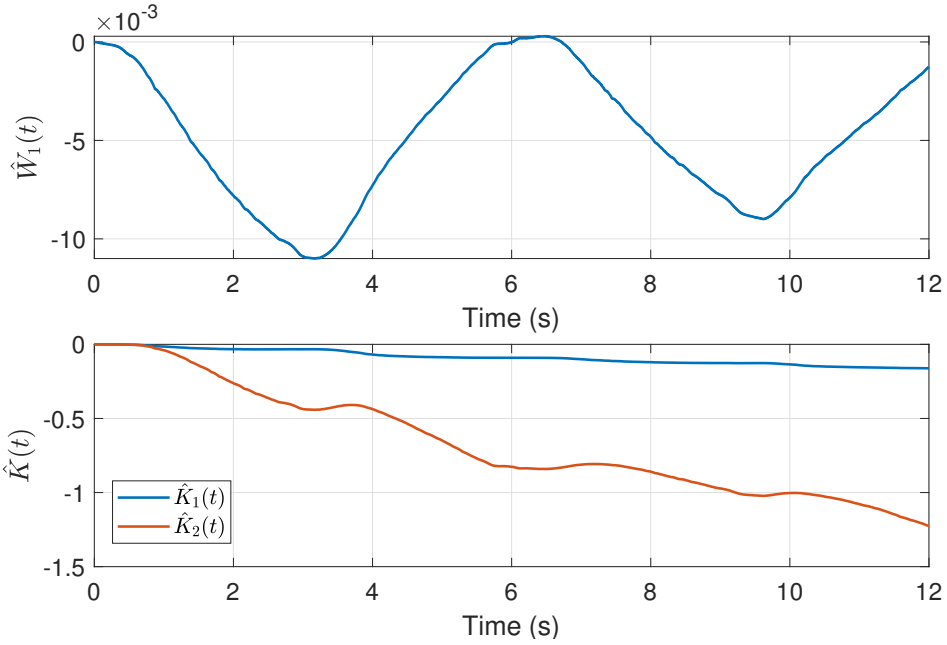

Figure 4.11: Evolution of the adaptive gains with the ADP-PI algorithm.

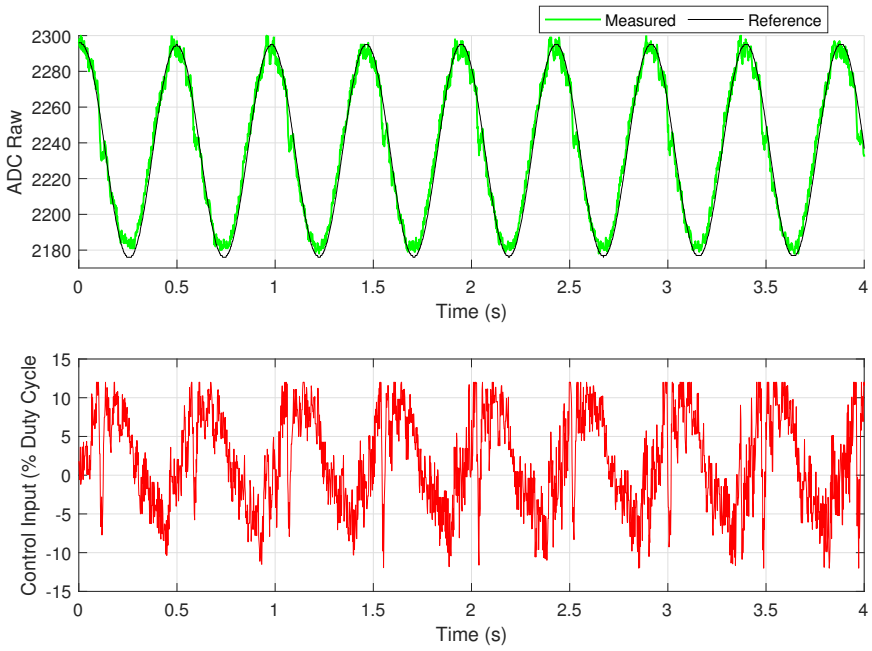

Figure 4.12: ADP-PI tracking control performance with a much faster and larger cosine trajectory.

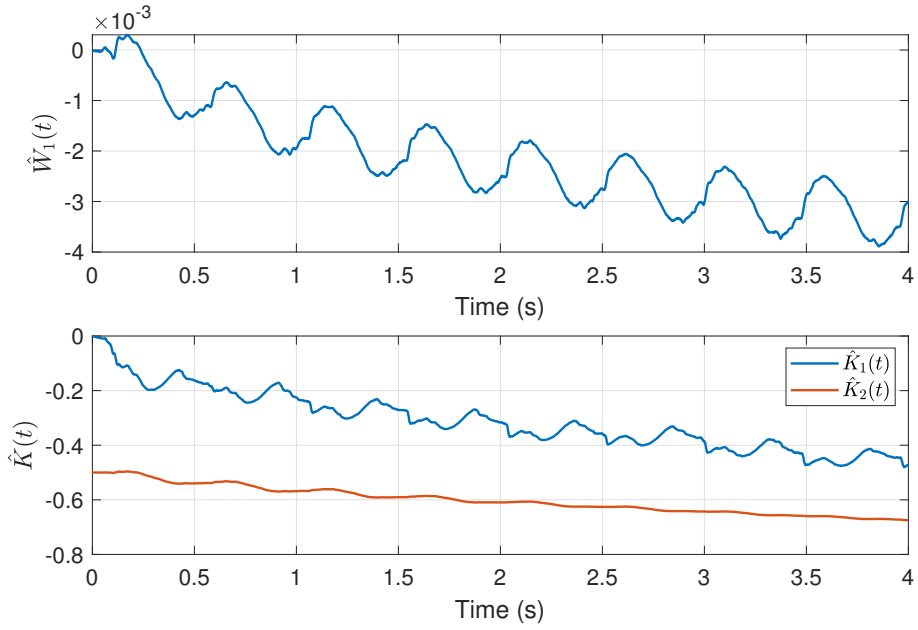

Figure 4.13: Adaptive gains over time with the ADP-PI controller.

4.4.2 Inverted Leg Results

The following section presents the results of the experiments with the inverted leg setup. For these experiments the ankle actuators were powered off while the knee actuator was tested. Due to the undesirable oscillations observed in the MRAC experiments with the actuator test stand, MRAC experiments were not conducted with the inverted leg setup. In this presentation of the results, the measured knee pitch angle is reported along with the force tracking data in order to provide additional context.

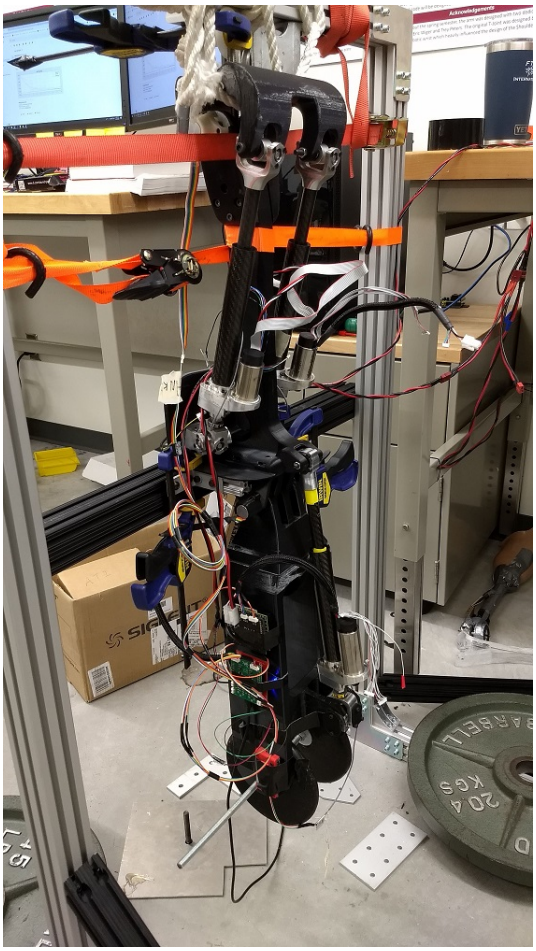

Figure 4.14: Inverted leg experimental setup. The actuator in this case is operating under the moving output condition, which is what would be encountered during the swing phase of walking.

PI Control

For PI control on the inverted leg, the following gains were used: $k_{\mathrm{p}} = 0.6$, $k_{\mathrm{i}} = 0.5$. Sinusoidal trajectories as well as a piecewise function consisting of a ramp-down and a ramp-up phase were used in the experiments. Figures 4.15 and 4.16 show results from a ramp and a cosine trajectory respectively.

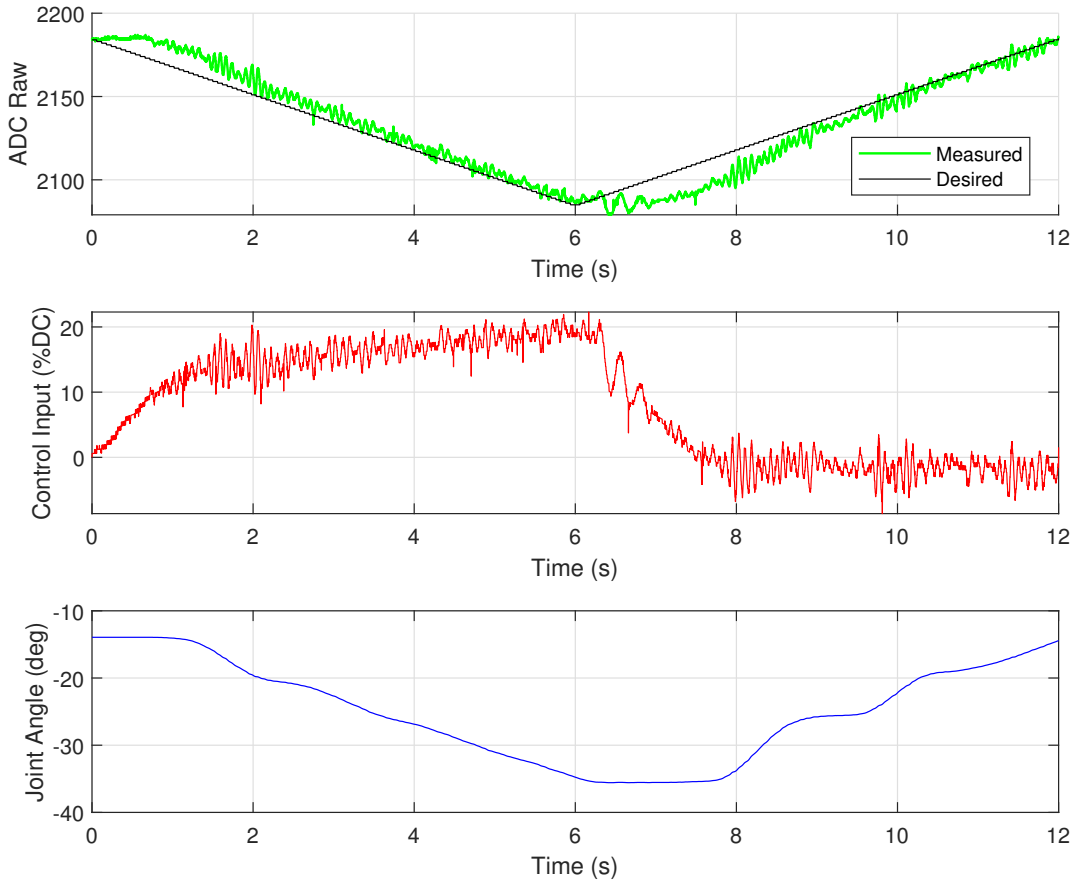

Figure 4.15: Tracking of a ramp down/ramp up trajectory with PI Control on the inverted leg.

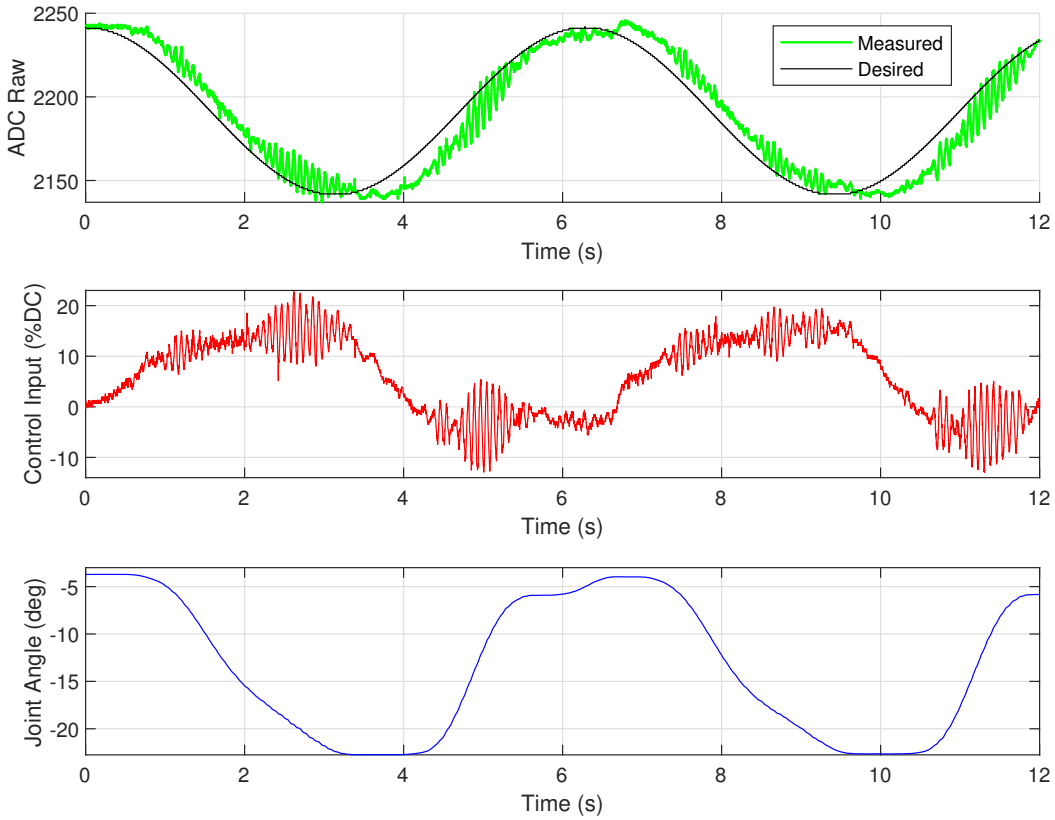

Figure 4.16: Tracking of 140 N peak-to-peak, 1 rad/s cosine trajectory with PI Control on the inverted leg. Chattering/oscillations can be observed when the direction of movement of the leg is changed, which are the points of maximum acceleration of the leg.

ADP-PI Control

The results from two experiments with the ADP-PI controller are presented in this section. Both experiments follow sinusoidal trajectories with 1 rad/s frequencies. In the first experiment the peak-to-peak amplitude of the desired trajectory was approximately 113 N, and the gains used were $\Gamma_K = \left[\begin{smallmatrix} .01 & 0 \\ 0 & .01 \end{smallmatrix}\right]$, $\Gamma_W = 1 \times 10^{-6}$, $k_\mathrm{p} = 0.6$, $k_\mathrm{i} = 0.5$, $a = -0.01$, and $b = 100$, and $\phi(x(t)) = 1.2(0.01x(t))$ was used. The adaptive gains $\hat{K}_1$ and $\hat{K}_2$ were bounded by ± 0.8 and ± 1.6 respectively.

In the second experiment, the peak-to-peak amplitude of the desired trajectory was 85 N, and the bounds on $\hat{K}_1$ were expanded to ± 1.2. An additional term was added to $\phi(x(t))$, resulting in $\phi(x(t)) = [1.2(0.01x(t))\sin(\omega t)]^{\mathrm{T}}$ where $\omega = 13.74 \cdot 2\pi$ rad/s. The gains were the same as the previous experiment with the exception of $\Gamma_W = \left[\begin{smallmatrix} 1e{-}6 & 0 \\ 0 & 1 \end{smallmatrix}\right]$.

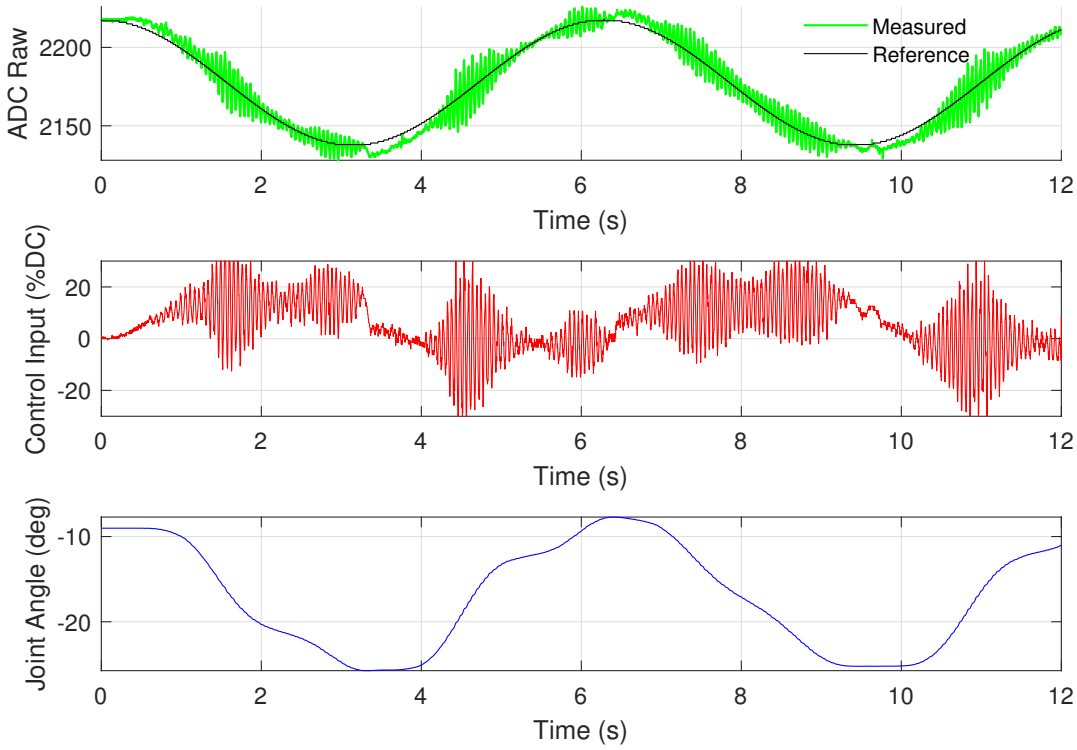

Figure 4.17: Tracking of 113 N peak-to-peak cosine trajectory with ADP-PI control on the inverted leg. High frequency oscillations are seen in the force controller and the resulting oscillations of the joint angle are very small.

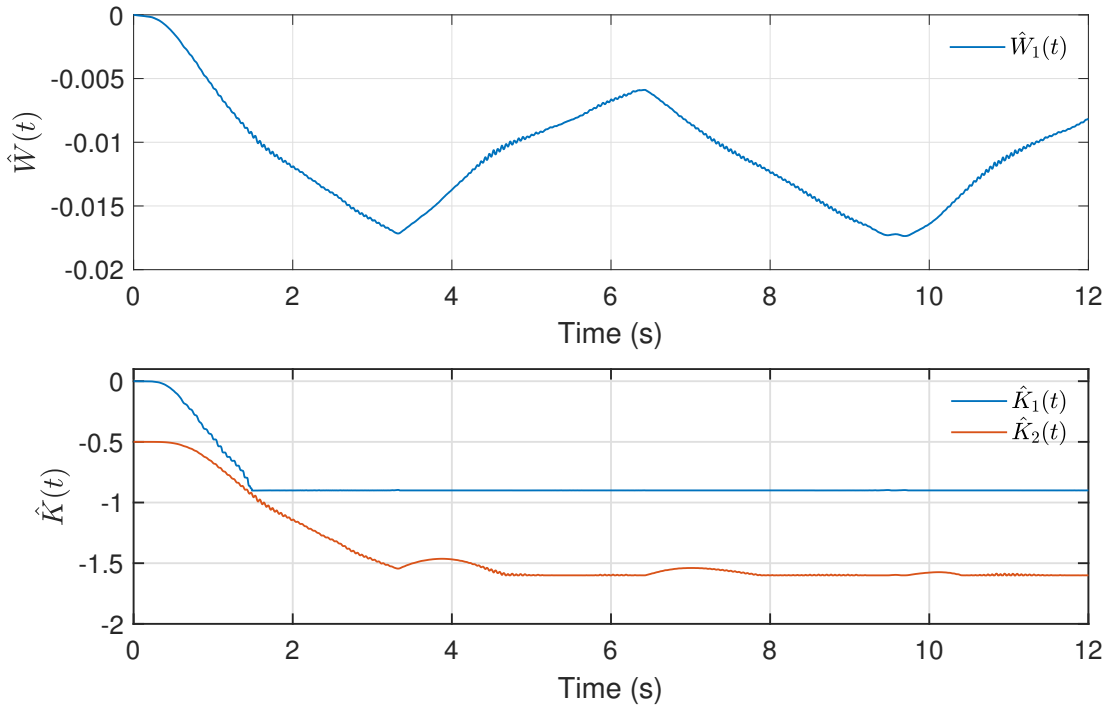

Figure 4.18: Adaptive gains during the ADP-PI experiment on the inverted leg. The adaptive proportional gain $\hat{K}_1(t)$ saturates at approximately 1.5 s and the adaptive integral gain $\hat{K}_2(t)$ saturates at 4.5 s.

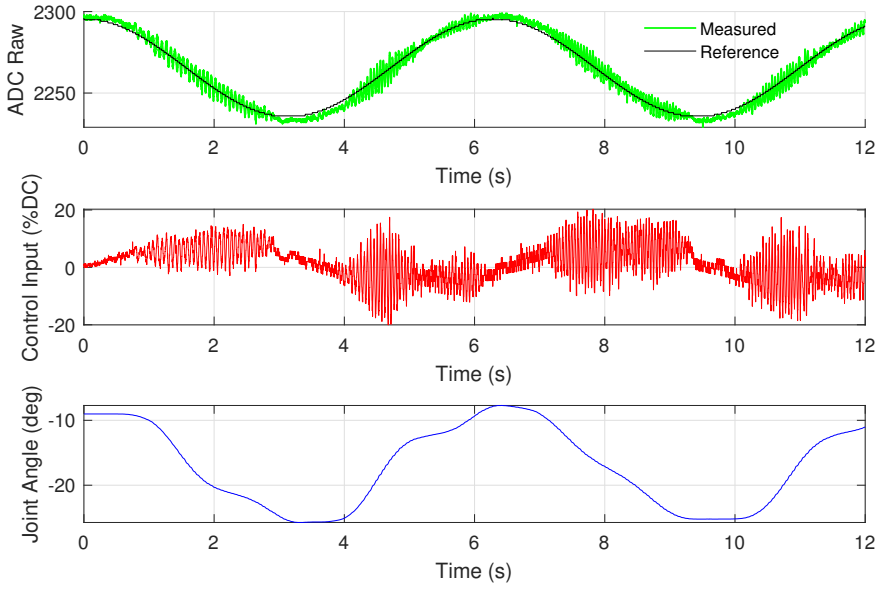

Figure 4.19: Tracking of an 85 N peak-to-peak cosine trajectory with ADP-PI control on the inverted leg, with an additional term added in $\phi(x(t))$. Note the continued presence of high frequency oscillations.

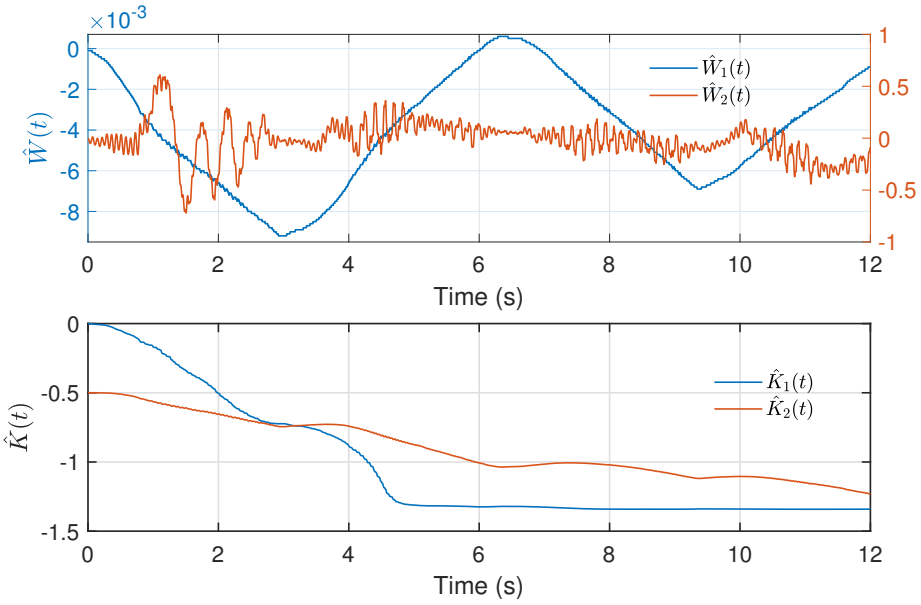

Figure 4.20: Adaptive gains from the ADP-PI experiment. Saturation of $\hat{K}_1$ occurs at around 4.5 s. Note that the saturation limits for $\hat{K}_1$ were expanded in this experiment.

Discussion of Results

Tracking performance with the actuator test stand was generally very good. The MRAC implementation suffered from some oscillations, which ultimately discouraged an MRAC implementation on the inverted leg. However minimal tracking errors were observed in the PI and ADP-PI controllers on the test stand. A noticeable decrease in force tracking performance was observed in the inverted leg experiments. This is likely due to the dynamics of the leg influencing the dynamics of the actuators, as well as issues with vibrations of the gantry, backlash in the knee joint, and poor clamping of the shin link. Oscillations in the inverted leg with the ADP-PI controller likely occurred because the adaptive gains in $\hat{K}(t)$ were converging towards higher values than the hardware could handle. In order to reduce oscillations, these gains were bounded. From the perspective of the adaptive controller, the ideal gains to drive the force measurement to the reference model may have been outside the bounds set on the parameters, resulting in poor tracking performance. Therefore it can be concluded that the first order dynamic model with matched nonlinearities used in the experiments was likely not sufficient to control the actuated leg, and in order to make use of the ADP-PI algorithm a more complicated model would have to be used. Due to the adequate performance of the PI controller and its simplicity, this controller was selected for use in the full prototype, and the ADP-PI algorithm was not pursued further.

Chapter 5

Modeling and Testing of the Knee Torque Controller

To validate the adaptive control framework developed in this thesis, a series of experiments were performed with an early prototype of THOR's leg. Due to hardware limitations at the time of testing, force feedback at the ankle joint was unavailable. Therefore, the experiments were conducted using a hybrid torque and position control scheme which enforces a desired torque at the knee joint while using high-gain position control at the ankle joint to regulate the ankle pitch and ankle roll joint angles. This chapter will cover the modifications made to the leg model, derivation of the modified adaptive controller, simulation results, and the experimental implementation and results. This hybrid controller was developed to serve as a proof of concept for the adaptive controller, and is not intended for use in locomotion in the finished robot.

5.1 Modified Knee Model

For the derivation of this control algorithm, we will make some modifications to the model introduced in Chapter 3. Rather than considering torque control of the knee joint as part of a multibody dynamics problem, we will assume that the dynamics of the shin link are negligible, and isolate the motion of the thigh link. Therefore the thigh will be modeled

as a single rigid body that is both rotating and translating through two-dimensional space, and the mass and acceleration of the shin link will be ignored. A schematic of the model is shown in Figure 5.1.

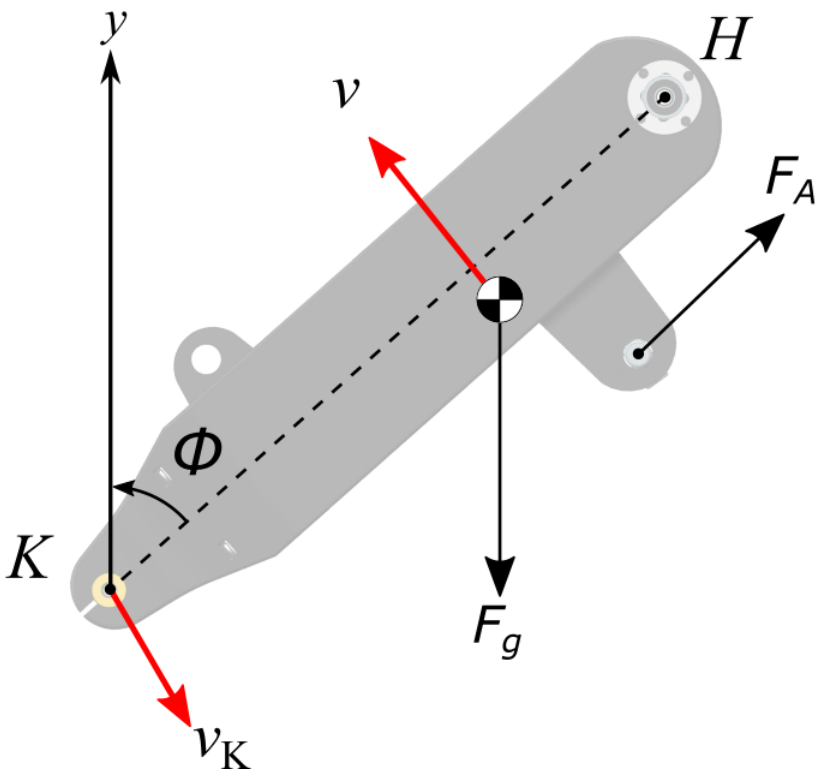

Figure 5.1: Free Body Diagram of the modified knee model.

Let $\phi(t)$ represent the orientation of the thigh link with respect to the y-axis of the world coordinate frame. This angle can be calculated in real time as $\phi(t) = q_1(t) + q_2(t)$. The angular acceleration of the thigh link is represented by $\alpha(t)$, while $v(t) \in \mathbb{R}^2$ is the linear velocity of the center of mass, and $v_K(t) \in \mathbb{R}^2$ is the linear velocity of point K, which is the location of knee joint, and also the point about which the thigh link rotates. The force due to gravity acts upon the center of mass in the negative y-direction with magnitude F_g. The actuator force passes through point A with magnitude $F_A(t)$. The direction of the actuator force varies with the knee angle, $q_2(t)$ as discussed in Chapter 3. The mapping from knee

actuator force $F_A(t)$ to its resultant torque about the knee joint also remains consistent with its definition in Chapter 3. The equations of motion of the thigh link are obtained through the angular form of Newton's Second Law as well as the principles of angular momentum [20], [21]. A diagram of the different vectors used to compute the cross products in the angular momentum analysis is given in Figure 5.2. Let $r_{\text{com}}(t) \in \mathbb{R}^2$ be the position vector from point K to the thigh center of mass, $r_x(t) \in \mathbb{R}^2$ be the position vector from the origin of the world frame, O, to the thigh center of mass, and $r_K(t) \in \mathbb{R}^2$ be the position vector from point O to point K. All of the aforementioned position vectors are expressed in the world coordinate frame.

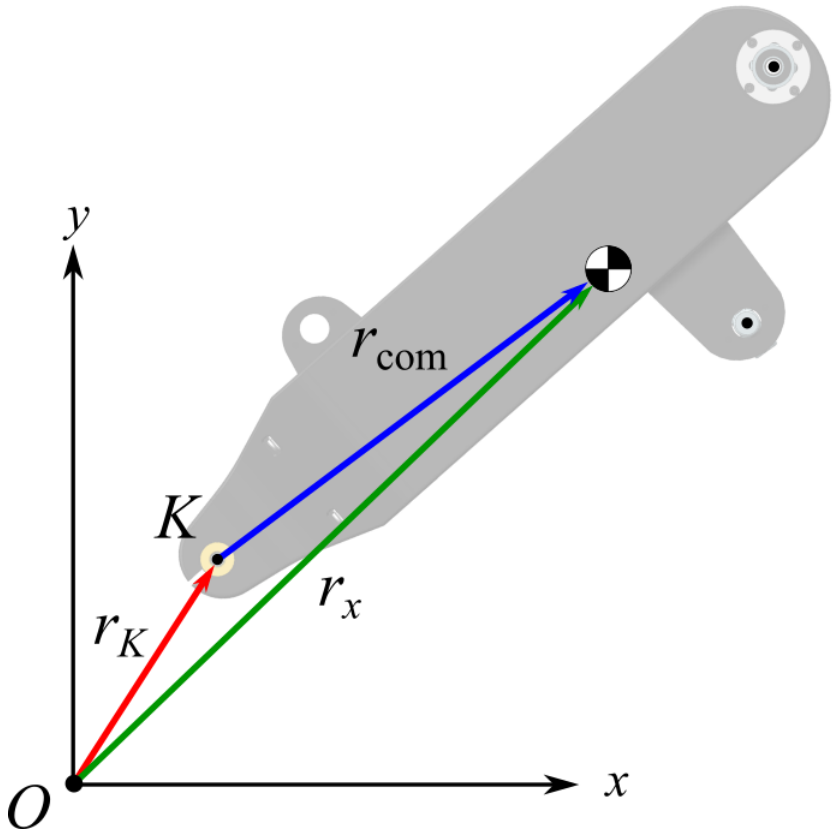

Figure 5.2: Diagram of the position vectors used in the angular momentum analysis of the modified knee model.

The angular momentum about point K is denoted by $L_K(t)$ and can be written as

$$L_K(t) = r_{\text{com}}(t) \times P(t), \tag{5.1}$$

where $P(t) = mv(t)$ is the linear momentum of the center of mass, and m is the mass of the thigh link. Differentiating (5.1) with respect to time yields

$$\dot{L}_K(t) = \dot{r}_{\text{com}}(t) \times P(t) + r_{\text{com}}(t) \times \dot{P}(t). \tag{5.2}$$

From Figure 5.2 it is clear that $r_{\text{com}}(t) = r_x(t) - r_K(t)$. Differentiating with respect to time we obtain $\dot{r}_{\text{com}}(t) = \dot{r}_x(t) - \dot{r}_K(t)$, which can equivalently be written as

$$\dot{r}_{\text{com}}(t) = v(t) - v_K(t). \tag{5.3}$$

Recognizing that the time derivative of the linear momentum of the center of mass is equal to the net external force acting on the center of mass, we observe that the quantity $r_{\text{com}}(t) \times \dot{P}(t)$ is equivalent to $r_{\text{com}}(t) \times F(t)$ where $F(t)$ is the net force resolved to the thigh link center of mass. This cross product is equivalent to the net external torque acting about point K, which we will label $\tau_K(t)$. Therefore we have

$$\tau_K(t) = r_{\text{com}}(t) \times \dot{P}(t). \tag{5.4}$$

By taking the sum of the torques about point K in the free body diagram given in Figure 5.1, we obtain the following expression for $\tau_K(t)$ in terms of the individual forces acting on the thigh link

$$\tau_K(t) = f_{\text{m}}(q_2(t))F_A(t) - F_{\text{g}} l_{\text{cm}_2 x} \cos(\phi(t)) + F_{\text{g}} l_{\text{cm}_2 y} \sin(\phi(t)), \tag{5.5}$$

where $q_2(t)$ is the knee joint angle, $f_{\mathrm{m}}(q_2(t))$ is the nonlinear function mapping actuator force to actuator torque for a given joint angle, $l_{\mathrm{cm}_2 x}$ and $l_{\mathrm{cm}_2 y}$ are the x and y locations of the thigh center of mass in the link coordinate frame. The system parameters $q_2(t)$, $f_{\mathrm{m}}(q_2(t))$, $l_{\mathrm{cm}_2 x}$ and $l_{\mathrm{cm}_2 y}$ are all consistent with their original definitions in Chapter 3. Substituting (5.3) and (5.4) into (5.2), we obtain

$$\begin{aligned} \dot{L}_K(t) &= v(t) \times P(t) - \big(v_K(t) \times P(t)\big) + \tau_K(t) \\ &= -\big(v_K(t) \times P(t)\big) + \tau_K(t). \end{aligned} \tag{5.6}$$

Note that the cross product $v(t) \times P(t)$ is equal to zero due to the two vectors being parallel. Substituting (5.6) in (5.5) and recognizing that $\dot{L}_K(t) = J\alpha(t)$ where J is the moment of inertia of the thigh link about the knee axis, we obtain the following equation of motion for the thigh link

$$J\alpha(t) = f_{\mathrm{m}}(q_2(t))F_A(t) - F_{\mathrm{g}} l_{\mathrm{cm}_2 x} \cos(\phi(t)) + F_{\mathrm{g}} l_{\mathrm{cm}_2 y} \sin(\phi(t)) - \big(v_K(t) \times P(t)\big). \tag{5.7}$$

Note that (5.7) accounts for both rotation and translation of the thigh link. This equation of motion will be used to derive the control law for the knee torque controller.

5.2 Controller Design

The derivation of the controller for the modified knee model is very similar to the controller in Section 3.3. Similar to the case of the 2DOF sagittal plane control problem, the knee actuator force is used as an input to regulate the net torque about the knee joint such that compliant trajectory tracking is achieved. Position and velocity reference models are designed with a certain ideal tracking performance in mind, an ideal controller is derived

under the assumption that the system parameters are known, and then the ideal controller is modified to be adaptive, following the convention of a Direct MRAC architecture.

5.2.1 Reference Model

For control of the modified knee model, we will use two reference systems. A standard second order system will be used for the position reference, and the velocity reference will be defined separately. The position reference system, $q_{\mathrm{r}}(t) = [q_{\mathrm{r}_1}(t) \;\; q_{\mathrm{r}_2}(t)]^{\mathrm{T}}$ has the following dynamics:

$$\dot{q}_{\mathrm{r}}(t) = \begin{bmatrix} \dot{q}_{\mathrm{r}_1}(t) \\ \dot{q}_{\mathrm{r}_2}(t) \end{bmatrix} = A_{\mathrm{r}} q_{\mathrm{r}}(t) + B_{\mathrm{r}} r(t), \tag{5.8}$$

where A_{r} and B_{r} are defined as

$$A_{\mathrm{r}} \triangleq \begin{bmatrix} 0 & 1 \\ -\omega_{\mathrm{n}}^2 & -2\zeta\omega_{\mathrm{n}} \end{bmatrix}, \quad B_{\mathrm{r}} \triangleq \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{5.9}$$

and $\omega_{\mathrm{n}} > 0$, $\zeta > 0$ are chosen based on a desired settling time and overshoot, and $r(t)$ is given by

$$r(t) = \omega_{\mathrm{n}}^2 q_{\mathrm{d}}(t) + 2\zeta\omega_{\mathrm{n}} \dot{q}_{\mathrm{d}}(t) + \ddot{q}_{\mathrm{d}}(t), \tag{5.10}$$

where the desired joint trajectory $q_{\mathrm{d}}(t)$ is a C^2 continuous function. Convergence of the reference model to the desired trajectory can be easily proven though analysis of the error dynamics. Let $e_{\mathrm{d}}(t)$ be

$$e_{\mathrm{d}}(t) = \begin{bmatrix} q_{\mathrm{r}_1}(t) \\ q_{\mathrm{r}_2}(t) \end{bmatrix} - \begin{bmatrix} q_{\mathrm{d}}(t) \\ \dot{q}_{\mathrm{d}}(t) \end{bmatrix}. \tag{5.11}$$

Differentiating (5.11) with respect to time results in

$$\begin{aligned}\dot{e}_{\mathrm{d}}(t) &= A_{\mathrm{r}} q_{\mathrm{r}}(t) + B_{\mathrm{r}} r(t) - \begin{bmatrix} \dot{q}_{\mathrm{d}}(t) \\ \ddot{q}_{\mathrm{d}}(t) \end{bmatrix} \\ &= A_{\mathrm{r}} e_{\mathrm{d}}(t). \end{aligned} \tag{5.12}$$

Since A_{r} is Hurwitz, the error dynamics in (5.12) are exponentially stable and therefore $q_{\mathrm{r}_1}(t) \to q_{\mathrm{d}}(t)$ and $q_{\mathrm{r}_2}(t) \to \dot{q}_{\mathrm{d}}(t)$ exponentially as $t \to \infty$. The state $q_{\mathrm{r}_1}(t)$ is used as the reference for position tracking. The position error is defined as

$$e(t) \triangleq q_2(t) - q_{\mathrm{r}_1}(t). \tag{5.13}$$

While we have shown that $q_{\mathrm{r}_2}(t)$ converges to the desired velocity $\dot{q}_{\mathrm{d}}(t)$, this state will not be used directly as the velocity reference model. Instead we will define the velocity reference $q_{\mathrm{rv}}(t)$ as

$$q_{\mathrm{rv}}(t) \triangleq q_{\mathrm{r}_2}(t) - \lambda_{\mathrm{p}} e(t), \tag{5.14}$$

where $\lambda_{\mathrm{p}} > 0$. The velocity error is defined as

$$e_{\mathrm{v}}(t) \triangleq \dot{q}_2(t) - q_{\mathrm{rv}}(t). \tag{5.15}$$

As with the reference model introduced in Section 3.3, (5.14) is defined in a manner similar to the velocity reference system used by Sadegh and Horowitz in [23]. By definition, (5.14) is guaranteed to converge to $q_{\mathrm{r}_2}(t)$ as $e(t) \to 0$. Since $q_{\mathrm{r}_2}(t) \to \dot{q}_{\mathrm{d}}(t)$ as $t \to \infty$, $q_{\mathrm{rv}}(t)$ will converge to $\dot{q}_{\mathrm{d}}(t)$ as long as $e(t)$ is guaranteed to converge to 0. The next step is to derive a feedback control law to drive position and velocity errors $e(t)$ and $e_{\mathrm{v}}(t)$ asymptotically to zero.

5.2.2 Ideal Controller

In order to drive the knee torque $J\alpha(t)$ to a desired joint torque $\tau_{\mathrm{d}}(t)$, we define the following ideal force input $F_A^*(t)$ as

$$F_A^*(t) = \frac{1}{f_{\mathrm{m}}(q_2(t))}\big(\tau_{\mathrm{d}}(t) + F_{\mathrm{g}} l_{\mathrm{cm}_2 x}\cos(\phi(t)) - F_{\mathrm{g}} l_{\mathrm{cm}_2 y}\sin(\phi(t)) + (v_K(t) \times P(t))\big), \quad (5.16)$$

where $f_{\mathrm{m}}(q_2(t)) > 0$ over the full range of motion of the knee joint. Substituting (5.16) into (5.7), the closed-loop dynamics become

$$J\alpha(t) = \tau_{\mathrm{d}}(t). \quad (5.17)$$

In order to achieve compliant trajectory tracking, the desired joint torque $\tau_{\mathrm{d}}(t)$ is chosen such that the joint behaves like a torsional spring and damper as it is displaced from the desired position. Define τ_{d}^* as the ideal desired torque,

$$\tau_{\mathrm{d}}^*(t) = J\dot{q}_{\mathrm{rv}}(t) - K_1 e(t) - K_2 e_{\mathrm{v}}(t), \quad (5.18)$$

where $\dot{q}_{\mathrm{rv}}(t) = \dot{q}_{\mathrm{r}_2} - \lambda_{\mathrm{p}}\dot{e}(t)$, and $K_1 > 0$ and $K_2 > 0$ are the desired stiffness and damping of the joint. The force input (5.16) and the desired torque (5.18) are functions of unknown parameters F_{g}, $l_{\mathrm{cm}_2 x}$, $l_{\mathrm{cm}_2 y}$, and J. Therefore, this control algorithm is not implementable in practice and the controller must be made adaptive.

5.2.3 Adaptive Controller

To design the adaptive controller, the unknown parameters of the system are lumped into ideal gains as follows,

$$\theta_1^* = F_{\mathrm{g}} l_{\mathrm{cm}_2 x}, \tag{5.19}$$

$$\theta_2^* = -F_{\mathrm{g}} l_{\mathrm{cm}_2 y}, \tag{5.20}$$

$$\theta_3^* = J. \tag{5.21}$$

The force control law and desired torque are redefined as functions of the adaptive parameters $\theta_1(t)$, $\theta_2(t)$, and $\theta_3(t)$, which are estimates of (5.19), (5.20), and (5.21) respectively. This results in the following

$$F_A(t) = \frac{1}{f_{\mathrm{m}}(q_2(t))}\big(\tau_{\mathrm{d}}(t) + \theta_1(t)\cos(\phi(t)) + \theta_2(t)\sin(\phi(t)) + (v_K(t) \times P(t))\big), \tag{5.22}$$

$$\tau_{\mathrm{d}} = \theta_3(t)\dot{q}_{\mathrm{rv}} - K_1 e(t) - K_2 e_{\mathrm{v}}(t). \tag{5.23}$$

The estimation errors are defined as

$$\tilde{\theta}_i(t) = \theta_i(t) - \theta_i^*, \qquad i = 1, 2, 3. \tag{5.24}$$

Substituting (5.22), (5.23), (5.19)-(5.21) and (5.24) into (5.7) yields the closed-loop dynamics

$$J\alpha(t) = \tau_{\mathrm{d}}(t) + \tilde{\theta}_1(t)\cos(\phi(t)) + \tilde{\theta}_2(t)\sin(\phi(t)). \tag{5.25}$$

The final step is to specify the update laws for the adaptive parameters in order to guarantee stability of the closed-loop system.

Theorem. Define the following adaptation laws,

$$\dot{\theta}_1(t) = -\gamma_1 e_{\mathrm{v}}(t)\cos(\phi(t)), \tag{5.26}$$

$$\dot{\theta}_2(t) = -\gamma_2 e_{\mathrm{v}}(t)\sin(\phi(t)), \tag{5.27}$$

$$\dot{\theta}_3(t) = -\gamma_3 e_{\mathrm{v}}(t)\dot{q}_{\mathrm{rv}}(t), \tag{5.28}$$

where γ_1, γ_2, $\gamma_3 > 0$ are the adaptation gains. Then, the closed-loop system consisting of (5.7), (5.8), (5.14), (5.22), (5.23), and (5.26)-(5.28) is Lyapunov Stable and $e(t)$ and $e_{\mathrm{v}}(t)$ are guaranteed to converge to zero as $t \to \infty$.

Proof. Using the definition of the velocity reference model (5.14), the error dynamics for position tracking can be rewritten as follows,

$$\begin{aligned}\dot{e}(t) &= \dot{q}_2(t) - q_{\mathrm{r}_2}(t) \\ &= \dot{q}_2(t) - \big(q_{\mathrm{rv}}(t) + \lambda_{\mathrm{p}} e(t)\big) \\ &= e_{\mathrm{v}}(t) - \lambda_{\mathrm{p}} e(t).\end{aligned} \tag{5.29}$$

Since we are neglecting the dynamics of the shin link, we will assume that the angular acceleration of the thigh link $\alpha(t)$ is approximately equal to the angular acceleration of the knee joint, i.e. $\alpha(t) \approx \ddot{q}_2(t)$. Leveraging this assumption, the definitions of the parameter errors (5.24), and substituting (5.25), the error dynamics for velocity tracking can be rewritten as

$$\begin{aligned}\dot{e}_{\mathrm{v}}(t) &= \ddot{q}_2(t) - \dot{q}_{\mathrm{rv}}(t) \\ &= \alpha(t) - \dot{q}_{\mathrm{rv}}(t) \\ &= \frac{1}{J}\big(\tau_{\mathrm{d}}(t) + \tilde{\theta}_1(t)\cos(\phi(t)) + \tilde{\theta}_2(t)\sin(\phi(t))\big) - \dot{q}_{\mathrm{rv}}(t).\end{aligned} \tag{5.30}$$

Next, consider the following Lyapunov function candidate [14],

$$V(t) = \frac{1}{2}Je_{\mathrm{v}}(t)^2 + \frac{1}{2}K_1e(t)^2 + \frac{1}{2\gamma_1}\tilde{\theta}_1(t)^2 + \frac{1}{2\gamma_2}\tilde{\theta}_2(t)^2 + \frac{1}{2\gamma_3}\tilde{\theta}_3(t)^2. \tag{5.31}$$

Taking the time derivative of (5.31) along the trajectories of (5.13), (5.15), and (5.26)-(5.28) results in

$$\dot{V}(t) = Je_{\mathrm{v}}(t)\dot{e}_{\mathrm{v}}(t) + K_1e(t)\dot{e}(t) + \frac{1}{\gamma_1}\tilde{\theta}_1(t)\dot{\tilde{\theta}}_1(t) + \frac{1}{\gamma_2}\tilde{\theta}_2(t)\dot{\tilde{\theta}}_2(t) + \frac{1}{\gamma_3}\tilde{\theta}_3(t)\dot{\tilde{\theta}}_3(t). \tag{5.32}$$

Substituting in the error dynamics (5.29) and (5.30) and rearranging terms we obtain

$$\begin{aligned}\dot{V}(t) &= Je_{\mathrm{v}}(t)\Big(\frac{1}{J}\big(\tau_{\mathrm{d}}(t) + \tilde{\theta}_1(t)\cos(\phi(t)) + \tilde{\theta}_2(t)\sin(\phi(t))\big) - \dot{q}_{\mathrm{rv}}(t)\Big) + K_1e(t)\big(e_{\mathrm{v}}(t) - \lambda_{\mathrm{p}}e(t)\big)\\ &\quad + \frac{1}{\gamma_1}\tilde{\theta}_1(t)\dot{\tilde{\theta}}_1(t) + \frac{1}{\gamma_2}\tilde{\theta}_2(t)\dot{\tilde{\theta}}_2(t) + \frac{1}{\gamma_3}\tilde{\theta}_3(t)\dot{\tilde{\theta}}_3(t)\\ &= e_{\mathrm{v}}(t)\big(\tau_{\mathrm{d}} - J\dot{q}_{\mathrm{rv}}(t) + \tilde{\theta}_1(t)\cos(\phi(t)) + \tilde{\theta}_2(t)\sin(\phi(t)) + K_1e(t)\big) - K_1\lambda_{\mathrm{p}}e(t)^2\\ &\quad + \frac{1}{\gamma_1}\tilde{\theta}_1(t)\dot{\tilde{\theta}}_1(t) + \frac{1}{\gamma_2}\tilde{\theta}_2(t)\dot{\tilde{\theta}}_2(t) + \frac{1}{\gamma_3}\tilde{\theta}_3(t)\dot{\tilde{\theta}}_3(t).\end{aligned} \tag{5.33}$$

Substituting in (5.23), (5.21) and (5.24) and simplifying results in

$$\begin{aligned}\dot{V}(t) &= e_{\mathrm{v}}(t)(\tilde{\theta}_1(t)\cos(\phi(t)) + \tilde{\theta}_2(t)\sin(\phi(t)) + \tilde{\theta}_3(t)\dot{q}_{\mathrm{rv}}(t) - K_2e_{\mathrm{v}}(t)) - K_1\lambda_{\mathrm{p}}e(t)^2\\ &\quad + \frac{1}{\gamma_1}\tilde{\theta}_1(t)\dot{\tilde{\theta}}_1(t) + \frac{1}{\gamma_2}\tilde{\theta}_2(t)\dot{\tilde{\theta}}_2(t) + \frac{1}{\gamma_3}\tilde{\theta}_3(t)\dot{\tilde{\theta}}_3(t)\\ &= -K_2e_{\mathrm{v}}(t)^2 - K_1\lambda_{\mathrm{p}}e(t)^2 + \tilde{\theta}_1(t)\big(\frac{1}{\gamma_1}\dot{\tilde{\theta}}_1(t) + e_{\mathrm{v}}(t)\cos(\phi(t))\big)\\ &\quad + \tilde{\theta}_2(t)\big(\frac{1}{\gamma_2}\dot{\tilde{\theta}}_2(t) + e_{\mathrm{v}}(t)\sin(\phi(t))\big) + \tilde{\theta}_3(t)\big(\frac{1}{\gamma_3}\dot{\tilde{\theta}}_3(t) + e_{\mathrm{v}}(t)\dot{q}_{\mathrm{rv}}(t)\big).\end{aligned} \tag{5.34}$$

Leveraging the assumption that the unknown parameters θ_1^*, θ_2^*, and θ_3^* do not change with time, we can substitute $\dot{\tilde{\theta}}_1(t) = \dot{\theta}_1(t)$, $\dot{\tilde{\theta}}_2(t) = \dot{\theta}_2(t)$ and $\dot{\tilde{\theta}}_3(t) = \dot{\theta}_3(t)$ into (5.34). Finally,

substitute in (5.26)-(5.28) to obtain

$$\dot{V}(t) = -K_2 e_{\mathrm{v}}(t)^2 - K_1 \lambda_{\mathrm{p}} e(t)^2 \leq 0. \tag{5.35}$$

Therefore, the closed-loop system is Lyapunov stable and (5.13) and (5.15) are guaranteed to converge to zero as $t \to \infty$ by the LaSalle-Yoshizawa Theorem [17].

5.3 Numerical Simulations

Initial testing of the adaptive controller was conducted through simulation. The closed-loop system was modeled in MATLAB/Simulink using geometric and inertial values obtained from the robot CAD model. A simple block diagram showing the general structure of the control algorithm is given in Figure 5.3. The simulation model treats the shin link as though it is rigidly fixed in the upright position ($q_1 = 0$), therefore isolating the dynamics of the thigh link. Consequently we have $v_{\mathrm{K}}(t) = 0, t \geq 0$.

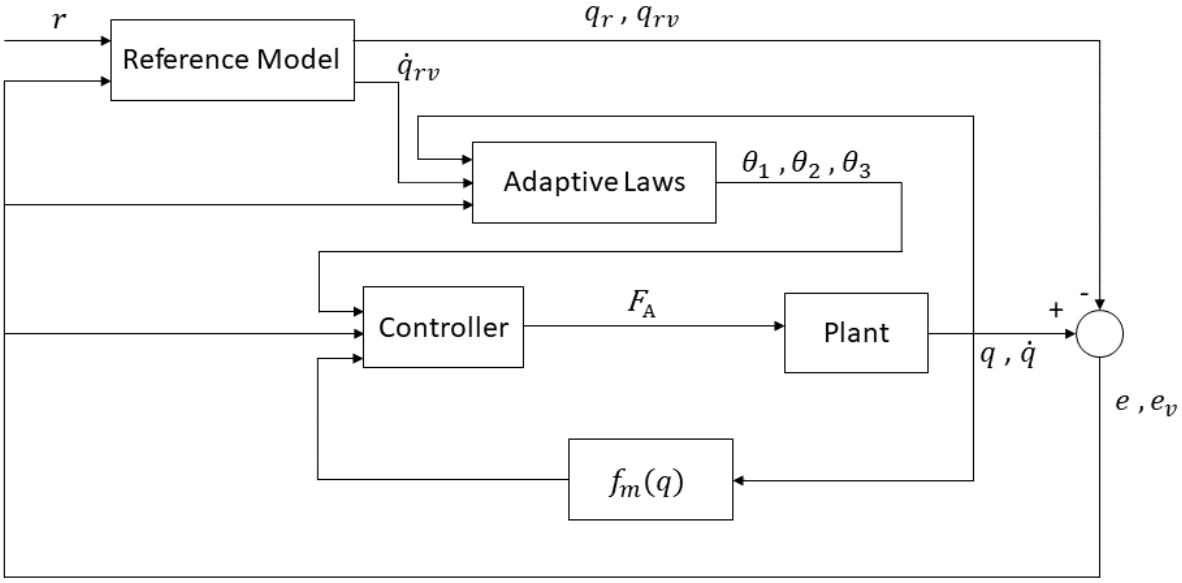

Figure 5.3: Block Diagram of the control system used in the simulations of the modified knee model.

5.3.1 Simulation Results

Simulation results for a sinusoidal trajectory with an amplitude of 20° and a frequency of 0.5 $\frac{\text{rad}}{\text{s}}$ are presented in Figures 5.4 - 5.8. The robot parameters were $J = 1.7346$ kgm^2, $l_{\text{cm}_2 x} = 0.0074$ m, $l_{\text{cm}_2 y} = 0.3838$ m, and $F_g = 9.81 \cdot 10.765$ N. The following gains were used in the simulation: $K_1 = 30$, $K_2 = 10$, $\lambda_p = 0.5$, $\gamma_1 = 300$, $\gamma_2 = 100$, $\gamma_3 = 0.1$, $\omega_{\text{n}} = 3$, $\zeta = 1$. The initial conditions were $\phi(0) = 0.05$ rad, $\dot{\phi}(0) = 0$ $\frac{\text{rad}}{\text{s}}$, $q_r(0) = \phi(0)$, $\theta_1(0) = 0.9$ Nm, $\theta_2(0) = -45$ Nm, and $\theta_3(0) = 1.9$ kgm^2.

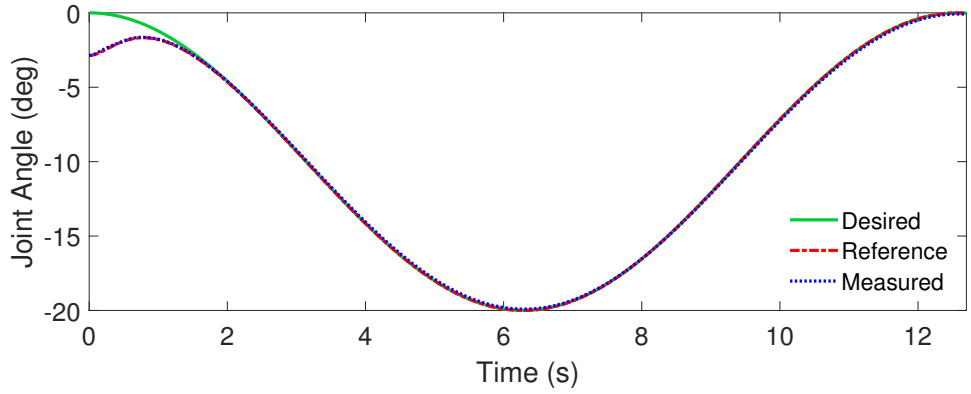

Figure 5.4: Simulation results showing position tracking performance of the knee joint for a 20 degrees amplitude, 0.5 $\frac{\text{rad}}{\text{s}}$ cosine trajectory.

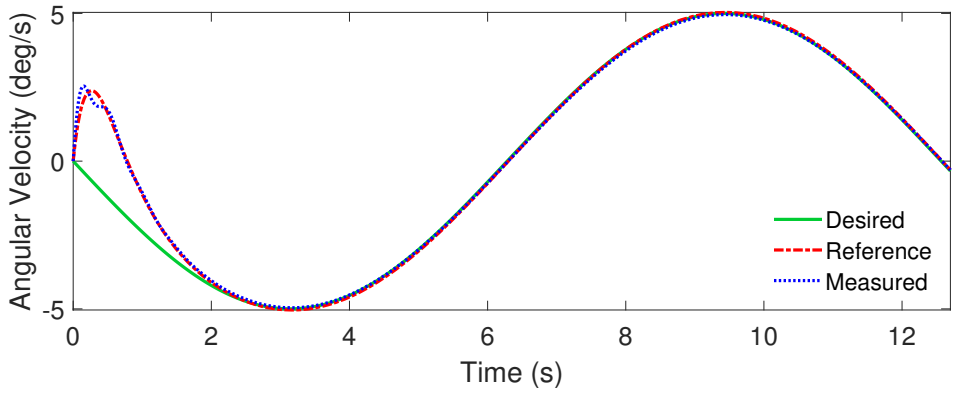

Figure 5.5: Velocity tracking performance of the knee joint.

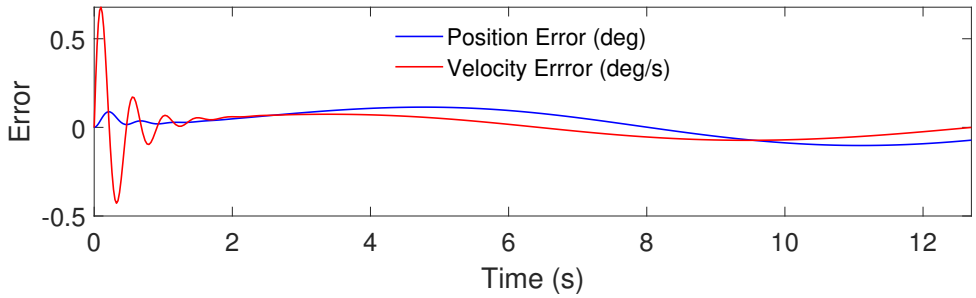

Figure 5.6: Position and velocity tracking error at the knee joint.

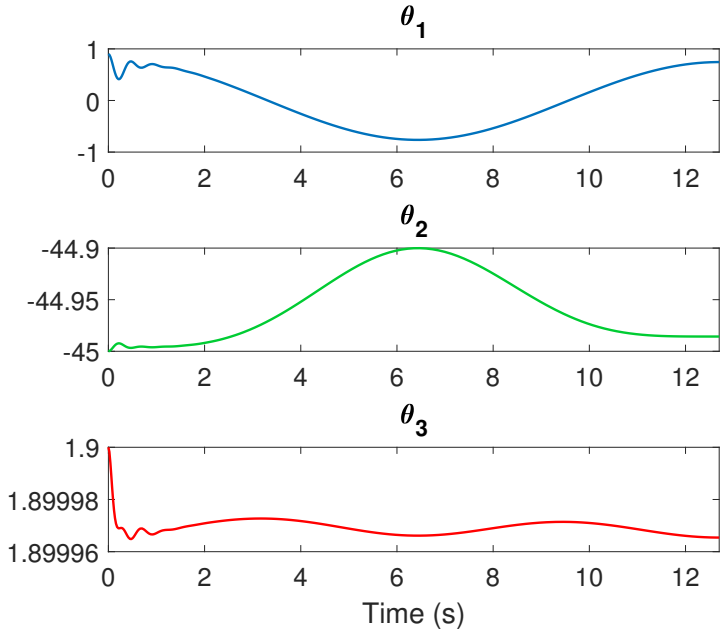

Figure 5.7: Evolution of the adaptive parameters during the simulation.

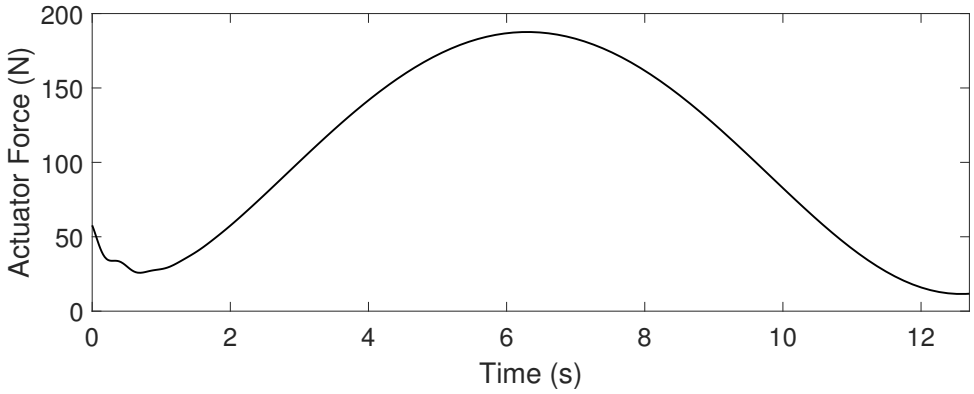

Figure 5.8: Actuator force input.

The simulation results show excellent position and velocity tracking performance, with tracking errors within $\pm 0.2^{\circ}$ and $\pm 0.7 \frac{\text{deg}}{\text{s}}$ of deviation from the reference models. The simulation results of this controller are comparable to the controller derived in Section 3.3 and therefore this controller was deemed a good candidate for an experimental solution.

5.4 Experimental Implementation

This section will provide a comprehensive discussion of the experiments conducted to validate the full control system consisting of an outer loop joint torque controller derived in Section 5.2 and the inner loop force controller introduced in Chapter 4, including both implementation and the results. As mentioned previously, a high-gain PID position controller was implemented at the 2DOF ankle mechanism in order to regulate ankle pitch and ankle roll during the experiments. A block diagram of the full control scheme used in the experiments is given in Figure 5.9.

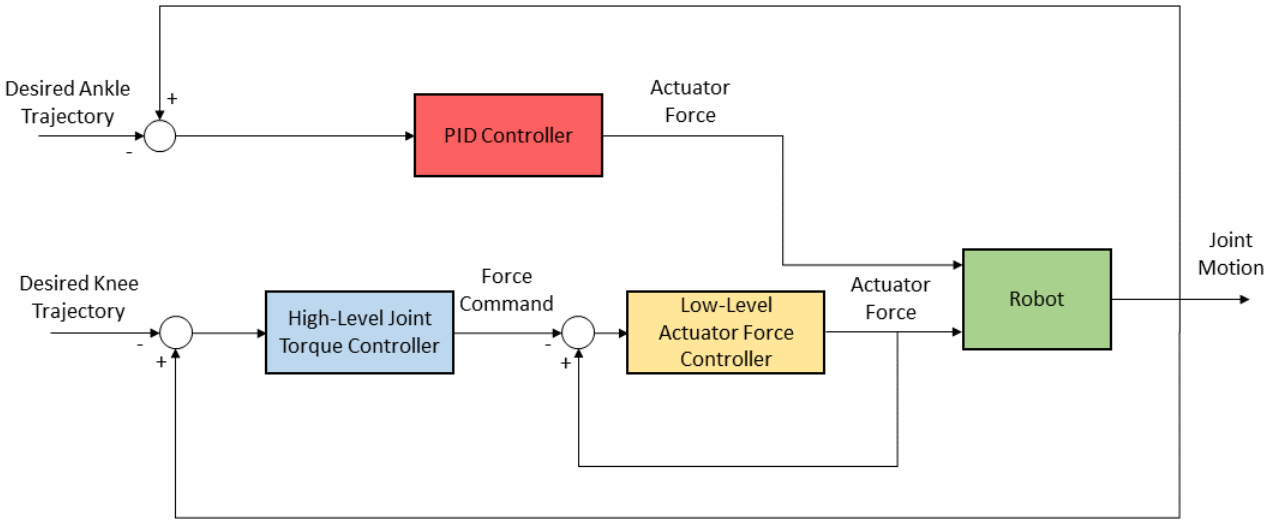

Figure 5.9: Block diagram showing the knee torque control and ankle position control systems used in the experiments.

The PI controller described in Section 4.1 was used as the inner loop low-level force controller to ensure fast and accurate tracking of the forces calculated in the high-level joint torque controller outer loop. The high-level control loop runs at 100 Hz while the low-level controller runs at 1000 Hz. This is done in order to give the force controller enough time to converge to the commanded forces in between time steps of the high-level controller.

5.4.1 2DOF Ankle Position Control

For all of the experiments with the knee torque controller, the ankle pitch and roll angles were regulated using a PID controller with position feedback. The ankle pitch motion is affected by the sum of the two actuator forces while the ankle roll motion is affected by the difference between the two forces. The controller works by evaluating the ankle pitch and ankle roll position errors, then determining control inputs for each joint. A system of two equations and two unknowns is then solved to find the commanded current in each actuator to achieve the desired control input.

Let $e_{\mathrm{A}}(t) = [e_{\mathrm{Ap}}(t)\ \ e_{\mathrm{Ar}}(t)]^{\mathrm{T}} \in \mathbb{R}^2$ be the vector of position errors, where $e_{\mathrm{Ap}}(t)$ and $e_{\mathrm{Ar}}(t)$ are the ankle pitch and ankle roll errors respectively. The position errors are defined with the following convention: $e = q - q_{\mathrm{d}}$, where q is the measured joint angle and q_{d} is the desired joint angle. Let $u(t) = [u_{\mathrm{p}}(t)\ \ u_{\mathrm{r}}(t)]^{\mathrm{T}} \in \mathbb{R}^2$ be the vector of control inputs, where $u_{\mathrm{p}}(t)$ and $u_{\mathrm{r}}(t)$ are the pitch and roll inputs respectively. The control law is then given by

$$u(t) = k_{\mathrm{P}} e_{\mathrm{A}}(t) + k_{\mathrm{I}} \int_0^t e_{\mathrm{A}}(t)\mathrm{d}t + k_{\mathrm{D}} \frac{\mathrm{d}e_{\mathrm{A}}(t)}{\mathrm{d}t}, \tag{5.36}$$

where $k_{\mathrm{P}} \in \mathbb{R}^{2\times 2}$, $k_{\mathrm{I}} \in \mathbb{R}^{2\times 2}$, $k_{\mathrm{D}} \in \mathbb{R}^{2\times 2}$, are diagonal matrices of the proportional, integral, and derivative gains used for each joint. The commanded current in each motor is calculated

as

$$i(t) = \begin{bmatrix} i_1(t) \\ i_2(t) \end{bmatrix} = \begin{bmatrix} -(u_{\mathrm{p}}(t) + u_{\mathrm{r}}(t)) \\ -(u_{\mathrm{p}}(t) - u_{\mathrm{r}}(t)) \end{bmatrix}. \tag{5.37}$$

The gains for the 2DOF ankle position controller were chosen to be relatively high in order to ensure that the ankle joint experienced minimal movement due to the dynamics of the knee joint. The ankle roll angle was enforced to be zero at all times during all experiments.

5.4.2 Velocity Estimation with the AIVSDE

The control laws defined in this chapter require velocity feedback from the joints. Since only the joint angle is provided by the optical encoders, direct velocity measurements are not available. One way of obtaining the velocity would be to implement a numerical derivative of the position measurements, such as using a backwards finite difference approximation. However this method is highly vulnerable to noise and the resulting velocity approximations will be delayed. Therefore in this implementation the velocity of each joint is approximated using the Adaptive Integral Variable Structure Derivative Estimator (AIVSDE) presented in [3]. Given an input signal $r(t)$, a signal $x_0(t)$ whose derivative is $x_1(t)$ is designed. The dynamics of $x_1(t)$ are chosen such that $x_0(t)$ converges to $r(t)$. Therefore $x_1(t)$ converges to $\dot{r}(t)$, and $x_1(t)$ is used in place of the time derivative of $r(t)$ in the control system. The $x_1(t)$ dynamics are given by

$$x_1(t) = k_{\mathrm{r}}(t)\mathrm{sign}(\sigma(t)) + k_{\mathrm{b}}\sigma(t) - \frac{k_1}{k_2}x_0(t), \tag{5.38}$$

$$\sigma(t) = k_2(r(t) - x_0(t)) + k_1 \int_0^t (r(\tau) - x_0(\tau))d\tau, \tag{5.39}$$

where $k_{\mathrm{b}}, k_1, k_2 > 0$ are constant tuning parameters, and $k_{\mathrm{r}}(t)$ is defined as follows,

$$\dot{k}_{\mathrm{r}}(t) = \begin{cases} \alpha_{\mathrm{r}} & \text{if} \quad |\sigma(t)| \geq \mu \\ 0 & \text{if} \quad |\sigma(t)| < \mu \end{cases}, \qquad k_{\mathrm{r}}(0) = k_{\mathrm{r}_0}, \tag{5.40}$$

where $\alpha_{\mathrm{r}}, \mu \geq 0$. The AIVSDE as defined by (5.38)-(5.40) has $\sigma(t)$ guaranteed to converge to zero in finite time, and the estimation error $e_{\mathrm{r}}(t) = r(t) - x_0(t)$ converges to zero asymptotically [3]. For the experiments discussed in this chapter, the following AIVSDE parameters were used: $k_1 = 10$, $k_2 = 20$, $k_{\mathrm{b}} = 5$, $\alpha_{\mathrm{r}} = 0.1$, $\mu = 1$, $x_0(0) = q(0)$, and $k_{\mathrm{r}_0} = 0.1$.

5.4.3 Loaded Squats

For the squatting experiments, the high-level controller gains were tuned for a single payload (20 lbs) and were not re-tuned for any of the further experiments. The intention was to verify the adaptive controller's ability to track the trajectory under various payloads. Experiments were conducted with payloads up to 70 lbs. The low-level force controller was modified slightly to accommodate the larger forces required for larger payloads. The PI control law was augmented with gain scheduling on the proportional component. The modified control law is as follows,

$$u(t) = k_{\mathrm{p,m}}(t)\, e_{\mathrm{f}}(t) + k_{\mathrm{I}} \int_0^t e_{\mathrm{f}}(t)\mathrm{d}t, \tag{5.41}$$

$$k_{\mathrm{p,m}}(t) = k_{\mathrm{p}} + \epsilon |e_{\mathrm{f}}(t)|, \tag{5.42}$$

where $u(t)$ is the motor current, $e_{\mathrm{f}}(t)$ is the force tracking error, k_{p} is the nominal proportional gain, k_{I} is the integral gain, and ϵ is a tuning parameter that determines how quickly the proportional component of the control input is increased with increasing error. A saturation limit was imposed on $k_{\mathrm{p,m}}(t)$ in order to preserve stability and reduce oscillations.

In order to achieve purely vertical movement of the hip, the desired ankle pitch is specified as a function of the measured knee pitch such that the displacement of the hip in the x-direction is equal to zero. This function was obtained by solving the inverse kinematics of the 2DOF leg numerically. Figure 5.10 gives a diagram of the angles used in the inverse kinematics for clarity.

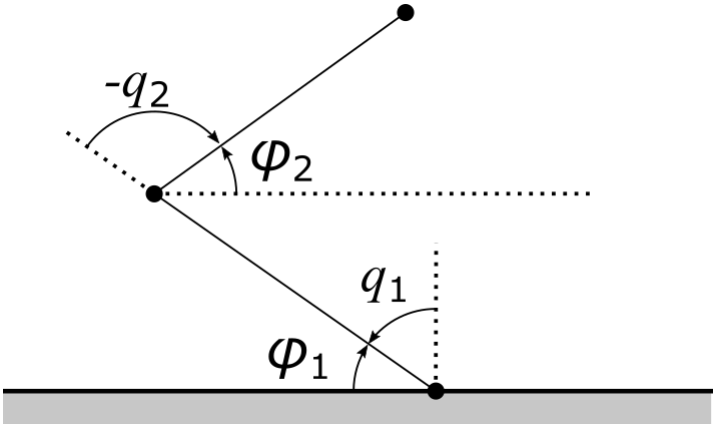

Figure 5.10: Angles used in the leg inverse kinematics for squatting.

The following constraint equation specifies zero x-displacement of the hip,

$$L_1 \cos(\phi_1) = L_2 \cos(\phi_2), \tag{5.43}$$

where ϕ_1 and ϕ_2 are given by

$$\phi_1 = \frac{\pi}{2} - q_1, \tag{5.44}$$

$$\phi_2 = \cos^{-1}\left(\frac{L_1}{L_2}\cos(\phi_1)\right). \tag{5.45}$$

While an explicit solution for q_1 in terms of q_2 does not exist, a relationship between q_1 and q_2 in which (5.43) is satisfied can be derived by solving the inverse kinematics numerically.

From the definition of ϕ_2, we know

$$q_2 = \phi_2 - \frac{\pi}{2} - q_1. \tag{5.46}$$

Therefore the relationship between q_1 and q_2 for a squatting motion can be obtained by substituting a range of values for q_1 into (5.46) and finding a best fit curve of the resulting data. In this case the relationship was found to be approximately linear, and therefore the desired ankle pitch $q_{1,\mathrm{d}}$ required to maintain zero x-displacement given an arbitrary trajectory of the knee pitch is as follows,

$$q_{1,\mathrm{d}}(t) = -0.4892 q_2(t). \tag{5.47}$$

Finally, the linear velocity of the knee joint was assumed to be negligible in order to make two significant simplifications: 1) the control law (5.22) is simplified by eliminating the need to perform vector cross products, 2) the need for the knee controller to receive velocity feedback from the 2DOF ankle controller is eliminated. While this assumption does not reflect reality, the knee torque controller was found to have satisfactory performance due to the velocities being relatively small.

5.4.4 Results

Position Regulation

The position regulation experiments were conducted with various setpoints for the knee joint while the ankle joint was maintained in the vertical configuration, $q_1 = 0$. Experiments were conducted both with and without disturbances. Disturbances were applied as demonstrated

in Figure 5.15: force was applied near the location of the hip connection point, in the direction of the positive x-axis of the world frame. Each disturbance lasted a few seconds, and the robot was allowed to return to the upright position before the next disturbance was applied. The following gains were used in the regulation experiments: $K_1 = 2$, $K_2 = 0.1$, $\lambda_{\mathrm{p}} = 2$, $\gamma_1 = 1$, $\gamma_2 = 10$, $\gamma_3 = 0.03$, $\zeta = 1$, $\omega_n = 3$, $k_{\mathrm{p}} = 0.23$, $k_{\mathrm{I}} = 0.15$. The results of an experiment with a setpoint of $q_{\mathrm{d}} = -4°$, the initial condition $q_2(0) = -1.8°$, and no disturbances are presented in Figures 5.11-5.14. Experimental results with disturbances and the setpoint $q_{\mathrm{d}} = -4°$ are given in Figures 5.16-5.18.

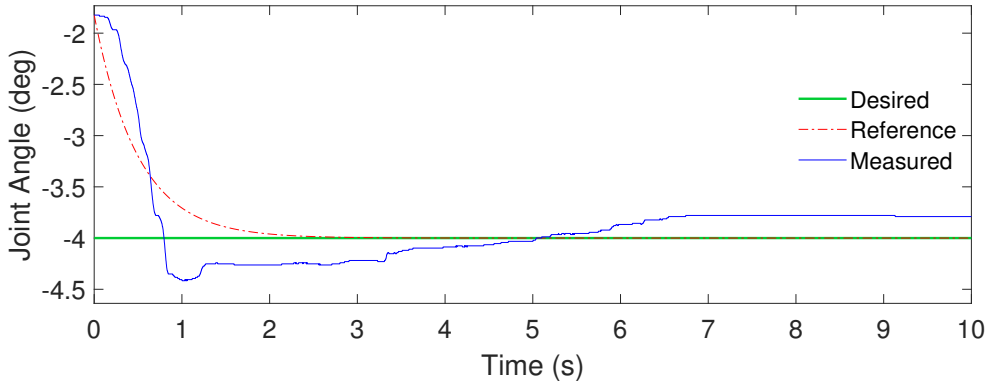

Figure 5.11: Position regulation performance with a setpoint of $-4°$.

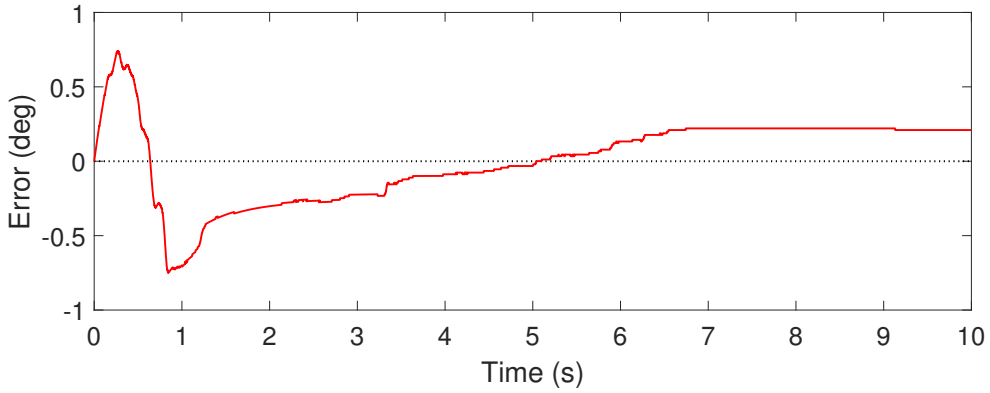

Figure 5.12: Error between the knee joint angle and the position reference model over time. The steady state error is approximately $0.2°$.

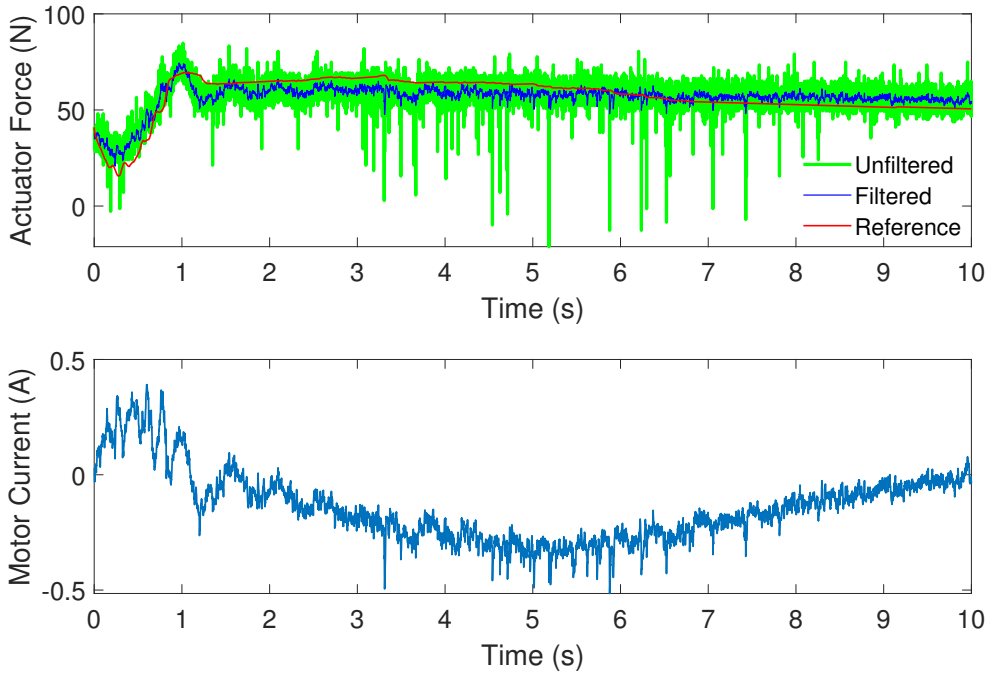

Figure 5.13: Actuator force and motor current during the position regulation experiment.

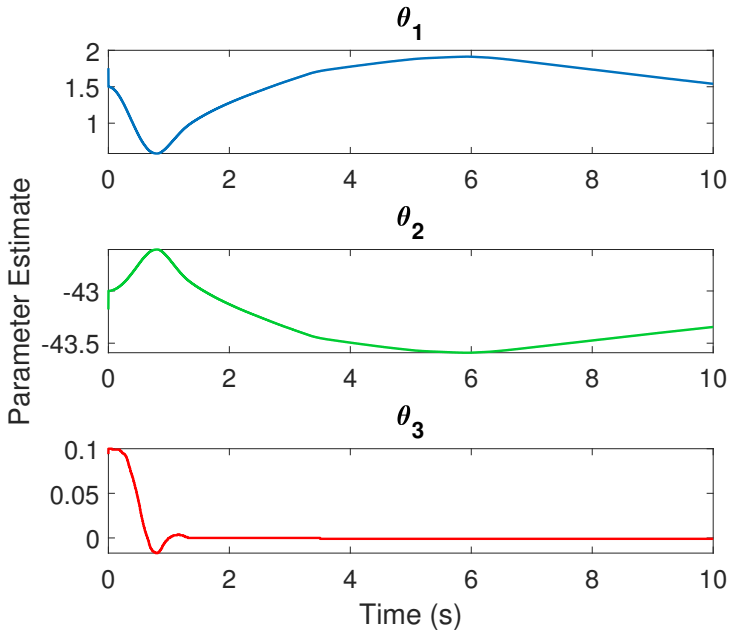

Figure 5.14: Evolution of the adaptive parameters during the position regulation experiment.

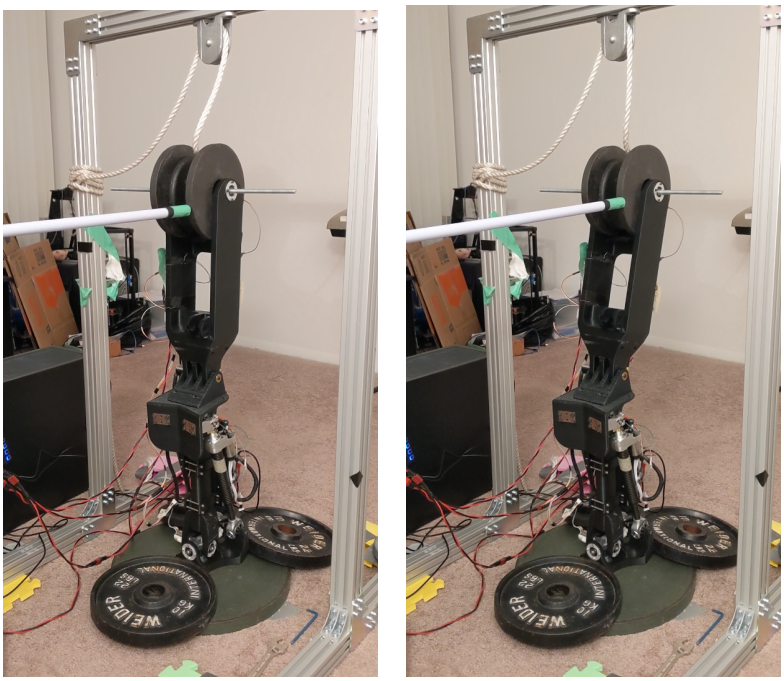

Figure 5.15: Experiment with compliant position regulation in which disturbances were applied to the leg as it attempted to maintain the straight configuration ($q_1 = q_2 = 0$). Note that only the knee torque controller has compliance, as the 2DOF ankle was operating with a high-gain position controller.

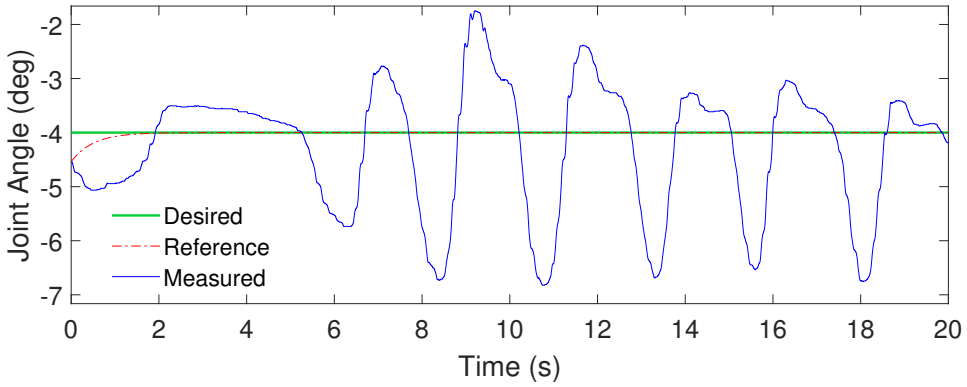

Figure 5.16: Position regulation performance with disturbances. The robot was pushed at approximately 5.5 s, 7.5 s, 10 s, 12.5 s, 15 s, and 17.5 s.

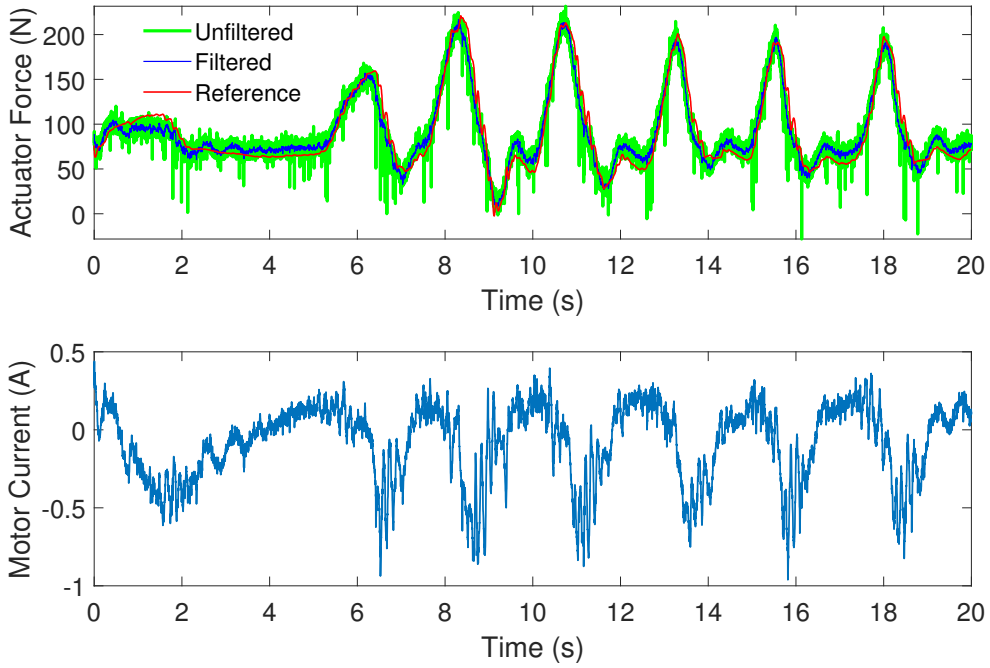

Figure 5.17: Actuator force and motor current during the position regulation experiment with disturbances.

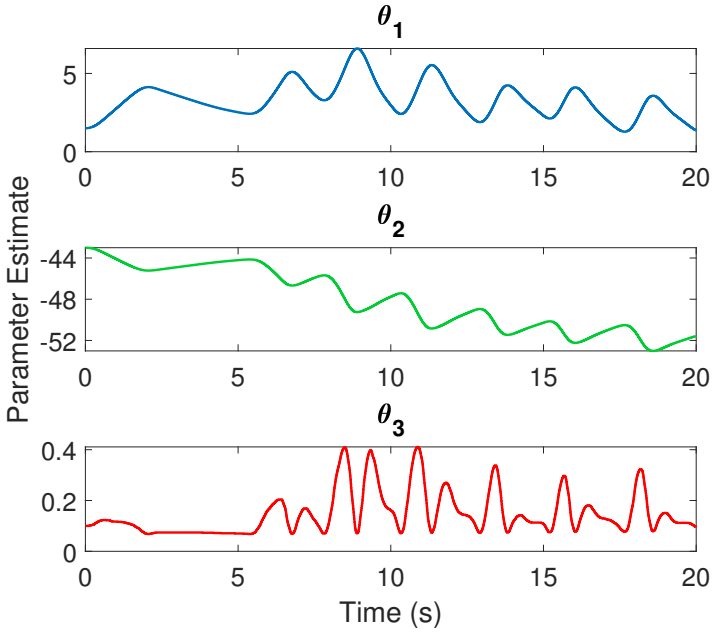

Figure 5.18: Adaptive parameters during the position regulation experiment with disturbances.

Loaded Squats

Experiments were performed with payloads of up to 70 lbs to test the ability of the adaptive controller to maintain tracking performance under different loading conditions. These trials also reflect the adaptive controller's ability to achieve the performance of a model-based controller with an unknown model. High-level torque control gains were tuned for a payload of 20 lbs and were not adjusted for the other trials. The knee and ankle joint trajectories were specified in order to maintain purely vertical movement of the hip. The following high-level controller gains were used: $K_1 = 17$, $K_2 = 1$, $\lambda_{\mathrm{p}} = 0.5$, $\gamma_1 = 1.25$, $\gamma_2 = 20$, $\gamma_3 = 0.03$, $\omega_n = 3$, and $\zeta = 1$. The low-level controller gains were $k_{\mathrm{p}} = 0.05$, $k_{\mathrm{I}} = 0.08$, $\epsilon = 10^{-3}$, and the saturation limit of $k_{\mathrm{p,m}}(t)$ was set to 0.1. Results for payloads of 20 lbs, 45 lbs, and 70 lbs are presented in this section. Figures 5.24 and 5.29 show the experimental setup with payloads of 45 lbs and 70 lbs respectively.

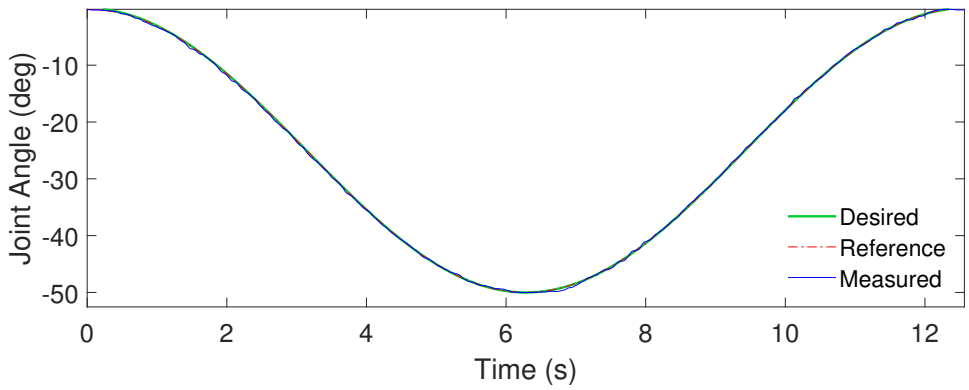

Figure 5.19: Tracking performance from squatting with a 20 lb payload.

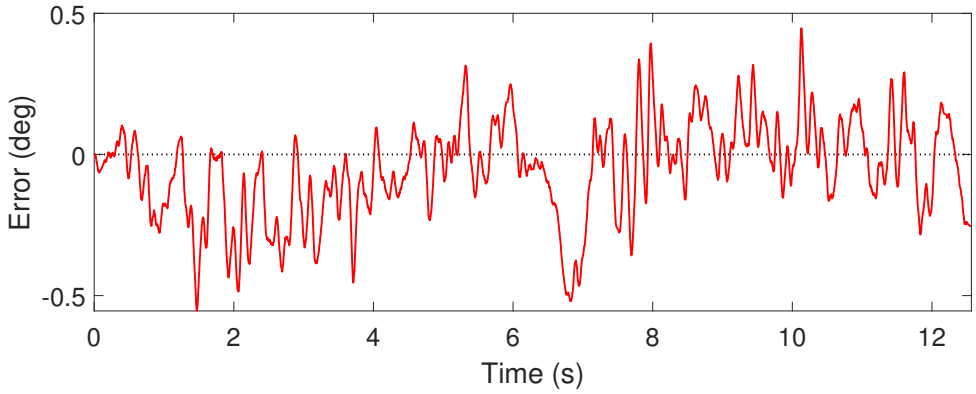

Figure 5.20: Position tracking error over time with a 20 lb payload. The most notable peak occurs when the knee joint is moving out from the bottom of the squat at approximately 6.5 s.

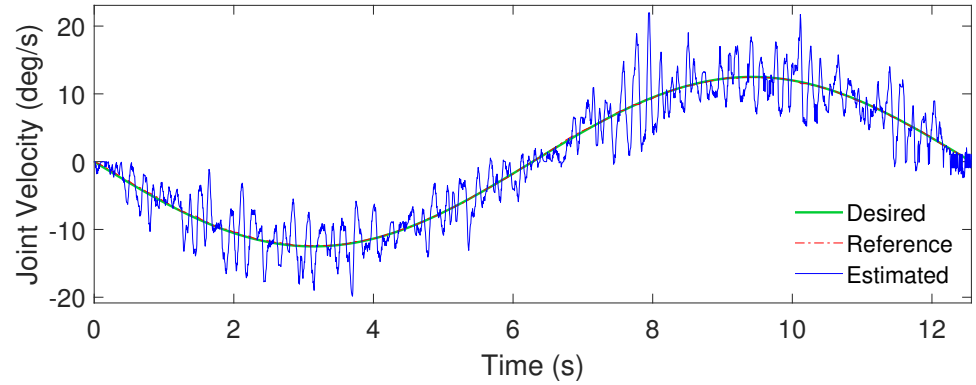

Figure 5.21: Velocity tracking performance while squatting with a 20 lb payload. The velocity signal shows vibrations occurring in the joint. These vibrations are believed to be caused by external noise corrupting the loadcell measurements, which negatively impacts the low-level force controller.

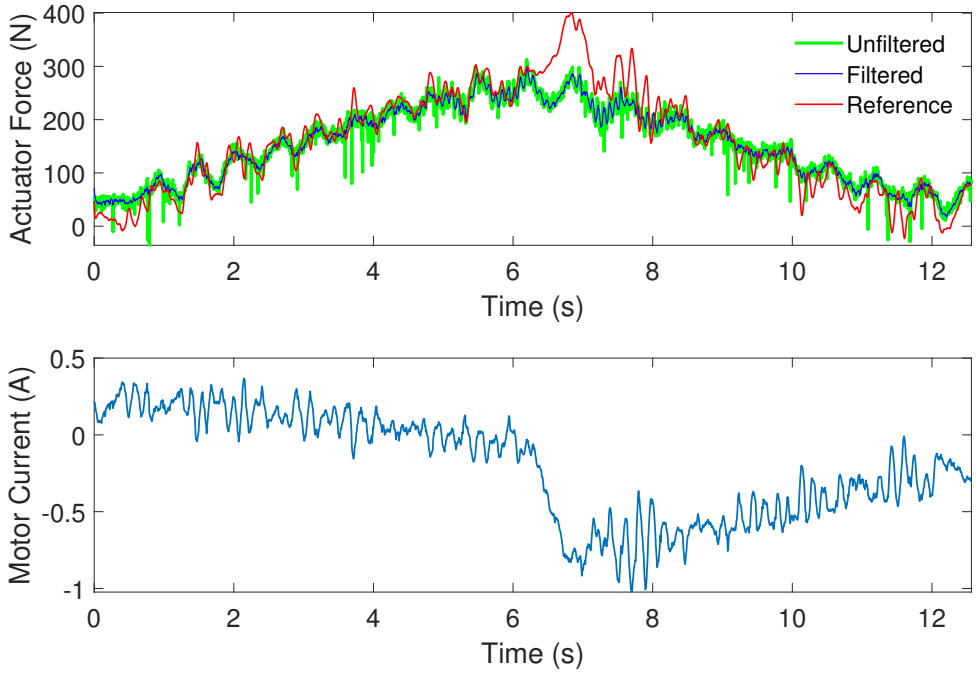

Figure 5.22: Actuator forces and motor current while squatting with a 20 lb payload.

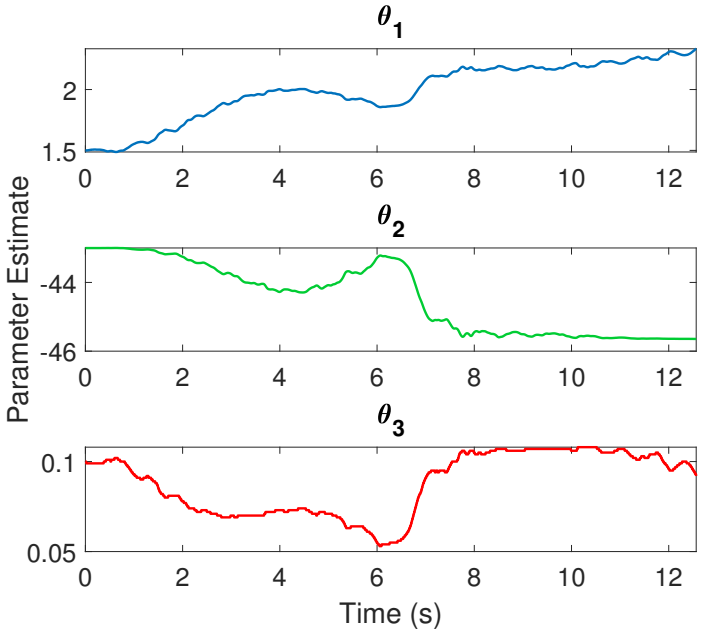

Figure 5.23: Adaptive parameters while squatting with a 20 lb payload.

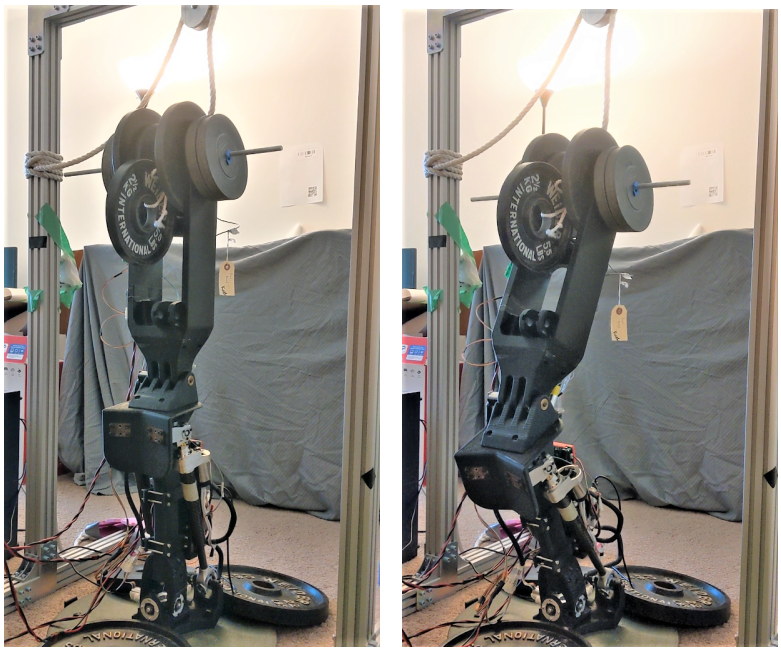

Figure 5.24: Experiment using the hybrid position/torque control scheme with a payload of 45 lbs at the beginning of the squat (left) and in the middle of the squat (right).

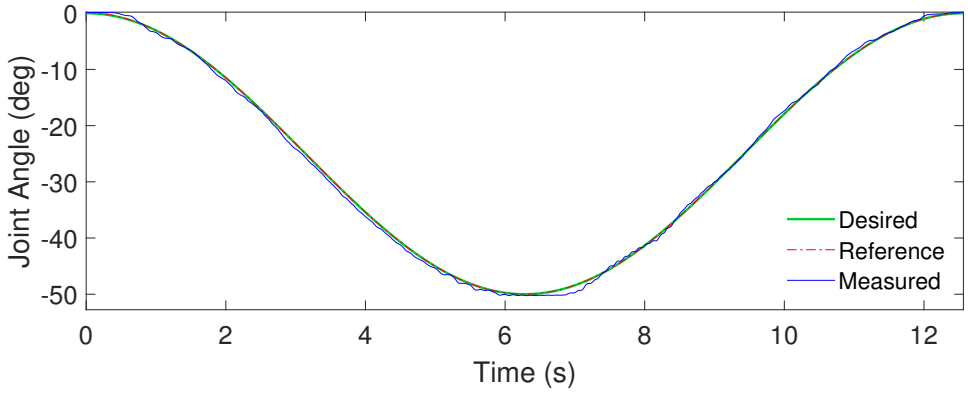

Figure 5.25: Position tracking performance from squatting with a 45 lb payload.

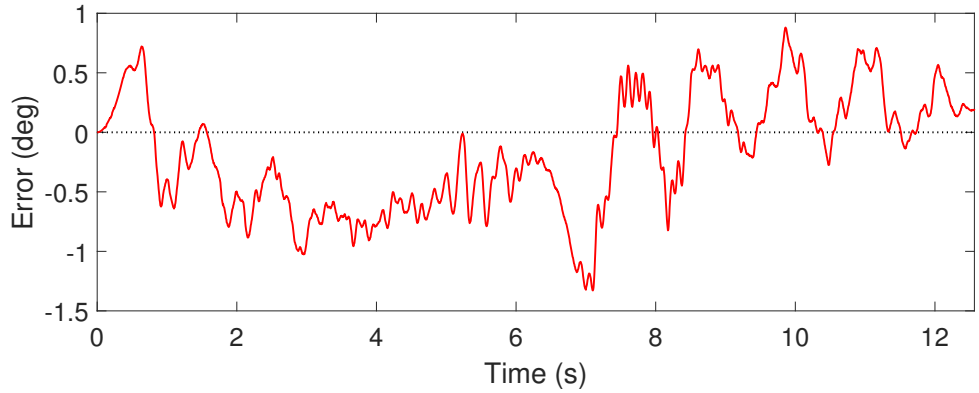

Figure 5.26: Position tracking error while squatting with a 45 lb payload.

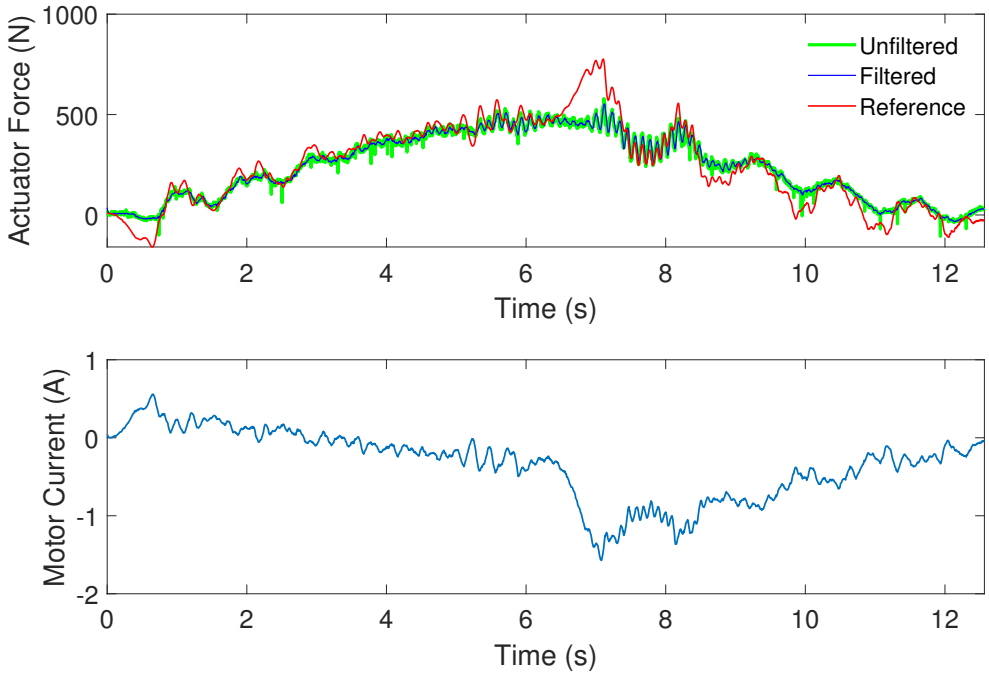

Figure 5.27: Force tracking and motor current while squatting with a 45 lb payload.

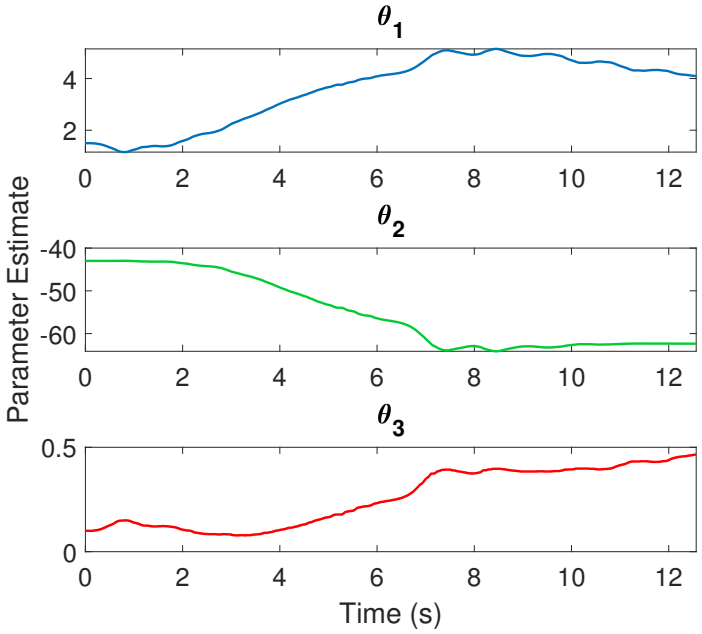

Figure 5.28: Adaptive parameters while squatting with a 45 lb payload.

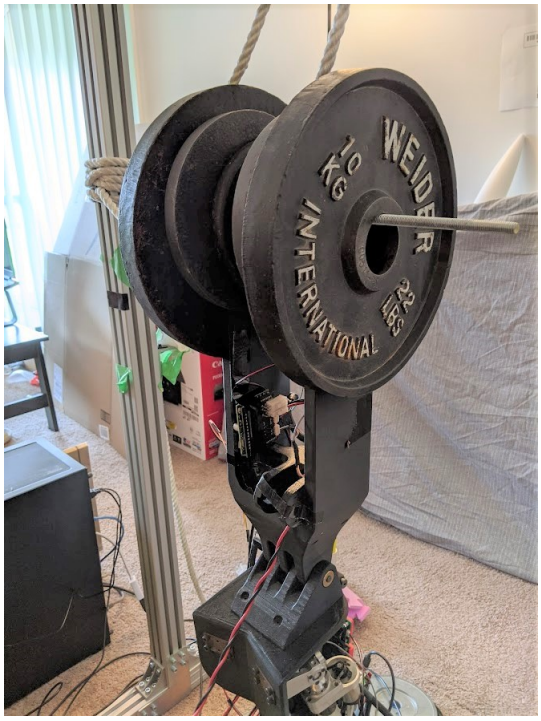

Figure 5.29: Experimental setup of the leg holding a total payload of 70 lbs.

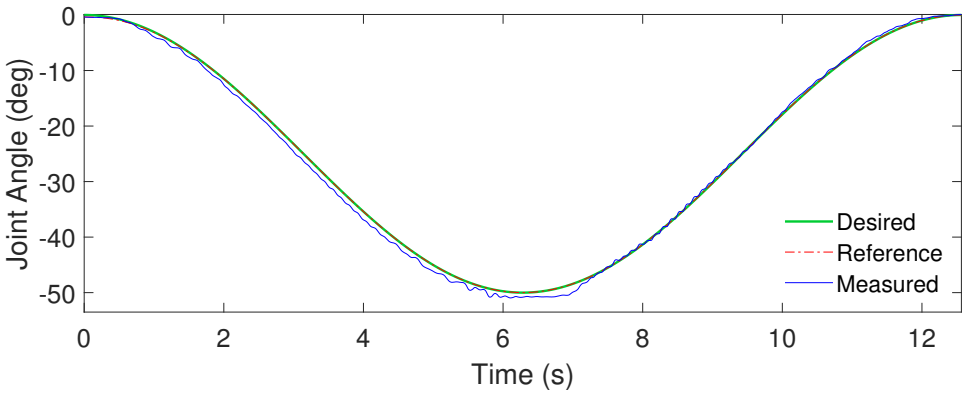

Figure 5.30: Position tracking performance while squatting with a 70 lb payload.

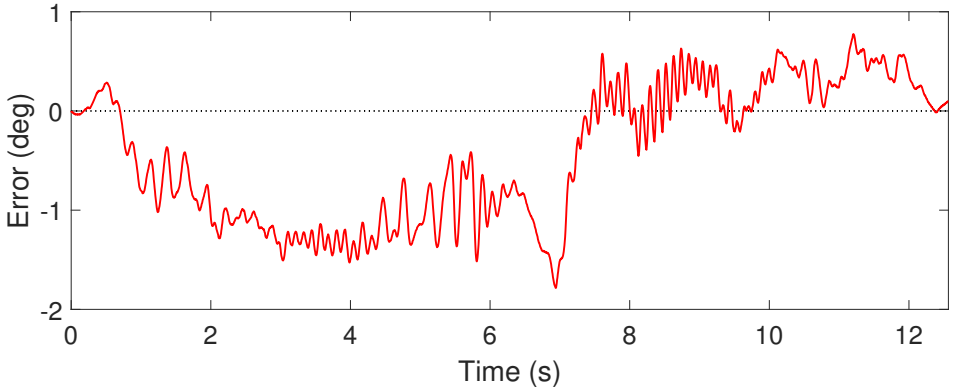

Figure 5.31: Position tracking error while squatting with a 70 lb payload.

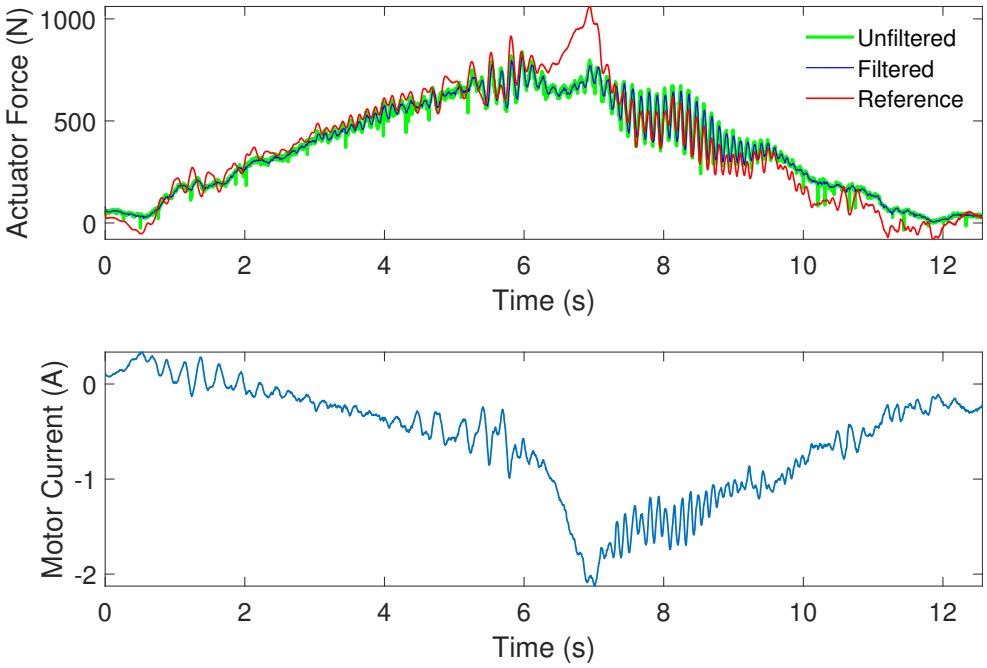

Figure 5.32: Actuator force tracking and motor current while squatting with a 70 lb payload.

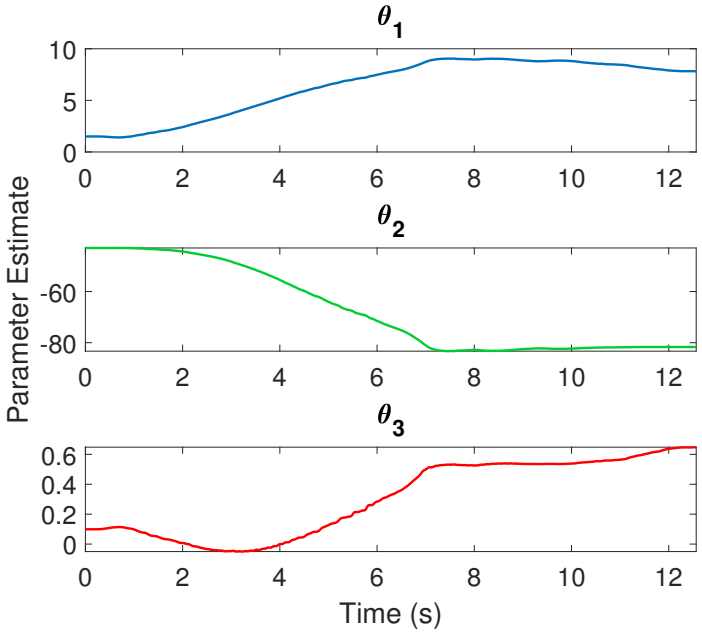

Figure 5.33: Evolution of the adaptive parameters while squatting with a 70 lb payload.

Discussion of Results

Overall the experiments showed good tracking performance with position errors within $\pm 2^{\circ}$. The tracking was also found to be consistent with different payloads, with minimal loss of performance with increasing weight. High frequency oscillations were observed in all of the experiments, most notably through the control input and velocity tracking. These oscillations were not significantly visible in the position tracking up until a 70 lbs payload was tested. These oscillations are likely caused by noise in the ADC measurements. Though software low-pass filtering was implemented by feeding raw measurements through a single-pole filter, hardware filtering would have been ideal in order to achieve proper anti-aliasing of the loadcell signals. Oscillations could have also been caused by the assumption of negligible shin link dynamics, which is of course not an accurate representation of reality. The low-level force controller showed some large tracking errors at the deepest points of the squats. This was due to the necessity of the high level controller to command very large forces to bring the leg back up from the bottom of the squats. As the high-level controller functions independently of the low-level force controller, the adaptive parameters are adjusted such that the high level controller commands whatever force is required to drive the joint trajectories to the reference model, regardless of the low-level force tracking performance. However, the generally good position tracking and the reasonably consistent performance under a wide range of payloads indicates that the adaptive controller is functioning as expected, and that the low-level controller is able to supply the forces needed to drive the robot dynamics to behave in the desired manner. Therefore the experimental results serve as a proof of concept of the control scheme developed in this chapter.

Chapter 6

Conclusions and Future Work

The goal of this thesis was to find a viable way to control a 3D-printed robot with unknown inertial properties such that compliant trajectory tracking could be accomplished under a variety of different operating conditions. An early prototype consisting of a single 3DOF leg was used to validate the controller design though experiments. In order to simplify the control problem while still maintaining a level of depth sufficient achieve the desired tracking performance, the 3DOF prototype was modeled as a 2DOF leg in the sagittal plane. In Chapter 3, the sagittal plane model was developed and an adaptive controller for high-level joint space impedance control was derived. Simulations were conducted with and without disturbances for tracking control with sinusoidal trajectories and for position regulation with step inputs. The proposed controller was found to perform quite well in simulation, so in Chapter 4 various low-level force control algorithms were derived and evaluated to find a good fit for the experimental implementation. PI, MRAC, and ADP-PI controllers were tested in two different configurations, and ultimately the PI controller was selected for implementation on the robot due to its effectiveness and simplicity.

In Chapter 5, the sagittal plane model was modified to include only a single degree of freedom due to hardware limitations at the time of testing, and a hybrid position control and torque control scheme was implemented in the experiments. An adaptive controller with very similar structure to the one derived in Chapter 3 was developed for use in the hybrid controller experiments. The adaptive torque controller used in the knee joint with

PI control for low-level force tracking was found to have satisfactory tracking performance with and without disturbances, and was also found to be fairly consistent with payloads up to 70 lbs. Therefore the ability of the adaptive controller to maintain accurate tracking regardless of the inertial properties of the robot was validated through the experiments, and the implementation was an overall success, though certain aspects of the design can be improved.

For future implementations of this controller, it is important to implement proper filtering and conditioning of the analog signals. Noise in the load cell measurements was found to significantly impact the low-level force controller performance and induce oscillations in the leg. Since the number of adaptive parameters required to approximate the robot model grows exponentially with each additional degree of freedom, the reparameterization of the dynamics presented in Chapter 3 will be difficult for higher order models. It may be useful to automate the reparameterization in a symbolic math program, potentially with some kind of recursive algorithm to find the nonlinear terms containing the state variables. If one wishes to reduce the number of adaptive parameters and the number of nonlinear functions required to provide full dynamics compensation, one may consider removing the Coriolis and centripetal terms from the adaptation algorithm, as well as any terms in the mass-inertia matrix that are off the main diagonal. This tactic was implemented in [11] and produced favorable results in simulation. Neglecting the Coriolis and centripetal terms may prove very useful as the nonlinear equations associated with them are relatively complicated, and the resulting torques are typically very small for walking that is not highly dynamic. If one wishes to simplify the adaptive controller even further, they may consider a near model-free approach as the authors did in [15]. By approximating the structure of the dynamics rather than considering the full equations of motion, the authors were able to greatly reduce the computational effort required to compute the control inputs. When adequate hardware and

computing resources are available, adaptive control has great potential for use in a final implementation of the robot due to its robustness and versatility.

Bibliography

[1] *AZBDC12A8 Servo Motor Drive*. Advanced Motion Controls, 2018. URL https://www.a-m-c.com/product/azbdc12a8/.

[2] Karl Johan Astrom and Richard M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, USA, 2008. ISBN 0691135762.

[3] Chih-Chiang Cheng and Ming-Wen Chang. Design of derivative estimator using adaptive sliding mode technique. In *2006 American Control Conference*, pages 5–pp. IEEE, 2006.

[4] Paola Jaramillo Cienfuegos, Adam Shoemaker, Robert W Grange, Nicole Abaid, and Alexander Leonessa. Classical and adaptive control of ex vivo skeletal muscle contractions using functional electrical stimulation (fes). *PloS one*, 12(3):e0172761, 2017.

[5] John J. Craig, Ping Hsu, and S. Shankar Sastry. Adaptive control of mechanical manipulators. *The International Journal of Robotics Research*, 6(2):16–28, 1987.

[6] Alessandro De Luca and Pasquale Lucibello. A general algorithm for dynamic feedback linearization of robots with elastic joints. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, volume 1, pages 504–510. IEEE, 1998.

[7] *FUTEK Model CSG110*. Futek, 2018. URL https://media.futek.com/content/futek/files/pdf/productdrawings/csg110.pdf.

[8] *LCM Series Tension and Compression Sensor Family Manual*. Futek, 2018. URL https://media.futek.com/docs/support/LCMSERIESMANUAL.pdf.

[9] *Gurley Models A19 & A20 Absolute Encoder*. Gurley Precision, 2018. URL https://www.gurley.com/s/A19-tp33.pdf.

[10] Michael Anthony Hopkins. *Dynamic Locomotion and Whole-Body Control for Compliant Humanoids*. PhD thesis, Virginia Tech, 2015.

[11] R. Horowitz and M. Tomizuka. An adaptive control scheme for mechanical manipulators. compensation of nonlinearity and decoupling control. *Journal of dynamic systems, measurement, and control*, 108(2):127–135, 1986.

[12] Petros A. Ioannou and Jing Sun. *Robust Adaptive Control*. PTR Prentice-Hall, Upper Saddle River, NJ, 1996.

[13] Paola Jaramillo, Adam Shoemaker, and Alexander Leonessa. Skeletal muscle contraction control and tracking using an adaptive augmented pi control approach. In *Converging Clinical and Engineering Research on Neurorehabilitation*, pages 135–139. Springer, 2013.

[14] Hassan Khalil. *Nonlinear Systems*. Prentice Hall, Upper Saddle River, New Jersey, 3rd edition, 2002.

[15] Said Ghani Khan, Guido Herrmann, Tony Pipe, Chris Melhuish, and Adam Spiers. Safe adaptive compliance control of a humanoid robotic arm with anti-windup compensation and posture control. *International Journal of Social Robotics*, 2(3):305–319, 2010.

[16] Coleman Knabe, John Seminatore, Jacob Webb, Michael Hopkins, Tomonari Furukawa, Alexander Leonessa, and Brian Lattimer. Design of a series elastic humanoid for the darpa robotics challenge. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 738–743. IEEE, 2015.

[17] M. Kristić, I. Kanellakopoullos, and P. Kotović. *Nonlinear and Adaptive Control Design*. Wiley, New York, 1995.

[18] Derek Frei Lahr. *Design and control of a bipedal robot*. PhD thesis, Virginia Polytechnic Institute and State University, 2014.

[19] *Maxon EC-4pole*. Maxon, 2017. URL https://www.maxongroup.com/medias/sys_master/root/8841184018462EN-251.pdf.

[20] James L. Meriam and L. Glenn Kraige. *Engineering Mechanics: Dynamics*, volume 2. John Wiley & Sons, 7th edition, 2012.

[21] Jonas Peter. Angular momentum about a moving point, 2017. URL https://physics.stackexchange.com/questions/346829/angular-momentum-about-a-moving-point.

[22] Jean-Baptiste Pomet, Laurent Praly, et al. Adaptive nonlinear regulation: Estimation from the lyapunov equation. *IEEE Transactions on automatic control*, 37(6):729–740, 1992.

[23] Nader. Sadegh and Roberto. Horowitz. Stability and robustness analysis of a class of adaptive controllers for robotic manipulators. *The International Journal of Robotics Research*, 9(3):74–92, 1990.

[24] Jean-Jacques E. Slotine and Weiping Li. On the adaptive control of robot manipulators. *The International Journal of Robotics Research*, 6(3):49–59, 1987.

[25] Mark. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, Hoboken, NJ, 2006.

[26] *Tiva™ TM4C123GH6PM Microcontroller DATA SHEET*. Texas Instruments, 2014. URL https://www.ti.com/lit/pdf/spms376.

[27] E. Westervelt, J. W. Grizzle, C. Chevallereau, H. C. Choi, and B. Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press, Boca Raton, 2007.

[28] Dan Zhang and Bin Wei. A review on model reference adaptive control of robotic manipulators. *Annual Reviews in Control*, 43:188–198, 2017.

Appendices

Appendix A

Symbolic Calculations in Matlab

This section includes a segment of the Matlab code used to derive the symbolic equations of motion of the sagittal plane model using the method of Lagrange.

```
close all
clc
clear
syms q1 q2;
syms dq1 dq2;
syms qrv1 qrv2;
syms dqrv1 dqrv2;

%joint angles and velocities
q =  [q1;q2];
dq = [dq1;dq2];
qrv = [qrv1; qrv2];
dqrv = [dqrv1; dqrv2];

%absolute angles
th1 = q1;
th2 = q1+q2;
%absolute velocities
```

```
dth1 = dq1;
dth2 = dq1+dq2;
syms lcm1x lcm1y L1 lcm2x lcm2y L2; %Link lengths
syms m1 J1 m2 J2 g;
robot_params = [L1 L2 lcm1x lcm1y lcm2x lcm2y m1 m2 J1 J2 g].';

%Anonymous Function for 2D rotation matrix (about z-axis)
rot = @(th) [cos(th) -sin(th);sin(th) cos(th)];

%Forward Kinematics
pcm1 = [0;0] + rot(th1)*[lcm1x;lcm1y];
p1 = [0;0] + rot(th1)*[0;L1];
pcm2 = p1 + rot(th2)*[lcm2x;lcm2y];
p2 = p1 + rot(th2)*[0;L2];

%Find Hip Jacobian
Ja = jacobian(p2,q);
%Find End-effector (Hip) Velocity
vcmH = Ja*dq;    vcmH = simplify(vcmH);

%Define CoM Velocities
Jcm1 = jacobian(pcm1,q); Jcm1 = simplify(Jcm1); vcm1 = Jcm1*dq; vcm1 =
    simplify(vcm1);
Jcm2 = jacobian(pcm2,q); Jcm2 = simplify(Jcm2); vcm2 = Jcm2*dq;    vcm2
    = simplify(vcm2);

%Calculate Kinetic Energy
```

```
K = 1/2*m1*transpose(vcm1)*vcm1+1/2*J1*dth1^2 + ...
    1/2*m2*transpose(vcm2)*vcm2+1/2*J2*dth2^2;
K = simplify(K);

%Find Potential Energy
V = g*(m1*pcm1(2)+m2*pcm2(2));
V = simplify(V);

%Find Mass-Inertia Matrix
dK_dqdot = jacobian(K,dq);
D        = jacobian(dK_dqdot,dq);
D        = simplify(D);

%Find Coriolis and Centrifugal terms
C = 0*D;
N = length(q);
for k=1:N
    for j=1:N
        sum = 0*g;
        for i=1:N
            sum = sum + 1/2*(diff(D(k,j),q(i)) + diff(D(k,i),q(j)) -
                diff(D(i,j),q(k)))*dq(i);
        end
        C(k,j) = sum;
    end
end
Cdq = C*dq;
```

```
Cdq = simplify(Cdq);

%Terms for feedforward control

Ddqrv = D*dqrv;
Ddqrv = simplify(Ddqrv);

Cqrv = C*qrv;
Cqrv = simplify(Cqrv);

%Find Gravity Matrix
G = jacobian(V,q).';

% Reparameterize robot dynamics as the product of a matrix of nonlinear
%functions of known states and a vector of unknown constants

%Reparamaterized Mass-Inertia Matrix * dqrv

W1 = [dqrv1      sin(q2)*(dqrv1+1/2*dqrv2)      cos(q2)*(dqrv1+1/2*dqrv2)
          dqrv2;
      0          1/2*dqrv1*sin(q2)                  1/2*dqrv1*cos(q2)
                          dqrv1+dqrv2];

theta1 = [J1+J2+m1*(lcm1x^2+lcm1y^2)+m2*(L1^2+lcm2x^2+lcm2y^2);
           2*L1*lcm2x*m2;
           2*L1*lcm2y*m2;
           J2+m2*(lcm2x^2+lcm2y^2)];
```

```
%Reparamaterized Coriolis and Centrifugal Terms Matrix * qrv
W2 = [qrv1*dq2*cos(q2)+qrv2*cos(q2)*(dq1+dq2)        -(qrv1*dq2*sin(q2)+
    qrv2*sin(q2)*(dq1+dq2));
       -dq1*qrv1*cos(q2)                              dq1*qrv1*sin(q2)];

theta2 = [L1*lcm2x*m2;
          L1*lcm2y*m2];

%Reparamaterized Gravity Torques Vector
W3 = [cos(q1) -sin(q1) -sin(q1+q2) cos(q1+q2);
       0          0      -sin(q1+q2)  cos(q1+q2)];
theta3 = [m1*g*lcm1x;
          m2*g*L1+m1*g*lcm1y;
          m2*g*lcm2y;
          m2*g*lcm2x];
```

Appendix B

Kinematics of the 2DOF Ankle Mechanism

In this section the contributions of each of the two actuators to their respective torques about the ankle roll and ankle pitch axes will be studied. Figure B.1 gives a diagram of the actuated mechanism. For this analysis, consider the vectors $OA_1 = [x_{A_1} \ y_{A_1} \ z_{A_1}]^{\mathrm{T}}$ and $OA_2 = [x_{A_2} \ y_{A_2} \ z_{A_2}]^{\mathrm{T}}$, which are the position vectors going from the ankle joint to the ends of actuators 1 and 2 respectively, and $OB_{1,q=0} = [x_{B_{1,0}} \ y_{B_{1,0}} \ z_{B_{1,0}}]^{\mathrm{T}}$ and $OB_{2,q=0} = [x_{B_{2,0}} \ y_{B_{2,0}} \ z_{B_{2,0}}]^{\mathrm{T}}$, which indicate displacements from the the ankle joint to the base of each actuator when the joint angles are zero. It then follows that the vectors from the ankle joint to the base of each actuator for nonzero joint angles, OB_1 and OB_2, are given by the following,

$$OB_1 = R_x(-q_{\mathrm{AP}})R_{\mathrm{y}}(q_{\mathrm{AR}})OB_{1,q=0}, \tag{B.1}$$

$$OB_2 = R_x(-q_{\mathrm{AP}})R_{\mathrm{y}}(q_{\mathrm{AR}})OB_{2,q=0}, \tag{B.2}$$

where q_{AP} and q_{AR} are the ankle pitch and ankle roll angles and $R_x(\theta) \in \mathbb{R}^{3\times3}$ and $R_y(\theta) \in \mathbb{R}^{3\times3}$ are rotation matrices about the x-axis and y-axis respectively. Then from vector addition we have the following expression for the actuator lengths,

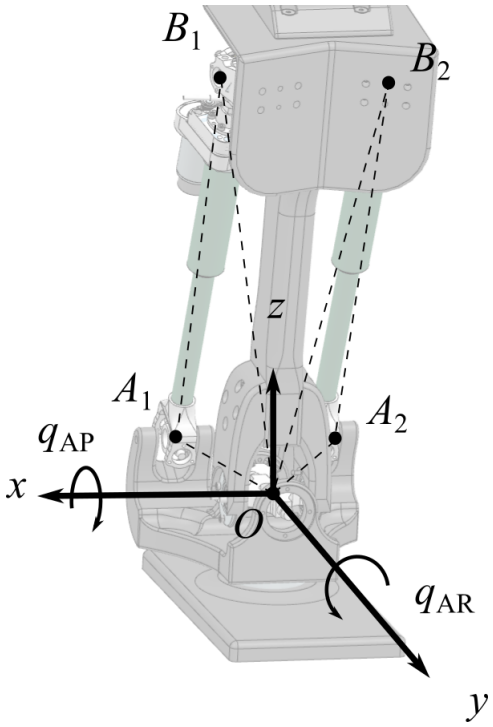

Figure B.1: Diagram of the 2DOF parallel actuated ankle mechanism.

$$A_1B_1 = OB_1 - OA_1, \tag{B.3}$$

$$A_2B_2 = OB_2 - OA_2, \tag{B.4}$$

where A_1B_1 is the length of actuator 1 and A_2B_2 is the length of actuator 2. The force vectors for the two actuators are written as follows,

$$F_{A_1} = f_1 \frac{A_1B_1}{||A_1B_1||}, \tag{B.5}$$

$$F_{A_2} = f_2 \frac{A_2B_2}{||A_2B_2||}, \tag{B.6}$$

where f_1 and f_2 are the magnitudes of the forces in the two actuators. The mapping from an actuator's force to its contribution to the ankle pitch and roll torques can be derived in two ways. The first method is to take the cross product of the moment arm and the force

vector as follows,

$$\tau_i = OB_i \times F_{A_i} \,, \tag{B.7}$$

where i is an indexing value representing the actuator considered. The torque vector can then be projected onto the ankle pitch and ankle roll axes to find the resultant ankle pitch and ankle roll torques. The alternative method would be to find the Jacobian matrix $J_i(q) = \frac{\partial A_i B_i}{\partial q}$ for each actuator and use the principle of virtual work, resulting in the following,

$$\tau_{A_i} = J_i(q)^{\mathrm{T}} F_{A_i} \,, \tag{B.8}$$

where $\tau_{A_i} = [\tau_{\mathrm{AP}_i} \ \tau_{\mathrm{AR}_i}]^{\mathrm{T}}$ is the vector of toques acting on the ankle pitch and roll due to actuator i.