Smart Exercise Adaptive Control of a Three Degree of Freedom Upper-limb Manipulator Robot

by

Aastav Sasha Sen

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2021

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

An adaptive velocity field controller for robotic manipulators is proposed in this thesis. The control objective is to cause the user to exercise in a manner that optimizes a criterion related to the user's mechanical power. The control structure allows for passive user-manipulator physical interaction while the adaptive algorithm identifies the user's biomechanical characteristics as a linear Hill based force-velocity curve defined at each pose of a repetitive exercise motion i.e. a Hill surface. The study of such a surface allows for the characterization of maximal effort exercise tasks and subsequently the control of exercises that is unique to each user. This allows for the intelligent characterization of a user's abilities such that repetitive exercises defined by velocity fields can be safely performed. Such a study involving a 3DOF manipulator operating in full 3D has not been conducted in literature to the best of author's knowledge. The proposed control structure is verified through experimentation on a unimanual setup of the BURT rehabilitation manipulator system involving a single user. The manipulator system includes friction, actuator/sensor noise, and unmodelled dynamics.

Acknowledgements

The completion of this thesis would not have been possible without the guidance of my supervisor, Dr. Soo Jeon. I owe him a debt of gratitude for giving me this opportunity.

In addition, I would like to thank all the researchers on whose work I was able to build.

I also thank my parents for their blessings and unconditional support.

Table of Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Symbol	Description
pHRI	Physical Human-Robot Interaction
PVFC	Passive Velocity Field Control/ Controller
PIC	Passive Impedance Control
BURT	Barrett Upper-Extremity Robotic Trainer
EMG	Electromyography
PBC	Passivity-Based Control
EL	Euler-Lagrange
DOF	Degree of Freedom
DH	Denavit-Hartenberg
RRR	3 revolute joint manipulator
SCARA	Selective Compliance Articulated Robot Arm
PAA	Parameter Adaptation Algorithm

Accents and Subscripts

Symbol	Description
a^{+}	positive quantity
a^{*}	optimal quantity
$\hat{a}$	estimate of quantity
$\tilde{a}$	error of estimate of quantity
a_0	initial quantity
$\dot{a}$	velocity quantity
$\ddot{a}$	acceleration quantity
$\bar{a}$	augmented system quantity
a_x	Cartesian /task space vector quantity
a_q	joint space vector quantity
a_k	quantity in period $[T_k, T_{k+1}]$
a_f	quantity related to the fictitious flywheel state
a_d	Cartesian /task space vector quantity related to the desired path
a_μ	scalar quantity related to the desired path
$\|\|a\|\|$	Euclidean norm of vector quantity
$\|a\|$	absolute value of quantity

a_h	quantity associated with Hill surface

Roman Symbols

Symbol	Description	Units
T	quantity of time	s
q_i	joint angle of joint i	rad
J	robot Jacobian matrix	m/rad
f_h	Hill model of user applied force	N
a	Hill parameter (maximum force)	N
b	Hill parameter (velocity drop off)	Ns/m
$\hat{F}$	stable filtered output of environmental forces	N
F	environmental forces	N
R	n dimensional configuration manifold	-
G	configuration space along x_d	-
$\mathbb{R}$	set of real numbers	-
M	robot Inertia matrix	kg
C	robot Coriolis matrix	kg/s
B	robot viscous friction affects	kg/s
D	damping matrix	kg/s

k	kinetic energy of manipulator system	J
E_{max}	maximum kinetic energy of augmented system	J
p	momentum	$kg\,m/s$
P	desired momentum	$kg\,m/s$
w	inverse compensation term	Nm
V	desired velocity/ speed	$m/s, rad/s$
m	quantity of mass	kg
I_i	inertia tensor of link i	$kgm2$
I	identity matrix	-
H	symmetric Hilbert-Schmidt kernel (discrete time)	-
h	single row of H	-
$\mathcal{H}$	discrete time filter (ZOH)	-
c	vector of unknown influence functions	-
L	length of desired contour	m
$\boldsymbol{S}$	state transition event	-
S	supply rate	W
s	Laplace transform complex variable	Hz
ν	window length (in segments) of discrete Gaussian approximation	-
N	resolution of the discrete kernel	-

e	measure of error	-
$\boldsymbol{P}$	PAA gain matrix	-
$\boldsymbol{d}$	PAA signal vector	-
Q	binary state	-
g	measure of optimality error	-

Greek Symbols

Symbol	**Description**	**Units**
μ	scalar path variable that parameterizes the desired contour	m
ϕ	regressor signal	-
Φ	desired regressor signal	-
ρ	filtered regressor signal	-
τ_i	applied torque on joint i	Nm
τ_e	environmental torques	Nm
$\hat{\tau}_e$	stable filtered output of environmental torques	Nm
τ_g	torques due to gravity	Nm
τ_{PVFC}	PVFC control torques (coupling + feedback)	Nm
τ_c	PVFC coupling control torques	Nm

τ_F	PVFC feedback control torques	Nm
τ_D	injected damping torques	Nm
α	tracked scalar multiple of reference velocity	-
θ	unknown parameter vector	-
$\aleph$	skew-symmetric matrix	-
λ	force observer cutoff frequency	rad/s
ω_c	force observer cutoff frequency	Hz
λ_a	PAA forgetting factor	-
λ_L	training velocity gain	-
ω_s	sampling frequency	Hz
Γ	PAA adaptation gain	-
Π	objective function	W
η	arbitrary continuous function	-
γ	PVFC feedback gain	-
γ_1	optimality error gain	-

Chapter 1
Introduction

The task of modelling and controlling robotic manipulation systems is heavily researched and, in general, well understood. However, the robustness and effectiveness of manipulator control algorithms go beyond the modelling of the robot and is also subject to the environment that the robot manipulator interacts with. This is especially true if the environment is as dynamic as an interacting human user. Unknown dynamic environments as well as unmodeled robot dynamics hinder the robustness and performance capabilities of manipulator control algorithms. The parameters that define this uncertainty are often difficult to pre-calculate or measure and this has led to a growth in the research activity of physical human-robot interaction (pHRI) solutions based on adaptive control and learning algorithms.

Power assisting devices can be found in many different applications, most notably, exercise/fitness equipment, rehabilitation of spinal cord injury or stroke patients, limb muscle enhancement of workers, and bionic limbs of amputee patient. Within the realm of pHRI, robotic rehabilitation is quickly becoming an important area of research as societal demand for the safe and effective treatment of physical disabilities grows. Rehabilitation robots work just like assistive ones by treating all signals and interactions from the human user as disturbances, although traditional control methods are still applicable to them, human intentions and movements should be encouraged and complemented rather than rejected. Cerebrovascular disorders and traumatic brain injury are at present the leading causes of disability resulting in partial of complete motor limitation in upper and lower limbs [1], [2]. Stroke is one of the leading causes of adult disability and the risk increases with age. Following the acute phase, all patients require continuous medical care. The theory of synaptic plasticity, which was later summarized as “neurons that fire together, wire together” [3] has been proposed for more than a century and points towards the importance of physiotherapy and rehabilitation in aiding patient recovery. However, rehabilitation is labor intensive, often necessitating one-on-one manual interaction with therapists [4]. Due to limited human and financial resources dedicated to rehabilitative practices, patient are usually sent home when they have partially recovered even though their motor functions are severely impaired and this is true even though there is evidence that patients would improve with further therapy [5]. This affects their quality of life and independence in daily activities. One of the motivations of rehabilitation robotics is a reduced need for labor and a reduced running cost of rehabilitation services as well providing patients with autonomy [6], [7]. With the rise of technology, haptic environments have also proven to allow for fast recovery with added entertainment. It is also desirable to have quantitative evaluation of motor abilities in the effort to obtain an objective evaluation of rehabilitation and pharmacological treatment effects; a task that could be aided through robotic intervention.

With the introduction of human-cooperative robots into the workplace to aid a person in the completion of a manual task two key aspects must be considered: the human-robot interface must be intuitive to use and the safety of the user with respect to injuries inflicted by collisions with the robot must be guaranteed [8]. Safety in human-robot interaction is usually addressed from the following perspectives: mechanical design (intrinsic safety), safety assessment, control and planning. Various causes can incur injury, especially in rehabilitative applications where paralysis or spasticity causes high stiffness of human joint and inability to react fast. Possible injuries include bone damage, joint damage, and soft tissue fracture, severity of which may range from minor to fatal [9]. This is why passivity is a desirable property because it is necessary to ensure a stable interaction with any unknown environment [10], [11]. Control laws must also provide asymptotic stability guarantees [12] and be well-defined in the robot workspace to ensure safety. The control algorithm covered in this thesis not attempts not only to guarantee this, but, also to demonstrate one form of the "assist-as-needed" paradigm, which is proved to be more effective for patient recovery [13].

In this thesis we will consider a novel implementation of self-optimizing passive velocity field control to enable the execution of repetitive exercise motions in 3 DOF (degrees of freedom) while ensuring the safety of the user. This control algorithm is part of the primary control paradigm in robot therapy development which is the active assist exercise [14] that emulates physician-based therapies and enables strength training. The algorithm also provides a solution to the need for quantitative methods of identifying a given user's biomechanical capabilities, allowing for the evaluation of the effectiveness of rehabilitative treatments and patient improvement. The experiment and analysis conducted in this thesis takes [15] and [16] as its main source of motivation. A similar solution has also been applied to rehabilitation exercise machines for wrist motions [17]. High level analysis (general passivity and stability of these systems) will not be done in this thesis. For an example of a theoretical framework that establishes the passivity of closed-loop upper-limb rehabilitative robotic systems and allows rigorous stability analysis of human–robot interactions refer to [18] and its references.

1.1 Background and Related Works

In this section, background works in the paradigms of rehabilitative exercise machines, passivity-based control, human biomechanical characterization and repetitive control are surveyed. Theories relevant to the method covered in this thesis are also introduced and later elaborated on.

1.1.1 Robotic Exercise Machines

One of the main concepts of rehabilitative control systems is to only aid the user in tasks in which they exhibit weakness. This is known as the assist-as-needed approach [19]. One of the challenges in designing assist-as-needed devices is to determine the physical strength capacity of human musculoskeletal system without extensively conducting experiments to probe the patient's capability envelopes. Another difficulty is when adopting the assist-as-needed approach, it is often required that energy be injected into the system. However, typical exercise machines lack the ability of injecting energy into the system, but rather, create exercise resistance through means of damping only (this includes exercise bands, cable weight systems, etc.). Other damping only devices include simple rehabilitation devices such as a knee brace that can provide variable resistance and intelligent joint motion control using magnetic/ electro-rheological fluid dampers [20], [21]. However, the use of actuated manipulators for exercise allows for energy to be injected into the system and this enables power assistance to be provided to patients that require such assistance due to severe injury or other reasons for severely impaired mobility. In this thesis we utilize the BURT manipulator manufactured by Barrett Medical Inc. (see Section 2.1), which has this capability of assisting and resisting. See [22] for other research conducted on this manipulator. For examples of implementation of various other rehabilitative devices refer to [23] and it's references.

One possible goal of exercise machine controllers is to maximize the user's power expenditure. This maximum power objective is generally difficult because typically traditional exercise machines do not incorporate user specific information in the machine functionality. Exercise based on manual adjustments by the user are also affected by the psychological state of the user, which may result in decreased performance. To maximize the user's power output, recent research has focused on closed-loop actuated exercise equipment that incorporates feedback from the user. For example, in [24] a desired trajectory signal is designed to seek the optimal velocity setpoint that will maximize the user's power output, while the controller is designed to ensure that the exercise machine tracks the resulting desired trajectory. Similarly, an optimal velocity that maximizes the user's power expenditure is set based on knowledge of the user's force-velocity-position characteristics. However, such an objective also presents limitations (elaborated on in Section 4.5), requiring the patient to have sufficient mobility such that they can exert power throughout an exercise motion, and thus, limiting the target population to sufficiently mobile patients requiring strength training. More recent work on repetitive control (introduced in Subsection 1.1.4) also involves learning the user's physical abilities with respect to the task and adjusting the provided resistance or assistance online. In [25], parameters of a muscle synergy model were identified using an estimation scheme to obtain a measure of the physical strength capacity of the user, which was confirmed through comparison to collected EMG data. In [16], a self-optimizing technique is implemented to learn online position and velocity dependent strength capacity of a patient. However, injecting energy into the human-robot system means that there is potential to injure the user. Thus, an additional objective for assist-as-needed exercise machines is to maintain passivity (introduced in Subsection 1.1.2).

1.1.2 Passivity-Based Control and Passive Velocity Field Control

Passivity-based control (PBC), introduced in [26], is a well-established technique that has shown to be very powerful in designing robust controllers that render a system as passive with respect to a desired storage function. This applies to physical systems described by Euler–Lagrange (EL) equations of motion, which includes but is not limited to mechanical manipulators [27]. In recent literature we have seen an ever-increasing predominance of control techniques that respect and effectively exploit the structure of the system over classical techniques that try to impose some predetermined dynamic behavior [28]. For regulation problems of mechanical systems, which can be stabilized by "shaping" only the potential energy, PBC preserves the EL structure and furthermore assigns a closed-loop energy function equal to the difference between the energy of the system and the energy supplied by the controller. Thus, we say that stabilization is achieved via energy balancing or rather, “energy-shaping” [29]. This being said, most controllers in the literature are not passive. Thus, the energy of the system may increase even if no environment force is present.

In the case of mechanical manipulators, PBC exploits the passivity and linearity in the parameters of the rigid robot dynamics. This is unlike inverse dynamics control where the modified inner loop control does not achieve a linear decoupled system even when the parameters are exactly known. Apart from ensuring safety of the manipulator’s physical interaction with its environment, passivity theorem involves intuitive energy shaping (over nonlinearity cancellation and high gain). Thus, exploiting the structure of the system to provide physical interpretations to the control action involving performance, not just stability. For a detailed explanation of passivity-based control in terms of interconnection and damping assignment the reader is referred to [29], [30].

The passivity-based control scheme by Slotine and Li [31] makes the tracking error system passive but does not achieve passivity of the closed loop system with respect to the power formed by taking the product of the environmental force and the robot’s velocity (supply rate). In contrast, passive velocity field control (PVFC, introduced by Li and Horowitz [32]) ensures that the closed-loop system remains passive with respect to the robot's velocity and the environmental force. This also contrasts classical passivity-based control laws (such as [26], [33]), which cannot guarantee passivity when considering external forces as the input. PVFC is able to achieve passivity by adding a virtual flywheel to exchange the mechanical energy with the real robot to form an ‘augmented’ system (see Section 3.2). Passive Impedance control (PIC) [34] is another algorithm designed to keep the robot’s passivity as seen from the environment. It achieves this by switching a scaling parameter of the robot’s desired velocity so that the robot can remain passive even for the desired time-varying impedance center.

Although this thesis will focus on the original control algorithm, following its initial discovery many researchers have extended the PVFC formulation. The PVFC approach has been applied to a wide variety of problems such as cooperative multiple manipulator systems in [35], robotic rehabilitation tasks in [17] and biped robots in [36]. The original paper is followed by an implementation of PVFC for a two DOF manipulator where the inner joint velocity controller uses an observer for friction compensation (hierarchical velocity field control) [37]. The PVFC control algorithm was later extended in [38] to include shaping of the potential energy of the closed loop system dynamics as well as its kinetic energy. Another extension is the implementation of a ‘danger’ field [39], a safety measure that explicitly shapes the control law which represents a scalar quantity that is unique with respect to a generic point in the workspace and depends on the robot’s state. Saito and Luo also extended the PVFC for the robot to adapt to the uncertain geometrically constrained environment [40]. Newer applications include compensating for dynamic uncertainties to robotic exoskeleton control using a neural network velocity field controller [41]. Other leading research has focused on maintaining safe and stable operation of the teleoperator system through passivity concepts. In [42], Lee and Li extended these results to define a nonlinear decomposition which achieves passivity of the master and the slave robots by decomposing the closed-loop teleoperator system into two sub-systems. Similar work was presented in [43] concerning teleoperator time-delay. Further improvements of passive decomposition with respect to teleoperators can be seen in [44] and its references.

1.1.3 Model of Human Biomechanical Characteristics

Without extensively probing limit conditions, estimation of the physical strength capacity of human limbs is challenging due to several reasons:

1. Human musculoskeletal systems are complicated, and it is difficult (if not impossible) to identify each muscle’s physical envelopes without extensive experiments to probe these limits. This procedure often requires advanced algorithms and sensors (such as EMG’s and subcutaneous sensors).

2. Multiple muscles are involved in one physical activity with different contributions of each muscle.

3. The physical capacity envelopes of human limbs also depend on the limb pose and the estimation scheme must provide a systematic way to compute the envelopes at the poses of interest.

In this thesis, a simplification of the muscle model along with the repetitive nature of exercise allows us to obtain an approximate muscle model such that the forces generated by the user during exercise can be estimated. Once the physical capabilities of an individual user are known, they can be used to determine an optimal exercise strategy. However, as muscle models are complex, a simplification of the biomechanical model of the user makes it more feasible to learn. This simplification takes the form of a Hill like curve, which a fundamental principle of skeletal muscle physiology. Forces generated by contracting skeletal muscle have been modeled by many previous investigators since Hill's pioneering work conducted on isolated frog muscles [45] where he discovered that the slower a skeletal muscle shortens the greater the force it can generate during contraction and vice versa. Thus, the force-velocity relationship can be evaluated by measuring the force produced by muscles at different shortening or lengthening velocities, or the velocity at which muscles shorten or lengthen against different isotonic or auxotonic forces. Then a mathematical function is fitted (in Hill's case this was a hyperbolic function). Thus, the user's physical capabilities are assumed to satisfy a force-velocity relationship that is dependent on the user's configuration (pose of the arm). This assumption is used to form a 'muscle model' by which the force applied by the user on the manipulator system is characterized. We are now able to summarize the factors that affect the force a muscle exerts on the exercise machine:

- *Effort level:* This corresponds to how muscle is activated primarily under voluntary control. The level at which the user exerts him/herself will greatly affect the force they exert on the manipulator.
- *Position dependence:* As the configuration of the exercise machine changes, the user's joint positions also change. This results in a nonlinear dependance of the produced muscle force on the generalized position of the exercise motion. To visualize this, we can imagine that a forward pushing motion in line with our chest is more natural and is able to generate more power than when the motion is in line with our forehead.
- *Velocity dependence:* According to the force-velocity Hill relationship, the joint torque decreases with the shortening velocity of the muscle. A typical linear force-velocity relationship is shown in (Figure 1.1). Thus, a muscle will be able to generate a greater amount of force against a stationary constraint as opposed to when it is shortening (there is velocity in the direction of motion).
- *Fatigue:* During continual exercise, it is expected that there will be a decrease in the capability of the muscle to produce force. There is also a degree of mental fatigue due to maintaining focus for extended periods of time on a task.

Although a force-velocity-position model is used to characterize the user's interaction force, we only obtain a lumped measure of all the affects. This is because the kinematic or dynamic train of the human user is excluded. This includes the Inertial, Coriolis, gravitational and frictional effects of the arm as well as the moment arm (which also changes over the path traced by the user during an exercise motion). Furthermore, the Hill curve assumes constant muscle activation. This is clearly not the case if the user is performing an exercise where fatigue and effort level change. This being said, it is assumed that the user exhibited force-velocity characteristic are still 'Hill like'. In the rest of the thesis, when we refer to this 'Hill like' force-velocity model simply as a Hill curve.

There are many variations of the Hill muscle model that have been proposed. A short synopsis of these models is given here, but the reader is suggested to refer to [46] for more detail. A profile for the Hill curve that useful in pHRI due to its simplicity suggests a linear relationship between the force applicable by the skeletal muscle and the velocity of contraction of that muscle. This corresponds to the linear model (Figure 1.1) [47] while other force-velocity models include the hyperbolic model (Figure 1.2) [48], as well as the double-hyperbolic model (Figure 1.3) [49]. These varying fits have important implications for testing as well as the perception of contraction and adaptations of muscle function. The benefit in the scope of robotic rehabilitation includes bio-inspired actuators and control systems that exhibit natural (muscle-like) behavior in order to mimic human biomechanics to allow for human-robot force interactions. Examples of this can be seen in the development of exoskeletal systems [50], robotic joint design [51] as well as the design of bio-inspired actuators [52].

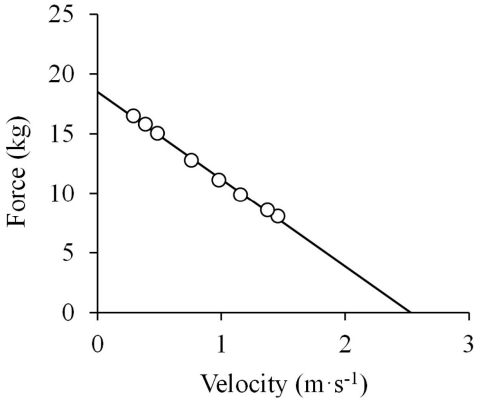

Figure 1.1: Example of force-velocity data obtained at a single position that suggests a linear Hill muscle model [46].

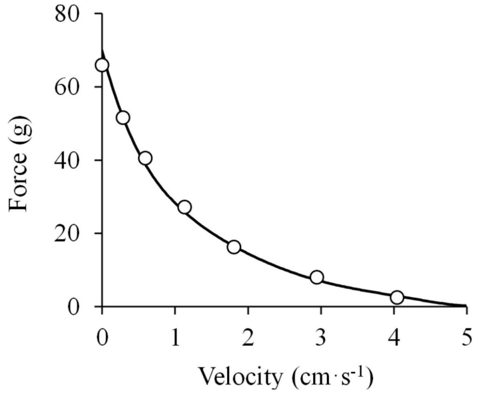

Figure 1.2: Example of force-velocity data obtained at a single position that suggests a hyperbolic Hill muscle model [46].

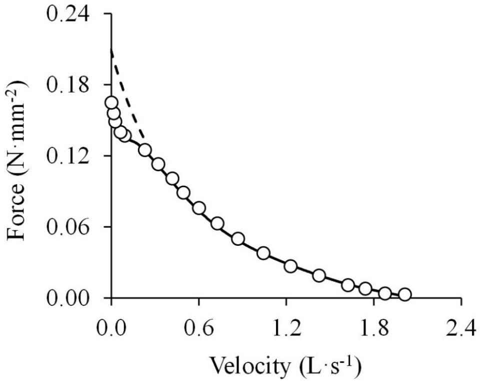

Figure 1.3: Example of force-velocity data obtained at a single position that suggests a double-hyperbolic Hill muscle model [46].

While a typical force-velocity relationship obtained from either in vitro muscles or isolated muscle groups can be described by a variety of equations, we are interested in exercise by robot rehabilitation systems that typically involve multi-joint upper limb motions (such as pulling or pushing). For such motions there exists evidence that the same relationship obtained from maximum-performance multi-joint movements could be approximately linear [53]. One such study involving maximal effort horizontal pulling at 3 different heights for seated subjects showed that force-velocity characteristics of such motions were approximately linear, and that the height of the pull had no significant effect on either static or dynamic performance [54], thus exhibiting the property of pose indifference and allowing for linear force-velocity curves to be fit to a variety of horizontal pulling motions. Other experiments involving maximal effort non-countermovement vertical jumps exhibit both hyperbolic and linear force-velocity relationships [55]. Consequently, this pattern also results in a relatively simple shape of the power-velocity relationship. Therefore, the linear force-velocity relationship together with the associated parabolic power-velocity relationship could provide both a new and simplified approach to studies of the design and function of human muscular system and its modeling, which is the model that is adopted in this thesis. As can be seen from Figure 1.4, the linear approximation is accurate for the range of velocities around maximal user power exertion even when the Hill model is hyperbolic, where μ in this case corresponds to a position along a curve or trajectory.

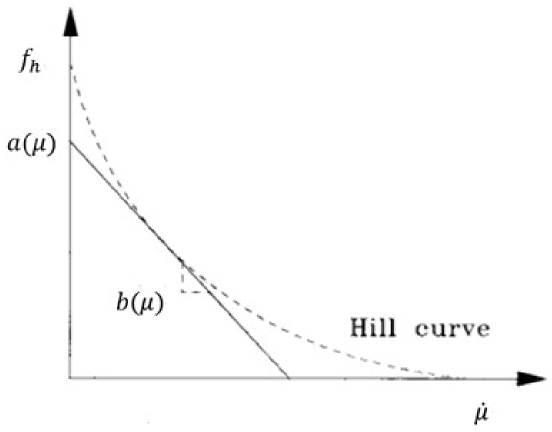

Figure 1.4: Illustrates how within a narrow range of velocities a hyperbolic Hill curve is approximately linear. This is true for all Hill muscle models within the region of maximum power generation. Figure adapted from [15].

$$f_h(\mu, \dot{\mu}) = a(\mu) - b(\mu)\dot{\mu} \tag{1.1}$$

$$P_{max} = \frac{a(0) \cdot b(0)}{4} \tag{1.2}$$

where $a(\mu) \in [\underline{a}(\mu), \infty)$ and $b(\mu) \in [\underline{b}(\mu), \overline{b}(\mu)]$, with $\underline{a}(x), \underline{b}(x), \overline{b}(x) > 0$. μ corresponds to some measure of position along a trajectory and represents different poses of the user's arm. f_h represents the force generated by the equivalent muscle and P_{max} refers to the region of maximum power generation. Defining a Hill curve at each pose along a trajectory results in a Hill surface. Although the simplified force-velocity relationship is linear, it varies non-linearly along each position in the trajectory. The resulting surface of the force-velocity relationship at each position along the desired trajectory would resemble Figure 1.5.

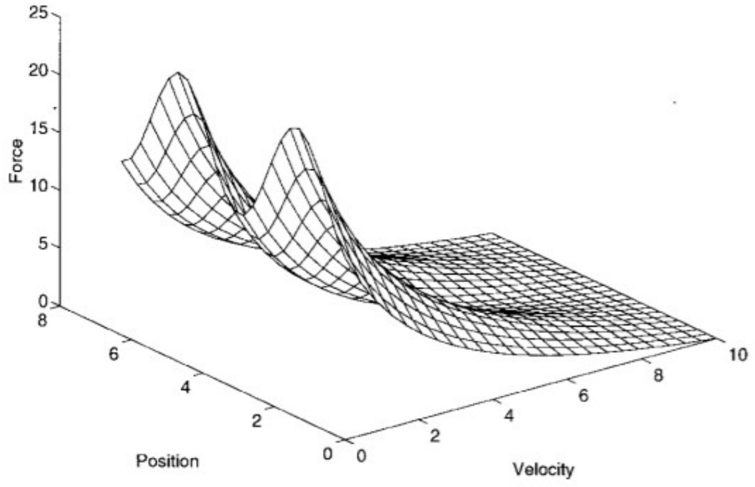

Figure 1.5: Sample Hill surface representing a hyperbolic Hill curve at each position along a trajectory. Notice that the Hill curves vary non-linearly along the trajectory defined by 'position' [15].

1.1.4 Repetitive Control and Learning Control

The nature of exercise motions considered within the scope of application of this thesis are repetitive. This motivates us to implement learning schemes in order to learn a model of the user's physical capabilities by viewing it as a repetitive external disturbance. The term repetitive control (synonymous to learning control) is used to describe algorithms that cancel errors that are periodic in time or position. This typically occurs when a system repeats a motion over and over (milling, thermal cycling, pick and place robots, etc.) or when a system is subject to a periodic disturbance (power systems, rotating machinery, etc.). Generally, when a robot task is periodic a learning or repetitive controller is shown to be able to enhance the performance of the system significantly [56]. Early application of repetitive control includes robot control, altitude stabilization of satellites, vibration compensation and more [57],

[58]. The reference [59] also provide a sufficient overview of repetitive control. Repetitive control was introduced for the first time to robotics in [56].

Repetitive control algorithms can be classified into two categories; internal and external model based repetitive control algorithms. Internal model based repetitive control states that if the exogeneous signal can be regarded as the output of an autonomous system, including the signal model in the closed loop system can assure complete rejection or perfect tracking of this signal. This is the most common approach to repetitive control (see [60], [61], [62]) where the input is designed such that it cancels the periodic disturbance with the goal of obtaining input and output sequences that converge to the reference/ desired input and output sequences. In one such example that aims to enhance the robustness of internal model repetitive control schemes, the repetitive update rule is modified to include the so-called Q-filter [63]. Unfortunately, the use of the Q-filter eliminates the ability of the tracking errors to converge to zero. Therefore, the trade-off between stability and tracking performance has been considered to be an important factor in the repetitive control system. On the other hand, external model repetitive control views the cancellation signal as being injected from outside the feedback loop and a parametric model of the plant is required. One such example that allows for selective cancellation of harmonics present in the disturbance is the basis function approach. In a majority of repetitive control scenarios, disturbances embody periodic functions that are defined by certain frequencies. Thus, a solution for repetitive disturbance rejection involves the online tuning of the gain for each harmonic of the periodic disturbance, where sinusoidal functions are most used as the basis functions [57]. Another external model based repetitive control algorithm is the ‘Learning’ algorithm that lacks the ability to selectively cancel harmonics but achieves short execution time and rapid disturbance cancellation. These are examples of ‘look up table’ methods [64]. This thesis will focus on the ‘Learning’ approach. A brief summary of internal and external model repetitive control above can be found in [57] and [58].

The ‘Learning’ approach takes the form of in integral transform that encodes the repetitive nature of the task. Messner and Horowitz have conducted extensive research on general classes of integral transforms [64] and derived stable adaptation algorithms for these controllers [65] for non-linear dynamic systems with unknown parametric uncertainties. This algorithm implies kernel functions in the update rule, specifically using Gaussian functions of time to approximate influence functions over a finite interval. However, difficulty lies in the practical application of their approach due to discretization of the integral representation. The reader is suggested to refer to [66] for further information on relevant schemes based on integral transforms. This scheme has been modified to fit a variety of repetitive disturbance cancellation tasks. For example, eliminating the need for joint velocity signals [67] or considering constraint surfaces in the workspace through hybrid force and position control [68].

The repetitive control algorithm is not enough on its own to ensure that unknown parameters of interest will converge to the real parameters. The requirement that ensures convergence is known as 'Persistence of Excitation' [69] which corresponds to an essential regressor signal vector property that is required for convergence of estimated parameters. This means that regressor signal needs to provide an amount of 'valuable information' such that beneficial parameters adaptations can be made. Often a reference generator can be used to modify the task such that direct influence over the regressor signal ensures that it is persistently exciting. This is the approach adopted in this thesis (introduced in Section 3.8) to ensure convergence of the unknown parameters. When a persistently exciting task is chosen and effectively executed by the manipulator, the learning controller is globally exponentially stable. In fact, all repetitive controllers are always exponentially stable [65].

1.2 Contributions and Organization

Given the current realm of algorithms outlined above, the contribution of this thesis is a novel application of the characterization of a user's biomechanics and subsequent control of an exercise motion defined by a velocity field. The controller also implements a novel solution to injecting dynamic damping. The procedure involves a user executing repetitive exercise motions while the self-optimizing control algorithm learns the user's force-velocity-position biomechanics such that it can inject sufficient damping to ensure constant power expenditure by the user. The end goal is to allow for safe robot-assisted exercise that requires no prior user specific knowledge (where only the robot model must be approximately known). The learning algorithm returns a Hill surface that allows for a better understanding of the user's disability and their improvement through exercise.

Chapter 2 presents the mathematical preliminaries, conventions, and notations used throughout the thesis, including a description of the BURT system. Chapter 3 considers the novel 3 DOF implementation of the PVFC and dynamic damping control architecture followed by validation through experimentation. Concluding remarks and further recommendations will be given in Chapter 4.

Chapter 2
Overview of Research Platform

The Barrett Upper-extremities Robotic Trainer, or BURT, is a rehabilitation and pHRI research robot designed by Barrett Technology as an off-the-shelf full 3D haptic device. Each BURT arm has three actuated rotational joints with position feedback from encoders, as well as a passive, unmeasured rotation joint at the end-effector to allow for wrist adduction and abduction. The BURT manipulator may be reconfigured, via quick-release screw mechanism, to the opposite handedness setup by flipping the device's forearm 180◦. The user is attached to the manipulator by a spherical handle at its end-effector. This section will discuss the kinematic and dynamic models of the BURT as a serial chain manipulator along with the kinematics and initialization of the 6DOF force-torque sensor mounted to the end-effector.

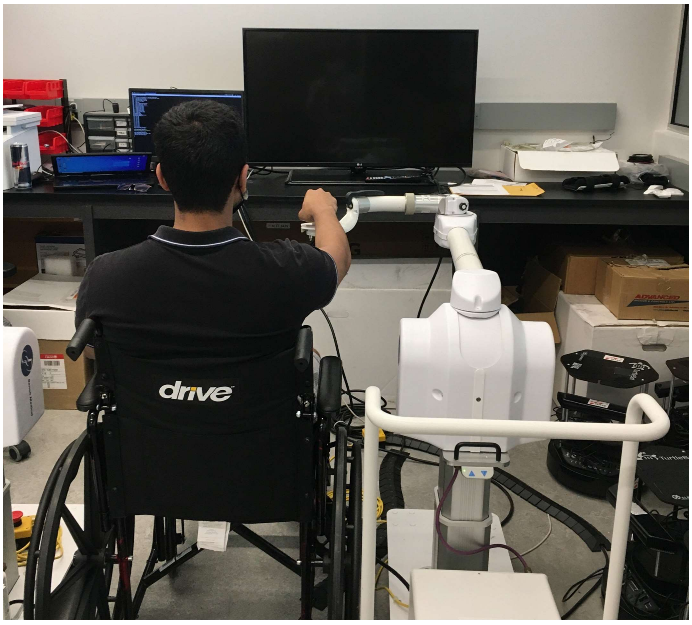

Figure 2.1: A user participating in an experiment on the Barrett Upper-extremities Robotic Trainer (BURT).

2.1 Kinematics

The well-known modified Denavit-Hartenberg (DH) convention can be used to specify coordinate frames along a kinematic chain. For a manipulator with three revolute joints (often denoted by RRR), such as the BURT, it is only required to specify three parameters for each link, i, known as DH parameters: link length, a_i, offset distance, d_i, and twist angle, α_i. The joint angle, q_i, is left as a variable which depends on robot pose. For a complete description of the coordinate frame assignment procedure, see [70]. Figure 2.2 shows how coordinate frames may be applied to the manipulator according to DH convention. Note that this pose is not actually attainable given the joint limits of the manipulator in either handedness but serves as a natural configuration for assigning $q = 0$. Positive values of q_i correspond with the right-hand rule about axis Z_i.

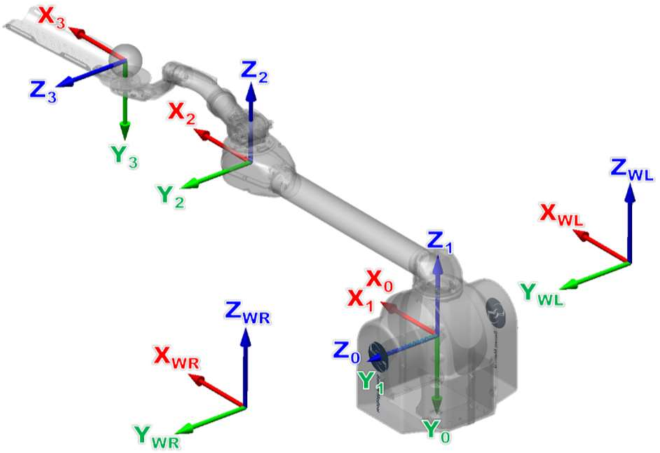

Figure 2.2: The coordinate frames assigned to the BURT system according the modified DH parameters. For a right handed setup, the world origin will be located to the right, respectively. In this configuration, $q_1 = q_2 = q_3 = 0$ (where q_i is the rotation of the joint about Z_i) [71].

The particular values for a, d, and α are provided in Table 2.1, as well as the manipulators joint limits. Note that the transformation described by the first row describes the rotation from world coordinates to the 0th coordinate frame. The 550mm translation (dependent on robot handedness) along the y axis is not included in this transformation. With this description of the robot's geometry, the forward kinematics problem may be solved. That is, given a particular manipulator configuration q in joint space, the position of the end-effector, x, in task-space can be determined. The forward kinematics problem always has a unique solution.

The opposite problem, to find the joint-space configuration q that achieves a given end-effector position, is known as the inverse kinematics problem. In general, the unique solution is not guaranteed and finding a closed-form analytic solution is not trivial for systems with larger DOF, n. For $n > 6$, a robot is said to have redundant DOF's, its end-effector may achieve any desired position and orientation in its workspace with infinitely many joint angles. Should x fall outside W, the robot's workspace, no solution to the inverse kinematics problem can be found. However, the joint limits eliminate all but one set of solutions to the BURT's inverse kinematics problem.

Table 2.1: Table of modified DH parameters and joint limits for the BURT robot

	D-H Parameters				**Joint Limits (rad)**	
Link (i)	a_i	α_i	d_i	q_i	$q_{\min i}$	$q_{\max i}$
0	0	$-\pi/2$	0	0	-	-
1	0	$\pi/2$	0	q_1	-1.01	+0.26
2	0.65	0	0.156	q_2	-0.96	+0.96
3	0.435	$-\pi/2$	0.069	q_3	-0.40	+2.84

The manipulator Jacobian is another useful kinematic tool that defines the dynamic relationship between the joint space and the task space (Cartesian space). This allows for joint velocities and joint torques to be converted to the end effector velocity and the end effectors force represented in Cartesian coordinates. Such a tool is required to allow for task space control of a manipulator with its dynamics defined in joint space. Formally, the Jacobian is a set of partial differential equations:

$$J = \frac{\partial x}{\partial q} \tag{2.1}$$

such that:

$$\dot{x} = J(q)\dot{q} \tag{2.2}$$

where $J \rightarrow R^{n\times n}$ is the linear Jacobian in the end effector frame, $\dot{x} \rightarrow R^3$ and $\dot{q} \rightarrow R^n$ represent the end effector velocity in Cartesian (task) space and the joint velocities respectively. Due to the virtual work theorem [27] the transpose of J allows for the transformation in the opposite direction when the manipulator is in static equilibrium such that:

$$F_x = J^{-T}\tau \tag{2.3}$$

F_x is the force vector applied to the end effector in task space and τ are joint torques.

2.2 Filtering Sensor Noise

The manipulator is equipped with encoders that return joint positions as well as approximated joint velocities. However, it is observed that although joint position measurements depict submillimeter accuracy, joint velocity measurements appear to include noise. Since the control input (see Sections 3.2, 3.7) utilizes the joint velocities and indirectly (through configuration dependent robot dynamics) the joint positions, reducing the high frequency noise in these signals result in greater stability and smoother operation.

To illustrate the magnitude and harmonics of the noise signals we can isolate the noise signals by recording and examining sensor output when the robot is not in motion. From this test it is observed that the noise of the joint positions is within the order of magnitude of 10^{-4} while the order of magnitude of the noise of the joint velocities (which is derived from the joint position information) is 10^{-3}. A discrete first order low pass filter (zero order hold) with a cutoff frequency of $10\ Hz$ was experimentally verified to minimize high frequency noise of the joint position and velocity signals without introducing significant delay. The discretized low pass filter is given by:

$$\mathcal{H}_{\dot{q}}(z) = \frac{0.06533}{z - 0.9347} \tag{2.4}$$

2.3 Dynamics

Let $q, \dot{q}, \ddot{q} \in R^3$ denote the joint positions, velocities and accelerations respectively. Then the dynamics of the manipulator system are given by the well-known formula:

$$M\big(q(t)\big)\ddot{q}(t) + C\big(q(t), \dot{q}(t)\big)\dot{q}(t) + B\dot{q}(t) + \tau_g\big(q(t)\big) = \tau(t) + \tau_e(t) \qquad (2.5)$$

where $M(q(t))\epsilon\Re^+$ is the generalized inertia of the three-bar linkage, $C(q(t), \dot{q}(t))$ represents the Coriolis and centripetal effects, B is the viscous damping coefficients acting at the joints, $\tau_g(q(t))$ are the torques due to gravity and $\tau(t)$ and $\tau_e(t)$ are the generalized torques generated by the motor and the environment (contains the user) respectively. Unmodeled dynamics include coulomb friction and unmodelled inertial effects.

For completeness, friction is included in the manipulator dynamics equation above, however, friction compensation is excluded in the formulation following in this thesis. For velocity field control with friction considered refer to [37]. Unfortunately, in [37] closed-loop passivity requirements are ignored. Also, gravity is included in the manipulator dynamics equation above, although gravity is excluded from the control formulation as it is compensated for by the robot firmware directly. Thus, in the formulation of the control and learning algorithm that follows gravity is excluded from the robot dynamics. Due to the fact that the system includes the feature of automatic gravity compensation and accurate calibration routines, details on the gravity compensation method will not be covered in this thesis. For information on the specific gravity compensation technique used refer to [72]. It follows that the robot dynamics equation considered in this thesis is:

$$M\big(q(t)\big)\ddot{q}(t) + C\big(q(t), \dot{q}(t)\big)\dot{q}(t) = \tau(t) + \tau_e(t) \qquad (2.6)$$

where all dissipative forces are treated as part of the external torques, $\tau_e(t)$. We shall assume that $M(q(t))$ is known, thus, $C(q(t), \dot{q}(t))$ is also known and is calculated by the Christoffel symbols [27]. $M(q(t))$ and $C(q(t), \dot{q}(t))$ are related by:

$$N\big(q(t), \dot{q}(t)\big) = \frac{d}{dq}M\big(q(t)\big) - 2C\big(q(t), \dot{q}(t)\big) \qquad (2.7)$$

where $N(q(t), \dot{q}(t))$ is a skew symmetric matrix. The fact that the time derivative of the matrix $M(q(t))$ and the matrix $C(q(t), \dot{q}(t))$ are related has been previously remarked by several authors including [73] [74] [27].

The Lagrangian method or the Newton-Euler method can be used to obtain a symbolic expression for the inertia matrix $M(q(t))$. This requires knowledge of the DH parameters and rigidbody parameters with a sufficient degree of accuracy (see Section 2.5). As the equation for the robot dynamics (Equation (2.6)) evaluated at $q, \dot{q}$ is required at every timestep, evaluating the inertia and Coriolis matrices as functions rather than calculating them in real time reduces the time to compute of the control loop. It is desirable that the dynamics model of the manipulator be computed quickly enough so that discretization effects do not degrade performance relative to the continuous-time, zero-delay ideal case.

2.4 System Architecture

The architecture of the BURT system consists of three subcomponents: the motherboard, a soft-realtime system running the user interface and user-specified control routines; the mainboard, a hard-realtime system responsible for low-level control (e.g. kinematics, gravity compensation, safety systems); and the robot itself, including the Barrett Puck motor drivers which provided closed-loop current control to faithfully actuate the commanded torque. Each of these subcomponents, shown in Figure 2.3, play an important role in the BURT's performance, but particular attention will be paid to the operation of the soft-realtime motherboard in this section.

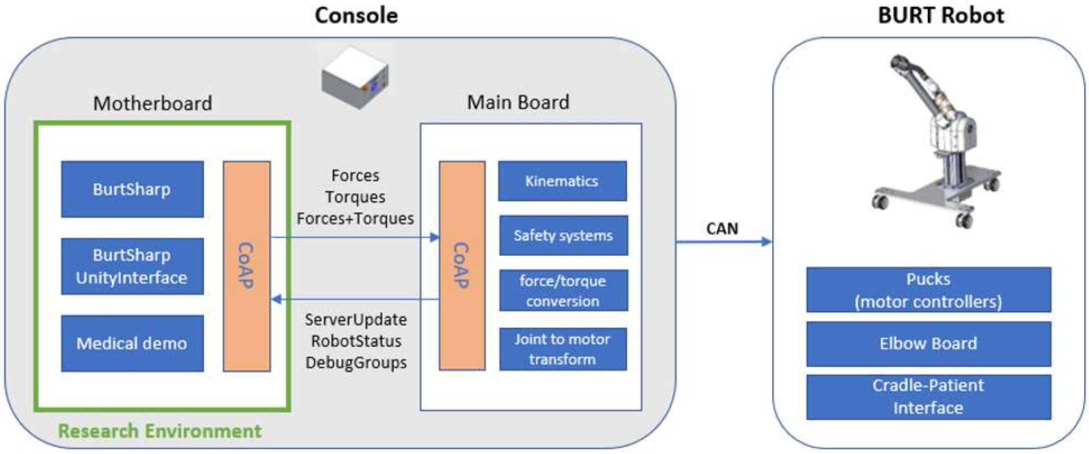

Figure 2.3: A block diagram representation of the BURT's system architecture [71].

The motherboard, which serves as the user interface for the robot, runs the Xubuntu operating system to support control development in C#/Mono with the BurtSharp library. It also allows for tight integration with the Unity game engine with the BurtSharpUnity library, which can be used to facilitate experimental development and quickly create virtual haptic environments. This integration with Unity is not used in this experiment, and the experiment takes the form of a console application. Since this is not a hard-realtime system, control inputs cannot be sent to the BURT with any guaranteed timing. Additionally, the hard-realtime mainboard requests control updates from the motherboard at a fixed rate of $500Hz$. The motherboard responds after some delay, determined by competing processes on the Xubuntu, issuing a new control command on average every $2.15ms$ ($\approx 465Hz$). This rate also determines the maximum data logging rate achievable with the system. The length of the delay between control cycle times is measured by the system and may be retrieved on the subsequent interaction of the control loop.

Occasionally, the time between responses from the motherboard may increase to $5.00ms$ (more than double the average control timestep), however, this is negligible with respect to ensuring smooth operation of the robot arm. Figure 2.4 shows the time between torque commands for the experiment.

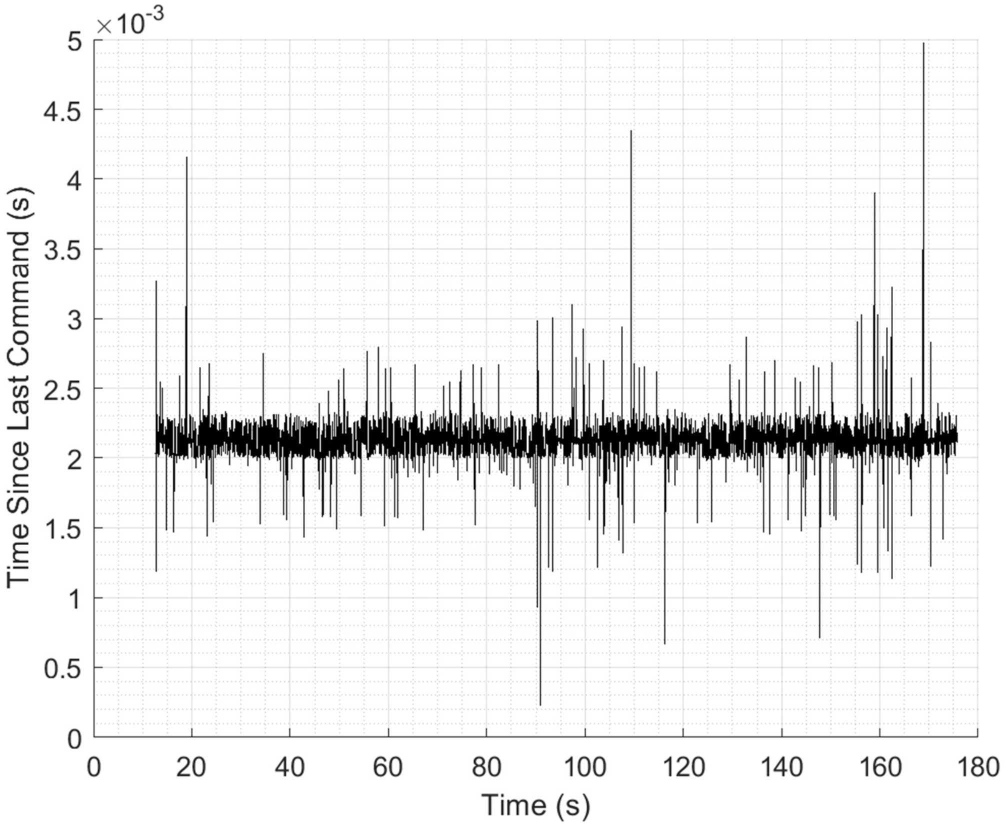

Figure 2.4: Typical delay between control cycle times during a $160s$ experimental trial while running the smart exercise controller presented in Chapter 3.

2.5 Rigid Body Parameters

Knowledge of the robot dynamics must be approximately available to implement any form of inverse dynamics based control. As discussed previously, since gravity and friction are ignored and the Coriolis matrix $C(q(t), \dot{q}(t))$ can be calculated from the Mass matrix $M(q(t))$ it is only required that we know the Mass matrix. This matrix can be symbolically generated if the DH parameters and the inertial parameters are known with a degree of accuracy. DH parameters are almost always provided with off the shelf manipulators or can be physically measured. Inertial parameters can often also be obtained from the manufacturer and it has also been shown that adaptive algorithms [75], [76], as well as direct measurement methods can be used to identify these parameters. For an example of generalized modelling and identification applied to a SCARA manipulator refer to [77]. Learning of rigidbody parameters is out of the scope of this thesis as they are manufacturer provided with a sufficient degree of accuracy.

Each of the three links of the manipulator arm have 10 constant parameters that are used to define that links dynamics. These parameters are the links mass (1 parameter), the links center of mass (3 parameters) and the links inertia tensor about its center of mass (6 parameters). These parameters are defined in Appendix A where m_i is the mass of each link, the center of mass of each link is measured from the joint axis in terms of displacement (in meters) and the inertia tensor is with respect to the center of mass of the link.

2.6 Drive Dynamics

Apart from the dynamics due to the rigidbody model of the robot manipulator, the dynamics due to the drive system is also considered to provide for a more accurate final dynamics model. As the drive system is not a direct drive system (where motors are mounted at the base of each link and drive that link with no transmission/ directly) but rather a cable driven system, the motor axis to joint axis (and vice versa) must be taken into account when considering motor inertias in the joint referenced dynamics model. The joint axis to motor axis transformation ($T_{q \to q_{motor}}$) allows for joint velocities, accelerations, and torques to be converted to motor velocities, accelerations, and torques respectively:

$$\dot{q} = T_{q_{motor} \to q}{}^{-1} \cdot \dot{q}_{motor}, \qquad \dot{q}_{motor} = T_{q \to q_{motor}} \cdot \dot{q} \tag{2.8}$$

The transformations for joint/motor accelerations as well as joint/motor torques is the same as the transformations to and from for joint/motor velocities. The joint to motor axis transformation as well as the drive dynamics (manufacturer provided) are defined in Appendix A. Note that although they are known, viscous motor friction affects are ignored in the dynamic model of the robot (just as joint coulomb and viscous friction is ignored) while motor inertial effects are included in the inertia matrix, $M(q(t))$.

Chapter 3
Smart Exercise Machine Controller

This section explores a novel experiment with the goal of learning a user's Hill surface along a desired trajectory defined by a reference velocity field that results in a repetitive exercise motion. Following this, an optimal reference velocity is commanded and damping is injected such that the user is made to execute repetitive exercise motions with maximal power output. Within the following section, an overview of the mathematical preliminaries concerning the self-optimizing smart exercise machine controller are also introduced.

3.1 Design of the Exercise Velocity Field

The existence of a velocity field encoding the task is a crucial part of PVFC. In this thesis, the velocity field is pre-defined and pre-calculated. Online generation of velocity fields is out of the scope of this thesis, and the readers are suggested to refer to [78], [79].

A joint space velocity field $V_q(q)$ is a map that assigns to each point $q = (q_1, q_2, q_3) \in R^3$ a vector forming the tangent space $\{(q, \dot{q})\}$. We consider repetitive exercise motions that are specified by a velocity field $V_x(x)$, that is designed prior to experimentation and defined in 3D space $x = (x_1, x_2, x_3) \in \mathbb{R}^3$. In our given manipulation task, the velocity field represents a repetitive counter-clockwise circle tracing motion. Thus, it is designed such that it converges to a circular trajectory where the position along the trajectory and velocity tangential to it are given by x_d and $\dot{x}_d$ respectively. The circular trajectory traces a path in Cartesian space defined by x_d. $x_d(x)$ refers to the closest point on the circular path from the position x. Similarly, $\dot{x}_d(x, \dot{x})$ refers to the velocity component of $\dot{x}$ that is tangent to x_d and is given by the vector projection:

$$\dot{x}_d(x, \dot{x}) = \frac{\dot{x} \cdot V_x(x_d(x))}{||V_x(x_d(x))||^2} V_x(x_d(x)) \qquad (3.1)$$

We define an arbitrary position on the path, x_{d_0}, that corresponds to the origin or 'starting position' of the path. We also define a scalar value, L, that represents the length of the path characterized by x_d.

For a circular path, L represents the circumference. Now we are in a position to define a scalar variable, $\mu\big(x_d(x)\big) = \mu(x) \in [0, L)$, that parameterizes the circular path such that $\mu(x)$ represents the length of the line segment along the path from x_{d_0} (where $\mu\big(x_{d_0}\big) = 0$) to $x_d(x)$. This is measured with respect to the counter-clockwise direction such that $\mu: 0 \to L$ as the path is traced according to the velocity field V_x. $\dot{\mu}$ is the travel speed along the circular path and is given by the scalar projection:

$$\dot{\mu}(x, \dot{x}) = \frac{\dot{x} \cdot V_x\big(x_d(x)\big)}{\big\|V_x\big(x_d(x)\big)\big\|} = \|\dot{x}_d(x, \dot{x})\| \tag{3.2}$$

where $\big\|V_x\big(x_d(x)\big)\big\|$ corresponds to the Euclidean norm of $V_x\big(x_d(x)\big)$. We can evaluate the desired speed at μ (V_μ) which is given by:

$$V_\mu(x) = \big\|V_x\big(x_d(x)\big)\big\| \tag{3.3}$$

Figure 3.1 shows a diagram outlining the relationship between the terms $x_{d_0}, x_d(x)$, $\dot{x}_d(x, \dot{x})$, $\mu(x)$, $\dot{\mu}(x, \dot{x})$ and a measure of the shortest distance from x to the path, $\|x - x_d(x)\|$.

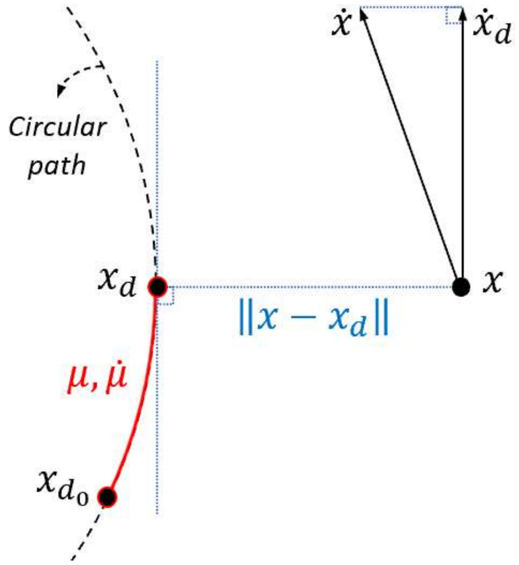

Figure 3.1: Diagram illustrating the scalar and vector values related to the parameterization of the desired contour.

In the remainder of the thesis the terms desired contour and desired path (used interchangeably) are given by the circular path traced by x_d and parameterized by μ. The term desired trajectory corresponds to the circle tracing task of achieving x_d, $V(x_d)$. Using forward kinematics and Equation (2.2), $x_d(q)$ and $\dot{x}_d(q, \dot{q})$ can be solved for. In the remainder of this thesis, x_d and $\dot{x}_d$ will correspond to $x_d(x)$ and $\dot{x}_d(x, \dot{x})$ respectively.

Figure 3.2 shows a velocity field for tracing a circle in Cartesian space, with vectors evaluated in the region around the desired contour. The timing of each point of the trajectory is not important when we consider the coordination and synchronization of an exercise task, thus, the velocity field is more suitable to specify the task than timed trajectories. This avoids the discrepancy between a timed trajectory task and a contour following task that has been attributed by [80] and others.

The design of the velocity field is specific to each user and should consider the range of motion of the user as well as the physical condition that they are in (for example, newly injured or recovering). An example is limiting large vertical exercise motion when the user has a severe shoulder injury. In such a case, the user should not overexert themselves to prevent further injury.

We will now define the velocity field task, ${V_x}^0$, which will be refer to as the exercise velocity field. ${V_x}^0(x)$ is designed such that it results in the counterclockwise tracing of a circle in the xy plane. ${V_x}^0$ is normalized, thus defines only the direction of the desired motion. The traced circle is defined by a center $({x_1}_c, {x_2}_c, {x_3}_c)$ and radius r_c. We propose the field as a summation of three separate velocity fields; one that defines concentric motion in the xy plane (T), one that produces normal motion within the xy plane towards a desired circular contour (N) and finally one that converges towards the z plane containing the circular contour (Z). The field is finally normalized such that it has the same magnitude throughout. The field is given by:

$$ {V_x}^0(x) = {V_x}^T + {V_x}^N + {V_x}^Z / \| {V_x}^T + {V_x}^N + {V_x}^Z \| \qquad (3.4) $$

such that $\| {V_x}^0(x) \| = 1$ for all x. Each component of the exercise velocity field is given by:

$$ {V_x}^T = \begin{bmatrix} {x_2}_c - {x_2}_p \\ {x_1}_p - {x_1}_c \\ 0 \end{bmatrix} \qquad (3.5) $$

$$V_x{}^N = \begin{bmatrix} \sigma_N \left(x_{1_c} - x_{1_p}\right)\left(\sqrt{\left(x_{1_p} - x_{1_c}\right)^2 + \left(x_{2_p} - x_{2_c}\right)^2} - r_c\right) \\ \sigma_N \left(x_{2_c} - x_{2_p}\right)\left(\sqrt{\left(x_{1_p} - x_{1_c}\right)^2 + \left(x_{2_p} - x_{2_c}\right)^2} - r_c\right) \\ 0 \end{bmatrix} \tag{3.6}$$

$$V_x{}^Z = \begin{bmatrix} 0 \\ 0 \\ \sigma_Z \left(x_{3_c} - x_{3_p}\right) \end{bmatrix} \tag{3.7}$$

where $(x_{1_p}, x_{2_p}, x_{3_p})$ corresponds to the end effector's Cartesian position. $x_{1_c} = 0.65\ m, x_{2_c} = -0.15\ m$ and $x_{3_c} = 0.3$ is taken to be the center of a circle with radius $r_c = 0.2\ m$ (Figure 3.2). σ_N and σ_Z are convergence gain factors.

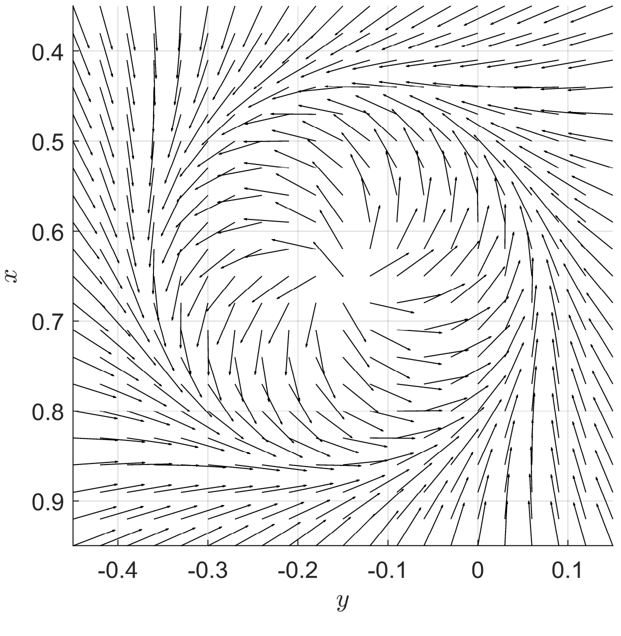

Figure 3.2: The normalized exercise velocity field in task space that results in the count-clockwise tracing of a circle in the xy plane with $x_{1_c} = 0.65\ m, x_{2_c} = -0.15\ m$ and $x_{3_c} = 0.3$ and $r_c = 0.2\ m$.

The convergence gain factors are $\sigma_N = 12$ and $\sigma_Z = 0.1$. The field is evaluated within the BURT manipulator workspace viewed from 'above' (top view of xy plane).

For a desired velocity field given by $V_x(x)$, the forward kinematics equations and the end effector's Cartesian position x can be represented in terms of joint positions, q, resulting in $V_x(q)$. The velocity field is then converted to joint space resulting in $V_q(q)$. This is done by multiplying the resulting field by the inverse of the Jacobian matrix. This results in the desired velocity field in joint space being:

$$V_q(q) = J^{-1}\, V_x(q) \tag{3.8}$$

In the rest of this thesis, V will refer to the desired joint velocities unless specified to be otherwise via a subscript i.e. $V = V_q$.

Any velocity field which results in a repetitive motion is a valid reference. This 'repetitiveness' of the resulting motion is necessary for the learning algorithm that will be introduced in Section 3.5. Some variations to the circular velocity field apart from its radius and location in the workspace are the convergence rate and the convergence profile. An increased convergence rate (increasing the values of σ_N and σ_Z) will result in faster convergence to the desired contour μ, but may cause overshooting. This problem of overshooting can be addressed by altering the convergence profile. For example, a cubic convergence profile (this would be achieved for convergence in the z plane by substituting $\left(x_{3_c} - x_{3_p}\right)$ for $\left(x_{3_c} - x_{3_p}\right)^3$ in Equation (3.7)) will achieve faster convergence to the region of μ while minimizing overshoot. However, since the PVFC control algorithm requires the covariant derivative of the velocity field (covered in Section 3.2), increasing the complexity of the velocity field will result in an increased computation time between control loop commands. This is especially true if the covariant derivative is computed in real time.

The velocity field in this thesis does not account for any consideration of environmental dynamics or constraints. For adaptive PVFC that learns the unconstrained desired velocity field due to the environment's dynamics (which is often initially unknown), please refer to [40].

3.2 Passive Velocity Field Control

It is important to ensure safe operation of an exercise machine which is operating under closed loop control. One way to promote the safety of a controlled exercise machine is to design the controller such that the machine behaves as a passive device with respect to the user. The primary motivation behind passivity-based control (PBC) schemes is that they result in error equations where the regressor (in an adaptive control scenario) is independent of the joint accelerations [26].

For a manipulator system, in the absence of potential energy, the storage function is the kinetic energy of the system. This is given by:

$$k(q,\dot{q}) = \frac{1}{2}\dot{q}^{T}M(q)\dot{q} \tag{3.9}$$

When the system is considered as an input output system with the environmentally applied torques (τ_e) being the input and the resulting joint velocities ($\dot{q}$) being the output (refer to Figure 3.3), the exercise machine is said to be passive with respect to the supply rate $S = \tau_e \cdot \dot{q}$ if:

$$\int_0^t \tau_e(\zeta) \cdot \dot{q}(\zeta) d\zeta \geq -\beta^2 \tag{3.10}$$

for all $t \geq 0$, where the supply rate is the power generated by the user except for an amount of initial energy β^2. The control goal of PVFC is to track the desired velocity field, V. V corresponds to the joint space velocity field generated by the reference generator (introduced in Section 3.8). The tracking of V is achieved by injecting the appropriate control input torque (τ) such that a scaled multiple of the desired velocity field is exactly tracked:

$$\dot{q} = \alpha V(q) \tag{3.11}$$

such that the velocity field tracking error given by:

$$e_\alpha = \dot{q} - \alpha V(q) \tag{3.12}$$

satisfies $\lim_{t \to \infty} e_\alpha = 0$. If we consider a situation where an appropriate control input (τ) is applied to the system such that a multiple of $V(q)$ is tracked (Equation (3.11)), even if there are no external torques applied ($\tau_e = 0$) the kinetic energy of the system may increase. Thus, the passivity relation will not generally be satisfied by the supply rate given by $S(\tau_e, \dot{q}\,) = \tau_e \cdot \dot{q}$ and the storage function being the kinetic energy. To overcome this difficulty, the manipulator system is extended to include the dynamics of a fictitious flywheel. Energy is exchanged between the real system and the fictitious flywheel such that interactions between the closed-loop system and its physical environment are rendered passive. This extended mechanical system is referred to as the augmented system.

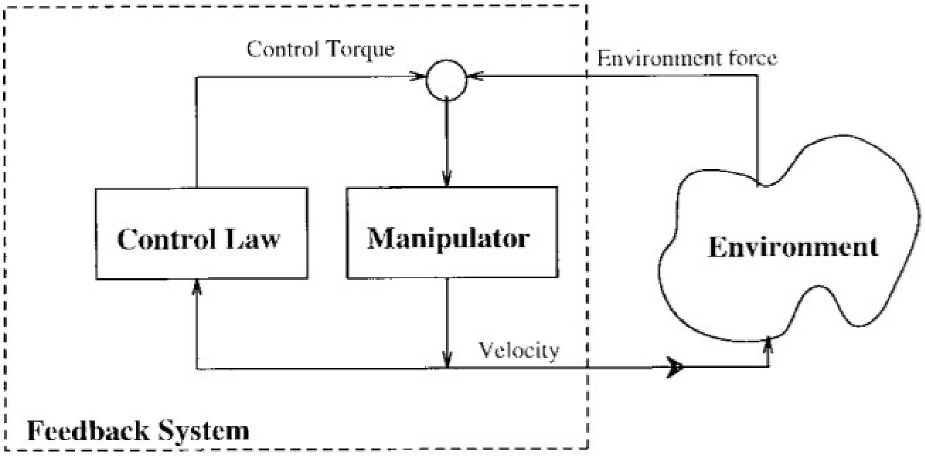

Figure 3.3: Block diagram of the interaction of the robot manipulator with the environment (contains the user) and the PVFC controller [32].

The configuration of the augmented mechanical system in the case of a n degree of freedom manipulator is denoted by:

$$\bar{q} = \begin{bmatrix} q^T & q_f \end{bmatrix}^T \tag{3.13}$$

where $q \in R^n$ and $\bar{q} \in R^{n+1}$. q_f is the angle of the fictitious flywheel which acts to store energy. The dynamics of the flywheel is given by:

$$m_f \dot{q}_f = \tau_f \tag{3.14}$$

where $m_f$is the mass of the fictitious inertia and τ_f is the coupling control input to the flywheel which is defined later (Equation (3.27)). Thus, the augmented system's dynamics can be written as:

$$\bar{M}(\bar{q})\ddot{\bar{q}} + \bar{C}(\bar{q},\dot{\bar{q}})\dot{\bar{q}} = \bar{\tau}_{PVFC} + \bar{\tau}_{D} + \bar{\tau}_{e} \tag{3.15}$$

where $\bar{q}$, $\dot{\bar{q}}$ and $\ddot{\bar{q}}$ are the configuration, velocity and acceleration of the augmented system respectively. The PVFC control input is given by $\bar{\tau}_{PVFC} = \left[{\tau_{PVFC}}^{T} \quad {\tau_{PVFC}}_{f}\right]^{T} \in R^{n+1}$ and the augmented environmental force is given by $\bar{\tau}_{e} = \left[{\tau_{e}}^{T} \quad 0\right]^{T} \in R^{n+1}$. $\bar{\tau}_{D}$ refers to injected damping that is introduced in Section 3.7. The Inertia and Coriolis matrices of the augmented system are:

$$\bar{M}(\bar{q}) = \begin{bmatrix} M(q) & 0 \\ 0 & m_f \end{bmatrix} \tag{3.16}$$

$$\bar{C}(\bar{q},\dot{\bar{q}}) = \begin{bmatrix} C(q,\dot{q}) & 0 \\ 0 & 0 \end{bmatrix} \tag{3.17}$$

The kinetic energy of the augmented system is given by:

$$\bar{k}(\bar{q},\dot{\bar{q}}) = \frac{1}{2}\dot{\bar{q}}^{T}\bar{M}(\bar{q})\dot{\bar{q}} \tag{3.18}$$

This modification of the manipulator system to an augmented mechanical system requires the formation of an augmented desired velocity field, $\bar{V}$, defined in joint space. This means that a desired flywheel velocity (V_f) must be defined in addition to the desired joint velocities of the manipulator system (V). To aid the following formulations, we define the value of the augmented system's total kinetic energy evaluated at the desired velocity field to be a constant, E_{max}. This value is assigned prior to the implementation.

$$\bar{k}\big(\bar{q},\bar{V}(\bar{q})\big) = \frac{1}{2}\,\bar{V}(\bar{q})^{T}\bar{M}(\bar{q})\bar{V}(\bar{q}) = E_{max} > 0 \tag{3.19}$$

where $\bar{k}\big(\bar{q},\bar{V}(\bar{q})\big)$ refers to the kinetic energy of the augmented system at the desired augmented velocity given by:

$$\bar{V}(\bar{q}) = \left[V(q)^{T} \quad V_f(q)\right]^{T} \tag{3.20}$$

Once E_{max} is specified, the desired velocity field for the fictitious inertia can be determined:

$$V_f(q) = \sqrt{\frac{2}{m_f}\left(E_{max} - \frac{1}{2}\, V(q)^T M(q) V(q)\right)} \tag{3.21}$$

where E_{max} is chosen to be large enough such that $V_f(q)$ has a real solution. Assuming that $\tau_e = 0$, the scaled multiple of the desired velocity field that is tracked is given by:

$$\alpha = \sqrt{\frac{\bar{k}(\bar{q}, \dot{\bar{q}})}{E_{max}}} \tag{3.22}$$

For proof refer to [32]. This means that when there is no interaction with the environment, the arm moves along the integral curves of a pre-defined velocity field with a speed proportional to the energy of the augmented system.

The value assigned to E_{max} affects the robustness of the controller as well as its response to environmental forces. On one hand, a low value of E_{max} allows for quick adaptation to new reference velocities as proportionally less energy must be added or taken away from the system to result in an increase or decrease in the tracked multiple of the desired velocity field (α). However, minimizing E_{max} negatively impacts the robustness with respect to the rate of energy dissipation and disturbances [81]. The suggested procedure to choose an appropriate E_{max} value is to allow the user to follow the desired velocity field in order to find the largest value of the manipulator's kinetic energy at which the user is still comfortable. One way of choosing the value of E_{max} for the first 'run' is to evaluate the manipulators kinetic energy at the joint velocity limits. Such a value will be sufficiently large if the manipulator is safely operating within these limits. Following this, the controller is implemented on the robot system and the value of E_{max} is lowered until the desired behavior is obtained.

The coupling control law consists of an inverse dynamics term and a feedback term that allow the velocity of the augmented system ($\dot{\bar{q}}$) to converge to a multiple of the desired velocity field ($\alpha\bar{V}(\bar{q})$). To easily present the control law, we will define the momentum $\bar{p}$, desired momentum $\bar{P}$, as well as the Levi-Civita connection associated with the desired momentum $\bar{w}$ below:

$$\bar{p}(\bar{q}, \dot{\bar{q}}) = \bar{M}(\bar{q})\dot{\bar{q}} \tag{3.23}$$

$$\bar{P}(\bar{q}) = \bar{M}(\bar{q})\bar{V}(\bar{q}) \tag{3.24}$$

$$\bar{w}(\bar{q}, \dot{\bar{q}}) = \bar{M}(\bar{q})\dot{\bar{V}}(\bar{q}) + \bar{C}(\bar{q}, \dot{\bar{q}})\bar{V}(\bar{q}) \tag{3.25}$$

The term $\bar{w}(\bar{q}, \dot{\bar{q}})$ is responsible for generating the inverse dynamics necessary to track the desired velocity field. The derivative of the desired velocity field can be calculated by:

$$\dot{\bar{V}}_i(\bar{q}) = \sum_{k=1}^{n+1} \frac{\partial \bar{V}_i(\bar{q})}{\partial \bar{q}_k} \dot{\bar{q}}_k \tag{3.26}$$

where $\dot{\bar{V}}_i(\bar{q})$ is symbolically computed prior to the experimentation and evaluated in real time. This approach becomes beneficial to reduce computational complexity, because computing the above derivative in real-time would not be possible with the firmware's processing capabilities. The final coupling control law is given by:

$$\bar{\tau}_{PVFC}(\bar{q}, \dot{\bar{q}}) = \bar{\tau}_c(\bar{q}, \dot{\bar{q}}) + \bar{\tau}_F(\bar{q}, \dot{\bar{q}}) \tag{3.27}$$

$$\bar{\tau}_c = \frac{1}{2\,E_{max}}\, \aleph_c\, \dot{\bar{q}} = \frac{1}{2\,E_{max}}\, (\bar{w}\,\bar{P}^T - \bar{P}\,\bar{w}^T)\, \dot{\bar{q}} \tag{3.28}$$

$$\bar{\tau}_F = \gamma\, \aleph_f\, \dot{\bar{q}} = \gamma\, (\bar{P}\,\bar{p}^T - \bar{p}\,\bar{P}^T)\, \dot{\bar{q}} \tag{3.29}$$

where $\bar{\tau}_c$ and $\bar{\tau}_F$ are the inverse dynamics and stabilizing feedback terms respectively. γ is a control gain factor that determines the convergence rate to the desired velocity field. It can be noted that the $\aleph_c$ and $\aleph_f$ matrices are skew symmetric [32]. It is this property that is the mechanism by which the kinetic energy of the augmented system is maintained at a constant value in the absence of environmental torques. Note that for this to be true it is important that $C(q(t), \dot{q}(t))$ is chosen such that $\dot{M}\big(q(t)\big) - 2C(q(t), \dot{q}(t))$ is skew symmetric [26]. For further details on relevant geometry, refer to [82]. The original generalized representation of the above formulation for a n degree of freedom mechanical manipulator can be found in [32]. Similarly, the full geometrical implications and derivation of the PVFC control solution can be found in [81].

Thus, we can conclude that PVFC realizes two main objectives:

1. The control torques take the form of skew symmetric matrices. Thus, the augmented system maintains passivity with respect to the power supplied by the environment.

2. The robot's actual velocity approaches a scaled multiple (α) of the desired velocity field no environmental forces are present. This multiple is determined by the amount of energy in the system.

However, this approach has the following problems (briefly mentioned in [34]):

1. The formulation above does not consider the uncertainties of the environmental geometric constraints.

2. It requires precise knowledge of the robot model to realize the nonlinear feedback control torque.

3. Although it maintains passivity, the contact task performance is guided and cannot be adjusted with respect to the environment dynamics intuitively due to high nonlinearity and complexity of its structure.

These limitations are addressed in Section 4.5.

3.3 Linear Parameterization of Unknown Functions

We wish to characterize the user force along the desired contour (parameterized by μ) to which the desired velocity field converges to. It is along this contour that we learn a model of the user's force-velocity characteristics that takes the form of a linear Hill curve at each value of μ i.e. a Hill 'surface'. Specifically, the Hill force applied by the user is given as:

$$f_h(\mu, \dot{\mu}(t)) = a(\mu) - b(\mu)\dot{\mu}(t) = \begin{bmatrix} 1 & -\dot{\mu}(t) \end{bmatrix} \begin{bmatrix} a(\mu) \\ b(\mu) \end{bmatrix} = \phi^T(t)\theta(\mu) \tag{3.30}$$

where $\theta(\mu) = [a(\mu) \quad b(\mu)]^T$ is the unknown parameter vector that lies in a convex set such that $a(\mu) \in [\underline{a}(\mu), \infty)$, $b(\mu) \in [\underline{b}(\mu), \overline{b}(\mu)]$ and $\phi^T(t)$ is a known regressor signal. Likewise, the known desired regressor signal defined at μ, $V_\mu(\mu(t))$ is given by:

$$\Phi^T(t) = [1 \quad -V_\mu(\mu(t))] \tag{3.31}$$

where $V_\mu(\mu(t))$ is the desired speed along μ. We assume that the user applied force when tracking the desired trajectory is periodic with respect to the position along the trajectory:

$$f_h(\mu, \dot{\mu}\,) = f_h(\mu + L, \dot{\mu}) \tag{3.32}$$

We require a discrete parameterization suitable for real time implementation on digital computers. This not only reduces the theoretical complexity, but a discrete time algorithm naturally limits the highest frequency components of the disturbance [83]. The discrete learning algorithm presented here can be shown to be equivalent to the form presented in [84]. It can also be found summarized in [85] where it is referred to as the 'Learning Algorithm' and can be shown to be equivalent to the continuous rule originally proposed in [64]. When implemented in the discretized form, the theoretical complexity of the original continuous rule is greatly reduced. To estimate the nonlinear functions $a(\mu)$ and $b(\mu)$, they are linearly parameterized:

$$\theta(\mu) = \begin{bmatrix} a(\mu) \\ b(\mu) \end{bmatrix} = h(\mu) \begin{bmatrix} c_a \\ c_b \end{bmatrix} \tag{3.33}$$

where $h(\mu) \in G$ is a vector with shape $[1 \times N]$. $c = [c_a \quad c_b]^T$ is a vector of unknown influence functions where $c_a, c_b \in G$ with shape $[N \times 1]$. $a(\mu), b(\mu)$ are scalar values of the unknown parameters at μ. G represents the configuration space along μ. We define $h(\mu)$ to be a discrete Gaussian function approximation with truncated tails which is approximated by ν segments. $\nu \leq N$ corresponds to the window length of the discrete Gaussian approximation such that the values of $h(\mu)$ that lie outside of this window length are zero. This means that ν and N can be chosen such that only a few entries of $h(\mu)$ need to be nonzero. We define that $h(\mu)$ has the same repetitive property as f_h (as in Equation (3.32)):

$$h(\mu) = h(\mu + L) \tag{3.34}$$

This is the mechanism that enables the learning of the unknown parameters along μ. As $\mu \rightarrow \mu + L$, the values of $h(\mu)$ are 'shifted'. Thus, $h(\mu)$ acts as a 'mask' over the influence functions, allowing for the generation of parameter estimates $a(\mu)$ and $b(\mu)$ along μ. The unknown parameter vectors a and b are expressed as:

$$a = Hc_a, b = Hc_b \tag{3.35}$$

where $H \in G \times G$ with shape $[N \times N]$. $a, b \in G$ are of the shape $[N \times 1]$ and contain N values of the unknown parameters defined along μ. H is symmetric and positive definite and is formed by stacking the vectors from $h(\mu) \rightarrow h(\mu + L)$. In fact, it can be shown that H must be positive definite to ensure asymptotic convergence of the parameter estimation error to zero [85]. H has the same function as the sufficiently smooth symmetric Hilbert-Schmidt kernel in the equivalent continuous time learning algorithm proposed in [64] that is used to linearly parameterize the unknown parameters. The elements of H are given by:

$$H_{ij} = H_{(i-1)(j-1) mod\, N} \tag{3.36}$$

For example, in a case of the Gaussian function being approximated by the window length of $\nu = 5$ with values $j_1, j_2, j_3 \in \mathbb{R}^+$ (where j_1 represents the central peak), if we choose to take $N = 5$ (resulting in all entries of $h(\mu)$ being nonzero) points along a trajectory of length L, the possible forms of $h(\mu)$ are as follows:

$$\begin{aligned}
\mu = 0: &\quad h(0) = [j_1 \quad j_2 \quad j_3 \quad j_3 \quad j_2] \\
\mu = L/5: &\quad h(L/5) = [j_2 \quad j_1 \quad j_2 \quad j_3 \quad j_3] \\
\mu = 2L/5: &\quad h(2L/5) = [j_3 \quad j_2 \quad j_1 \quad j_2 \quad j_3] \\
\mu = 3L/5: &\quad h(3L/5) = [j_3 \quad j_3 \quad j_2 \quad j_1 \quad j_2] \\
\mu = 4L/5: &\quad h(4L/5) = [j_2 \quad j_3 \quad j_3 \quad j_2 \quad j_1]
\end{aligned} \tag{3.37}$$

In this case H will be given by:

$$H = \begin{bmatrix} j_1 & j_2 & j_3 & j_3 & j_2 \\ j_2 & j_1 & j_2 & j_3 & j_3 \\ j_3 & j_2 & j_1 & j_2 & j_3 \\ j_3 & j_3 & j_2 & j_1 & j_2 \\ j_2 & j_3 & j_3 & j_2 & j_1 \end{bmatrix} \tag{3.38}$$

which is symmetric and positive definite. Thus, the unknown parameters are represented as a linear combination of basis functions that are the eigenvectors of H. The structure of H means that we need only store a single row, $h(\mu)$. Additionally, there is a benefit of reduced additions and multiplications since ν and N can be chosen such that only a few entries of H are non-zero ($\nu \ll N$), leading to a reduced computation time. Although this learning algorithm does not allow for selective frequency identification, the rate at which different frequencies can be learned can be adjusted to some degree. This is done by altering the window length (ν) of the Gaussian or using another kernel shape [85].

Now that the model of the user's Hill force (f_h) is expressed as a discrete parameterization along μ, we require a measurement of the actual force applied by the user along this contour. The access to this measurement is necessary in order to learn the user's Hill surface (f_h) as this allows us to 'fit' f_h to the measured user applied force using adaptive techniques (introduced in Section 3.5). As the user applied force is the primary component of the environmental torques τ_e (apart from disturbances due to friction and model uncertainties, which are also part of the environment), τ_e is assumed to be sufficiently representative of the user applied force. Thus, the user applied force is assumed to be given by the scalar projection of the environmental force along the desired circular contour:

$$F_\mu(x,t) = F_x(t) \cdot {V_x}^0(x_d(x)) \tag{3.39}$$

where $F_x(t)$ represents the environmental force in task space and is given by $F_x(t) = J^{-1} \cdot \tau_e(t)$. $F_\mu(x,t)$ is scalar and corresponds to the component of $F_x(t)$ that is tangent to $V_x(x_d(x))$. Note that the calculation of this scalar projection is only true for a normalized velocity field i.e., $\|{V_x}^0\| = 1$ (which is the case in this thesis, refer to Section 3.1).

Issues arise when we approximate the user applied force along μ with F_μ. The surface defined by $f_h(\mu,\dot{\mu})$ is only characterized along μ, $\dot{\mu}$, thus, the user is required to adhere to the desired circular contour for the Hill parameters to be representative of the user's biomechanical characteristics along μ. Deviations from the desired contour result in different poses of the user's arm and the use of different muscle groups that exhibit force-velocity-position characteristics that are not representative of $f_h(\mu,\dot{\mu})$. The further the user deviates from the desired contour, the less representative their force characteristics are expected to be of $f_h(\mu,\dot{\mu})$. This takes into account that although PVFC 'guides' the user to x_d, $\dot{x}_d$, due to the difficulty of the task and imperfect tracking by PVFC the user is still expected to deviate from this (the reasons for this are elaborated on in Section 4.5). Thus, with increasing distance from x_d, the exhibited force-velocity-position characteristics are expected to become less representative of $f_h(\mu, \dot{\mu})$. This distance is given by $\|x - x_d\|$, which is the closest point distance (or Euclidean distance) from the end effector to the desired contour. Thus, given $\|x - x_d\|$ remains sufficiently small, $F_\mu(x, t)$ is representative of $f_h(\mu, \dot{\mu})$. The value of $\|x(t) - x_d(t)\|$ is investigated through experimentation (Chapter 4).

This is also true for the 'direction' in which the user applies the force. The user applied force must be tangent to μ as $f_h(\mu, \dot{\mu})$ is only characterized tangent to μ. However, since the user can move in 3D, F_x is not necessarily tangent to μ. Consider that the user tracks the desired contour counter-clockwise and the user applied force ($F_x(t)$) is tangent to the desired contour such that $F_\mu(x, t) = \|F_x(t)\|$. When this happens, the environmental force best represents the user's force-velocity-position characteristics defined by f_h. This appears to be true even when $F_\mu(x, t) \cong \|F_x(t)\|$ (which occurs when the user applied force is near tangent to the desired contour) as the same muscle groups are involved in generating the component of force that is tangent to the desired contour. However, if $F_\mu(x, t) \ll \|F_x(t)\|$ (the user applied force is far from tangent to the desired contour) then it is not representative of f_h. This situation is avoided by injecting damping (introduced in Section 3.7), the resulting updates of the parameter estimate vector (introduced in Section 3.5) and specific instructions to the user (Section 4.2).

Due to the lack of force and torque sensors, we do not have access to τ_e. In the following section we will introduce a force observer that is used to obtain a filtered approximation of the environmental torques and consequently an estimate of the user applied force along the desired contour ($\hat{F}_\mu$).

3.4 Force Observer

One way to estimate the user force is to design a force observer based on the manipulator's joint positions and velocities. Since the motor applied torques ($\tau = \tau_{PVFC} + \tau_D$), as well as the robot parameters $M(t)$ and $C(t)$ are known, we can obtain $\hat{\tau}_e$. $\hat{\tau}_e$ is the stable filtered output of the environmental torques τ_e and is given by:

$$\hat{\tau}_e = \frac{\lambda}{\lambda + s}\tau_e \qquad \lambda > 0 \tag{3.40}$$

where λ corresponds to the cutoff frequency of the low-pass filter (rad/s) and τ_e represents the true environmental torques. Using Equation (2.6), $\hat{\tau}_e$ can be written as:

$$\hat{\tau}_e = \lambda M(q)\dot{q} - \frac{\lambda}{\lambda + s}(\tau - C(q,\dot{q})\dot{q} + \lambda M(q)\dot{q}) \tag{3.41}$$

Due to model uncertainties, friction and joint position and velocity measurements that appear to contain noise, $\hat{\tau}_e$ acts as an estimate of τ_e. Notice that the force observer does not require knowledge of the joint accelerations. The filtered force in task space is given by $\hat{F}_x(t) = J^{-1} \cdot \hat{\tau}_e(t)$ and its component tangent to the desired contour by $\hat{F}_\mu(x,t) = \hat{F}_x(t) \cdot {V_x}^0(x_d(x))$ (as in Equation (3.39)). $\hat{F}_x(t)$ is the estimate of the user applied force in task space ($F_x(t)$) and $\hat{F}_\mu(x,t)$ is the estimate of the user applied force along the desired contour ($F_\mu(t)$). As mentioned in Section 3.3, $\hat{F}_\mu(t)$ is representative of f_h given $\|x - x_d\|$ is sufficiently small and $\hat{F}_\mu(x,t) \cong \|\hat{F}_x(t)\|$.

The filtered regressor vector $\rho(t)$ is given by:

$$\rho = \frac{\lambda}{\lambda + s}\phi \tag{3.42}$$

$\frac{\lambda}{\lambda+s}$ acts as the first order low pass filter. Notice that the DC gain is 1, meaning that only filtering occurs at a cut-off frequency of $\omega_c = \frac{\lambda}{2\pi}\ Hz$. The filter has an effect of reducing high frequency noise with the intention of capturing smooth user applied force data that are by nature at low frequencies. The frequencies of interest usually correlate with the pace of the exercise motion. Specifically, the frequency of the exercise motion at the highest reference velocity (see Subsection 3.8.1) is much smaller than $1Hz$, so the cutoff frequency is taken as $\omega_c = 1\ Hz$. Note that the average sampling frequency of the BURT controller is around $\omega_s \approx 465\ Hz$. The discretized filter is given by:

$$\mathcal{H}_{\mathcal{F}}(z) = \frac{0.01342}{z - 0.9866} \tag{3.43}$$

3.5 Online Parameter Adaptation

The injected damping (introduced in Section 3.7) that generates the resistive force that opposes the user requires an estimate of the user's force. Using the certainty equivalence theorem [69], the estimate of the user applied force is given by:

$$\hat{f}_h(\mu(t), \dot{\mu}(t)) = \hat{a}(\mu(t)) - \hat{b}(\mu(t))\dot{\mu}(t) = \begin{bmatrix} 1 & -\dot{\mu}(t) \end{bmatrix} \begin{bmatrix} \hat{a}(\mu(t)) \\ \hat{b}(\mu(t)) \end{bmatrix} = \phi^T(t)\hat{\theta}(\mu(t)) \qquad (3.44)$$

where the estimate of the Hill parameters is $\hat{\theta}(\mu(t))$, which is given by:

$$\hat{\theta}(\mu) = \begin{bmatrix} \hat{a}(\mu) \\ \hat{b}(\mu) \end{bmatrix} = h(\mu) \begin{bmatrix} \hat{c}_a \\ \hat{c}_b \end{bmatrix} \qquad (3.45)$$

where the estimate is used as if it were the true value of the unknown parameter vector. However, the use of the certainty equivalence based estimate may generate unwanted transients and achieve suboptimal performance. This assumes that the learning algorithm is able to identify the unknown parameters relatively quickly and with sufficient accuracy such that these negative effects are mitigated.

The parameter estimates are updated using a discrete version of the parameter adaptation algorithm presented in [64]. An overview of this discrete time learning algorithm can be found in [85]. The adaptation law extracts information from the input (prediction) and output (tracking) error. This is known as Composite Adaptive Control and was first introduced in [86]. The PAA (Parameter Adaptation Algorithm) is given by:

$$\dot{\hat{\theta}}(t) = Proj_{\hat{\theta}}\left[-\boldsymbol{P}(t)\hat{\theta}(t) + \boldsymbol{d}(t) + \Phi(t)e_1(t)\right] \qquad (3.46)$$

$$\dot{\boldsymbol{P}}(t) = -\lambda_a \boldsymbol{P}(t) + \rho(t)\rho^T(t) \qquad (3.47)$$

$$\dot{\boldsymbol{d}}(t) = -\lambda_a \boldsymbol{d}(t) + \rho(t)\left|\hat{F}_\mu(x(t), t)\right| \qquad (3.48)$$

with the initial condition of $\boldsymbol{P}_0 = [0]_{2\times2}$ and $\boldsymbol{d}_0 = [0]_{2\times1}$, where $\lambda_a > 0$ is a forgetting factor. $e_1(t) = \dot{\mu}(t) - V_\mu(\mu(t))$ and $\Phi(t) = \begin{bmatrix} 1 & -V_\mu(\mu(t)) \end{bmatrix}^T$. The PAA enables the learning of a Hill surface that characterizes the user's force along μ, given that the task is sufficiently executed. Two types of error signals are used in this parameter adaptation algorithm; the 'prediction' error signal as well as the 'output' error signal:

- The *prediction error signal,* $-\boldsymbol{P}(t)\hat{\theta}(t) + \boldsymbol{d}(t) = -\boldsymbol{P}(t)\tilde{\theta}(t)$ (which can be found by multiplying both sides of Equations (3.47) and (3.48) by θ), enables us to obtain a direct measurement of the parameter estimate error ($\tilde{\theta} = \hat{\theta} - \theta$). The prediction error along with the filtered gain matrix update law enables the parameter error to vanish asymptotically.
- The *output error signal*, $\Phi(t)e_1(t)$, is only related to the parameter error ($\tilde{\theta}$) through the dynamics of the system. This is the adaptation term that allows for the user to asymptotically exercise at the desired velocity.

As mentioned in Section 3.3, if $F_\mu(x,t) \ll \|F_x(t)\|$ then it is not representative of f_h (the true Hill surface). This includes the situation where $F_\mu(x,t) < 0$. This occurs when the user deviates greatly from the desired contour and attempts to return to it, or if PVFC 'pulls' the user when an increase in the reference velocity occurs (indicating that the reference velocity is set too high, or the user is unable to sufficiently track it). This being said, when $\hat{F}_\mu(x,t) < 0$ the learning algorithm adjusts $\hat{\theta}(\mu(t))$ such that $\hat{f}_h(\mu(t), \dot{\mu}(t))$ increases, additional damping is injected (refer to Section 3.7) and $\hat{F}_\mu(x,t)$ approaches $\|\hat{F}_x(t)\|$ in subsequent cycles. This is achieved by taking the absolute value of $\hat{F}_\mu(x(t),t)$ (in Equation (3.48).

The inclusion of a projection operator ($Proj_{\hat{\theta}}$ in Equation (3.46)) ensures that the parameter estimates remain in a convex set. The projection operator is applied prior to updating the influence functions (Equation (3.51)) such that the adaptation step is not carried out if the bounds are broken. The bounds are given by:

$$\hat{a}(\mu,t) \geq \underline{a}(\mu) \ldots \text{and} \ldots \;\; \hat{b}(\mu,t)\ \epsilon\ [\underline{b}(\mu), \overline{b}(\mu)] \qquad (3.49)$$

$$0 < \underline{a}(\mu) - V_\mu(\mu)\overline{b}(\mu) \leq \underline{a}(\mu) - \underline{V}_{\,\mu}\overline{b}(\mu) \qquad (3.50)$$

where $\underline{a}(\mu)$ corresponds to the lower bound of $a(\mu)$ and similarly $\underline{b}(\mu)$, $\overline{b}(\mu)$ correspond to the lower and upper bounds of $b(\mu)$ respectively. Additionally, $\underline{V}_{\,\mu}$ is the lowest reference speed. The bounds are determined experimentally. In the application of the learning algorithm the influence functions are updated directly. This is done discretely at the control frequency in real time:

$$\dot{\hat{c}} = \Gamma\, h(\mu)\, \dot{\hat{\theta}}(t) \tag{3.51}$$

where Γ corresponds to a constant adaptation gain.

3.6 Optimal Exercise Strategy

The mechanism of learning a model of the user's biomechanical characteristics and using this knowledge to achieve an optimal exercise strategy is what makes this exercise controller 'intelligent'. The exercise performed by the user is assumed to lie on a Hill surface such that the exercise trajectory can be defined by $\mu(t)$, $\dot{\mu}(t)$ and $f_h(\mu(t), \dot{\mu}(t))$. In this thesis, the specific criteria used to generate the optimal exercise trajectory is the Opti-poteric criteria (Opti = maximum, poter = is Latin for power). This is obtained by maximizing the function:

$$\Pi\big(F_\mu, \dot{\mu}\big) = F_\mu \cdot \dot{\mu} \tag{3.52}$$

for all $t \geq 0$ where Π is the mechanical power generated by the equivalent muscle. Maximizing Π means that the user must exercise with maximum caloric consumption. F_μ is the environmental force tangent to the desired contour that is assumed to satisfy $F_\mu = f_h(\mu, \dot{\mu})$.

The goal of the adaptive controller is to identify and track the opti-poteric velocity field that maximizes the instantaneous power produced by the user along the desired contour. If the functions $a(\mu)$ and $b(\mu)$ are well known, using Equation (3.30):

$$\Pi(f_h(\mu, \dot{\mu}), \dot{\mu}) = f_h(\mu, \dot{\mu}) \cdot \dot{\mu} = a(\mu)\dot{\mu} - b(\mu)\dot{\mu}^2 \tag{3.53}$$

$$\frac{d}{dt}\Pi(f_h(\mu, \dot{\mu}), \dot{\mu}) = 0 = a(\mu)\ddot{\mu} - 2b(\mu){V_\mu}^*(\mu)\ddot{\mu} \tag{3.54}$$

where ${V_\mu}^*(\mu)$ is the optimal speed at each value of μ. Thus, the optimal condition is given by:

$$V_\mu{}^*(\mu) = \frac{1}{2}\frac{a(\mu)}{b(\mu)} \tag{3.55}$$

$$F_\mu{}^*(\mu) = \frac{a(\mu)}{2} \tag{3.56}$$

where $F_\mu{}^*(\mu)$ is the optimal scalar user generated force at each μ such that Π is maximized if exercise is executed according to $F_\mu{}^*(\mu)$, $V_\mu{}^*(\mu)$. In joint space the optimal reference is given by:

$$V^*(x) = V_q{}^*(x) = V_\mu{}^*(\mu(x)) \cdot V_q{}^0(x_d(x)) \tag{3.57}$$

Similarly, using forward kinematics $V^*(q)$ can be found. This is subject to the constraint that the user is able to track the desired contour and $V^*(q)$ sufficiently well such that Π is maximized. Since the parameters $a(\mu)$ and $b(\mu)$ are unknown, using the certainty equivalence theorem the estimate of the optimal speed is given by:

$$\hat{V}_\mu{}^*(\mu(t)) = \frac{1}{2}\frac{\hat{a}(\mu(t))}{\hat{b}(\mu(t))} \tag{3.58}$$

The same procedure as Equation (3.57) is used to calculate $\hat{V}^*(x(t))$ and using forward kinematics $\hat{V}^*(q(t))$ can be obtained.

There is a possibility that the user will only be able to approximately adhere to the desired contour. For example, the user may repeatedly trace an ellipse rather than a circle. As long as this motion can be sufficiently parameterized with respect to the path μ and the motion exhibits repeatability, the identified Hill estimates are representative of this motion. Thus, Π can still be maximized given V is sufficiently well tracked.

3.7 Dynamic Damping

The user is continuously supplying energy to the closed loop system while exercising. This means that with only PVFC torques applied, the kinetic energy of the augmented system will increase without bound. To maintain a velocity during exercise, damping is injected into the closed loop system to regulate its energy. This injected damping is felt by the user as a resistive force.

The injected damping is applied in the opposite direction to the end effector velocity such that it is felt by the subject as 'conventional damping'. This refers to damping in mechanical systems comprising of only passive elements that is proportional to the systems velocity. The damping matrix must satisfy:

$$\bar{\tau}_D = -\begin{bmatrix} \hat{f}_{h_q}(q, V) \\ 0 \end{bmatrix} \tag{3.59}$$

where $\hat{f}_{h_q}$ is the estimate of the users Hill force in joint space and is given by:

$$\hat{f}_{h_q}(q, \dot{q}) = J^T \left(\hat{f}_h(\mu(q), \dot{\mu}(\dot{q})) \cdot \frac{\dot{x}(q, \dot{q})}{\|\dot{x}(q, \dot{q})\|} \right) \tag{3.60}$$

where $\dot{x}(q, \dot{q})$ is calculated by Equation (2.2). Thus, $\hat{f}_{h_q}(q, \dot{q})$ represents the estimate of the user's Hill force along the desired circular contour applied in the opposite direction of the end effectors velocity in joint space. If μ is adhered to, $V_\mu(\mu)$ sufficiently tracked and $\hat{f}_h$ is sufficiently identified such that is it representative of f_h, the observed torques exerted on the manipulator system by the user ($\hat{\tau}_e(t)$) are opposed by the estimate of these torques ($\hat{f}_{h_q}$). Along with the certainty equivalence theorem, we can use the fact that $\alpha = 1$ when the reference tracked by PVFC is exactly equal to the desired velocity field such that the injected damping takes the following form:

$$\bar{\tau}_D(\bar{q}, \dot{\bar{q}}) = -D(\bar{q}, \dot{\bar{q}})\, \dot{\bar{q}} \tag{3.61}$$

where the damping matrix $D(\bar{q}, \dot{\bar{q}})$ is given by:

$$D(\bar{q},\dot{\bar{q}}) = \begin{bmatrix} 0 & \dfrac{\hat{f}_{h_q}(q,\dot{q})}{\bar{V}_f}\left(\dfrac{\dot{\bar{q}}^T\bar{M}\dot{\bar{q}}}{2E_{max}}\right) \\ (\alpha-1)\dfrac{\hat{f}_{h_q}(q,\dot{q})}{\bar{V}_f}\left(\dfrac{\dot{\bar{q}}^T\bar{M}\dot{\bar{q}}}{2E_{max}}\right) & 0 \end{bmatrix} = \begin{bmatrix} 0 & \dfrac{\hat{f}_{h_q}(q,\dot{q})}{\bar{V}_f}\alpha^2 \\ (\alpha-1)\dfrac{\hat{f}_{h_q}(q,\dot{q})}{\bar{V}_f}\alpha^2 & 0 \end{bmatrix} \qquad (3.62)$$

When $\alpha < 1$, energy is added to the fictitious storage element and when $\alpha > 1$ energy is removed from this storage element. Notice that $D(\bar{q},\dot{\bar{q}})$ is skew symmetric when $\alpha = 0$ (a hypothetical scenario) and as α increases the matrix turns positive definite (where the matrix is positive definite when $\alpha > 1$). Thus, apart from resisting the user with a scaled estimate of their force, the damping matrix has the purpose of regulating $\alpha \to 1$ while maintaining passivity of the closed loop system.

Notice that in Equation (3.60), $\hat{f}_h$ is applied in the opposite direction to $\dot{x}$. On the other hand, applying the damping force in the opposite direction of the desired velocity field will have the effect of pushing the user away from the desired contour when the user deviates from the desired velocity direction. In attempt to adapt and reposition such that the desired contour is traced, any slight deviation will result in the user once again being pushed away from the desired contour. This results in a shaking about the desired contour that makes tracking the reference difficult if not impossible.

3.8 Reference Generator

It is not guaranteed that $\hat{V}^*$ converges to V^*. First, an accurate estimate of the Hill parameters ($\hat{\theta}$) must be obtained. With use of a binary state machine (the excitation supervisor, introduced later in Subsection 3.8.2) switching between a 'training' state (during which the Hill parameters are updated) and a 'control' state (during which the optimal exercise is instructed) ensures that the user's Hill parameters are sufficiently well characterized before the optimal exercise condition is commanded. This is synonymous with the exploitation versus exploration component of reinforcement learning, where an objective function is used to replace a 'reward' function. The final instructed velocity field at time T_k (elaborated on in Subsection 3.8.2) is given by V_k. This is 'fed' into the PVFC controller and represents the task that user is instructed to follow during the period $[T_k, T_{k+1}]$.

In the following subsection we will define the reference velocity that is instructed during the training state. Following this, the binary state machine that transitions between the training and control states is introduced. Finally, the condition which determines which state is transitioned to is presented.

3.8.1 Training Velocity Field

This subsection will specify a training behavior (which will be refer to as ${V_\mu}^{tr}(t)$) which is designed to provide sufficient excitation such that $\hat{F}_q(t)$ is representative of $\hat{f}_{h_q}$. ${V_\mu}^{tr}(t)$ corresponds a training speed along the desired contour. To enable Hill parameter convergence ($\tilde{\theta} \rightarrow 0$), the training speed must satisfy:

$$\int_{T_0}^{T_0+T_{max}} \phi(\zeta)\phi^T(\zeta)\, d\zeta \geq \eta(e) I_{2N} \tag{3.63}$$

where ϕ is the regressor, η is a continuous function of a measure of parameter error and $|\dot{\mu}(t) - V_\mu(t)| \leq e$ for all $t \in [T_0, T_0 + T_{max}]$. This is a well-known condition for sufficient excitation in adaptive control [69]. This means that when the speed $\dot{\mu}(t)$ tracks ${V_\mu}^{tr}(t)$ adequately then sufficient information is contained in $\left|\hat{F}_\mu(x(t), t)\right|$ such that $\tilde{\theta} \rightarrow 0$.

As our approximation of the hill curve is linear, it should be clear that the training velocity should visit at least two different velocities such that a force-velocity line can be fit. Thus, the training velocity is designed to alternate between two different velocities, V_{high} and V_{low} (where $0 < V_{low} < V_{high}$). It is ensured that there are smooth transitions between V_{high} and V_{low} to ensure stable PVFC input torques. The smooth transitions also ensure that the required motion is practical (the user can track this naturally). This results in the training velocity taking the following form:

$${V_\mu}^{tr}(t) = L_1(t) V_{high} + L_2(t) V_{low} \tag{3.64}$$

$$\frac{d}{dt}\begin{pmatrix} L_1 \\ L_2 \end{pmatrix} = -\lambda_L \begin{pmatrix} L_1 \\ L_2 \end{pmatrix} + \lambda_L p(t) \qquad \lambda_L \gg 0 \tag{3.65}$$

$$p(t) = \begin{cases} (1 \quad 0)^T, & if\ t\ mod\ (T_1 + T_2) < T_1 \\ (0 \quad 1)^T, & if\ t\ mod\ (T_1 + T_2) \geq T_1 \end{cases} \tag{3.66}$$

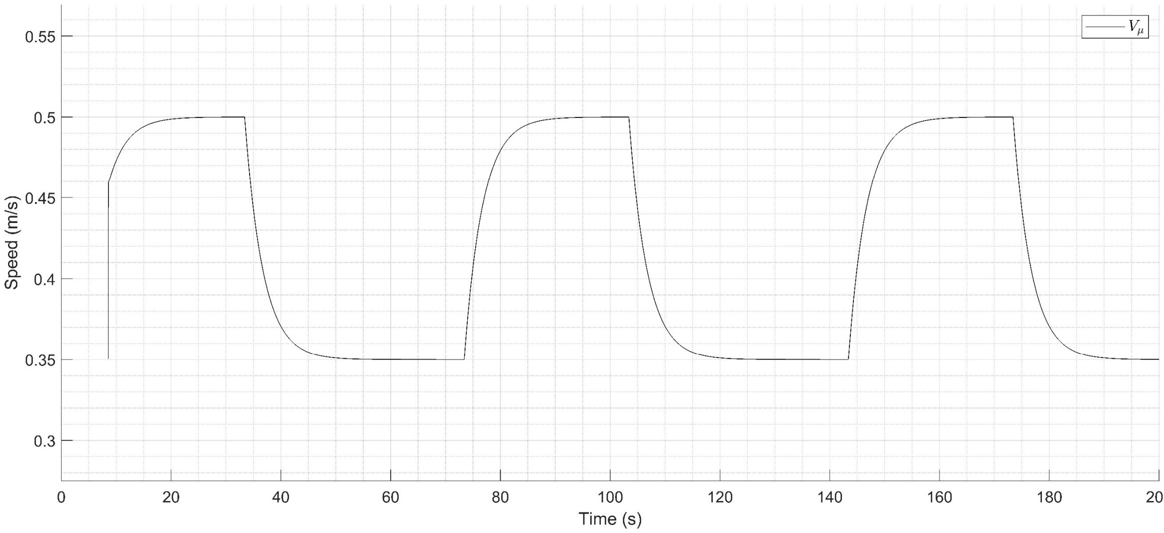

Figure 3.4: Reference training velocity signal generated for a high reference velocity of 0.5 m/s (for a period of $30s$, T1), a low reference velocity of 0.35 m/s (for a period of $40s$, T2) with $\lambda_L = 0.3$ plotted for a duration of $290s$.

where $T_1 > 2n\pi r_c/V_{hig}$ is the time to complete n cycles of the desired circular contour at V_{hig} before a switch to V_{low} is initiated. Similarly, $T_2 > 2n\pi r_c/V_{low}$. In practice, these time periods are increased in order to provide time in which the user can adjust to the new velocity. The training velocity in task space is given by:

$$V_x{}^{tr}(q,t) = V_\mu{}^{tr}(t) \cdot V_x(q) \tag{3.67}$$

taking into account the $V_x(q)$ was designed such that $\|V_x(q)\| = 1$ (Section 3.1). This is converted to joint space using the robot Jacobian (as done in Equation (3.8)) to result in $V^{tr}(q,t)$. The training velocity above satisfies the excitation condition (Equation (3.63)), for proof see [87]. The assigned parameters that define the training velocity field generator are given in Table 3.1.

Table 3.1: Training Velocity Field Generator Assigned Parameters

${L_1}_0$	${L_2}_0$	V_{high} (m/s)	V_{low} (m/s)	λ_L	T_1 (s)	T_2 (s)
0.0	1.0	0.30	0.60	0.3	30	40

3.8.2 Excitation Supervisor

The excitation supervisor is a logic unit that allows for the switching between the training velocity field and the optimal velocity field. This results in two possible states that the excitation supervisor could have, the 'train' state in which the training velocity V^{tr} is tracked and the 'control' state where an estimate of the optimal velocity $\hat{V}^*$ is tracked. The value of an internally generated optimality error signal e_{opt} (introduced in Subsection 3.8.3) is used to control the switching between the two velocity fields. This subsection defines the instructed reference velocity in the 'train' and 'control' states and during transition between these states. The timing associated with these transitions is also outlined here.

$T_k \to k = 0,1,2 \ldots$ represents the transition times and Q_k represents the state at time T_k. The initial state is $Q_{k=0} = 'train'$. The transition times T_k as well as the conditions that trigger a specific transition will be defined in the following subsection.

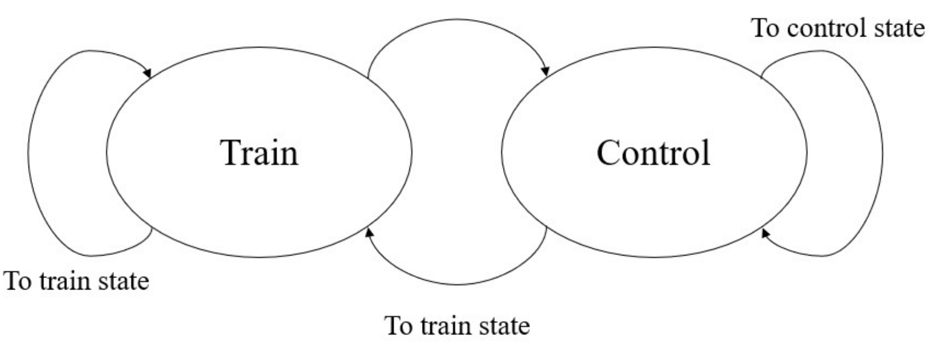

Figure 3.5: A flowchart of the excitation supervisor that contains the two possible states 'train' and 'control' (of Q) as well as all transitions between them. Whether we are in the 'train' or 'control' state decides whether the training velocity or the estimate of the optimal velocity is tracked (respectively).

All following velocity fields are presented in joint space such that $V_k = \left(V_q\right)_k$. After each transition time T_k, the following events take place:

1. The state changes from $Q_{k-1} \rightarrow Q_k$.

2. The instructed estimate of the optimal velocity field is updated:

$$\hat{V}_k^{\ *}(q) = \begin{cases} \hat{V}_{k-1}^{*}(q), & if\ Q_{k-1} = control \\ \hat{V}^{*}(q(T_k)), & if\ Q_{k-1} = train \end{cases} \qquad (3.68)$$

where $\hat{V}_k^{\ *}$ represents the instructed estimate of the optimal velocity field which is only updated by the estimate of the optimal velocity field (using the current parameter estimates, $\hat{\theta}(\mu(t))$) in the training phase. This avoids updating the instructed optimal velocity field while it is being tracked by the user (in the control phase). This insures safety against 'live' updates of $\hat{V}_k^{\ *}$ in the control phase.

3. The velocity field selected to be tracked after time T_k is:

$$V_k(q,t) = \begin{cases} V^{tr}(q,t), & if\ Q_k = train \\ \hat{V}_k^{\ *}(q), & if\ Q_k = control \end{cases} \qquad (3.69)$$

where V_k is the instructed reference velocity field that is 'fed' into the PVFC controller and represents the task that user is instructed to follow during the period $[T_k, T_{k+1}]$.

4. If a switch from train to control or vice versa is initiated by the excitation supervisor, the transition must be smooth. This is done by linearly interpolating between $V_{k-1}(q,t)$ and $V_k(q,t)$ over the period $[T_k, T_k + T_{trans}^k]$, and remaining at $V_k(q,t)$ from $(T_k + T_{trans}^k, T_{k+1}]$ i.e. until the next transition time. T_{trans}^k is a constant time frame during which the transition occurs:

$$V_k(x,t) = \left(\frac{t - T_k}{T_{trans}^k}\right) V_k + \left(\frac{T_{trans}^k - t + T_k}{T_{trans}^k}\right) V_{k-1} \tag{3.70}$$

for $t \in [T_k, T_k + T_{trans}^k]$. No smoothing is necessary if no transition between control and train occurs at time T_k. For the experiment (Chapter 3) $T_{trans}^k = 10\ s$. This must be chosen such that $T_{trans}^k < T_1$.

5. Finally, the next transition time is defined as:

$$T_{k+1} = \begin{cases} T_k + T_{max} + T_{trans}^k, & if\ Q_k = train \\ T_k + T_{dwell}, & if\ Q_k = control \end{cases} \tag{3.71}$$

where $T_{max} = T_1 + T_2$ ensures that V_{high} and V_{low} are both visited during the training phase, and T_{dwell} is the amount of time the exercise motion takes to complete n turns. A suitable choice for T_{dwell} is the value of T_2 with additional time for the user to adapt to the new velocity. If the training velocity field is chosen to be tracked an infinite number of times, this ensures sufficient excitation such that the parameter estimates converge to the true parameter values asymptotically ($\tilde{\theta} \to 0$) (refer to Appendix II of [16]).

3.8.3 State Transition

In this subsection, an error signal that is used to determine which state transition is made at time T_k is defined. The error signal that is defined in this subsection, given by e_{opt}, is a computed value that represents the accuracy of the parameter estimates as well as the degree at which the optimal objective is satisfied.

As we have seen previously (Equation (3.52)), the objective function takes the form of the power exerted by the user on the manipulator system which is given by:

$$\Pi\big(\mu, \dot{\mu}, \hat{\theta}(\mu)\big) = \dot{\mu}\big[\hat{a}(\mu) - \hat{b}(\mu)\dot{\mu}\big] \tag{3.72}$$

when $f_h(\mu, \dot{\mu})$ is representative of F_μ. Using Equation (3.55), the estimate of the optimal value of the objective function can be derived to be:

$$\Pi^*(\mu) = \frac{1}{4}\frac{\hat{a}(\mu)^2}{\hat{b}(\mu)} \tag{3.73}$$

e_{opt} is given by:

$$e_{opt}(k) = \gamma_1 g_1(k) + g_2(k) \tag{3.74}$$

where $\gamma_1 > 0$ is a gain factor. $g_1(k)$ is given by:

$$g_1(k) = \begin{cases} \max\limits_{\mu,t}\left|\Pi^*(\mu) - \Pi\big(\mu, \hat{V}^*_{k-1}, \hat{\theta}(\mu)\big)\right|, & Q_{k-1} = train \\ 0, & Q_{k-1} = control \end{cases} \tag{3.75}$$

where the max is taken over $t \in [T_{k-1}, T_k)$. $\Pi^*(\mu)$ is taken from Equation (3.) and $\Pi\big(\mu, V^*_{k-1}, \hat{\theta}(\mu)\big)$ from Equation (3.). $g_1(k)$ checks the consistency of the current parameter estimates and the previous estimate of the optimal velocity field. If the value of $g_1(k)$ is small, this indicates $V^*_{k-1} \cong V^*_k$ and that the change in the parameter estimates ($\hat{\theta}$) between time T_{k-1} and T_k is small. $g_2(k)$ is a measure the parameter estimation error and is given by:

$$g_2(k) = \begin{cases} \left|\left|\hat{\theta}(\mu(T_k)) - \boldsymbol{P}^{-1}(T_k)\boldsymbol{d}(T_k)\right|\right|^2, & if\ Q_{k-1} = train \\ g_2(k-1), & if\ Q_{k-1} = control \end{cases} \tag{3.76}$$

which can be shown to be equal to $||\tilde{\theta}(T_j)||^2$ (refer to Section 3.5, the prediction error signal) where T_j is the last time after V^{tr} was selected.

The transition event is triggered at time T_k if e_{opt} is less than some predefined tolerance for suboptimality. This entails that the parameter estimates closely represent the true parameter values and we switch to (or stay in) the control phase.

$$\boldsymbol{S}(k) = \begin{cases} control, & if\ e_{opt}(k) \le e_{opt\ bound} \\ train, & otherwise \end{cases} \tag{3.77}$$

where $\boldsymbol{S}(k)$ represents the transition event at time T_k. $e_{opt\ bound} > 0$ represents a tolerance that is chosen during initial experiments. In application, the user's Hill surface varies due to fatigue, change in levels of motivation and other factors. By appropriately setting γ_1 and $e_{opt\ bound}$ the excitation supervisor can be made insensitive to small variations in the parameter estimates while still adjusting to respond to large changes.

3.9 Summary of Controller

The closed-loop system dynamics with PVFC and injected damping is given by the equation:

$$\bar{M}(\bar{q})\ddot{\bar{q}} + \bar{C}(\bar{q},\dot{\bar{q}})\dot{\bar{q}} = \bar{\tau}_{PVFC} + \bar{\tau}_D + \bar{\tau}_e \tag{3.78}$$

A block diagram of the proposed smart exercise control scheme can be seen in Figure 3.6. This block diagram shows all the components of the proposed controller that were outlined in previous sections.

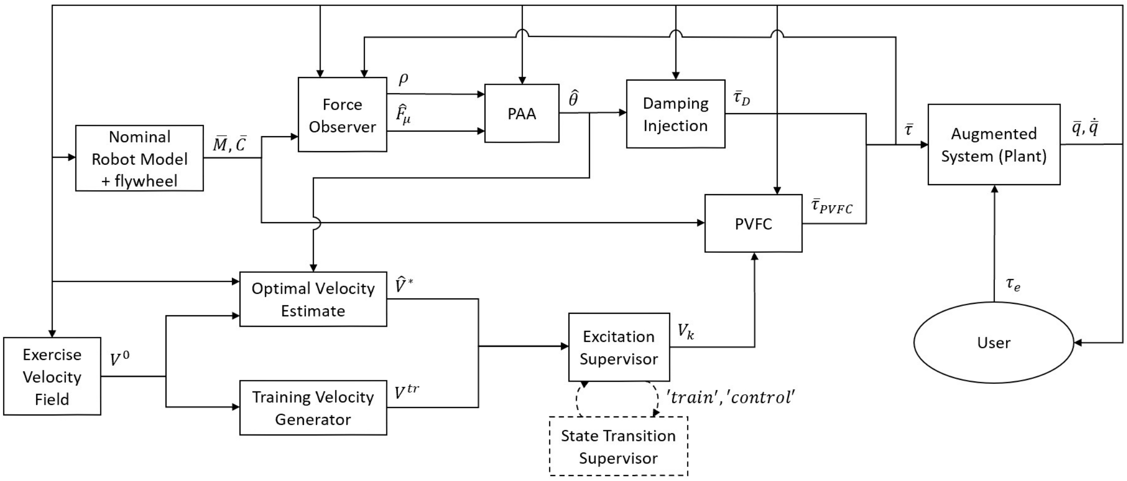

Figure 3.6: Block diagram of the proposed smart exercise control scheme. Illustrates how signals are routed between the various components of the controller. The State Transition Supervisor decides whether to 'train' or 'control'. Signals routed to the State Transition Supervisor that are used to calculate e_{opt} are excluded to simplify the diagram.

Chapter 4
Verification of the Smart Exercise Machine Controller on the BURT Manipulator

Determining accurate Hill parameter estimates for multi-joint movements is difficult, but useful due to its potential applications in the scope of rehabilitation. Such knowledge could provide quantitative methods for evaluating a patient's performance in a dynamic task, as well as ensuring safe human-robot interactions. Unfortunately, the study of multi-joint force-velocity-position characteristics is generally limited to mechanical systems composed of fully passive elements (such as conventional exercise machines) or simplistic low degree of freedom manipulators. This presents obvious limitations. Notably, these machines are limited in their ability to command a variety of different exercise motions. In addition to this, identification of a patient's biomechanical characteristics is often used for diagnoses as opposed to generating intelligent exercise routines. This is generally difficult to do with machines that are comprised of mainly passive elements as it often involves extensive calibration 'by hand'. Apart from this, the potential of interacting with dynamic haptic environments that behave differently depending on the user's physical capabilities is a relatively new field of research.

In this experiment, a subject is attached to the end-effector of the BURT robot system and performs a repetitive exercise motion during which the subject's force-velocity-position characteristics is learned in an online manner. A desired velocity field that converges to a repetitive exercise motion defines the task. If the user sufficiently follows the task, their Hill curve is identified and used (in real time) to generate opposing forces such that the user is required to exercise with maximal caloric consumption. This main experiment is preceded by an initial experiment that evaluates the efficacy of the PVFC controller (defined in Section 3.2).

In this chapter, relevant background work is first discussed, the experimental setup is then defined, the results along with a discussion of the findings are presented, and finally, the limitations encountered are addressed.

4.1 Relation to Background Work

Previous work on self-optimizing control schemes that achieve the Opti-poteric exercise criteria has been conducted on a single link manipulator [15], [16]. In the single link case, the Hill curve of the user depends only on the position and angular velocity of a single joint. However, in our formulation we attempt to learn a damping assignment structure capable of handling a similar multi-joint exercise motion on a mechanical manipulator with 3 DOF. This allows us to specify arbitrary 3D exercise trajectories (represented as velocity fields), thus, allows for a far greater number of potential exercise routines.

The case of a single link manipulator results in the desired contour being a circle from which the user cannot deviate. Although this greatly simplifies the user experience and lowers the difficulty of executing the task, this entails the obvious problem when it comes to obtaining an accurate Hill surface along the desired contour. At certain poses, the user is able to apply non tangential forces and still obtain tangential motion. Thus, the same resulting end effector motion can arise from a variety of muscle activations, all that would otherwise exhibit different force-velocity-position characteristics. Enabling the user to move freely in Cartesian space eliminates this problem and ensures that the identified Hill surface at each position along the desired contour is representative of a specific activation of muscle groups. On the other hand, this increases the difficulty of conducting the experiment as the user must follow the desired velocity field sufficiently well while experiencing opposing forces.

The following implementation ignores the inertial and Coriolis dynamics of the user in the closed loop system such that the $M(q(t))$ and $C(q(t), \dot{q}(t))$ matrices are independent of the user and represent the manipulator dynamics only. This exclusion of the user from the closed loop dynamics allows multiple patients to use the exercise scheme without any required user specific calibration. The motivation here is to create a smart exercise strategy that has greater potential for application on active rehabilitation systems that frequently train a variety of patients. Apart from this advantage, excluding user dynamics adds comparative value to any user specific Hill surfaces that are identified through the proposed scheme. The advantage of allowing for valuable comparisons between the biomechanics of different patients is to help judge a patient's ability, improvement through rehabilitative processes, and possibly allow for the quantitative diagnosis of upper limb incapacities through exercise routines. However, it should be noted that knowledge of user specific biomechanics identified on one manipulator do not translate to other manipulators. The identified biomechanics are also subject to model uncertainties, friction, and other 'environmental' affects (which are treated as environmental forces and thus are components of the force observer output). This is elaborated on in Section 4.5 and observed in the results (Section 4.3).

In the original work, a dynamic damping term is implemented to ensure that the force observer always returns a positive and opposing force to the motion of the user. This is achieved by ensuring that the damping matrix is positive definite. This was changed such that the damping matrix is positive definite when the energy of the augmented system exceeds the desired energy ($\alpha > 1$) and near skew symmetric when the energy of the augmented system is near zero (see Section 3.7). This damping formulation regulates the energy of the system about $\alpha = 1$ while maintaining passivity with respect to the closed-loop system. Regulating α about 1 avoids instabilities concerned with low and high values of α as mentioned in [81].

4.2 Experimental Setup

A preliminary experimental trial (referred to as the first experimental trial) is used to evaluate the effectiveness of PVFC in tracking the desired velocity field, adhering to the desired contour defined by μ, and finally, its ability to maintain stability in face of dissipative forces. This preliminary experiment is useful in aiding in the discussion of the main results concerning the smart exercise controller.

For the main experiment (referred to as the second experimental trail), the proposed control scheme was verified by experimentation on a single user. The desired velocity field traces a circle in the horizontal plane such that the height of the motion remains constant throughout the desired trajectory. The height of the motion is calibrated to line up with the user's right shoulder. The user is seated in an upright position and conducts the experiment with their right arm (on a right-handed setup of the BURT, see Chapter 2). The user is instructed to exercise at a constant effort level until this can no longer be maintained, at which time the experiment ends.

Initially, the user is allowed a habituation phase during which the user conducts the velocity field tracking task without resistive forces. During this phase, the user is instructed to track to the best of their ability the desired motion. The purpose of the habituation phase is to familiarize the user with the desired exercise motion and become habituated to the motion prior to conducting the experiment with damping. The training velocity reference is tracked during this phase to familiarize the user with the pacing during the training phase. This pre-experiment habituation reduces the error caused by the user's unfamiliarity with the task.

The subject is provided with a simple interface that indicates the task following error. A single number that is updated every 'cycle' ($\mu: 0 \rightarrow L$) is shown on screen that indicates the mean of the absolute value of the subject's distance from the desired contour in the last 'cycle' (given by $\frac{1}{L}\sum_{\mu=0}^{L}||x - x_d||$). This is only intended to notify the user (and the empiricist) of large deviations due to fatigue, reduced effort level or growing transgression from the desired contour. An interface is not necessary as the user is 'guided' along the exercise motion by PVFC.

The values assigned to various parameters are indicated in Table 4.1 (for parameter values associated with the training velocity reference generator refer to Table 3.1).

Table 4.1: Values Assigned Prior to Conducting Experiments

Parameterization		**PVFC**	
Kernel Length, N	512	E_{max} $(joules)$	40
Window Length, ν	100	γ	0.18
Learning Scheme		$\dot{q}_{f_0}$ (rad/s)	5
$c_a(\cdot)_0$	0.1	$\dot{q}_0$ (rad/s)	$[0 \quad 0 \quad 0]^T$
$c_b(\cdot)_0$	0.15	m_f (kg)	1
Γ	0.0001	**Reference Generator**	
λ_{PAA}	0.95	T_{trans}^{k} (s)	10

State Transition Supervisor		$T_{sp}^{k}\ (s)$	10
γ_1	200	$Q_{k=0}$	$'train'$
$e_{opt\ bound}$	100		

Table 4.2: Desired Velocity Field Specifications

$x_{1_c}\ (m)$	$x_{2_c}\ (m)$	$x_{3_c}\ (m)$	$r_c\ (m)$	σ_N	σ_Z
0.65	-0.15	0.3	0.2	12	0.1

During initialization of the PVFC algorithm the augmented system must contain a non-zero amount of energy (mentioned in Section 3.2). For this reason, the flywheel was given an initial velocity. Note that since friction was not compensated for, the robot should not be left without user interaction such that significant energy is dissipated. This may cause divergence of the PVFC input torques. Thus, the user is also instructed to always exert power in the direction of the reference velocity field (to the best of their ability).

4.3 Results

In this section, the results of two experimental trials are presented. In the first trial, the performance of PVFC is assessed. In the second trial, dynamic damping is injected along with the self-optimizing control strategy to learn the subjects Hill parameters and optimize their power output during the exercise task. The results are quantitatively presented in this section. A discussion of these results follows in Section 4.4.

In the **first experimental trial** the effectiveness of the PVFC algorithm is in its ability to track the reference velocity field with and without user input is investigated. The reference speed is assigned a constant value of $V_\mu = 0.4\ m/s$. The figures below correspond to the time period of 40s to 75s. During this time, the controller operates both with and without the presence of a user.

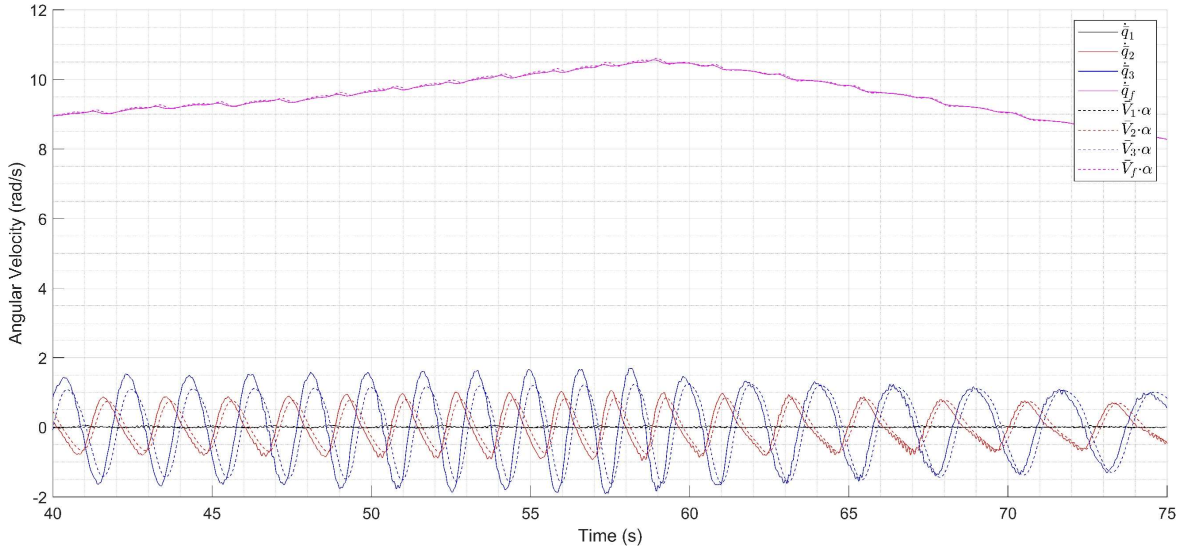

Figure 4.1: Desired augmented joint velocities scaled with α and plotted alongside the augmented joint velocities to illustrate the tracking ability of PVFC with and without user input. For the first experimental from $40s$ to $75s$.

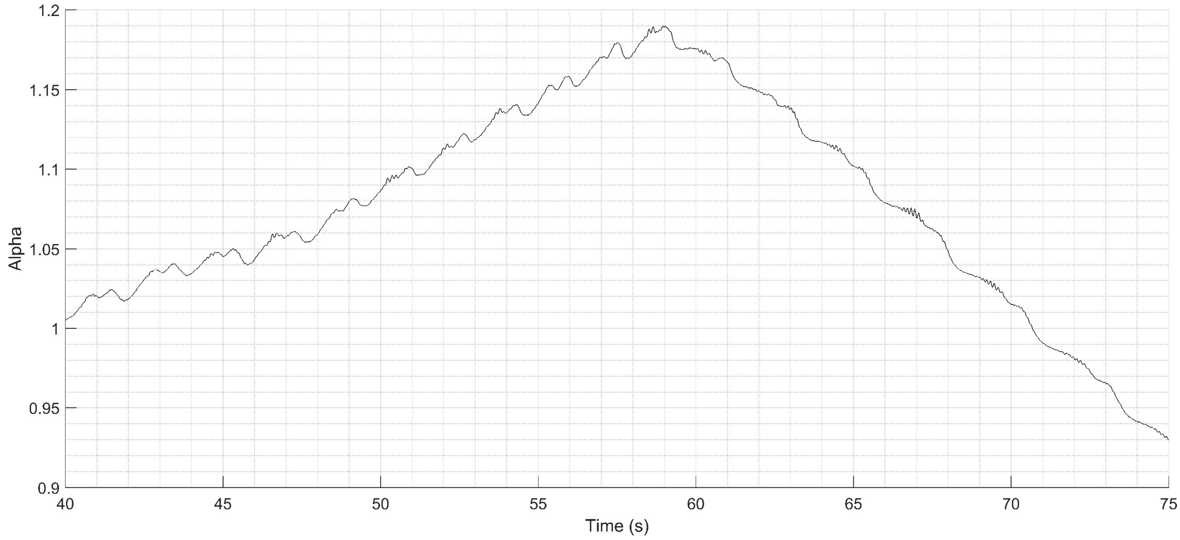

Figure 4.2: The tracked scalar multiple of the desired joint velocities with and without user input when only PVFC is active. For the first experimental trial from 40s to 75s.

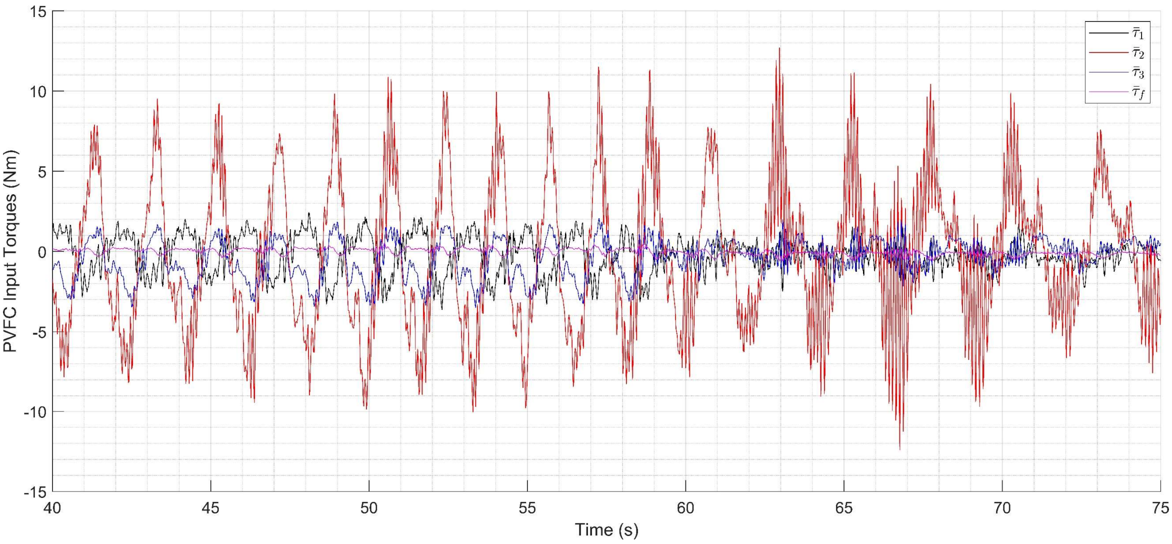

Figure 4.3: Input torques generated by PVFC with and without user input. Consists of the summation of coupling control torques and feedback control torques. For the first experimental from 40s to 75s.

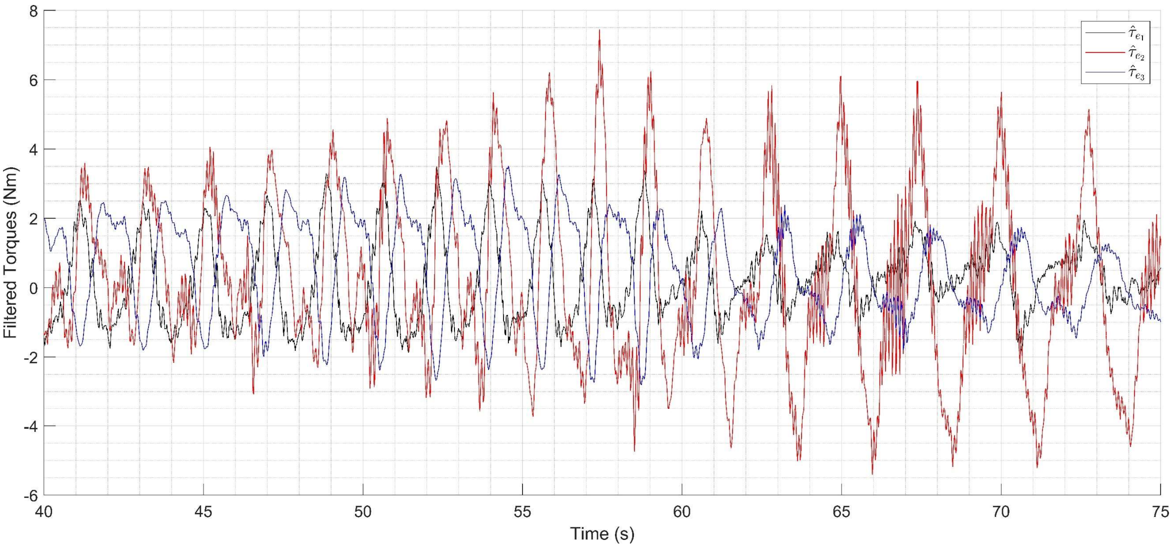

Figure 4.4: Output of the force observer, indicative of the user input torques (and model uncertainties + frictional forces), collected with and without user input during PVFC control. For the first experimental trial from 40s to 75s.

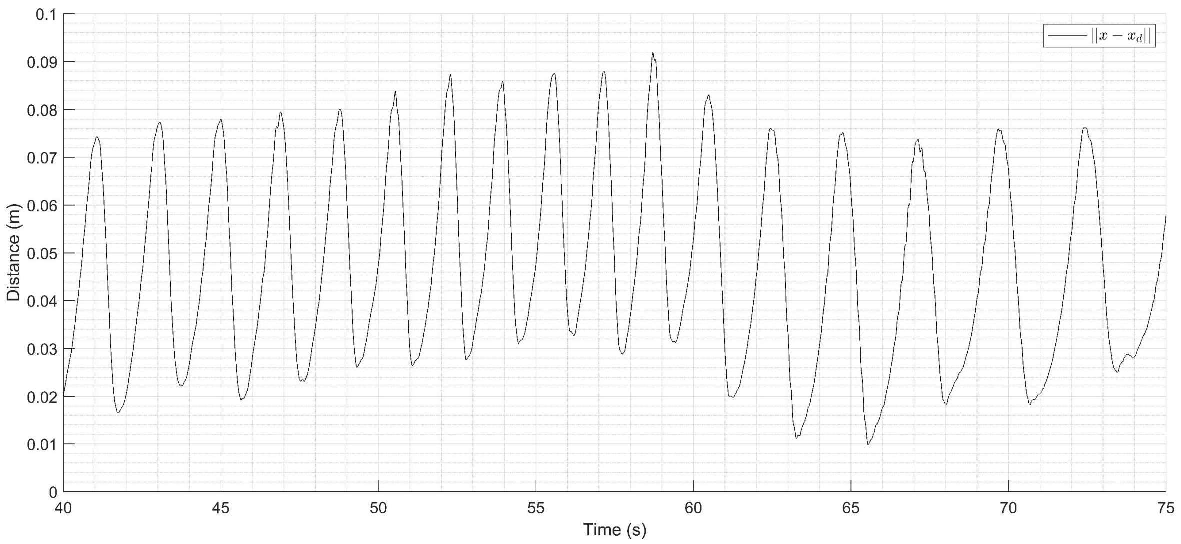

Figure 4.5: Euclidean distance between the end effector pose in task space and the closest point along the desired trajectory observed during PVFC control with and without user input. During the first experimental trial from 40s to 75s.

As there was no damping injected in this experimental trial, during the period of time between 40s and 55s the energy of the augmented system is seen to increase indefinitely. This is due to the user supplying power to the closed-loop system. The quantity of energy added by the user during this sample time period can be measured to be ${\alpha_{55s}}^2 E_{max} - {\alpha_{40s}}^2 E_{max} = 8.0504\ Joules$. This can be deduced from the plot of α (Figure 4.2). Despite this, the PVFC algorithm is able to sufficiently redirect the user such that the reference joint velocities (given by V_q) are tracked. This is done by storing excess energy in the fictitious flywheel, as can be seen from the increasing flywheel velocity during this period (Figure 4.1). The tracking ability during this time period can be quantified by the mean values of $|\bar{e}_\alpha|$ (measure of joint velocity tracking error) that are given by 0.01939, 0.22997, 0.53220, 0.02727 m/s corresponding to $|\bar{e}_{\alpha_1}|$, $|\bar{e}_{\alpha_2}|$, $|\bar{e}_{\alpha_3}|$, $|\bar{e}_{\alpha_f}|$ respectively.

Upon 'letting go' (no force applied to the system by the user) during the period of time between 60s and 75s, the PVFC controller exhibits better tracking ability of the reference joint velocities when compared to the 40s to 55s time period. The tracking ability during this time period can be quantified by the mean values of $|\bar{e}_\alpha|$ that are given by 0.01348, 0.15784, 0.32323, 0.01301 m/s corresponding to $|\bar{e}_{\alpha_1}|$, $|\bar{e}_{\alpha_2}|$, $|\bar{e}_{\alpha_3}|$, $|\bar{e}_{\alpha_f}|$ respectively. The stored energy of the flywheel is able to compensate for the dissipation of energy from the mechanical system such that the reference joint velocities of the manipulator system can still be tracked sufficiently well. During this time period, the significant friction and 'sticking' exhibited by the second joint is the main source of tracking error. The manipulator system dissipates energy at an average rate of -1.37622 $Joules/s$ with a standard deviation of 2.14178 $Joules/s$.

The tracking of the desired circular contour given by x_d is observed to be better when there is no user interaction force present (as expected, as this is seen as a disturbance to the controller). The maximum value of $\|x - x_d\|$ when the user is interacting with the system is 0.0920 m which is 46.0% of r_c, while the maximum value of $\|x - x_d\|$ when no user interaction force present (after stabilizing) is 0.0762 m which is 38.1% of r_c.

In the **second experimental trial** the ability of the smart exercise controller to learn the user's Hill parameters and command the opti-poteric exercise regime is investigated. The validity of the online estimates is also investigated. The figures below correspond to an experiment conducted for a duration of 360s.

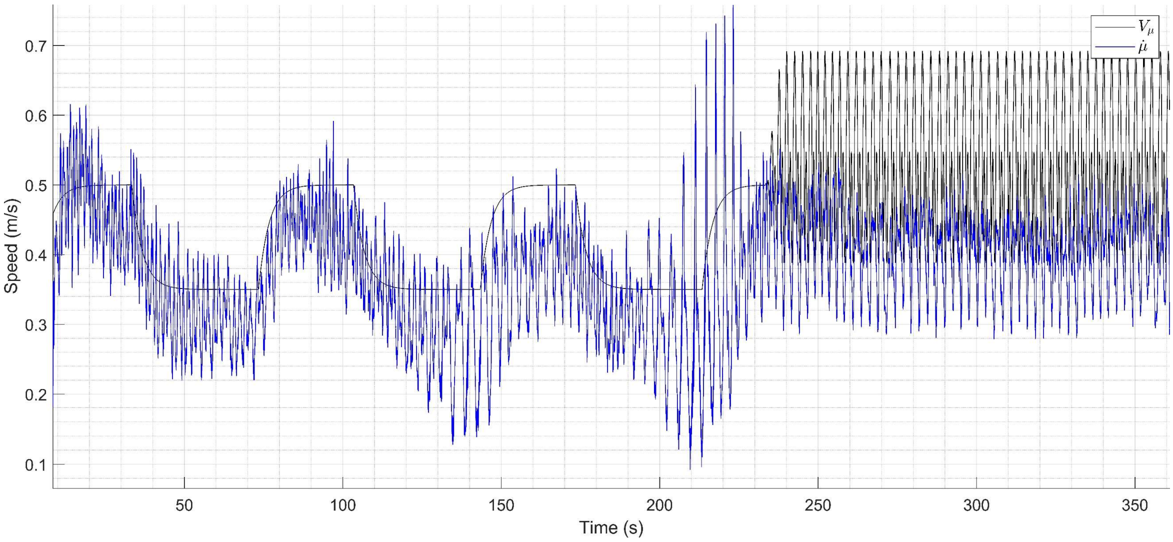

Figure 4.6: The desired speed against the projected travel speed along the parameterized contour during the second experimental trial for the length of the experiment (360s).

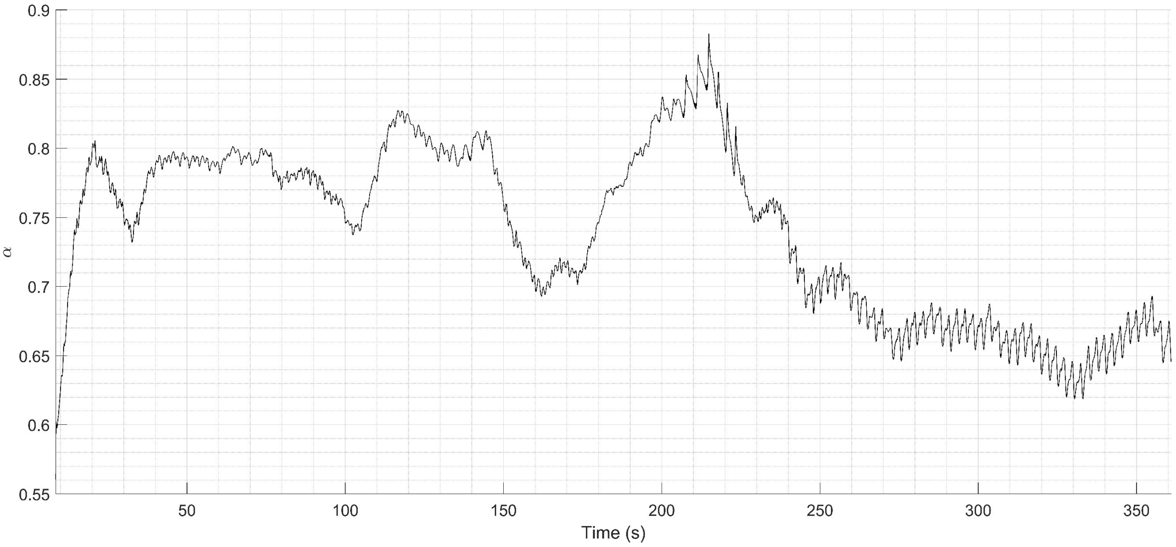

Figure 4.7: The tracked scalar multiple (α) of the reference joint velocities during the second experimental trial plotted for the length of the experiment ($360s$). Representative of the energy of the augmented system.

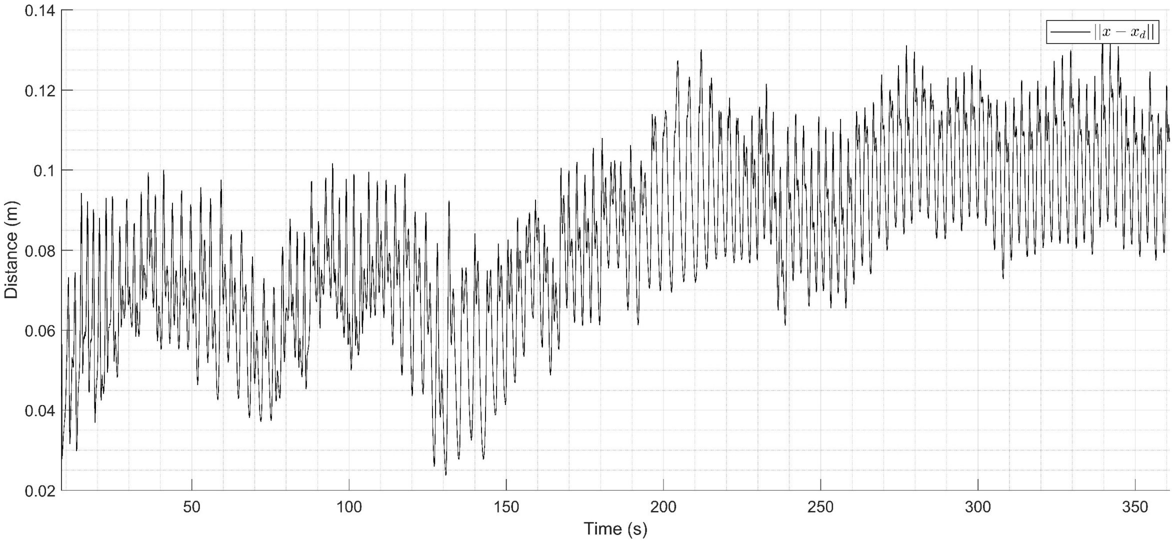

Figure 4.8: Euclidean distance between the end effector pose in task space and the closest point along the desired trajectory observed during the second experimental trial plotted for the length of the experiment ($360s$).

The following figures display data from the second experimental trial which correspond to a single ‘cycle’ of the reference training velocity (V_{low} → transition → V_{high}) during the time period of $172s$ to $236s$. These are followed by figures that are associated with the identified force-velocity-position characteristics of the user as well as power output during the control phase.

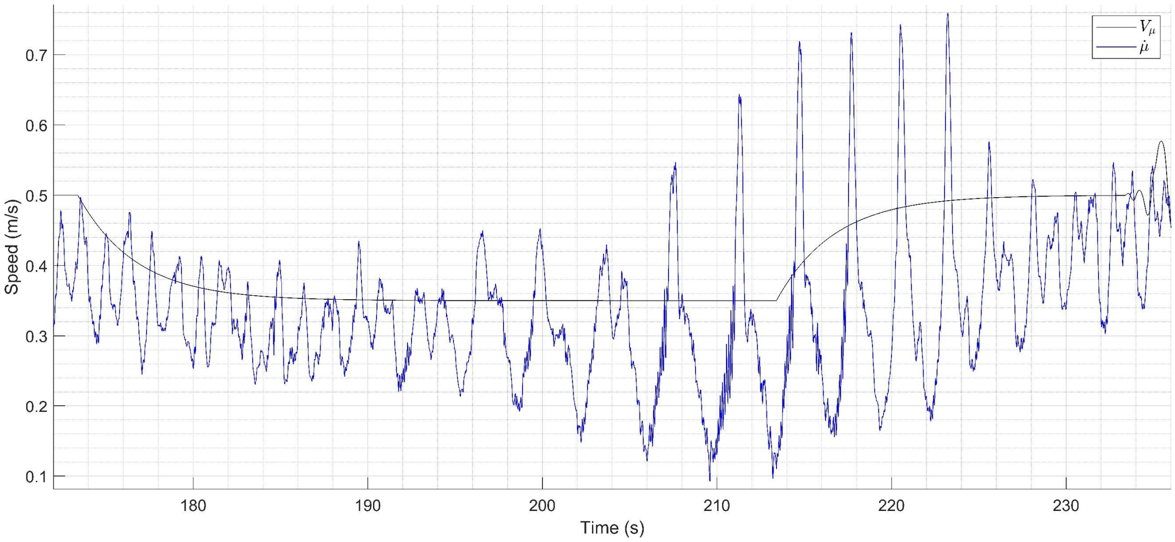

Figure 4.9: The desired speed plotted against the projected travel speed along the parameterized contour during the second experimental trial. For the length of a single ‘cycle’ of the reference training velocity from $172s$ to $236s$.

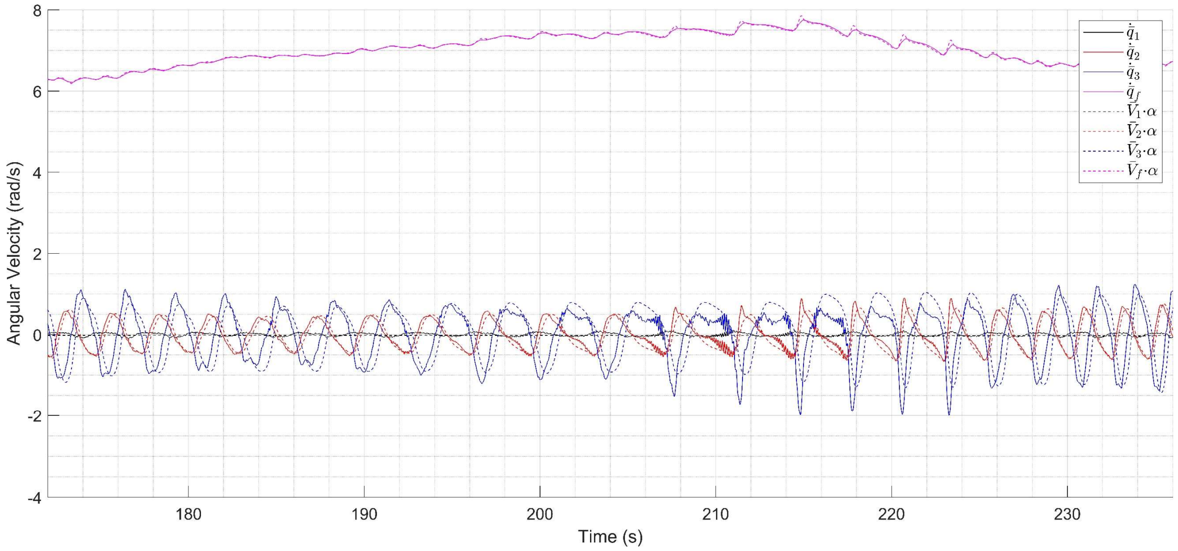

Figure 4.10: Desired augmented joint velocities scaled with α and plotted alongside the augmented joint velocities to show tracking ability during the second experimental trial. Plotted for the length of a single 'cycle' of the reference velocity from $172s$ to $236s$.

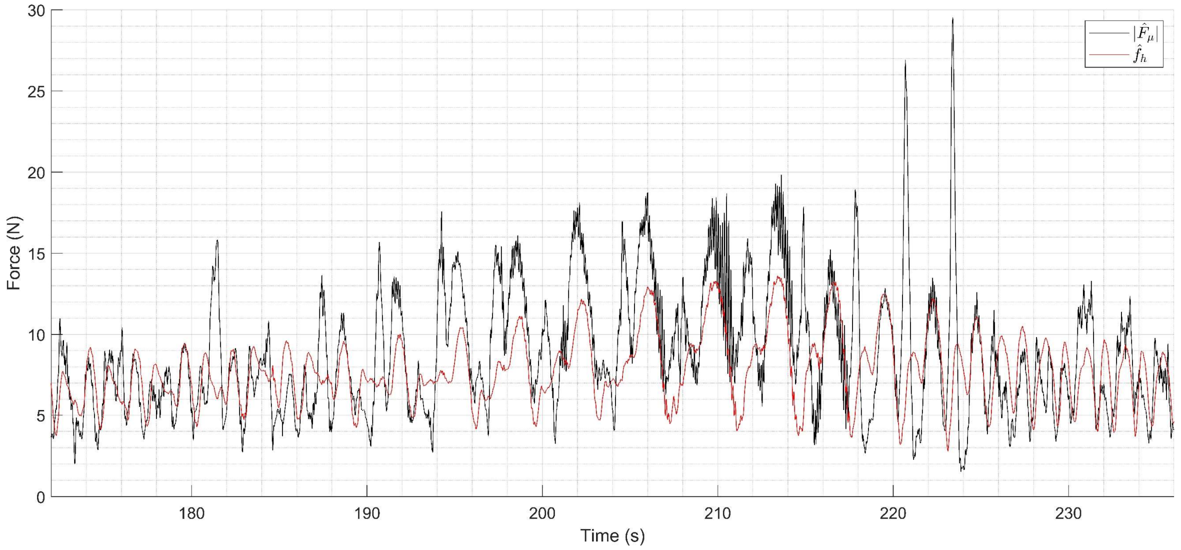

Figure 4.11: The estimated user applied Hill force plotted against the absolute value of the projection of the force observer output along μ. Illustrates the degree at which the estimate of the user applied force is able to characterize the force observer output. For the second experimental trial during the length of a single 'cycle' of the reference velocity from $172s$ to $236s$.

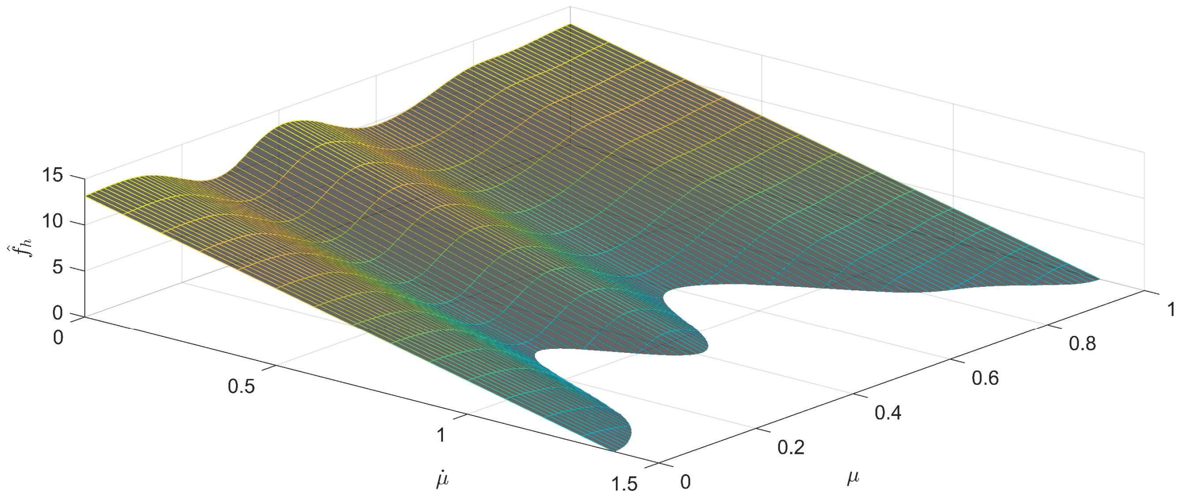

Figure 4.12: The estimated Hill surface along the desired trajectory μ that generates the estimate of the user applied force. Evaluated at the end of the second experimental trial.

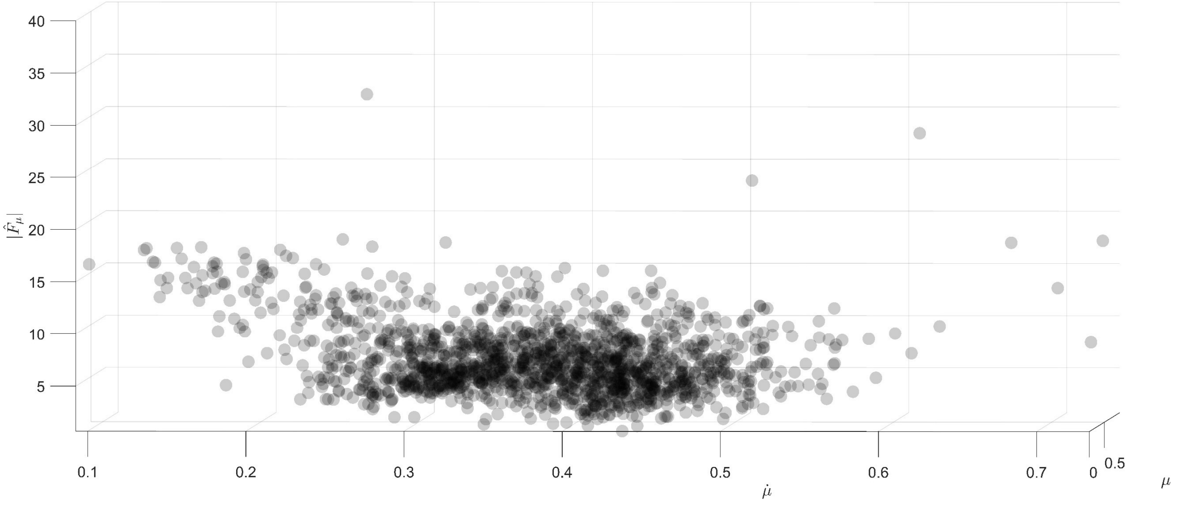

Figure 4.13: 3D scatter plot of force-velocity-position data that is representative of the user's true Hill surface along μ (subject to model uncertainties and deviations from μ) where the user generated force is approximated by the absolute value of the projection of the force observer output along μ. The data was collected over the course of the second experimental trial for its full duration of $360s$.

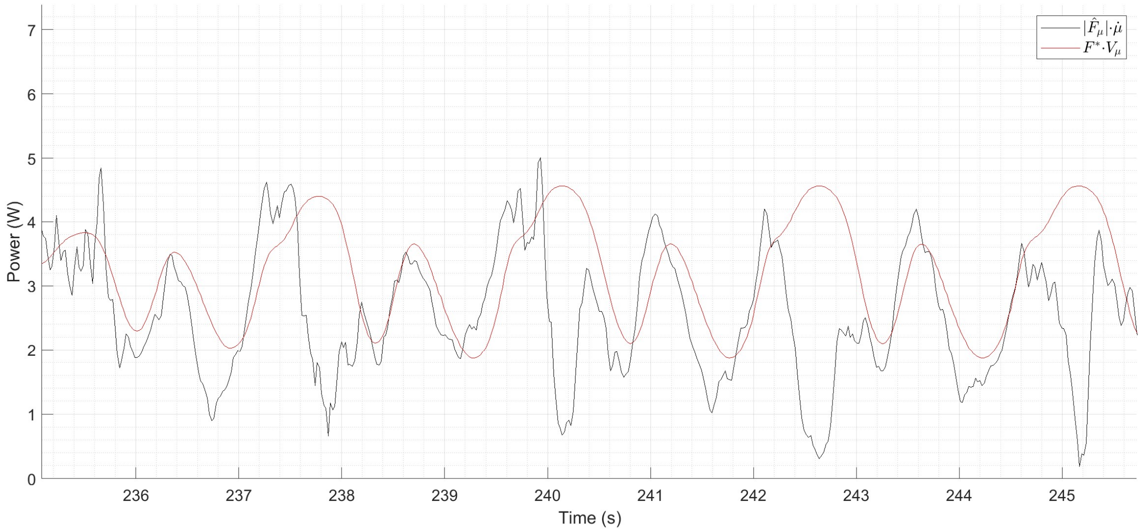

Figure 4.14: The user's power output along the desired contour (parameterized by μ) plotted against the actual power output along μ during the initial regime of the control phase (from $240s$ to $250s$) of the second experimental trial.

The user's ability to track the desired contour (given by x_d) is observed to progressively decline during the training phase ($0s$ to $234s$). The maximum value of $||x - x_d||$ during the training phase is given by $0.07277m$ which represents 36.38% of r_c. During the control phase, although oscillating, the average value of the contour tracking error remains approximately constant at $0.09923m$ which represents 49.62% of r_c with a maximum value of $0.135008m$ which represents 67.50% of r_c.

The reference speed over the course of the second experimental trial is sufficiently tracked with an MAPE of 0.28987%, however, underdamped convergence behavior (represented by oscillations of $\dot{\mu}$ about V_μ) is observed as in the first experimental trial (as can be seen in Figure 4.6. Additionally, in the training phase, the ability of the user to track the high reference speed (given by V_{high}) differs from their ability to track the low reference speed (given by V_{low}). This phenomenon is illustrated in Figure 4.9 (during an isolated cycle). The MAPE of the speed tracking task (that is given by $|\dot{\mu} - V_\mu|$) is 0.28795% when the reference speed is given by V_{low} and 0.22777% when the reference speed is given by V_{high} through the duration of the training phase. The tracking ability during the training phase can also be quantified by the mean values of $|\bar{e}_\alpha|$ (measure of joint velocity tracking error) that are given by 0.04120, 0.12541, 0.38896, 0.01394 m/s corresponding to $|\bar{e}_{\alpha_1}|$, $|\bar{e}_{\alpha_2}|$, $|\bar{e}_{\alpha_3}|$, $|\bar{e}_{\alpha_f}|$ respectively. Similarly, the tracking ability during the control phase can also be quantified by the mean values of $|\bar{e}_\alpha|$ that are given by 0.04174, 0.11565, 0.40113, 0.01832 m/s corresponding to $|\bar{e}_{\alpha_1}|$, $|\bar{e}_{\alpha_2}|$, $|\bar{e}_{\alpha_3}|$, $|\bar{e}_{\alpha_f}|$ respectively.

From Figure 4.11 we observe that the estimated user applied force is representative of the force observer output with a MAPE of 0.52697% through the duration of the experiment. However, it is also observed that the estimation of the user applied force (represented by the output of the force observer) shows greater predictive value at low reference speeds than at high reference speeds (as can be seen in Figure 4.11). The MAPE of the force estimate is 0.30793% when the reference speed is given by V_{low} and 0.41208% when the reference speed is given by V_{high} through the duration of the training phase.

4.4 Discussion

This section will discuss the findings presented in the results section in relation to expected behavior of the controller (mentioned in Chapter 3) and observed differences.

4.4.1 Passive Velocity Field Control

PVFC is able to sufficiently track the reference velocity field and successfully achieves $e_\alpha \rightarrow 0$. The inability to obtain better tracking is due to friction, a limitation on the maximum stable value of the gain factor (γ), and most notably, model uncertainties.

During the period of time in which the user is stably adding energy to the system the velocity of the flywheel increases (Figure 4.1). This is to be expected as PVFC successfully maintains the kinetic energy of the manipulator system such that the desired reference velocities can be sufficiently tracked. On the other hand, during the period of time in which there is no interaction force from the user the controller reduces the energy stored in the flywheel in order to counteract dissipative forces and track the desired velocity field. These dissipative forces are a result of friction and model uncertainties. As expected, superior tracking is achieved when there is no user interaction (as this acts as a disturbance to PVFC).

Instability in the generated PVFC input torques is observed (refer to Figure 4.3) with increasing joint velocities. Although the controller's ability to withstand disturbances improves as the amount of energy in the system increases, the effects of model uncertainties grow in a quadratic manner with $\dot{\bar{q}}$. Also, this instability is partially due to the stabilizing feedback term being quadratic with $\dot{\bar{q}}$ which entails a greater presence of noise in the control input.

Model parameter uncertainties, unmodelled dynamics and friction are observed to be significant sources of error and limit the tracking ability of the controller. Specifically, it is observed that joint three exhibits the greatest tracking error and appears to undershoot the reference (Figure 4.1). This behavior is typical of significant parameter uncertainties of the third link resulting in imprecise tracking exhibited by the inverse compensation mechanic in PVFC. Due to this, the true shape of the path traced by the end-effector is slightly elliptical. Also, since model uncertainties are considered by the controller as environmental forces, the control input torques (Figure 4.3) appear in the force observer output (Figure 4.4) even when no user interaction is present.

During manual tuning of the various parameters, it is observed that the stable range of γ is exceptionally narrow ($\gamma \in [0.09, 0.2]$). Low values of γ result in a divergence of the coupling control torques ($\bar{\tau}_c$) due to the lack of stabilization. At higher values, the system exhibits unstable 'shaky' motion of the end effector due to amplified model uncertainties and the dependance of the feedback term on noisy joint velocities. This presents a limitation as increasing this value will improve the controller's ability to track the reference joint velocities. The largest possible value of γ that resulted in sufficient tracking while ensuring that the contact with the user is smooth and contains minimal vibrations was chosen ($\gamma = 0.18$). The importance of assigning appropriate values to γ as well as to E_{max} is discussed in Section 3.2.

4.4.2 Smart Exercise Machine Controller

The controller achieves sufficient tracking of the scalar multiple (α) of the reference velocity (such that $\dot{q} \rightarrow \alpha \bar{V}_q$, as can be seen in Figure 4.10). However, this is limited by the presence of friction, model uncertainties and unmodelled dynamics (mentioned in Subsection 4.4.1) that cause the controller to oscillate about the reference (Figure 4.6). The same behavior is observed with respect to adhering to the desired contour (Figure 4.8) and is also observed in the first experimental trail. The error associated with tracking the reference velocity and the desired circular contour increases as the Hill parameters are learned due to the increased difficulty in executing the task with injected damping. Decreased task tracking performance could also be attributed to increased fatigue and decreased effort level as the experiment progresses.

The adaptive component of the smart exercise scheme effectively learns the force characteristics exhibited by the force observer to achieve online regression that results in a smooth Hill surface (refer to Figure 4.12). The estimate of the user applied force generated by this surface is able to characterize the force observer output (Figure 4.11). However, force-velocity-position data (illustrated in Figure 4.13) appears to contain outliers that deviate from the expected linear model at large values of $\dot{\mu}$. Thus, in these high velocity regions it is observed that the force estimate is less accurate. This is primarily due to a greater difficulty in tracking the task at high velocities that is exhibited by the user, where the user appears to deviate from the contour and oscillate to a greater extent about the reference velocity in these regions. The task is also observed to become less repetitive in the sense that deviations from the task vary between cycles. This intensifies as time progresses and is more significant at high velocities. Additionally, there is a greater presence of filter delay as well as amplification of friction and model uncertainties at higher velocities. Another reason for this deviation from the expected linear profile at large values of $\dot{\mu}$ could be due to the true multi-joint Hill model exhibiting hyperbolic behavior. As mentioned in Subsection 1.1.3 and addressed in the founding paper [15], the linear Hill model is only expected to remain valid about the region of maximal power output. These erroneous high magnitude outliers are learned with greater weighting than true user force characteristics exhibited at large values of $\dot{\mu}$ (due to their larger magnitude), resulting in the estimated Hill force at high speeds to be greater than what is suggested by collected force-velocity-position data that is non-erroneous (Figure 4.13).

When $e_{opt} < e_{opt\ bound}$ at transition time $T_k = 234s$, the optimal reference joint velocities are commanded. At this time the estimate of the optimal velocity field is instructed, and the user is forced to exercise according to the maximum power criteria (Figure 4.14). As the Hill surface approximately characterizes the environmental force at this time, the controller forces the user to exercise near the desired power output level. This is achieved to a greater extent at lower travel speeds along the desired contour and is limited by the accuracy of the Hill surface.

Variations in tracking ability of the desired speed along μ and the ability to adhere to the circular contour are observed. This means that the user cannot maintain the exact same exercise motion throughout, thus, the true Hill surface varies. It is expected that the desired power output level will be more representative of the optimal with further training. The ability to train further, however, is naturally limited by the users increasing level of fatigue and possibly decreasing effort level with continued exercise.

4.5 Limitations and Drawbacks

Although proven to be effective in identifying a patient's Hill parameters and utilizing these parameters to maximize the patient's caloric consumption, the experiment is limited not only by its specific objective, but also by its dependency on effective execution by the user. Apart from this, limitations of the hardware and the experimental method also present sources of error. In this section, the limitations observed through experimentation are addressed.

In haptic experiments, it can often be difficult for the user to comprehend the task. This is even more true for tasks that involve motion in 3D space and are not based only on position (such as point to point movement or timed trajectory tracking). For this reason, visual or auditory (or both) feedback is often used to indicate to the user the degree at which they are adhering to the task, such that the user can adapt during the experiment and quickly learn the task. This is especially true if the user begins to deviate from the task due to reasons such as fatigue and decreased motivation. However, this experiment lacks an immersive game type interface. This increases the difficulty of understanding the task potentially resulting in decreased user performance. Although the habituation phase serves the purpose of familiarizing the user with the task prior to running the experiment, it is observed that the user's performance in executing the task decreases with time. This may indicate that the user 'forgets' the task or may lose focus on sufficiently tracking the task due to exercise fatigue.

As previously mentioned (throughout Chapter 3), the successfulness of the experiment depends largely on the user's ability to execute the desired task. Generally, it is expected that some users will have a greater ability to adhere to the task due to the fact that they may be familiar with haptic environments, have naturally superior hand eye coordination when compared to other users, or exhibit superior athletic ability. Other users may find the task unnatural. It is also plausible that some patients may be incapable of exercising consistently with sufficient execution of the task due to their physical condition (such as recovering from injury). Thus, it is expected that the proposed exercise controller will not apply equally to all patients and may not be applicable to some.

There are several limitations that arise from the decision to use PVFC as the 'base' for the smart exercise framework presented in this thesis (briefly mentioned in Section 3.2). Although the PVFC component of the proposed smart exercise scheme 'guides' the user towards the desired contour, the controller may not achieve perfect tracking of the task (as observed in the experimental results, Section 4.3). Relying primarily on force feedback to indicate the task results in the user tracking the motion achieved by the PVCF torques, rather than the true task. This is observed in the experiments conducted in this thesis and is the primary reason that a prior experiment is conducted to evaluate the effectiveness of the PVFC controller in executing the task. PVFC also requires the robot's precise model to realize the nonlinear feedback control (Equation (3.27)). An approximate model is sufficient as model parameter uncertainties are treated by the PVFC algorithm as environmental forces (which it still maintains passivity with respect to), however, this entails that significant robot model inaccuracies can negatively impact the controller's robustness and result in decreased performance. Furthermore, dynamics that are not accounted for appear in the force observer output (elaborated on in the following paragraph). Apart from this, it is shown in [81] that the controller's ability to withstand disturbances improves as the amount of energy in the system increases. This entails that PVFC may not be a suitable choice for a controller when applied to low velocity tasks and lightweight manipulators. Additionally, the controller of a robot should be designed to absorb the environmental uncertainties and act softly with small contact force as not to injure the environment (containing the user). This being said, it is difficult to design the contact performance (i.e. mechanical impedance) of a PVFC-controlled robot (this is addressed in [34]). This may result in discontinuous contact between the robot and its environment such that the robot keeps 'punching' the environment [88]. This behavior was observed during the tuning of the values of γ and E_{max}, where large values of γ and small values of E_{max} exhibit underdamped 'shaky' behavior when environmental forces were introduced. Although the PVFC algorithm is designed to maintain passivity as to not injure the user, there is potential for the controller to diverge if stability and robustness criteria are not met. For example, slacking that leads to a sudden deviation from the velocity field can result in divergence as this produces spiking control input torques.

Another significant source of error encountered in the experiment concerns the nature of the force observer. The force observer is prone to uncertainty in the dynamic model such that the controller's input torques (and the unmodelled dynamics themselves) appear in the force observer's output. This is true for all environmental forces (such as friction) that are not accounted for in the robot dynamics model. Due to this, experiments of a similar nature should consider the use of a force sensor, which would result in the Hill surface being more representative of the user's biomechanical characteristics. This being said, the identified Hill surface still holds value in applications towards generating exercise strategies to satisfy power-based objectives (such as in this thesis), although it may not be representative of the user's biomechanics.

The exercise scheme presented in this thesis attempts to achieve the maximal power output objective. This objective requires the patient to have sufficient mobility such that they can exert power throughout an exercise motion, and thus, limits the target population to sufficiently mobile patients requiring strength training. Additionally, this may not result in a realistic rehabilitation exercise and designing exercises based on some power criteria may not provide for valuable exercise strategies. The controller does not directly shape the resistance force profile as it is automatically set by the criterion of maximizing the user's caloric consumption. For example, upon identifying an area of weakness within the exercise motion, that area is assigned a high reference control velocity and low opposing damping force. Such a reference task may not be desirable if motion should be limited in this identified area of weakness. This being said, the learned Hill surface can be used to generate a variety of power-based exercises.

The hardware also presents limitations with respect to its physical design and components. Joint velocity sensor data was found to contain noise (refer to Section 2.2). This is the primary source of instrumental error. Apart from this, the manipulator is relatively lightweight and this presents a difficulty in implementing the controller. For lightweight manipulators, such as the cable driven BURT platform (designed to reduce inertial affects), even at relatively high velocities the manipulator contains a low amount of kinetic energy. This decreases the effectiveness of the PVFC algorithm, as it exhibits decreased stability at low levels of kinetic energy (of the augmented system). Thus, for lightweight manipulators PVFC is not ideal because stability and robustness properties improve with increasing kinetic energy of the system (see pg. 1356 and 1357 of [81]).

Chapter 5
Conclusions, Recommendations and Future Work

5.1 Conclusions

This study presented a novel implementation of a self-optimizing control system on an active 3 DOF upper-limb robotic manipulator system that includes friction, actuator/sensor noise, and unmodelled dynamics. The control system generates reference velocity signals to ensure that sufficient information exists such that a biomechanical model of the user's force-velocity-position characteristics can be identified. This biomechanical model takes the form of a linear Hill surface which is initially unknown and is identified online using a discretized learning scheme. Following identification of the unknown Hill parameters, appropriate damping is injected and an optimal reference velocity is commanded such that the user is forced to exercise according to a maximum power output criteria. The novel damping matrix proposed allows for regulation of the systems energy while maintaining passivity of the closed-loop system. Experimental results conducted on a unimanual setup of the BURT manipulator with a single subject indicate that the control algorithm was able to identify the unknown Hill parameters and approximately command the optimal exercise strategy with limitations, thus, demonstrating the efficacy and utility of the proposed control algorithm.

5.2 Recommendations and Future Work

As discussed in Section 4.5, there exist limitations to the control algorithm proposed and experiment conducted in this thesis suggesting more room for improvement. In this section, suggestions to future work that improve on or carry forward this work are addressed.

The smart exercise controller could be made more robust by improving the estimate of the joint velocities. Currently, phase lag in the velocity estimate from low-pass filtering leads to instabilities in the PVFC control inputs, the output of the force observer, and the injected damping. Noise that remains in the signal adversely affect the stability of the proposed controller. The use of external sensors or improved joint position signal processing would result in better parameter estimates and improved stability as well as robustness of the controller. This would allow for a wider range of reference velocities to be tracked effectively, giving greater freedom with respect to assignable parameters (such as $\gamma, E_{max},$ etc.), greater accuracy of the identified Hill surface and improved safety.

As mentioned briefly in Subsection 1.1.2, although this thesis focuses on implementing the original PVFC scheme to achieve the velocity field tracking objective, many useful extensions presented by papers following the original concept also show potential. Since friction is known to limit the stability and robustness of the controller [81], extending the controller to incorporate friction compensation (addressed in [37]) would improve results. The existence of model uncertainties negatively impacts stability and robustness properties of the controller as forces coming from model uncertainties are treated as environmental disturbances. Thus, the effectiveness of the controller is affected by the accuracy of robot model parameters. Implementing an adaptive algorithm to identify the manipulators dynamics with greater accuracy (such is the case in [89]) has the potential to improve the controller's effectiveness in tracking the desired task with more desirable stability and robustness properties.

The results are limited by the chosen experimental method (mentioned in Section 4.5). It is known in literature that effective visual cues are able to assist an individual in a manual control task [90]. However, as mentioned in Section 4.2, no immersive interface was used in the experiment. This results in the user paying less attention to the degree at which they are successfully executing the task as the experiment progresses. This is especially true as the user becomes fatigued due to maintaining exercise intensity throughout the experiment, which in turn results in less attention being paid towards a non-captivating interface. Thus, results can be improved by implementing a more immersive interface that could potentially indicate the user's position in 3D space with respect to the desired contour as well as the user's task space velocity with respect to the desired task space velocity. If such an interface proves to be immersive enough to captivate the user and inspire continuous focus, additional information can also be displayed on the interface.

The biomechanical model used to model the user interaction force is a linear Hill-like curve. This is chosen due to its simplicity, that in turn simplifies the adaptation algorithm, However, with a wider range of velocities the Hill-like model is expected to exhibit force-velocity non-linearity. This behavior has been observed in similar studies involving multi-joint exercise motions (mentioned in Subsection

1.1.3). Thus, a possible extension to this work is to adapt the proposed control scheme to consider alternate Hill-like force-velocity relationships and other biomechanical models. Additionally, the Hill-like curve is fit to a lumped measure of the user interaction force. Thus, there is no knowledge of which muscle groups are active (muscle activations) during the task. Such information can only be provided by EMG (or subcutaneous) sensors. This is the approach suggested for future work to ensure that muscle activations are sufficiently repetitive, and, to give insight into which muscle groups are active during a specified exercise task.

Apart from the potential applications of the proposed smart exercise controller within the field of rehabilitation robotics, the concept of identifying a user's physical capabilities and applying this to haptic environments (in the form of virtual reality games with force feedback) is a new and emerging field of research. This enables creators of such environments to provide an experience uniquely tailored towards each user. For example, a potential application of a user's known force-velocity-position characteristics could be to provide game difficulty that scales uniquely to each user. Possible applications of the specific controller investigated in this thesis would include game tasks defined by repetitive trajectories such as a pick and place game where trajectory variation is minimal. The application of learning algorithms in haptic environments presents an interesting potential extension of work such as the controller proposed in this thesis.

References

[1] P. A. Wolf and J. C. Grotta, “Cerebrovascular disease.,” *Circulation*. 2000. doi: 10.1161/01.cir.102.suppl_4.iv-75.

[2] K. T. Khaw, “Epidemiology of stroke,” *Journal of Neurology Neurosurgery and Psychiatry*. 1996. doi: 10.1136/jnnp.61.4.333.

[3] C. J. Shatz, “The developing brain,” *Scientific American*, 1992, doi: 10.1038/scientificamerican0992-60.

[4] “Post-Stroke Rehabilitation,” *Journal of the American Academy of Nurse Practitioners*, 1995, doi: 10.1111/j.1745-7599.1995.tb01128.x.

[5] D. T. Wade and R. Langton Hewer, “WHY ADMIT STROKE PATIENTS TO HOSPITAL?,” *The Lancet*, 1983, doi: 10.1016/S0140-6736(83)91861-5.

[6] J. Johnson, V. Pearson, and L. McDivitt, “Stroke rehabilitation: assessing stroke survivors’ long-term learning needs.,” *Rehabilitation nursing : the official journal of the Association of Rehabilitation Nurses*, 1997, doi: 10.1002/j.2048-7940.1997.tb02110.x.

[7] L. Ferrucci *et al.*, “Recovery of functional status after stroke a postrehabilitation follow-up study,” *Stroke*, 1993, doi: 10.1161/01.STR.24.2.200.

[8] J. Heinzmann and A. Zelinsky, “Quantitative Safety Guarantees for Physical Human-Robot Interaction,” *The International Journal of Robotics Research*, 2003, doi: 10.1177/02783649030227004.

[9] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, “Safety analysis for a human-friendly manipulator,” *International Journal of Social Robotics*, 2010, doi: 10.1007/s12369-010-0053-z.

[10] A. van der Schaft, *L2-Gain and Passivity Techniques in Nonlinear Control*. 2000.

[11] S. Stramigioli, “Energy-Aware robotics,” 2015. doi: 10.1007/978-3-319-20988-3_3.

[12] T. S. Tadele, T. De Vries, and S. Stramigioli, “The safety of domestic robotics: A survey of various safety-related publications,” *IEEE Robotics and Automation Magazine*. 2014. doi: 10.1109/MRA.2014.2310151.

[13] L. L. Cai *et al.*, “Implications of assist-as-needed robotic step training after a complete spinal cord injury on intrinsic strategies of motor learning,” *Journal of Neuroscience*, 2006, doi: 10.1523/JNEUROSCI.2266-06.2006.

[14] L. Marchal-Crespo and D. J. Reinkensmeyer, “Review of control strategies for robotic movement training after neurologic injury,” *Journal of NeuroEngineering and Rehabilitation*. 2009. doi: 10.1186/1743-0003-6-20.

[15] P. Y. Li and R. Horowitz, “Control of smart exercise machines-part I: Problem formulation and nonadaptive control,” *IEEE/ASME Transactions on Mechatronics*, 1997, doi: 10.1109/3516.653048.

[16] P. Y. Li and R. Horowitz, “Control of smart exercise machines-part II: Self-optimizing control,” *IEEE/ASME Transactions on Mechatronics*, 1997, doi: 10.1109/3516.653049.

[17] A. Erdogan, A. C. Satici, and V. Patoglu, “Passive velocity field control of a forearm-wrist rehabilitation robot,” 2011. doi: 10.1109/ICORR.2011.5975433.

[18] J. Zhang and C. C. Cheah, “Passivity and Stability of Human-Robot Interaction Control for Upper-Limb Rehabilitation Robots,” *IEEE Transactions on Robotics*, 2015, doi: 10.1109/TRO.2015.2392451.

[19] W. E.T., C. V., R. D.J., and B. J.E., “Optimizing compliant, model-based robotic assistance to promote neurorehabilitation,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2008.

[20] J. Nikitczuk, B. Weinberg, and C. Mavroidis, “Control of electro-rheological fluid based resistive torque elements for use in active rehabilitation devices,” *Smart Materials and Structures*, 2007, doi: 10.1088/0964-1726/16/2/021.

[21] S. Dong, K. Q. Lu, J. Q. Sun, and K. Rudolph, “Rehabilitation device with variable resistance and intelligent control,” *Medical Engineering and Physics*, 2005, doi: 10.1016/j.medengphy.2004.09.009.

[22] C. Adans-Dester, A. O’Brien, R. Black-Schaffer, and P. Bonato, “Upper Extremity Rehabilitation with the BURT Robotic Arm,” *Archives of Physical Medicine and Rehabilitation*, 2019, doi: 10.1016/j.apmr.2019.10.148.

[23] R. Colombo *et al.*, “Robotic techniques for upper limb evaluation and rehabilitation of stroke patients,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2005, doi: 10.1109/TNSRE.2005.848352.

[24] X. T. Zhang, D. M. Dawson, W. E. Dixon, and B. Xian, “Extremum-seeking nonlinear controllers for a human exercise machine,” *IEEE/ASME Transactions on Mechatronics*, 2006, doi: 10.1109/TMECH.2006.871896.

[25] S. Chen, J. Yi, and T. Liu, “Strength capacity estimation of human upper limb in human-robot interactions with muscle synergy models,” 2018. doi: 10.1109/AIM.2018.8452447.

[26] R. Ortega and M. W. Spong, “Adaptive motion control of rigid robots: A tutorial,” *Automatica*, 1989, doi: 10.1016/0005-1098(89)90054-X.

[27] M. W. Spong, S. Hutchinson, and M. Vidyasagar, “Robot modeling and control,” *IEEE Control Systems*. 2006. doi: 10.1109/MCS.2006.252815.

[28] R. Ortega and E. García-Canseco, “Interconnection and damping assignment passivity-based control: A survey,” *European Journal of Control*, 2004, doi: 10.3166/ejc.10.432-450.

[29] R. Ortega, A. J. Van der Schaft, I. Mareels, and B. Maschke, “Putting energy back in control,” *IEEE Control Systems Magazine*, 2001, doi: 10.1109/37.915398.

[30] J. C. Willems, “Dissipative dynamical systems part I: General theory,” *Archive for Rational Mechanics and Analysis*, 1972, doi: 10.1007/BF00276493.

[31] J. J. E. Slotine and W. Li, “Adaptive Manipulator Control: A Case Study,” *IEEE Transactions on Automatic Control*, 1988, doi: 10.1109/9.14411.

[32] P. Y. Li and R. Horowitz, “Passive velocity field control of mechanical manipulators,” *IEEE Transactions on Robotics and Automation*, 1999, doi: 10.1109/70.782030.

[33] N. Sadegh and R. Horowitz, “Stability and Robustness Analysis of a Class of Adaptive Controllers for Robotic Manipulators,” *The International Journal of Robotics Research*, 1990, doi: 10.1177/027836499000900305.

[34] Y. Kishi, Z. W. Luo, F. Asano, and S. Hosoe, “Passive impedance control with time-varying impedance center,” 2003. doi: 10.1109/CIRA.2003.1222169.

[35] M. Yamakita, K. Suzuki, X. Z. Zheng, M. Katayama, and K. Ito, “Extension of passive velocity field control to cooperative multiple manipulator systems,” 1997. doi: 10.1109/iros.1997.648974.

[36] M. Yamakita, F. Asano, and K. Furuta, “Passive velocity field control of biped walking robot,” 2000. doi: 10.1109/robot.2000.846492.

[37] J. Moreno and R. Kelly, “Manipulator Velocity Field Control with Dynamic Friction Compensation,” 2003. doi: 10.1109/CDC.2003.1271747.

[38] J. Moreno-Valenzuela, “On passive velocity field control of robot arms,” 2006. doi: 10.1109/cdc.2006.377393.

[39] A. M. Zanchettin, B. Lacevic, and P. Rocco, “Passivity-based control of robotic manipulators for safe cooperation with humans,” *International Journal of Control*, 2015, doi: 10.1080/00207179.2014.956338.

[40] Y. Saitoh, Zhiwei Luo, and K. Watanabe, “Environmental adaptive PVFC for a robot manipulator,” 2003. doi: 10.1109/sice.2002.1195217.

[41] H. J. Asl, T. Narikiyo, and M. Kawanishi, “Neural network velocity field control of robotic exoskeletons with bounded input,” 2017. doi: 10.1109/AIM.2017.8014208.

[42] D. Lee and P. Y. Li, “Passive coordination control of nonlinear bilateral teleoperated manipulators,” 2002. doi: 10.1109/robot.2002.1013732.

[43] D. Lee and P. Y. Li, “Toward robust passivity: A passive control implementation structure for mechanical teleoperators,” 2003. doi: 10.1109/HAPTIC.2003.1191255.

[44] D. Lee and P. Y. Li, “Passive tool dynamics rendering for nonlinear bilateral teleoperated manipulators,” 2002. doi: 10.1109/robot.2002.1013733.

[45] H. A. V., “The heat of shortening and the dynamic constants of muscle,” *Proceedings of the Royal Society of London B: Biological Sciences*, 1979.

[46] J. Alcazar, R. Csapo, I. Ara, and L. M. Alegre, “On the shape of the force-velocity relationship in skeletal muscles: The linear, the hyperbolic, and the double-hyperbolic,” *Frontiers in Physiology*, 2019, doi: 10.3389/fphys.2019.00769.

[47] M. F. Bobbert, L. J. R. Casius, and A. J. Van Soest, “The relationship between pedal force and crank angular velocity in sprint cycling,” *Medicine and Science in Sports and Exercise*, 2016, doi: 10.1249/MSS.0000000000000845.

[48] “The dynamic constants of human muscle,” *Proceedings of the Royal Society of London. Series B - Biological Sciences*, 1940, doi: 10.1098/rspb.1940.0010.

[49] K. A. Edman, “Double-hyperbolic nature of the force-velocity relation in frog skeletal muscle.,” *Advances in experimental medicine and biology*, 1988.

[50] D. J. Hyun, H. Park, T. Ha, S. Park, and K. Jung, “Biomechanical design of an agile, electricity-powered lower-limb exoskeleton for weight-bearing assistance,” *Robotics and Autonomous Systems*, 2017, doi: 10.1016/j.robot.2017.06.010.

[51] D. Paluska and H. Herr, “The effect of series elasticity on actuator power and work output: Implications for robotic and prosthetic joint design,” *Robotics and Autonomous Systems*, 2006, doi: 10.1016/j.robot.2006.02.013.

[52] S. Schmitt, D. F. B. Haeufle, R. Blickhan, and M. Günther, “Nature as an engineer: One simple concept of a bio-inspired functional artificial muscle,” *Bioinspiration and Biomimetics*, 2012, doi: 10.1088/1748-3182/7/3/036022.

[53] S. Jaric, “Force-velocity Relationship of Muscles Performing Multi-joint Maximum Performance Tasks,” *International Journal of Sports Medicine*. 2015. doi: 10.1055/s-0035-1547283.

[54] D. W. Grieve and J. van der Linden, “Force, speed and power output of the human upper limb during horizontal pulls,” *European Journal of Applied Physiology and Occupational Physiology*, 1986, doi: 10.1007/BF00422745.

[55] A. Timothy and T. Hardyk, “FORCE- AND POWER-VELOCITY RELATIONSHIPS IN A MULTI-JOINT MOVEMENT,” no. December, 2000.

[56] S. Arimoto, S. Kawamura, and F. Miyazaki, “Bettering operation of Robots by learning,” *Journal of Robotic Systems*, 1984, doi: 10.1002/rob.4620010203.

[57] L. Cuiyan, Z. Dongchun, and Z. Xianyi, “A survey of repetitive control,” 2004. doi: 10.1109/iros.2004.1389553.

[58] G. Hillerström and K. Walgama, “Repetitive Control Theory and Applications - A Survey,” *IFAC Proceedings Volumes*, 1996, doi: 10.1016/s1474-6670(17)57870-2.

[59] R. W. Longman, “Designing Iterative Learning and Repetitive Controllers,” in *Iterative Learning Control*, 1998. doi: 10.1007/978-1-4615-5629-9_7.

[60] G. Heinzinger, D. Fenwick, B. Paden, and F. Miyazaki, “Stability of Learning Control with Disturbances and Uncertain Initial Conditions,” *IEEE Transactions on Automatic Control*, 1992, doi: 10.1109/9.109644.

[61] T. J. Jang, C. H. Choi, and H. S. Ahn, “Iterative learning control in feedback systems,” *Automatica*, 1995, doi: 10.1016/0005-1098(94)00064-P.

[62] K. L. Moore, *Iterative Learning Control for Deterministic Systems*. 1993.

[63] S. Hara, Y. Yamamoto, T. Omata, and M. Nakano, “Repetitive Control System: A New Type Servo System for Periodic Exogenous Signals,” *IEEE Transactions on Automatic Control*, 1988, doi: 10.1109/9.1274.

[64] W. Messner, R. Horowitz, W. W. Kao, and M. Boals, “A New Adaptive Learning Rule,” *IEEE Transactions on Automatic Control*, 1991, doi: 10.1109/9.67294.

[65] R. Horowitz, W. Messner, and J. B. Moore, “Exponential Convergence of a Learning Controller for Robot Manipulators,” *IEEE Transactions on Automatic Control*, 1991, doi: 10.1109/9.85074.

[66] R. Horowitz, “Learning control of robot manipulators,” *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 1993, doi: 10.1115/1.2899080.

[67] K. Kaneko and R. Horowitz, “Repetitive and adaptive control of robot manipulators with velocity estimation,” *IEEE Transactions on Robotics and Automation*, 1997, doi: 10.1109/70.563643.

[68] D. Jeon and M. Tomizuka, “Learning Hybrid Force and Position Control of Robot Manipulators,” *IEEE Transactions on Robotics and Automation*, 1993, doi: 10.1109/70.246053.

[69] P. Ioannou and B. Fidan, *Adaptive Control Tutorial*. 2006. doi: 10.1137/1.9780898718652.

[70] B. Siciliano and O. Khatib, *Springer handbook of robotics*. 2016. doi: 10.1007/978-3-319-32552-1.

[71] Barrett LLC, “BURT Support Documentation.” Newton, Massachusetts, 2018.

[72] C. Dellin, “Newton-Euler First-Moment Gravity Compensation Newton-Euler Formulation Conventions.” Barrett, pp. 1–15, 2008.

[73] S. Arimoto and F. Miyazaki, “STABILITY AND ROBUSTNESS OF PID FEEDBACK CONTROL FOR ROBOT MANIPULATORS OF SENSORY CAPABILITY.,” 1984.

[74] D. Koditschek, “NATURAL MOTION FOR ROBOT ARMS.,” 1984. doi: 10.1109/cdc.1984.272106.

[75] P. K. Khosla and T. Kanade, “PARAMETER IDENTIFICATION OF ROBOT DYNAMICS.,” 1985. doi: 10.1109/cdc.1985.268838.

[76] F. Pfeiffer, “Parameter identification for industrial robots,” 1995. doi: 10.1109/robot.1995.525483.

[77] C. G. Kang, W. W. Kao, M. Boals, and R. Horowitz, “Modeling identification and simulation of a two link scara manipulator,” 1988.

[78] A. Erdogan and V. Patoglu, “Online generation of velocity fields for passive contour following,” 2011. doi: 10.1109/WHC.2011.5945493.

[79] P. Y. Li and R. Horowitz, “Passive velocity field control (PVFC): Part II - application to contour following,” *IEEE Transactions on Automatic Control*, 2001, doi: 10.1109/9.948464.

[80] G. T. C. Chiu and M. Tomizuka, “Coordinated position control of multi-axis mechanical systems,” *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 1998, doi: 10.1115/1.2805413.

[81] P. Y. Li and R. Horowitz, “Passive velocity field control (PVFC): Part I - Geometry and robustness,” *IEEE Transactions on Automatic Control*, 2001, doi: 10.1109/9.948463.

[82] J. E. Marsden and T. S. Ratiu, “Introduction to Mechanics and Symmetry,” *Physics Today*, 1998, doi: 10.1063/1.2808303.

[83] M. Tomizuka, T. C. Tsao, and K. K. Chew, “Discrete-time domain analysis and synthesis of repetitive controllers.,” 1988. doi: 10.23919/acc.1988.4789842.

[84] M. Tomizuka, K. K. Chew, and W. C. Yang, “Disturbance rejection through an external model,” *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 1990, doi: 10.1115/1.2896180.

[85] C. Kempf, M. Tomizuka, R. Horowitz, and W. Messner, “Comparison of Four Discrete-Time Repetitive Control Algorithms,” *IEEE Control Systems*, 1993, doi: 10.1109/37.248004.

[86] J. J. E. Slotine and W. Li, “Composite adaptive control of robot manipulators,” *Automatica*, 1989, doi: 10.1016/0005-1098(89)90094-0.

[87] P. Y. Li, “Self optimizing control and passive velocity field control of intelligent machines,” *UNIVERSITY of CALIFORNIA at BERKELEY*, vol. 2, pp. 1–16, 1995.

[88] S. Cao and Z. Luo, “On energy-based robust passive control of a robot manipulator,” 2013. doi: 10.1109/sii.2013.6776723.

[89] P. Y. Li, “Adaptive passive velocity field control,” 1999. doi: 10.1109/acc.1999.783145.

[90] A. R. Ferber, M. Peshkin, and J. E. Colgate, “Using kinesthetic and tactile cues to maintain exercise intensity,” *IEEE Transactions on Haptics*, 2009, doi: 10.1109/TOH.2009.22.

Appendix A

BURT System Parameters

$$m_1 = 1.19281201\,kg \tag{5.1}$$

$$Center\ of\ Mass_1 = \begin{bmatrix} -0.00104829 \\ -0.00064515 \\ 0.01317876 \end{bmatrix} m \tag{5.2}$$

$$I_1 = \begin{bmatrix} 0.00599505 & -0.00000102 & 0.00001198 \\ -0.00000102 & 0.00275749 & 0.00001038 \\ 0.00001198 & 0.00001038 & 0.00689899 \end{bmatrix} kg\,m^2 \tag{5.3}$$

$$m_2 = 3.93585733\,kg \tag{5.4}$$

$$Center\ of\ Mass_2 = \begin{bmatrix} 0.11286337 \\ 0.00551130 \\ 0.09481159 \end{bmatrix} m \tag{5.5}$$

$$I_2 = \begin{bmatrix} 0.01282591 & 0.00974708 & 0.0268492 \\ 0.00974708 & 0.20831282 & 0.0013364 \\ 0.0268492 & 0.0013364 & 0.20404863 \end{bmatrix} kg\,m^2 \tag{5.6}$$

$$m_3 = 1.42108209\,kg \tag{5.7}$$

$$Center\ of\ Mass_3 = \begin{bmatrix} 0.15846971\ \mathrm{m} \\ -0.03529706\ \mathrm{m} \\ 0.03577011\ \mathrm{m} \end{bmatrix} m \tag{5.8}$$

$$I_3 = \begin{bmatrix} 0.00681472 & -0.01201551 & -0.00099897 \\ -0.01201551 & 0.03901587 & 0.00118192 \\ -0.00099897 & 0.00118192 & 0.04394678 \end{bmatrix} kg\,m^2 \tag{5.9}$$

$$Ja2ma = \begin{bmatrix} 8.51 & -15.19 & 0 \\ -8.51 & -15.19 & 0 \\ 0 & 0 & -9.57002 \end{bmatrix} \tag{5.10}$$

Table A: Viscous friction and inertia properties of motors (manufacturer provided)

	$I_{motor}\ (kg\ m^2)$	$F_{(viscous)motor}\ (N\ m\ s)$
Motor 1	0.0018685481	0.137482
Motor 2	0.0018933111	0.115084
Motor 3	0.0004372562	0.224504