12-2006

# HAPTICS IN ROBOTICS AND AUTOMOTIVE SYSTEMS

Erhun Iyasere
*Clemson University*, oiyaser@clemson.edu

# HAPTICS IN ROBOTICS AND AUTOMOTIVE SYSTEMS

---

A Thesis
Presented to
the Graduate School of
Clemson University

---

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Electrical Engineering

---

by
Erhun Iyasere
December 2006

---

Accepted by:
Dr. Darren Dawson, Committee Chair
Dr. John Wagner
Dr. Ian Walker

ABSTRACT

Haptics is the science of applying touch (tactile) sensation and control to interaction with computer applications. The devices used to interact with computer applications are known as haptic interfaces. These devices sense some form of human movement, be it finger, head, hand or body movement and receive feedback from computer applications in form of felt sensations to the limbs or other parts of the human body. Examples of haptic interfaces range from force feedback joysticks/controllers in video game consoles to tele-operative surgery. This thesis deals with haptic interfaces involving hand movements. The first experiment involves using the end effector of a robotic manipulator as an interactive device to aid patients with deficits in the upper extremities in passive resistance therapy using novel path planning. The second experiment involves the application of haptic technology to the human-vehicle interface in a steer-by-wire transportation system using adaptive control.

# DEDICATION

I dedicate this work to my parents, Godwin and Florence Iyasere. This thesis exists because of their support and high expectations.

# ACKNOWLEDGMENTS

I would first like to thank my advisor, Dr. Darren D. Dawson for his support and guidance. I would also like to thank Dr. John R. Wagner for his encouragement and help with my technical writing skills. I would also like to thank Dr. Ian D. Walker for serving on my committee. His enthusiasm about robotics in ECE 693 helped shape my thesis choice and vastly increased my overall technical knowledge.

Special thanks go to my laboratory partner, Ninad Pradhan for his assistance and altruism. My thanks also go to the Clemson Research Team of Jesse Black, Ryan Lusso and Thomas Cash. Their support in time of need was greatly appreciated.

It would be unfair to mention just a few names, but I express my thanks to all my friends at Clemson. Finally, this acknowledgment would be incomplete without thanking my parents, my brother and my sister, whose motivation and support at all times, has made me what I am.

TABLE OF CONTENTS

Table of Contents (Continued)

LIST OF TABLES

LIST OF FIGURES

List of Figures (Continued)

List of Figures (Continued)

List of Figures (Continued)

# CHAPTER ONE

# INTRODUCTION

Haptics has emerged into an innovative technology and has attracted the attention of researchers in the robotics and automotive communities because of the multi-disciplinary nature of its applications. These applications range from force feedback joysticks in video game consoles to driver/vehicle interfaces in steer-by-wire vehicles. Figure 1.1 illustrates the technology sharing that occurs between the robotics and automotive systems. The design and control of robotic systems using haptic technology has created interest due to concepts such as bilateral tele-operated robots and smart exercise machines. Similarly, the design and control of haptic interfaces has created interest due to the evolution of concept hybrid electric and x-by-wire vehicles, whose design demands the fabrication and design of new innovative steering systems to replace the conventional systems that already exist.

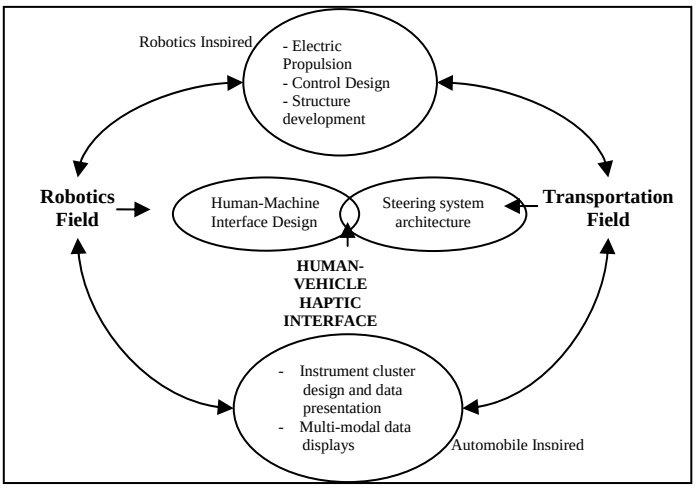


Figure 1.1 Technology sharing between the robotics and automotive disciplines

In this thesis, the application of haptics in robotics is studied with the objective of designing a novel path planning and control framework for passive resistance therapy using a robotic manipulator. In the experiment, a paradigm is presented for safe path generation and control for a robotic manipulator such that it provides programmable passive resistance therapy to patients with deficits in the upper extremities. When the patient applies an interaction force at the robot's end effector, a dynamic path generator time parameterizes any therapist-specified contour in the robot's workspace-thus the robot mimics the dynamics of a passive impedance whose anisotropy vector can be continuously reconfigured. The proposed algorithm is easily implementable because it is robust to uncertainty in the robot dynamics. Moreover, the proposed strategy also guarantees user safety by maintaining the net flow of energy during the human interaction from the user toward the manipulator, thus maintaining passivity.

Also present in this thesis, is a study of the application of haptics in automotive systems with the objective of explaining the dynamic interactions between the human operator and the steering system of an automobile. In this experiment, a full state feedback controller is designed to provide the desired force feedback on the steering wheel to reflect the tire/road interface forces and simultaneously synchronize the motion of the directional control assembly with the motion of the steering wheel. For the force feedback control design, a target system is used to generate the reference signal for the displacement of the primary system. This type of approach is motivated by the impedance control concept detailed in [12]. The controller adapts for parametric uncertainties in the system while ensuring global asymptotic tracking for the "driver experience error" and the "locked error". The target system selected to generate the reference signal is shown have been validated against existing test data, thus increasing the realism of the steer-by-wire steering system to a near emulation

of a conventional steering system. The configurability of the parameters of the target system has the added benefit of changing the steering feel of the steer-by-wire system to suit the operator's need.

## Thesis Organization

This thesis discusses haptic technology in robotics and automotive systems by presenting two studies and simulations. The first chapter consists of an introduction to both studies. Chapter two discusses in detail, the simulation showing the infusion of haptic technology in robotics: A Novel Path Planning and Control Framework for Passive Resistance Therapy with a Robot Manipulator. The simulation results are shown in Chapter three. Chapter four discusses in detail the application of haptic technology in automotive systems: An Adaptive Tracking Controller for a Steer-By-Wire Haptic Interface. The experimental results and a proposed steer–by-wire vehicle configuration are shown in Chapter five. Chapter six concludes this thesis. The programs running the simulation and nomenclature list are listed out in the appendix.

# CHAPTER TWO

# A NOVEL PATH PLANNING AND CONTROL FRAMEWORK FOR PASSIVE RESISTANCE THERAPY WITH A ROBOT MANIPULATOR

Typically, robots are used for simple, repetitive tasks in structured environments isolated from humans. However, the last decade has seen a surge in active research in the area of human robot interaction. Bilateral tele-operated robots [9, 18, 19, 40], smart exercise machines [21, 22], human assist gantry cranes [38], rehabilitation robots [5, 16, 24, 25], and steer-by-wire applications [34, 35] are among the multitude of application areas that drive this research. A common objective of the control algorithm design in all human robot interface applications is to rigorously ensure user safety. Approaches based on passivity ensure that the net flow of energy during the human robot interaction is from the user to the machine [1, 21].

The framework that is created in this study attempts to cast the robot as a reconfigurable passive exercise machine and is inspired by the desire to provide passive resistance therapy to patients affected by dystrophies in the muscles of the upper extremities — these patients need to target specific groups of muscles in order to regain muscle tone [2]. As stated in [2], moderate (sub-maximal) resistance weight lifting, among other treatments, may improve strength in slowly progressive neuromuscular diseases such as Lou Gehrig's Disease (ALS), Spinal Muscular Atrophy, etc. Along any desired curve of motion in 3D space that satisfies a criterion of merit, motion is permitted against a programmable apparent

inertia [17] when the user "pushes" at the end-effector; force applied in all other directions is penalized.

The strategy proposed in this study achieves semi-global asymptotically stable path following for a 3-link revolute robot manipulator in the presence of uncertainty in the robot dynamics. Specifically, given a desired curve of motion that optimizes therapist-established merit criteria, a generator is designed based on an anisotropic force-velocity relationship that generates a bounded desired trajectory in the robot workspace. The inputs into the generator are the patient's interaction force applied at the end-effector as well as the desired impedance parameters (For seminal work done by Hogan on impedance control, the reader is referred to [10]). The reference trajectory generator is carefully designed in order to ensure that the relationship between the patient applied interaction force and the desired end-effector velocity satisfies a passivity constraint. Next, a control strategy is crafted using a Lyapunov based argument in order to obtain the companion objectives of driving the robot end-effector tracking error to zero and ensuring that a filtered error signal nulls out rapidly. This convergence of the filtered error signal allows us to ensure that the interaction of the user with the robot is passive, i.e., energy always flows from the user to the robot manipulator. Additionally, a readily satisfiable mild assumption on the differentiability of the robot dynamics allows us to generate a control strategy that is continuous; this has significant implications in terms of implementability of the control algorithm. As an aside, the control mechanism has the interesting feature of being able learn the unknown robot dynamics.

This chapter is organized as follows. In Section 1, the standard robot task-space dynamics is presented. Section 2 of the chapter presents details of the path generation algorithm. In Section 3, the error systems, measurement constraints, and the assumptions under which the analysis is valid are defined. In Section 4, the design of the control strategy

is presented. Section 5 analyzes the stability of the closed-loop systems in addition to demonstrating the accomplishment of control objectives.

## Robot Dynamics

The end-effector position of a 3-link, revolute direct drive robot manipulator in an inertial frame $I$, denoted by $x(t) \in \Re^3$, is defined as follows

$$x = f(q) \tag{2.1}$$

where $q(t) \in \Re^3$ denotes the link position, and $f(q) \in \Re^3$ denotes the robot forward kinematics. Based on (2.1), the differential relationships between the end-effector position and the link position variables can be calculated as follows

$$\begin{aligned} \dot{x} &= J(q)\dot{q} \\ \ddot{x} &= \dot{J}(q)\dot{q} + J(q)\ddot{q} \end{aligned} \tag{2.2}$$

where $\dot{q}(t)$, $\ddot{q}(t) \in \Re^3$ denote the link velocity and acceleration vectors, respectively, and $J(q) = \dfrac{\delta f(q)}{\delta q} \in \Re^{3\times 3}$ denotes the manipulator Jacobian. The dynamic model for the 3-link robot manipulator is assumed to be in the following form [37]

$$M(q)\ddot{q} + V_m(q,\dot{q})\dot{q} + G(q) = \tau_q + J^T \bar{F} \tag{2.3}$$

where $M(q) \in \Re^{3\times 3}$ represents the inertia matrix, $V_m(q,\dot{q})$ represents the centripetal-Coriolis matrix, $G(q)$ represents the gravity effects, $\bar{F}$ represents the user applied force expressed in $I$, and $\tau_q(t)$ represents the torque input vector.

After utilizing (2.1) and (2.2), one can transform the joint space dynamics into the task-space as follows

$$\bar{M}(x)\ddot{x}+\bar{V}_m(x,\dot{x})\dot{x}+\bar{G}(x)=\bar{\tau}+\bar{F} \tag{2.4}$$

where $\bar{M}(x)=\left(J^T\right)^{-1}MJ^{-1},\bar{V}_m(x,\dot{x})=-\left(J^T\right)^{-1}MJ^{-1}\dot{J}J^{-1}+\left(J^T\right)^{-1}V_mJ^{-1}\in\Re^{3\times3}$ denote, respectively, transformed inertia and centripetal-Coriolis matrices, $\bar{G}(x)=\left(J^T\right)^{-1}G\in\Re^3$ represents gravity effects, $\bar{\tau}(t)=\left(J^T\right)^{-1}\tau_q\in\Re^{3\times3}$ represents the torque input vector expressed in $I$. Motivated by the subsequent stability analysis and control design, the following property is stated:

Property 1: The inertia matrix is symmetric and positive-definite, and satisfies the following inequalities

$$\underline{m}\|\xi\|^2\le\xi^T\bar{M}(\cdot)\xi\le\bar{m}(x)\|\xi\|^2 \qquad \forall\xi\in\Re^3 \tag{2.5}$$

where $\underline{m}\in\Re$ denotes a positive constant, $\bar{m}(x)$ denotes a positive non-decreasing function, while $\|\cdot\|$ denotes the standard Euclidean norm.

## Path Planning and Desired Trajectory Generator

It is well known that stretching, range of motion, and timely surgical correction of spinal deformities may enhance functional use of the extremities for patients with neuromuscular disorders (NMDs). In slowly progressive NMDs, moderate resistance weight lifting is known to improve muscle strength and cardiovascular performance [2]. Motivated by this, a 3-tier path generation and control strategy that is readily implementable on a real robot is presented. The objective is the generation of robot end-effector motion (when pushed by a patient) along a therapist specified path while ensuring that the device behaves as a passive and programmable impedance. The control strategy satisfies the desired properties of (a) guiding the user along contours that provide optimal rehabilitation, (b) generation of

contours that stay away from kinematic singularities, physical joint limits, and obstacles, and (c) time parameterization of the contours in a fashion that conforms to passivity requirements.

## Path Planning: Tier 1

In this thesis, it is assumed that a physical therapist has specified a desired curve of motion $\overline{r}_d(s) \in \Re^3$ given as follows

$$\overline{r}_d(s) = \left[ \overline{r}_{dx}(s) \quad \overline{r}_{dy}(s) \quad \overline{r}_{dz}(s) \right]^T \tag{2.6}$$

where $s \in \Re$ is the length of the curve, while $\overline{r}_{dx}(s)$, $\overline{r}_{dy}(s)$, and $\overline{r}_{dz}(s) \in \Re$ represent the respective coordinates in an inertial frame $I$ (say fixed to the base of the robot). However, this therapist specified contour may not be practicable with a real robot because of joint limits, singularities, and obstacles – one would then like to ensure that the therapist specified path is followed with fidelity until a singularity/joint limit/obstacle is nearby at which instance the robot smoothly veers away from that path and rejoins the original path away from the singularity/joint limit/obstacle.

To that end, one could utilize the virtual potential field concept of Khatib [13] that suggested generation of repulsion functions that grow larger as the robot nears an obstacle and becomes singular at the obstacles. However, a real robot actuator can generate only bounded torques [31]; hence, there is a motivation to design bounded repellers to take care of obstacles. In order to avoid kinematic singularities, the maximization of the Yoshikawa manipulability measure [41] is chosen:

$$\Psi_1(q_d) = \det\left( J(q_d) J(q_d)^T \right) \geq 0 \tag{2.7}$$

where $q_d(s) \in \Re^3$ is a vector of desired robot joint variables, $J(\cdot)$ has been previously introduced in (2.2). For dealing with joint limits, the measure below is chosen

$$\Psi_2(q_d) = \prod_{i=1}^{3} \alpha_i \left(1 - \frac{q_{di}}{q_{di\max}}\right)\left(\frac{q_{di}}{q_{di\min}} - 1\right) \geq 0 \tag{2.8}$$

where $q_{di}$, $q_{di\max}$, $q_{di\min} \in \Re$ denote, respectively, the desired joint angle variable, joint upper, and joint lower limits for the $i^{th}$ joint while $\alpha_i \in \Re$ is a positive constant. In order to avoid obstacles, the measure below is chosen

$$\Psi_3(q_d) = \prod_{i=1}^{n_o} \prod_{j=1}^{3} \left( \left\| r_{dj} - O_i \right\|^2 - R_i^2 \right) \geq 0 \tag{2.9}$$

where $O_i \in \Re^3, R_i \in \Re$ denote the position and the radius of the $i^{th}$ obstacle, $n_o \in \Re$ denotes the number of the obstacles, and $r_{dj} = f_j(q_d)$ where $r_{dj}(s) \in \Re^3$, $j = 1,2,3$ denote the position of the end point of the $j^{th}$ link, and $f_j(\cdot)$ denote the corresponding forward kinematics. The potential function is now defined

$$\Psi(q_d) = \gamma_1 \exp\left( -\gamma_2 \prod_{i=1}^{3} \psi_i(q_d) \right) \tag{2.10}$$

where $\gamma_1$, $\gamma_2 > 0$ are adjustable constants that, respectively, characterize the size and radius of influence of the potential function. This function satisfies the properties of boundedness as well as maximality at the obstacles. By utilizing the virtual field generated by the potential function above, one can dynamically generate a modified contour $r_d(s) \in \Re^3$ as follows

$$r_d'(s) = -\gamma_3 \left[ r_d(s) - \overline{r}_d(s) \right] - \gamma_4 \nabla \psi \left( f^{-1}(r_d) \right) + \overline{r}_d'(s) \tag{2.11}$$

where the notation $(\cdot)'$ denotes a derivative with respect to $s, \gamma_3, \gamma_4$ are tunable parameters, and $\nabla \psi(\cdot) \in \Re^3$ denotes the gradient vector of $\Psi(\cdot)$. The dynamic equation above acts like

a filter that smoothly drives $r_d(s)$ away from the nominal contour $\overline{r}_d(s)$ near obstacles/singularities/joint limits. In the above equation, $\gamma_3$ provides the rate along $s$ at which the modified contour veers away from (or toward) the original contour when it encounters a change in potential field. The constant $\gamma_4$ is a steady-state constant that amplifies or diminishes the impact of the potential function on changes in the desired contour. The result of this algorithm is a desired contour that avoids singularities, joint limits, and obstacles. Note here that the filtering process of (2.11) renders $s$ an arbitrary parameter that does not necessarily represent the length of the contour $r_d(s)$. Also, the steps involved in Tier 1 are completed offline.

Time Parameterization of Contour $r_d(s)$: Tier 2

In this section, the modified desired contour $r_d(s)$ is time parameterized such that a passivity relation holds between the desired velocity and the applied user interaction force at the robot end-effector. To begin, $\mathcal{F} \triangleq \left(\begin{bmatrix} u(s) & p(s) & b(s) \end{bmatrix}\right)$, known as the Frenet frame, is defined to be a rotating frame associated with the curve $r_d(s)$ such that

$$u(s) = \frac{r_d'(s)}{|r_d'(s)|} \quad p(s) = \frac{u'(s)}{|u'(s)|} \quad b(s) = u(s) \times p(s) \tag{2.12}$$

such that $\Gamma(s) = \begin{bmatrix} u(s) & p(s) & b(s) \end{bmatrix} \in SO(3)$. The relationship between the coordinate frames $\mathcal{F}$ and $I$ is depicted in Figure 2.1.

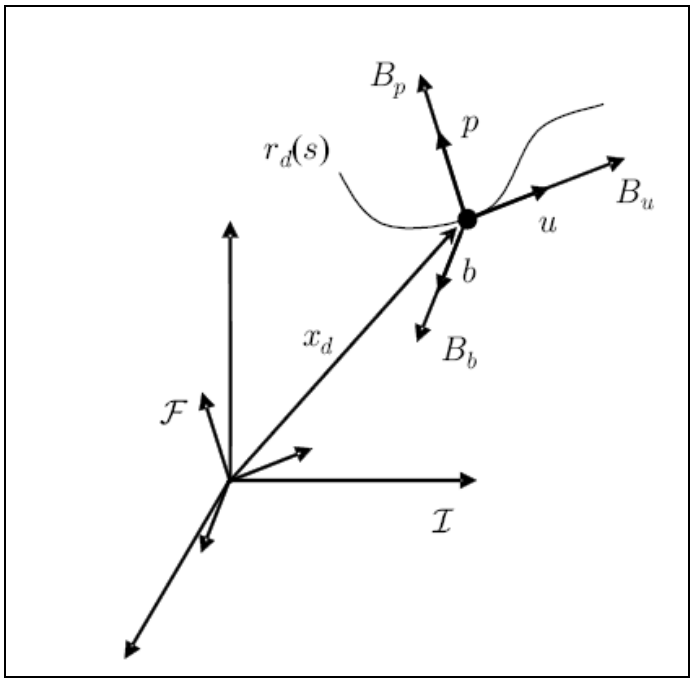


Figure 2.1 Relationship between Inertial and Frenet Frame (Zhang, 2004)

Also, the curvature $\kappa(s)$ and torsion $\tau(s)$ associated with the curve $r_d(s)$ is defined as follows [8]

$$\kappa(s) = \frac{\left| r_d'(s) \times r_d''(s) \right|}{\left| r_d'(s) \right|^3} \quad \tau(s) = \frac{r_d'(s) \cdot \left( r_d''(s) \times r_d'''(s) \right)}{\left| \kappa(s) \right|^2} \tag{2.13}$$

Furthermore, the vector $\omega(s) \triangleq \begin{bmatrix} -\tau(s) & 0 & -\kappa(s) \end{bmatrix} \in \Re^3$ and $\left| \omega(s) \right|_\times$ as the anti-symmetric matrix associated with that vector are defined. Since $(u, p, b)$ define a basis, a general desired velocity vector $\overline{v}_d \triangleq v_{d1} u + v_{d2} p + v_{d3} b = \Gamma v_d$ and an applied user force $\overline{F} \triangleq F_u u + F_p p + F_b b = \Gamma F$ in the inertial frame $I$ are defined where $v_d, F \in \Re^3$ are obviously defined. Since the robot acts as an anisotropic impedance, the direction of which

continuously varies as the desired curve of motion $r_d(s)$, $m_d$ is defined to be a scalar mass parameter and damping coefficients $B_u, B_p, B_b$ are considered along the directions $u, p, b$ such that the damping force $\bar{F}_d$ expressed in $I$ is given as

$$\bar{F}_d \triangleq -B_u v_{d1} u - B_p v_{d2} p - B_b v_{d3} b = -\Gamma B v_d \tag{2.14}$$

where $B \triangleq diag\{B_u, B_p, B_b\}$. By applying Newton's second law to this mass-damper system, the following can be obtained

$$m_d \dot{\bar{v}}_d = \bar{F} + \bar{F}_d$$

which can be written out as follows

$$m_d \dot{v}_d + m_d \dot{s}[\omega]_\times + B v_d = F \tag{2.15}$$

where the formulae of Frenet [15] have been utilized and $\dot{s}(t)$ is yet to be defined. Additionally, the kinematics of the problem can be expressed as follows

$$\dot{x}_d \triangleq v_{d1} u + v_{d2} p + v_{d3} b \tag{2.16}$$

where $x_d(t)$ denotes the time parameterized representation of the desired contour (expressed in the coordinates of $I$) traced by the robot end-effector. Since the intention is for motion to occur along the curve $r_d(s)$, a low tangential damping $B_u$ and very large normal and binormal damping $B_p$ and $B_b$ are designated such that the kinematic constraint is imposed. Under such conditions, the effective motion is governed via the following set of equations

$$\begin{aligned} &\dot{x}_d(t) = v_{d1}(t) u(s(t)) \\ &m_d v_{d1}(t) + B_u v_{d1}(t) = \bar{F}(t) \cdot u(s(t)) \\ &\dot{s}(t) = \frac{|v_{d1}(t)|}{|r_d'(s(t))|} \end{aligned} \tag{2.17}$$

where the first two equations in (2.17) are obtained by applying the kinematic constraint on (2.15) and (2.16) while the last equation expresses the relationship between the time rate of change of the arbitrary parameter $s(t)$ in terms of a known velocity $(v_{d1})$ along the curve $(r_d)$.

Proof of Passivity

In order for a user to exercise safely in conjunction with the robot, the robot must act as a passive device, i.e., the work done by the user force is always positive (minus finite stored initial energy if any). With that objective in mind, it is first demonstrated that there is a passive relationship between the interaction force $\bar{F}(t)$ and the desired end-effector velocity $\dot{x}_d(t)$, i.e.

$$\int_{t_0}^{t} \bar{F}^T \dot{x}_d dt \geq -c_1 \tag{2.18}$$

where $c_1 \in \Re$ is a positive constant, and the fact that $\Gamma^T \Gamma = I_3$ has been utilized. To prove (2.18), a Lyapunov function is defined

$$V = \frac{1}{2} m_d v_d{}^T v_d \geq 0 \tag{2.19}$$

After taking the time derivative of (2.19) along the desired dynamics of (2.15), one obtains

$$\dot{V} = -v_d{}^T B v_d + v_d{}^T F \tag{2.20}$$

where the fact that $\left([\omega]_\times v_d\right) \perp v_d$ has been utilized. After rearranging terms in the above equation and integrating both sides, one can obtain

$$\int_{t_0}^{t} \bar{F}^T \dot{x}_d dt = V(t) - V(t_0) + \int_{t_0}^{t} v_d{}^T B v_d dt \tag{2.21}$$

After utilizing the fact that $V(t), v_d{}^T B v_d \geq 0$, an lower-bound for the left hand side of the above equation is obtained as follows

$$\int_{t_0}^{t} \bar{F}^T \dot{x}_d dt \geq -V(t_0) = -c_1 \tag{2.22}$$

which proves (2.18). In the sequel, the passivity of the robot will be shown by utilizing (2.22) and the yet to be proved $\mathcal{L}_1$ stability property of the end-effector velocity tracking error.

Control Problem Formulation

Given the desired robot end-effector trajectory $x_d(t)$ (obtained via on-line solution of (2.17)), the primary control objective is to asymptotically drive the end-effector trajectory tracking error

$$e_1 \triangleq x_d - x \tag{2.23}$$

to zero while compensating for uncertainties in the system dynamics. Motivated by the subsequent control design strategy, additional tracking error variables $e_2(t), r(t) \in \Re^3$ are defined as follows

$$e_2 \triangleq \dot{e}_1 + e_1 \tag{2.24}$$

$$r \triangleq \dot{e}_2 + e_2 \tag{2.25}$$

the secondary control objective is to preserve the passivity of the robot for safety of user operation in the sense that

$$\int_{t_0}^{t} \bar{F}^T \dot{x} dt \geq -c_2 \tag{2.26}$$

where $\dot{x}(t)$ is the velocity of the robot and $\bar{F}(t)$ is the interaction force with both variables expressed in $I$ while $c_2 \in \Re$ is a positive constant. The control challenge is to obtain the

companion objectives mentioned above while utilizing only measurements of the end-effector position, velocity, and the interaction force. Given these measurements, $e_1(t), e_2(t)$ are measurable variables while $r(t)$ is unmeasurable. Motivated by the ensuing control development and stability analysis, the following set of assumptions can be made:

Assumption 1 The transformed inertia and gravity matrices denoted, respectively, by $\bar{M}(x)$, and $\bar{G}(x)$ are uncertain but known to be second order differentiable with respect to $x$ while the unknown centripetal-Coriolis matrix $\bar{V}_m(x, \dot{x})$ is known to be second order differentiable with respect to $x$ and $\dot{x}$.

Assumption 2 $\bar{F}(t) \in \mathcal{L}_\infty$ is a measurable interaction force exerted by the human operator at the end-effector.

Assumption 3 The reference trajectory $x_d(t)$ is continuously differentiable up to its fourth derivative such that $x_d^{(i)}(t) \in \mathcal{L}_\infty$, $i = 0, 1, 2, 3, 4$.

Assumption 4 The desired curve $r_d(s)$ is analytic along the parameter $s$ (at least the first three partial derivatives along $s$ exist and are bounded such that $r_d(s), r_d'(s), r_d''(s), r_d'''(s) \in \mathcal{L}_\infty$ ).

Assumption 5 The skew symmetric matrix $[\omega]_\times$ is continuously differentiable up to its second derivative such that $[\omega]_\times^{(i)} \in \mathcal{L}_\infty$, $i = 0,1,2$.

Assumption 6 During the control development, the assumption that the minimum singular value of the manipulator Jacobian, denoted by $\sigma_m$ is greater than a known small positive constant $\delta > 0$ will be made, such that $\max\left\{\left\|J^{-1}(q)\right\|\right\}$ is known a priori and all kinematic

singularities are always avoided (This is easily ensured by the algorithm introduced earlier in section 1). It is also noted that since the only concern is with revolute robot manipulators, the kinematic and dynamic terms denoted by $M(q), V_m(q,\dot{q}), G(q), f(q), J(q),$ and $J^{-1}(q)$ are bounded for all possible $q(t)$ (*i.e.*, these kinematic and dynamic terms only depend on $q(t)$ as arguments of trigonometric functions). From the preceding considerations, it is easy to argue that $\bar{M}(x), \bar{V}_m(x,\dot{x}), \bar{G}(x) \in \mathcal{L}_\infty$ for all possible $x(t)$.

Control Design: Tier 3

As a primary step, the system is partially feedback linearized by designing the control signal $\bar{\tau}(t)$ as follows

$$\bar{\tau} = -\bar{F} + \bar{\tau}_a \tag{2.27}$$

where $\bar{\tau}_a(t) \in \Re^3$ is a yet to be designed auxiliary control signal and Assumption 2 has been taken advantage of. Additionally, the system representation of (2.4) is simplified by defining a generalized variable $\bar{B}(x,\dot{x}) \in \Re^3$ as follows

$$\bar{B} = \bar{V}_m(x,\dot{x})\dot{x} + \bar{G}(x) \tag{2.28}$$

The utilization of (2.27) and (2.28) allows one to succinctly rewrite (2.4) as follows

$$\bar{M}\ddot{x} + \bar{B} = \bar{\tau}_a \tag{2.29}$$

Given (2.23 - 2.25) and (2.29), one can obtain the open-loop tracking error dynamics as follows

$$\bar{M}\dot{r} = -\frac{1}{2}\dot{\bar{M}}r - e_2 - \dot{\bar{\tau}}_a + N \tag{2.30}$$

where $N(\cdot) \in \Re^3$ is an aggregation of unknown dynamic terms that is explicitly defined as follows

$$N \triangleq \bar{M}\left(\ddot{x}_d + \ddot{e}_1 + \dot{e}_2\right) + \dot{\bar{M}}\left(\dot{x}_d + \frac{1}{2}r - e_1\right) + e_2 + \dot{\bar{B}} \tag{2.31}$$

In order to take advantage of the known structure of the uncertainty in the robot dynamics, N ( ·) is rewritten as a sum of two auxiliary signals $N_1(t, x, \dot{x}, \ddot{x})$ and $N_2(z)$ as follows

$$\begin{aligned} N = & \underbrace{\bar{M}(x)\ddot{x}_d + \dot{\bar{M}}(x,\dot{x})\dot{x}_d + \dot{\bar{B}}(x,\dot{x},\ddot{x})}_{N_1(\cdot)} + \\ & \underbrace{\bar{M}(x)\left(\ddot{e}_1 + \dot{e}_2\right) + \dot{\bar{M}}(x,\dot{x})\left(\frac{1}{2}r - \ddot{e}_1\right) + e_2}_{N_2(\cdot)} \end{aligned} \tag{2.32}$$

where $z(t) = \left[e_1^T(t) \quad e_2^T(t) \quad r^T(t)\right]^T$ defines a composite error vector. Motivated by the structure of $N_1(\cdot)$ in (2.32), a desired variable $N_{1d}(t)$ is defined as follows

$$N_{1d}(t) \triangleq N\left(x_d, \dot{x}_d, \ddot{x}_d, \dddot{x}_d\right) = \bar{M}\left(x_d\right)\ddot{x}_d + \dot{\bar{M}}\left(x_d, \dot{x}_d\right)\dot{x}_d + \dot{\bar{B}}\left(x_d, \dot{x}_d, \ddot{x}_d\right) \tag{2.33}$$

From Assumptions 1, 3, and 6, it is easy to see that. After adding and subtracting $N_{1d}(t)$, $\dot{N}_{1d}(t)$ to the right-hand side of (2.30), the result is

$$\bar{M}\dot{r} = -\frac{1}{2}\dot{\bar{M}}r - e_2 - \dot{\tau}_a + \tilde{N} + N_{1d} \tag{2.34}$$

where $\tilde{N} \triangleq N - N_{1d}$ is an unmeasurable error signal. After extensive algebraic manipulations (See Appendix A), it can be shown that $\tilde{N}(\cdot)$ can be upper bounded as follows

$$\tilde{N} \le \rho\left(\|z\|\right)\|z\| \tag{2.35}$$

where the notation $\|\cdot\|$ denotes the standard Euclidean norm, $\rho\left(\|z\|\right) \in \Re$ is a positive non-decreasing function while $z(t) \in \Re^9$ has been previously defined below (2.32). Based on the

structure of (2.34), (2.35) as well as the subsequent stability analysis, the following implementable continuous control law is proposed to achieve the stated control objectives

$$\begin{aligned}\overline{\tau}_a &= \left(k_s+1\right)e_2\left(t\right)-\left(k_s+1\right)e_2\left(t_0\right)+ \\ &\int_{t0}^{t}\left[\left(k_s+1\right)e_2\left(\tau\right)+\left(\beta_1+\beta_2\right)\mathrm{sgn}\left(e_2\left(\tau\right)\right)\right]d\tau\end{aligned} \tag{2.36}$$

where $k_s, \beta_1, \beta_2$ are constant positive control gains, and $\mathrm{sgn}\left(\cdot\right)$ denotes the standard signum function. After taking the time derivative of (2.36) and substituting for $\overline{\tau}_a\left(t\right)$ into (2.34), the following closed loop system is obtained

$$\overline{M}\dot{r} = -\frac{1}{2}\dot{\overline{M}}r - e_2 - \left(k_s+1\right)r - \left(\beta_1+\beta_2\right)\mathrm{sgn}\left(e_2\right) + \tilde{N} + N_{1d} \tag{2.37}$$

Stability Analysis

Before presenting the main result of this section, the following two lemmas are stated. that will be invoked later.

Lemma 1 *Let the auxiliary function* $L_1\left(t\right)\in\Re$ *be defined as follows*

$$L_1 \triangleq r^T\left(N_{1d}-\beta_1\,\mathrm{sgn}\left(e_2\right)\right) \tag{2.38}$$

*If the control gain* $\beta_1$ *is selected to satisfy the sufficient condition*

$$\beta_1 > \left\|N_{1d}\left(t\right)\right\| + \left\|\dot{N}_{1d}\left(t\right)\right\| \tag{2.39}$$

*then*

$$\int_{t_0}^{t} L_1\left(\tau\right)d\tau \le \zeta_{b1} \tag{2.40}$$

*where the positive constant* $\zeta_{b1}\in\Re$ *is defined as*

$$\zeta_{b1} \triangleq \beta_1\left\|e_2\left(t_0\right)\right\|_1 - e_2^T\left(t_0\right)N_{1d}\left(t_0\right) \tag{2.41}$$

*where the notation* $\|\eta\|_1 \triangleq \sum_{r=1}^{n} |\eta r| \ \ \forall \eta \in \Re^n$ *denotes the 1-norm.*

Proof: The proof is presented in Appendix B. □

Lemma 2 *Let the auxiliary function be defined as follows*

$$L_2 \triangleq \dot{e}_2^T \left(-\beta_2 \operatorname{sgn}\left(e_2\right)\right) \tag{2.42}$$

*It is then easy to show that*

$$\begin{aligned} \int_{t_0}^{t} L_2\left(\tau\right) d\tau &= \int_{t0}^{t} e_2^T \left(-\beta_2 \operatorname{sgn}\left(e_2\right)\right) d\tau \\ &= \beta_2 \left\|e_2\left(t_0\right)\right\|_1 - \beta_2 \left\|e_2\left(t\right)\right\|_1 \le \beta_2 \left\|e_2\left(t_0\right)\right\| \triangleq \zeta_{b2} \end{aligned} \tag{2.43}$$

The main stability result for the proposed controller is now stated in the following Theorem.

Theorem 1 *The control law of (2.36) ensures that all system signals are bounded under closed-loop operation and asymptotic tracking is obtained in the sense that*

$$e_i^{(j)}\left(t\right) \to 0 \ \ as\, t \to \infty \ \forall i = 1,2; \ \ j = 0,1. \tag{2.44}$$

Proof: The proof is presented in Appendix C. □

The passivity of the robot manipulator is now proven. Integrating both sides of the bottom expression of (2.63), the following result is obtained

$$\int_{t_0}^{t} \left\|e_2\left(\tau\right)\right\|_1 d\tau \le \frac{V_1\left(t_0\right)}{\beta_2} \Rightarrow e_2\left(t\right) \in \mathcal{L}_1$$

Since $e_1\left(t\right)$ is related to $e_2\left(t\right)$ through a transfer function that is strictly proper and stable, one can use Lemma A.8 of [30] to conclude that $e_1\left(t\right) \in \mathcal{L}_1$. Now, utilizing (2.24), one can also state that $\dot{e}_1\left(t\right) \in \mathcal{L}_1$. The work done by the interaction force on the robot is denoted by $W\left(t\right)$ and given by

$$W = \int_{t_0}^{t} \overline{F}^T \dot{x} d\tau = \int_{t_0}^{t} \overline{F}^T \dot{x}_d d\tau - \int_{t_0}^{t} \overline{F}^T \dot{e}_1 d\tau \tag{2.45}$$

where (2.23) has been utilized. Since the first term on the right hand side of (2.45) has been lower bounded as in (2.22), attention is shifted to the second term. The second term can now be upper-bound as follows

$$\int_{t_0}^{t} \overline{F}^T \dot{e}_1 d\tau \leq \sup_t \left\{ \left\| \overline{F}(t) \right\| \right\} \sup_t \left\{ \int_{t_0}^{t} \left\| \dot{e}_1(t) \right\|_1 d\tau \right\} \leq c_3 \tag{2.46}$$

where the fact that $\dot{e}_1(t) \in \mathcal{L}_1$ as well as Assumption 2 has been utilized to justify the existence of the supremum functions defined above, and $c_3$ is a positive constant. One can now utilize the lower bound of (2.22) and the upper-bound of (2.46) in order to lower-bound $W(t)$ as $W(t) \geq -c_1 - c_3 = -c_2$; this satisfies the passivity control objective of (2.26).

CHAPTER THREE

REHABILITATION ROBOT

SIMULATION RESULTS

Figure 3.1 illustrates the graphical representation of the Path Planning and Control algorithm. As previously stated in previous chapter, the steps involved in Tier 1 include planning a path to avoid singularities, joint limits and obstacles while attempting to track the desired trajectory. These steps can be computed offline and cut down on the computation time. The real time simulation consists on an impedance generator which computes the desired end-effector position, a non-linear compensator which computes the control torque based on the error between the desired and actual end effector kinematics.

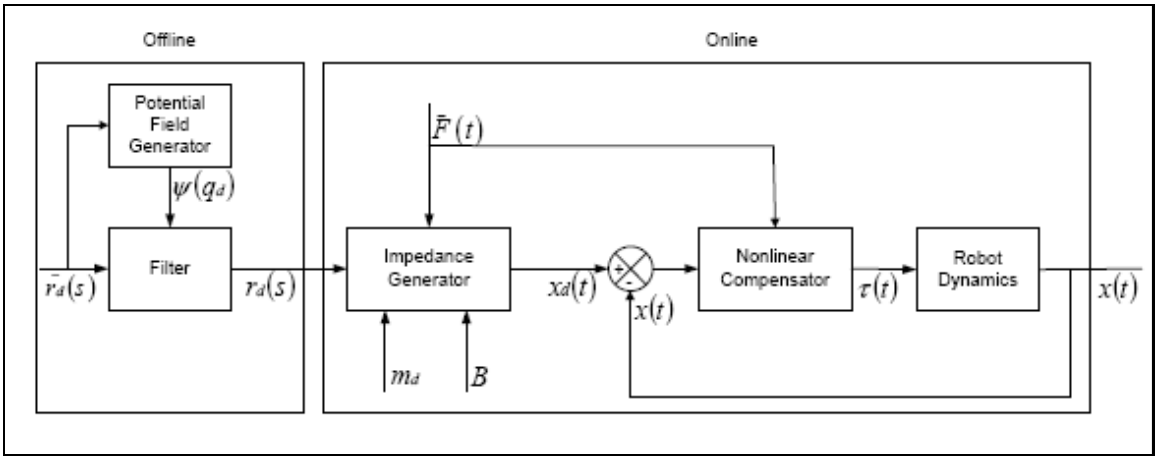


Figure 3.1 Graphical Representation of Path Planning
and Control Algorithm (Zhang, 2004)

Two-Link Planar Elbow Arm

Numerical simulations were performed to illustrate the performance of the proposed reference generator and control law of (2.17), (2.27), and (2.36) with a two-link planar elbow arm (shown in Figure 3.2.) whose inertia matrix $M(q)$ can be expressed in terms of its elements as follows

$$\begin{aligned} m_{11} &= (m_1 + m_2) l_1^2 + m2l_2^2 + 2m_2 l_1 l_2 \cos q_2 \\ m_{12} &= m_{21} = m_2 l_2^2 + m_2 l_1 l_2 \cos q_2 \\ m_{22} &= m_2 l_2^2 \end{aligned} \tag{2.47}$$

while the centripetal Coriolis vector can be expressed in the following manner

$$V_m(q, \dot{q})\dot{q} = \begin{bmatrix} -m_2 l_1 l_2 \left(2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2\right) \sin q_2 \\ m_2 l_1 l_2 \dot{q}_1^2 \sin q_2 \end{bmatrix} \tag{2.48}$$

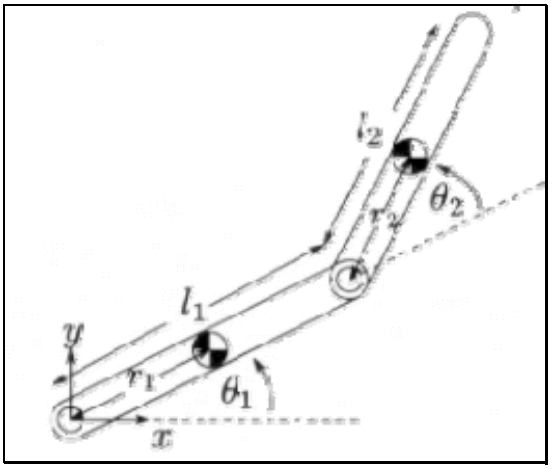


Figure 3.2 Two Link Manipulator Configuration

The mass and length parameters of the manipulator are specified as follows

$$m_1 = 2.08[\text{kg}] \quad m_2 = 0.168[\text{kg}] \quad l_1 = 1.5[\text{m}] \quad l_2 = 1.2[\text{m}]$$

The initial configuration of the two-link robot is chosen as $q_1(0) = 0.334[\text{rad}],\ q_2(0) = 0.7[\text{rad}]$. The desired contour is specified by a unit circular path $\bar{r}_d(s) = \left[\cos(s) \quad \sin(s)\right]^T$. The initial conditions and parameters for the reference generator are chosen as follows

$$x_d(0) = [1.6 \quad 1.5]^T[\text{m}] \qquad s(0) = 0$$
$$m_d = 0.1[\text{kg}] \qquad B = diag\{1, 1000\}\left[\text{Ns}^{-1}\right]$$

The parameters for the obstacle is chosen as follows

$$O_1(0) = [-0.5 \quad 1.65]^T[\text{m}] \quad R_1 = 0.5[\text{m}]$$

The interaction force applied at the end-effector by a user was chosen to be $F = [2 \quad 2]^T[\text{N}]$. The joint limit for all joints are set as $q_{\text{dimax}} = 2\pi[\text{rad}]$ and $q_{\text{dimin}} = \frac{\pi}{16}[\text{rad}]$. The parameters in (2.8), (2.10) and (2.11) are chosen as follows

$$\alpha_1 = 1 \qquad \alpha_2 = 4 \quad \gamma_1 = 0.1$$
$$\gamma_2 = 0.5 \quad \gamma_3 = 4 \quad \gamma_4 = 10$$

For best transient performance, the control gains specified in (2.36) are chosen to be $k_s = 99, \beta_1 + \beta_2 = 1$. The measure $\psi_1$ defined in (2.7) is depicted in Figure 3.3. as one closed contour is traced. Corresponding to the second dip in $\psi_1$ in Figure 3.3., illustrated in Figure 3.4., a snapshot of the 2-link manipulator veering away from the dashed circular contour $\bar{r}_d$ in order to avoid the kinematic singularity $(q_2 = \pi)$. Next, the measure $\psi_3$ defined in (2.9) is shown in Figure 3.5 — by employing the dip in $\psi_3$, the algorithm is able to steer the robot

away from the physical obstacle marked by the solid circle in the robot workspace as can be seen in the snapshot of Figure 3.6. In Figure 3.7, one can see the evolution of the measure $\psi_2$ - as the measure starts close to 0, since $q_2$ is close to the lower joint limit $\frac{\pi}{16}[\text{rad}]$, the measure $\psi_2$ then increases and decreases based on how close the joint angles are to their limits, the snapshot in Figure 3.8 shows how our algorithm utilizes the dip in the measure $\psi_2$ in order to avoid the $q_1 = \frac{\pi}{16}[\text{rad}]$ joint limit.

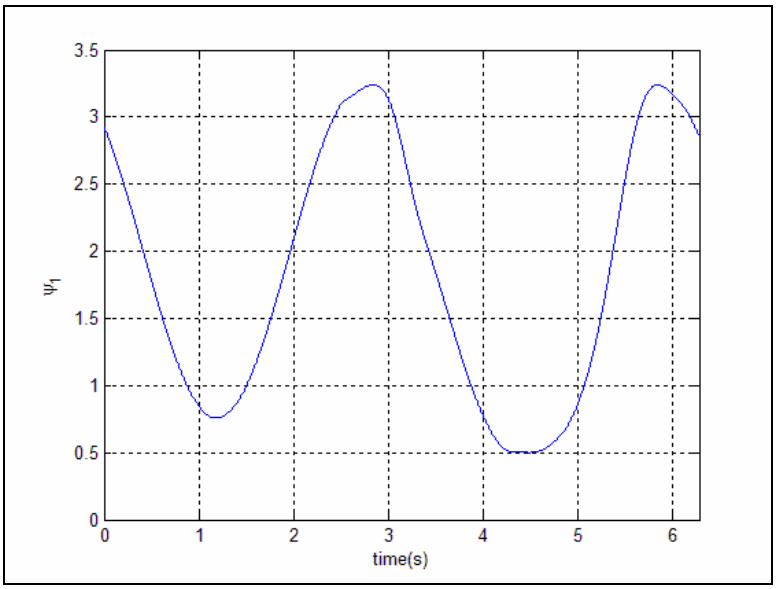


Figure 3.3 Manipulability Metric $\Psi_1$ for Avoiding Kinematic Singularities

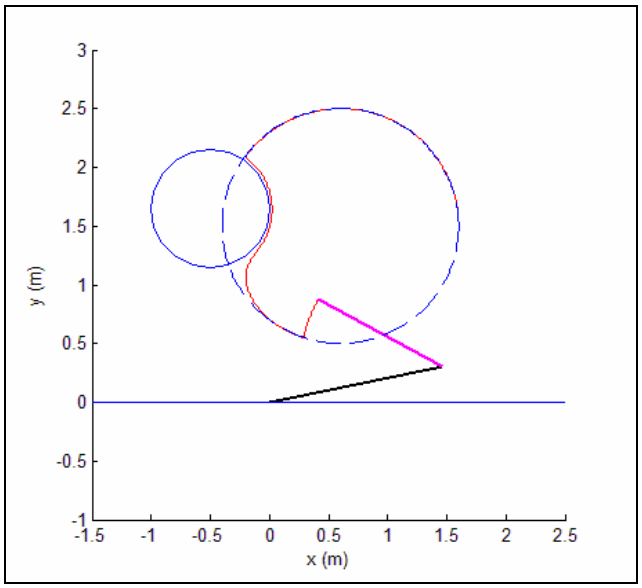


Figure 3.4 Example of the 2-link manipulator using the $\Psi_1$ Metric to Avoid a Kinematic Singularity

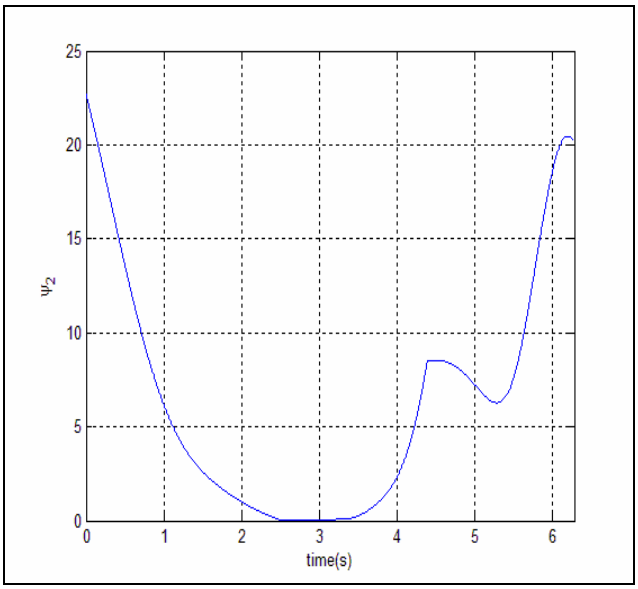


Figure 3.5 Metric $\Psi_3$ for avoiding Obstacles in the Robot Workspace

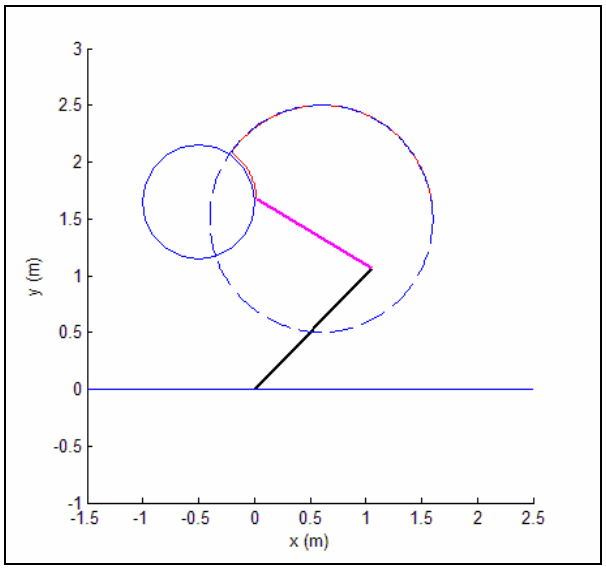


Figure 3.6 Example of a 2-link Manipulator using the $\Psi_3$ metric to Avoid an Obstacle

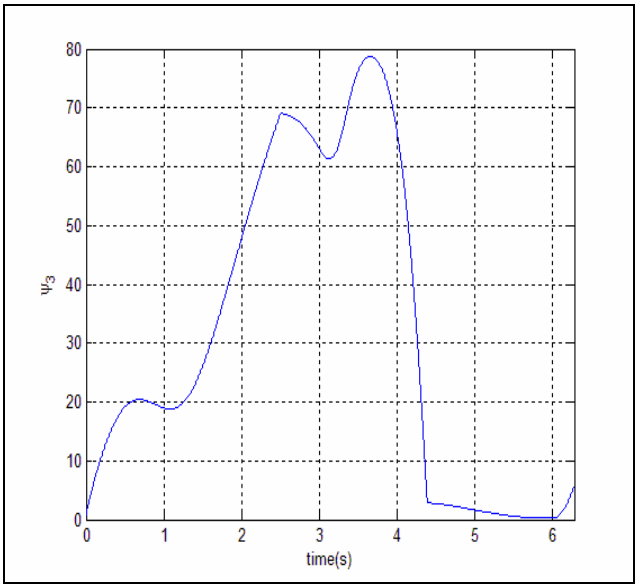


Figure 3.7 Metric $\Psi_2$ for avoiding joint limit singularities

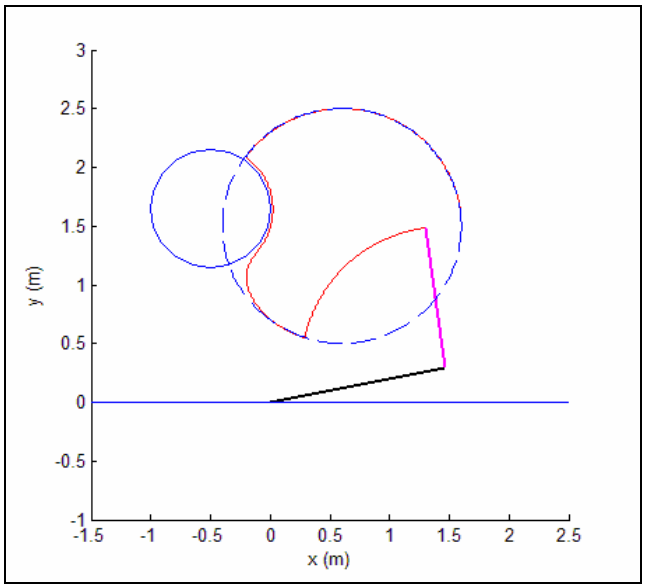


Figure 3.8 Example of a 2-link Robotic Manipulator using the $\Psi_2$ metric to avoid a joint limit for the first joint

The tracking error $e_1(t)$ is depicted in Figure 3.9 and the control torque input $\overline{\tau}(t)$ is depicted in Figure 3.10. Figure 3.11 shows the robot end-effector tracing the modified desired contour $r_d$ as the user applies interaction force at the end-effector. It should be noted that in Figures 3.4, 3.6, 3.8, and 3.11, the solid circle denotes an obstacle, the dashed circle denotes the nominal contour $\overline{r}_d(s)$, the thin solid curve denotes the time parameterization $x_d(t)$ of the modified contour $r_d(s)$, and the thick solid curve denotes the actual trajectory of the robot end-effector $x(t)$.

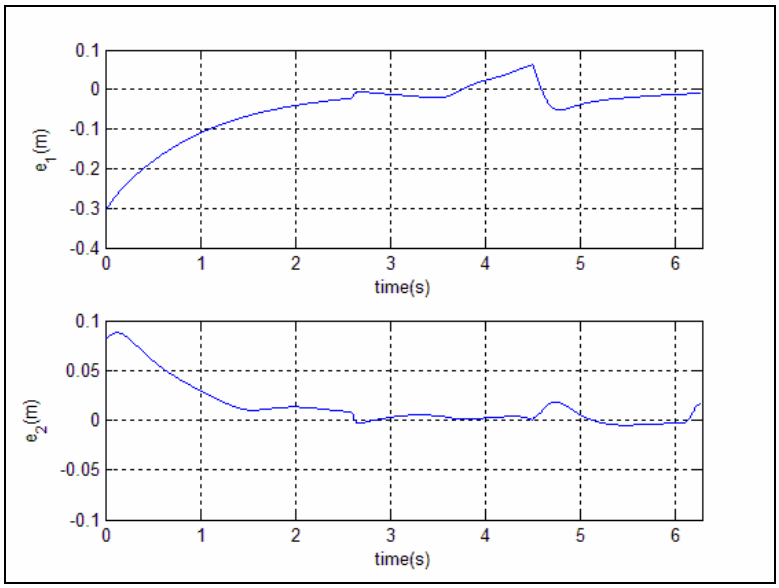


Figure 3.9 Errors between the Desired and Actual End-effector Trajectories

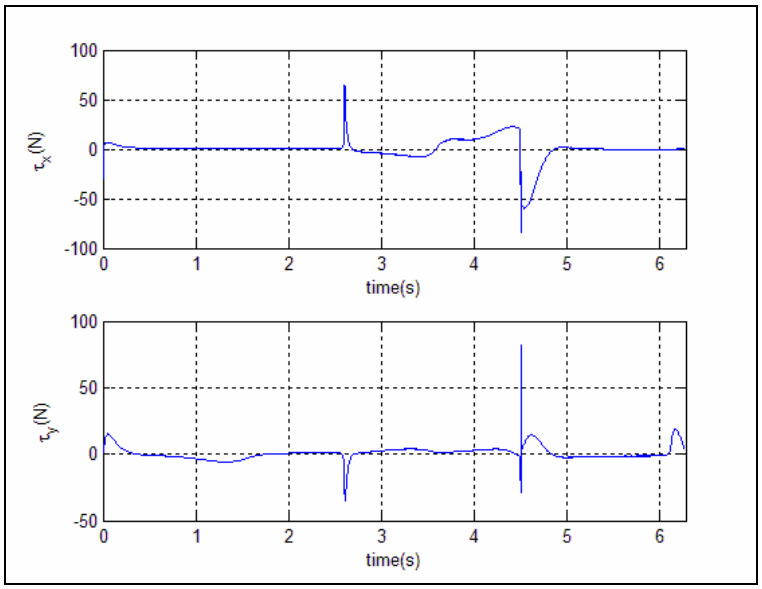


Figure 3.10. Control Torque $\bar{\tau}(t)$

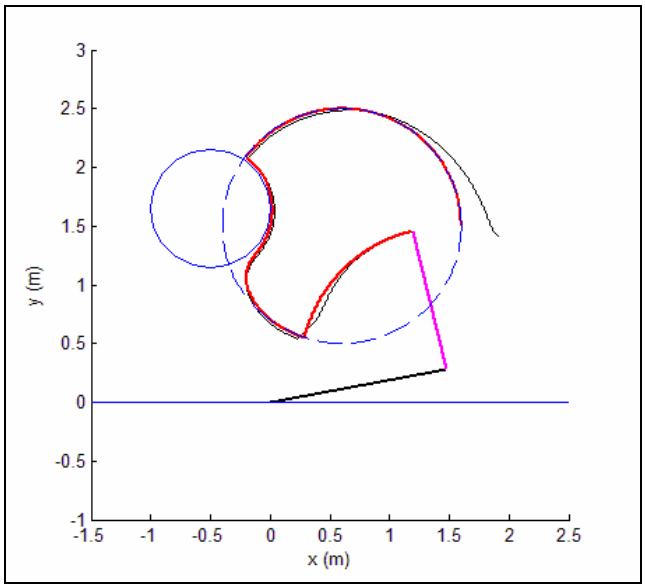


Figure 3.11 A plot of a 2-link Manipulator tracking the Desired Trajectory with User Interaction Force

Three Degree-of-Freedom Elbow Arm Manipulator

Numerical simulations were performed to illustrate the performance of the proposed reference generator and control law of (2.17), (2.27), and (2.36) with a three-link elbow arm manipulator (shown in Figure 3.12) whose inertia matrix $M(q)$ can be expressed in terms of its elements as follows

$$
\begin{aligned}
m_{11} &= \frac{1}{2}m_2a_2^2\cos(2q_2) + \frac{1}{2}m_2a_2^2 + \frac{1}{2}m_3a_3^2\cos(2q_2+2q_3) + \frac{1}{2}m_3a_3^2 \\
&\quad + m_3a_3a_2\cos(q_3) + m_3a_3a_2\cos(2q_2+q_3) + \frac{1}{2}m_3a_2^2\cos(2q_2) + \frac{1}{2}m_3a_2^2 + I_1 \\
m_{12} &= m_{21} = 0 \\
m_{13} &= m_{31} = 0 \\
m_{22} &= m_2a_2^2 + 2m_3a_3a_2\cos(q_3) + m_3a_2^2 + m_3a_3^2 \\
m_{23} &= m_{32} = m_3a_3a_2\cos(q_3) + m_3a_3^2 \\
m_{33} &= m_3a_3^2
\end{aligned}
\qquad (2.47b)
$$

while the centripetal Coriolis vector and Gravity effects can be expressed in the following manner

$$
V_m(q,\dot{q})\dot{q} = \begin{bmatrix} V_{m11} \\ V_{m12} \\ V_{m13} \end{bmatrix} \quad ; \quad \mathrm{G} = \begin{bmatrix} 0 \\ \mathrm{m_2a_2}g\cos(q_2) \\ m_3a_3g\cos(q_2+q_3) \end{bmatrix}
$$

where $V_{m11}$, $V_{m12}$, $V_{m13}$ are shown in Appendix G.

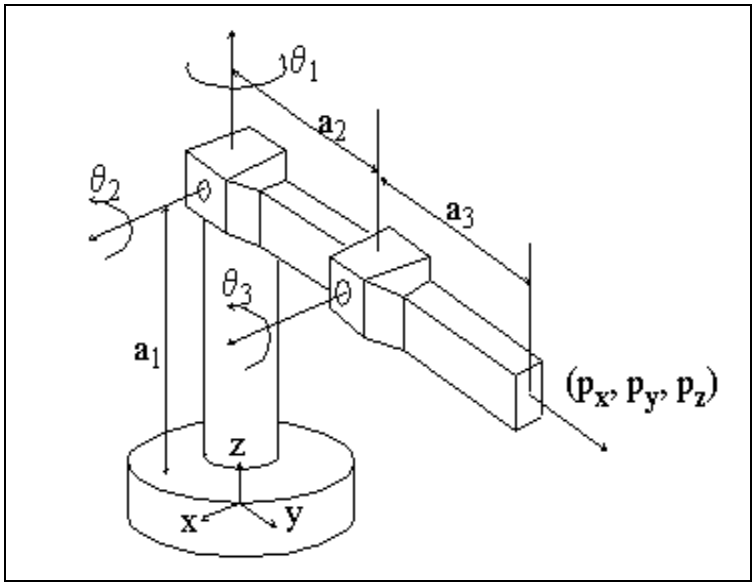


Figure 3.12: The three-link elbow robotic manipulator

The mass and length parameters of the manipulator are specified as follows

$I_1 = 2.08\left[\text{kg-m}^2\right]$ $m_2 = 2.08[\text{kg}]$ $m_3 = 0.168[\text{kg}]$ $a_1 = 1.5[\text{m}]$ $a_2 = 1.2[\text{m}]$ $a3 = 0.5[m]$

The initial configuration of the two-link robot is chosen as $q_1(0) = \frac{\pi}{3}[\text{rad}]$, $q_2(0) = 5.3904[\text{rad}]$, $q_3(0) = 1.9462[\text{rad}]$. The desired contour is specified by a skewed circular path $\bar{r}_d(s) = \left[\cos(s) \quad \sin(s) \quad \frac{\sin(s)}{2}\right]^T$. The initial conditions and parameters for the reference generator are chosen as follows

$$x_d(0) = [1 \quad 0 \quad 1]^T[\text{m}] \qquad s(0) = 0$$
$$m_d = 0.1[\text{kg}] \qquad B = diag\{1,100,100\}\left[\text{Ns}^{-1}\right]$$

The parameters for the obstacle is chosen as follows

$$O_1(0) = \left[-\frac{\sqrt{2}}{2} \quad -\frac{\sqrt{2}}{2} \quad 1\right]^T[\text{m}] \quad R_1 = 0.5[\text{m}]$$

The interaction force applied at the end-effector by a user was chosen to be $F = [2 \quad 1 \quad 2]^T[\text{N}]$. The joint limit for all joints are set as $q_{\text{dimax}} = 2\pi[\text{rad}]$ and $q_{\text{dimin}} = \frac{\pi}{16}[\text{rad}]$ for $i = 2,3$. The parameters in (2.8), (2.10) and (2.11) are chosen as follows

$$\alpha_1 = 1 \qquad \alpha_{2,3} = 4 \quad \gamma_1 = 0.5$$
$$\gamma_2 = 0.5 \qquad \gamma_3 = 4 \quad \gamma_4 = 10$$

For best transient performance, the control gains specified in (2.36) are chosen to be $k_s = 149, \beta_1 + \beta_2 = 15$. The measure $\psi_1$ defined in (2.7) is depicted in Figure 3.13 as one closed contour is traced. The measure is always positive which indicates that the robotic manipulator is never configured to a kinematic singularity. The depressions in the measure

indicate occurrences when the manipulator is close to a singularity. Next, the measure $\Psi_3$ defined in (2.9) is shown in Figure 3.14 — by employing the dip in $\psi_3$, our algorithm is able to steer the robot away from the physical obstacle marked by the solid circle in the robot workspace as can be seen in the snapshot of Figure 3.15. In Figure 3.16, one can see the evolution of the measure $\Psi_2$. The measure is always positive which indicates that the robotic manipulator never reaches its joints' limits. The depressions in the measure indicate occurrences when the manipulator is close to some or all of its joints' limits.

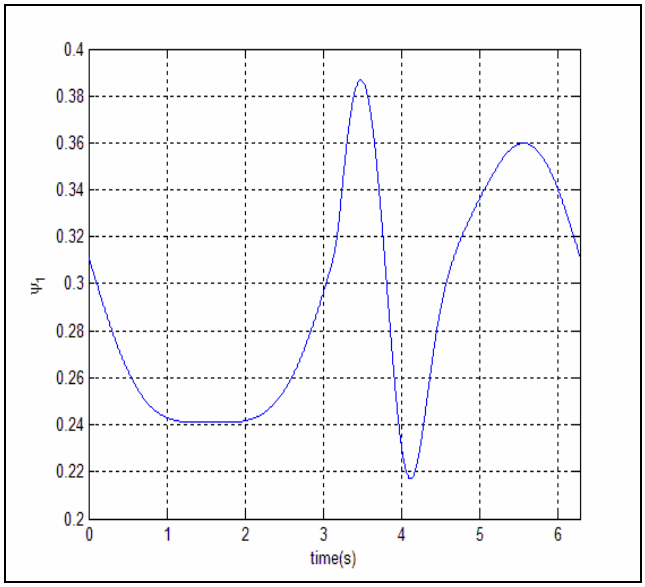


Figure 3.13 Manipulability metric $\Psi_1$ for avoiding kinematic singularities

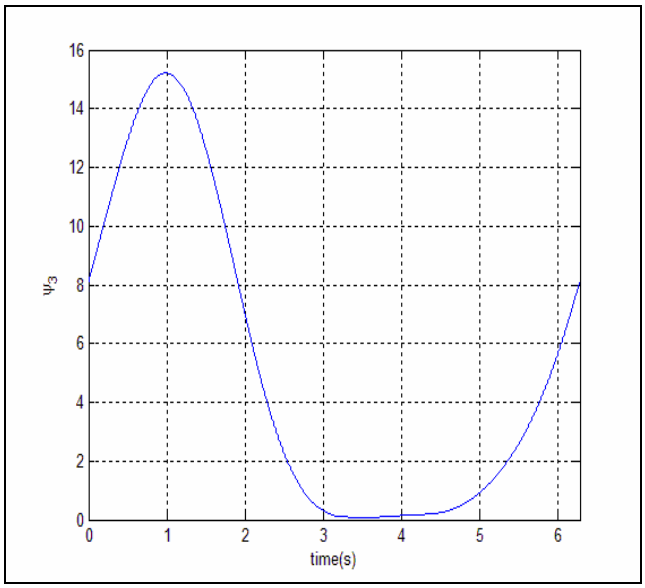


Figure 3.14 Metric $\Psi_3$ for avoiding obstacles in the robot workspace

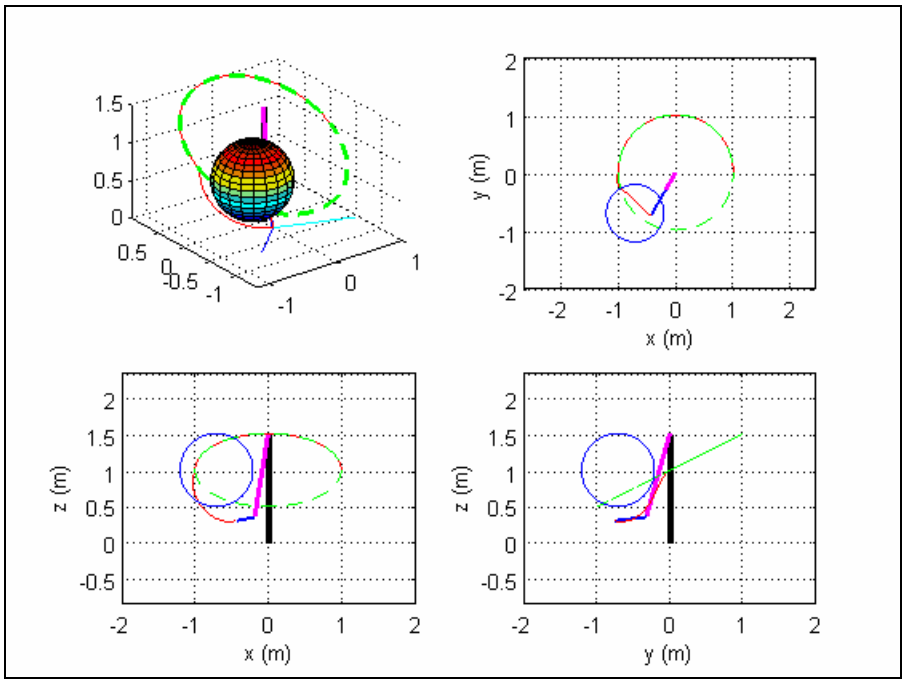


Figure 3.15 Example of a 3-link Manipulator using
the $\Psi_3$ Metric to avoid an Obstacle

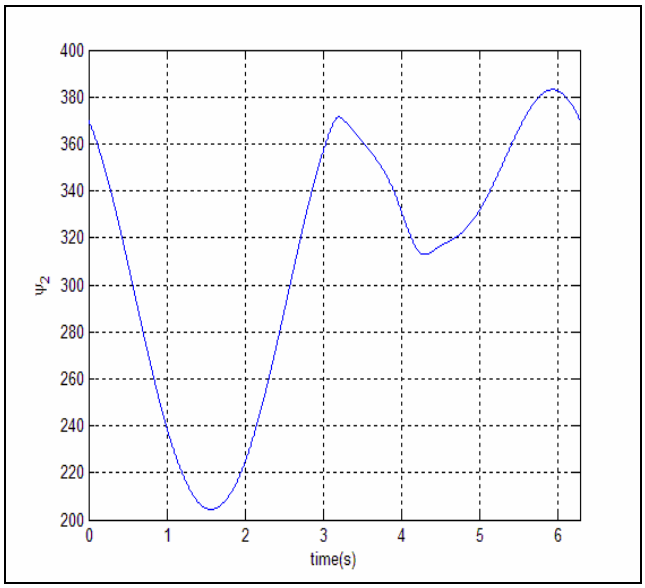


Figure 3.16 Metric $\Psi_2$ for avoiding joint limit singularities

The tracking error $e_1(t)$ is depicted in Figure 3.17 and the control torque input $\overline{\tau}(t)$ is depicted in Figure 3.18. Figure 3.19 shows the robot end-effector tracing the modified desired contour $r_d$ as the user applies interaction force at the end-effector. It should be noted that in Figures 3.15 and 3.19, the sphere denotes an obstacle, the thick dashed skewed circle denotes the nominal contour $\overline{r}_d(s)$, the thin dashed curve denotes the time parameterization $x_d(t)$ of the modified contour $r_d(s)$, and the solid curve denotes the actual trajectory of the robot end-effector $x(t)$.

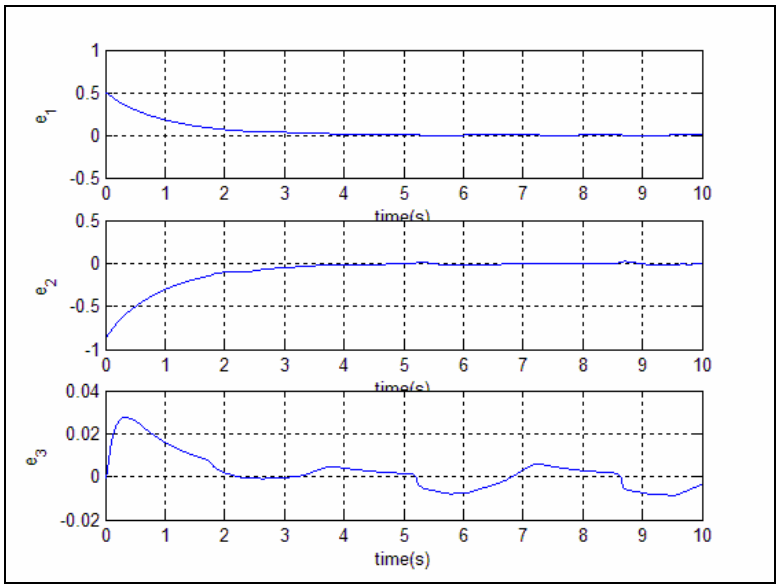


Figure 3.17 Errors between the Desired and Actual End-effector Trajectories

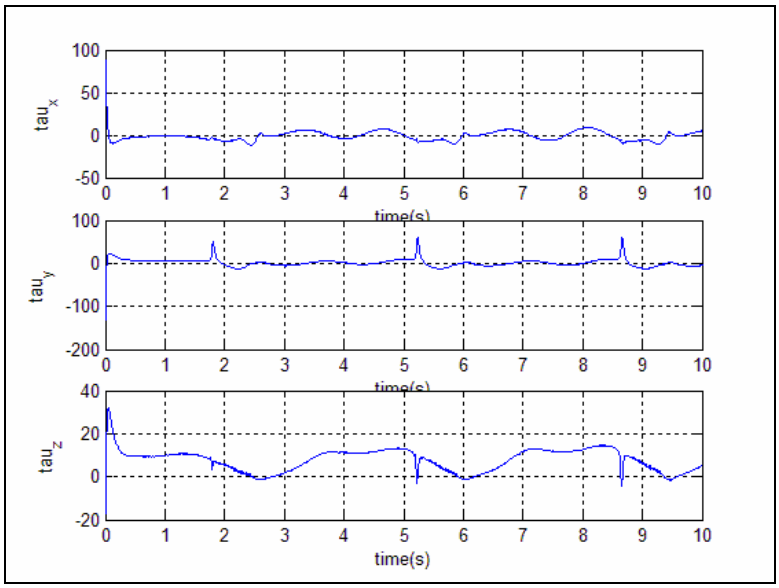


Figure 3.18. Control Torque $\bar{\tau}(t)$

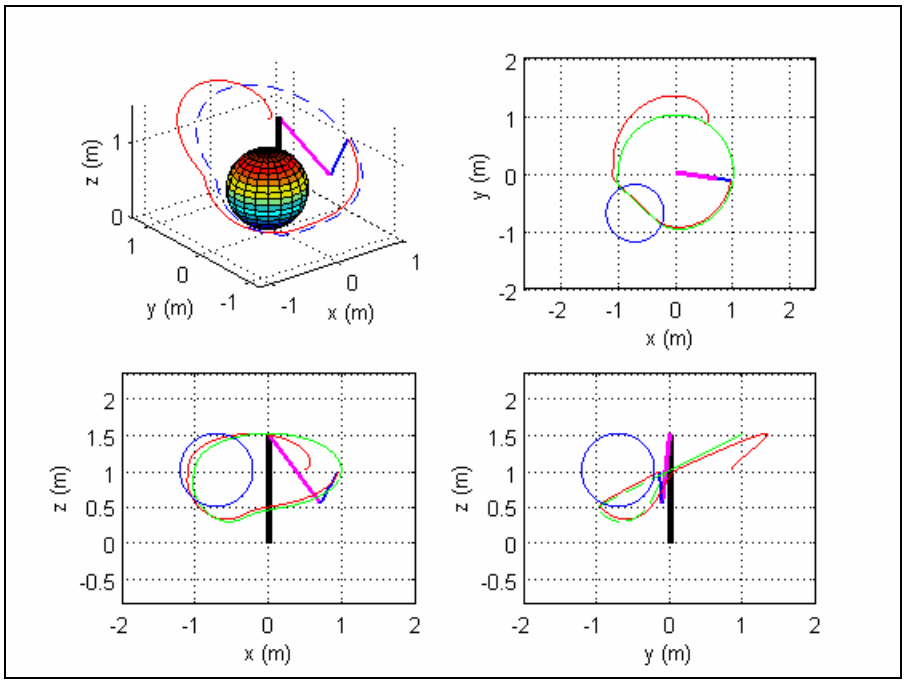


Figure 3.19. A plot of a 3-link Elbow Manipulator tracking the Desired Trajectory with User Interaction Force

# CHAPTER FOUR

# A HAPTIC INTERFACE CONTROLLER FOR STEER-BY-WIRE VEHICLE USING MODEL-REFERENCE ADAPTIVE CONTROL (MRAC)

In recent years, military conflicts and energy shortages have conspired to create a need for an alternate means of providing energy to automotive vehicles other than petroleum products, which in turn creates the need for an alternative to the conventional hydraulic power steering system. Examples such as hybrid electric vehicles (HEV) featuring hydrogen, fuel cells, electric motors, solar cells, and/or internal combustion engines are being designed and introduced into the world market. Although the concepts of electric and specifically steer-by-wire steering systems have been explored in vehicular research, attention must be focused on the haptic interface for a new kind of steering system. In the robotics field, tele-operation of robotic manipulators has been well studied (as in Chapters II and III) as it permits the introduction of human intelligence and decision making capabilities into a possibly hostile remote environment (such as a nuclear power plant and possibly a different planet). The concept of force feedback has thus been judged to be a necessity.

An operator functioning within a remote driving environment primarily depends on visual feedback to make meaningful maneuvers. The "feel" of the road, due to both the vehicle acceleration forces (i.e., G-forces) and the tire/road forces, plays a very prominent role in creating the driving experience as stated by Liu *et al.* The physiological effect of these forces has been documented by Godley *et al.* to be the most important stimulus after visual feedback for optimal driver performance. An appropriate magnitude is important for force feedback to be valuable to the driver. For instance, excessive feedback results in the need for

large driver forces to steer the system which defeats the purpose of emulating the steering "feel" experienced while operating a typical vehicle in addition to increasing driver fatigue. Conversely, minute feedback results in poor driver response which also defeats the purpose of easing the driver experience in terms of safety and comfort. Hence, it is essential for the control strategy to ensure that the road "feel" provided by the force feedback be configurable.

Many researchers (such as Proca *et al.*) have worked on establishing dynamic models and performing experiments to identify system parameters with the intention of providing simulated force feedback. Detailed modeling of the conventional, electric, and steer-by-wire steering systems is presented by Post *et al.* After making appropriate simplifying assumptions, these models have been utilized in this thesis to provide the system model. The Bavaria Motor Works company (BMW) has designed and produced an active front steering (AFS) system that varies the steering system behavior based on vehicle parameters. Present day simulators such as the Clemson Vehicle Steering Simulator, already use the virtual environment concept to evaluate commercial and concept steering models.

The general concept of the proposed steer-by-wire haptic-interface control architecture is presented in Figure 4.1. Flow of information in a steering system is bi- directional. Hence providing force feedback only handles one of the two issues that from the utilization of a steer-by-wire system architecture (wherein the steering column and directional assembly are not physically connected). The other equally important feature of the steer-by-wire system involves the actuation of the directional control assembly to translate the driver's steering wheel commands into road wheel movement. In this thesis, a model-reference adaptive control approach is utilized to provide the desired force feedback on the steering wheel to reflect the tire/road interface forces and simultaneously synchronize the motion of the

directional control assembly with the motion of the steering wheel. This approach consists of four parts: A plant (in this case the physical steering system) containing unknown parameters, a reference model (which as been validated with experimental data) for compactly specifying the ideal plant response (which in this thesis is the angular position of the steering wheel and rack), a feedback control law containing adjustable parameters, and an adaptation mechanism that adjusts the parameters such that the response of the plant under adaptive control becomes the same as that of the reference model. This type of approach is motivated by the impedance control concept detailed in [8]. The controller adapts for parametric uncertainties in the system while ensuring global asymptotic tracking for the "driver experience error" and the "locked error".

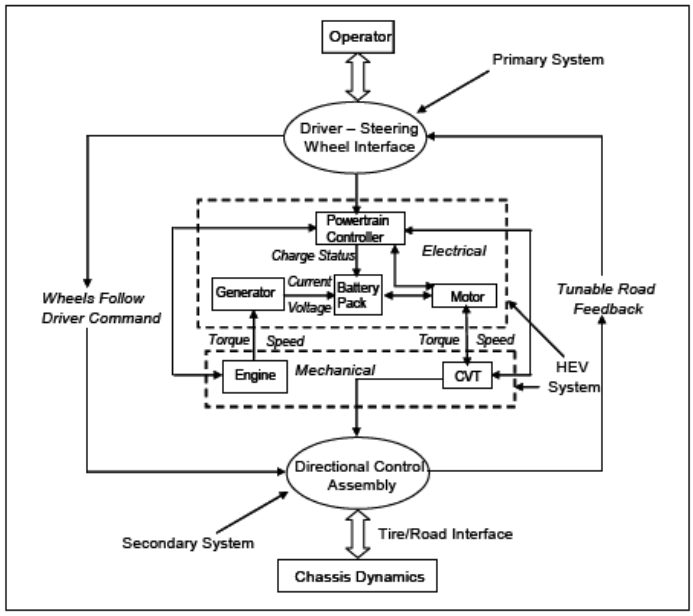


Figure 4.1 Steer-by-wire Interface Architecture for a Typical Hybrid Vehicle (Setlur, 2003)

This chapter is organized as follows. In Section 2, the control system objectives are stated. In Section 3, the models for the servo-motor based steering system and the reference model are presented followed by the open-loop tracking dynamics. In Section 4, an adaptive, tracking controller is presented along with the corresponding closed-loop error system. The stability analysis is discussed in Section 5.

## Control Problem Statement

The steer-by-wire haptic interface control objective is twofold. First, the driver's steering angle commands must be accurately followed; this requires the torque control input provided to the drive motor be designed such that the angular position of the directional control assembly accurately tracks the input. Second, the driver must be given a realistic "virtual driving experience". To this end, a reference model, or target dynamics for the driver input device, should be designed to generate the desired angular position of the driver input device. The reference model must also be chosen appropriately to provide a realistic steering "feel". The control torque provided by the feedback motor must then be designed to ensure that the response of the driver input device follows that of the reference system. The reader is referred to Figure 4.2 for definition of the driver interface and the directional control assembly.

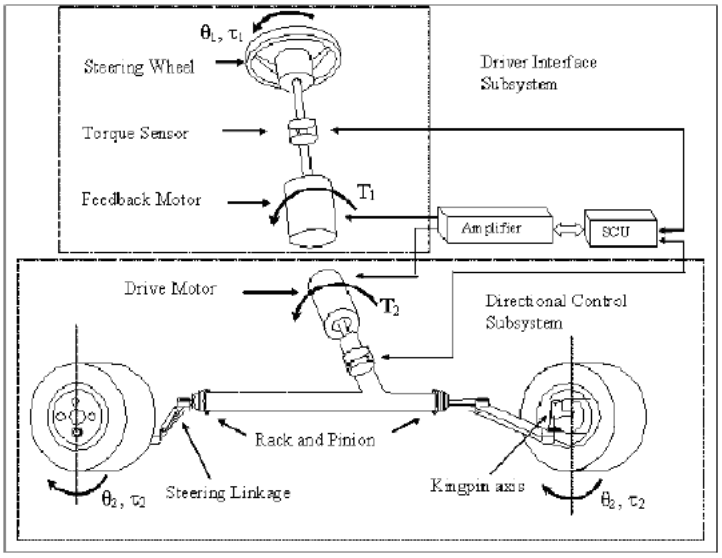


Figure 4.2 Driver interface and directional control subsystems
in a steer-by-wire steering system with nomenclature (Setlur, 2003)

## Dynamic Model Development

Detailed models for the conventional and power assisted steering systems have been explored by several authors (e.g., [26], [27]). The steer-by-wire system involves the partial or total removal of the steering column present in a conventional steering system and the introduction of two servo motors, a motor at the steering wheel, as well as a motor at the rack and pinion to control the road wheels. The steering system is separated into two subsystems: the primary and the secondary subsystems. The primary system, consists of the driver input device (steering wheel) and a servo motor to provide the driver with force feedback, the feedback motor must possess enough torque to adequately translate the aligning torque and friction felt at the road wheels to the driver. The secondary subsystem is composed of the directional control assembly (e.g. rack and pinion system) and a servo motor that provides the necessary torque input to drive this assembly and steer the vehicle.

The drive motor must be powerful enough to adequately turn the road wheels in high torque input maneuvers such as parking and turn from a standstill.

## Steering System Model Formulation

In general, the steering system dynamics, may be expressed in a simplified form as

$$I_1\ddot{\theta}_1 + N_1\left(\theta_1, \dot{\theta}_1\right) = \alpha_1\tau_{sw} + T_1 \tag{4.1}$$

$$I_2\ddot{\theta}_2 + N_2\left(\theta_2, \dot{\theta}_2\right) = \alpha_2\tau_{road} + T_2 \tag{4.2}$$

where $\theta_1(t),\ \dot{\theta}_1(t),\ \ddot{\theta}_1(t) \in \Re$ denote the angular position, velocity, and acceleration, respectively, of the driver input device, $I_1, I_2 \in \Re$ represent the inertias of the driver input device and the vehicle directional control assembly, respectively. $N_1\left(\theta_1, \dot{\theta}_1\right) \in \Re$ is an auxiliary nonlinear function that describes the dynamics on the driver side, $\tau_{sw}(t) \in \Re$ denotes the driver input torque, $T_1(t) \in \Re$ represents a control input torque applied to the driver input device, $\theta_2(t),\ \dot{\theta}_2(t),\ \ddot{\theta}_2(t) \in \Re$ denote the angular position, velocity, and acceleration, respectively, of the vehicle directional control assembly, $N_2\left(\theta_2, \dot{\theta}_2\right) \in \Re$ is an auxiliary nonlinear function that is used to describe the dynamics of the vehicle directional control assembly, $\tau_{road}(t) \in \Re$ represents the reaction torque between the actuator on the directional control assembly and mechanical subsystem actuated by the directional control assembly consisting of the aligning torque at the tires and tire spin inertia, and $T_2(t) \in \Re$ denotes a control input torque applied to the directional control assembly. The constants $\alpha_1, \alpha_2 \in \Re$ are scaling factors that could arise due to gearing in the system.

Remark 1 *The damping and friction effects modeled by* $N_1(\cdot)$ *and* $N_2(\cdot)$ *are assumed to be linearly parameterizable as*

$$N_1\left(\theta_1,\dot{\theta}_1\right)=Y_{N1}\left(\theta_1,\dot{\theta}_1\right)\phi_{N1} \tag{4.3}$$

$$N_2\left(\theta_2,\dot{\theta}_2\right)=Y_{N2}\left(\theta_2,\dot{\theta}_2\right)\phi_{N2} \tag{4.4}$$

*where* $Y_{N1}(\cdot)\in\Re^{1\times p}$, $Y_{N2}(\cdot)\in\Re^{1\times q}$ *are regression matrices containing the measurable signals, and* $\phi_{N1}\in\Re^{p\times 1}$, $\phi_{N2}\in\Re^{q\times 1}$ *are constant matrices containing the unknown parameters in the model* $N_1(\cdot)$ *and* $N_2(\cdot)$. *Further, it is also assumed that if* $\theta_1(t),\dot{\theta}_1(t)\in\mathcal{L}_\infty$ *then* $N_1\left(\theta_1,\dot{\theta}_1\right)\in\mathcal{L}_\infty$ *and if* $\theta_2(t),\dot{\theta}_2(t)\in\mathcal{L}_\infty$ *then* $N_2\left(\theta_2,\dot{\theta}_2\right)\in\mathcal{L}_\infty$.

Reference Model Development

The second control objective is the provision of road "feel" to the driver. To satisfy this goal, impedance control concepts [8] used for robot manipulator position/force control problems have been applied to the problem. This means that the performance of the automotive system has been characterized by its input impedance, the resistance offered by the system to the angular position of the input device (driver steering wheel) with the mathematical representation being $\left(z(t)=\dfrac{\tau(t)}{\theta(t)}\right)$. This Laplace transform counterpart of this transfer function allows the tuning of component parameters to match a desired phase and magnitude, as a function of frequency of the input device. To address this issue, a non linear power steering model [28] was implemented. Additionally, the reference model must be capable of emulating multiple steering configurations depending on the preference of the operator such as power-assist level, the steering column stiffness and the steering ratio.

The power steering system is comprised of the steering wheel and column, a torsion bar, a rack and pinion assembly, and the road wheel linkages. The power assist torque is computed based on relationship function with respect to the magnitude of the torsion bar. A lumped parameter modeling approach is used to simplify the system dynamics as shown in Figure 2.7. Specifically, the hypothetical reference model structure is defined as follows [26]: The input to the steering system is the driver input torque command $\tau_{sw}$, resulting in the angular displacement of the steering wheel $\theta_{sw}$, given by

$$\ddot{\theta}_{sw} = \frac{1}{I_{sw}}\left[\tau_{sw} - B_{sc}\left(\dot{\theta}_{sw} - \dot{\theta}_{sp}\right) - K_{sc}\left(\theta_{sw} - \theta_{sp}\right) - \tau_{fr,sc}\right] \tag{4.5}$$

where $\theta_{sw}, \dot{\theta}_{sw}, \ddot{\theta}_{sw} \in \Re$ denote the desired angular position, velocity, and acceleration, respectively, of the driver input device, and $\theta_{sp}$ represents the spool valve angular displacements. The parameters $I_{sw}, B_{sc}, K_{sc}, \text{and } \tau_{fr,sc}$ represent the lumped steering wheel and column inertia, damping, stiffness, and the dry friction, respectively. The steering column and torsion bar stiffness act as two linear springs in series because the spool valve is modeled as an element with negligible inertia.

The angular displacement of the spool valve $\theta_{sp}$ is a result of the torsion bar windup and is formulated as

$$\dot{\theta}_{sp} = \dot{\theta}_{sw} + \frac{1}{B_{sc}}\left[K_{sc}\left(\theta_{sw} - \theta_{sp}\right) - K_T\left(\theta_{tbar}\right)\right] \tag{4.6}$$

where $K_T$ denotes the torsion bar's stiffness and $\theta_{tbar}$ its angular displacement. The torsion bar twist results in the transmission of driver input torque to the pinion gear of the rack and pinion system. This pinion torque is transformed into the rack force that is resisted by the

feedback forces from the tire-road interface consisting of the aligning torque and tire-spin inertia. The governing equation for the rack displacement is formulated as

$$\ddot{y}_{rack} = \frac{1}{M_{rack}}\left[\frac{K_T}{R_p}\left(\theta_{tbar}\right) - B_{rack}\dot{y}_{rack} - \frac{2K_L}{N_L}\left(\frac{y_{rack}}{N_L} - \theta_{rw}\right) - F_{fr,rack} + F_{boost}\right] \tag{4.7}$$

where $y_{rack}$ is the rack displacement, $\theta_{rw}$ is the angular displacement of the front road wheels, and $F_{boost}$ is the power assist force modeled to be dependent on the torsion bar displacement. The parameters $K_L$ and $N_L$ are constants which represent the steering linkage stiffness and ratio of the steering wheel angle to road wheel angle, respectively. The terms $M_{rack}$, $B_{rack}$ and $F_{fr,rack}$ denote the rack's mass, damping and inherent friction modeled to be dependent on the rack velocity.

The torsion bar twist, which measures the relative displacement between the spool valve and the pinion gear, is formulated as

$$\theta_{tbar} = \theta_{sp} - \frac{y_{rack}}{R_p} \tag{4.8}$$

where $R_p$ denotes the radius of the pinion gear. Finally, the governing equation of motion for the wheel and linkage assembly is expressed as

$$\ddot{\theta}_{rw} = \frac{1}{I_w}\left[K_L\left(\frac{y_{rack}}{N_L} - \theta_{rw}\right) - B_w\dot{\theta}_{rw} - \tau_{fr,kp} - \tau_{fb}\right] \tag{4.9}$$

where $\tau_{fr,kp}$ and $\tau_{fb}$ denote the kingpin friction and aligning torques at the tire-road interface, respectively, and $I_w$ and $B_w$ denote the lumped inertia and damping of the wheel and linkage assembly, respectively.

By expressing this entire model as

$$I_{sw}\theta_{sw} + N_T\left(\theta_{sw}, \dot{\theta}_{sw}\right) = N_f\left(\tau_{sw}, \tau_{road}\right) \tag{4.10}$$

$N_T(\cdot)$ and $N_f(\cdot)$ can be designed to simulate the desired driving experience by adjusting the parameters that make up $N_T(\cdot)$ and $N_f(\cdot)$. This ensures that the reference model is capable to emulating multiple steering configurations as previously stated. Hence the dynamics given by (4.5)-(4.10) can now essentially function as a trajectory generator for the control design purposes of the physical steering system.

## Open-Loop Error System Development

To quantify the mismatches between the target system and the primary system or driver experience tracking error, as well as the primary and the secondary system or locked tracking error, filtered error signals, are defined as

$$r_1 = \dot{e}_1 + \mu e_1 \tag{4.11}$$

$$r_2 = \dot{e}_2 + \mu e_2 \tag{4.12}$$

where $\mu_1, \mu_2 \in \Re$ represent positive control gains, and the error signals $e_1(t), e_2(t) \in \Re$ are

$$e_1 = \theta_{sw} - \theta_1 \tag{4.13}$$

$$e_2 = \theta_1 - \theta_2 \tag{4.14}$$

After taking the first time derivative of (4.11) and (4.12), and substituting the dynamics in (4.1), (4.2) and (4.5), the open loop error systems are

$$I_1 \dot{r}_1 = Y_1 \phi_1 - T_1 \tag{4.15}$$

$$I_2 \dot{r}_2 = Y_2 \phi_2 - T_2 \tag{4.16}$$

where $Y_1(\cdot) \in \Re^{1\times r}$, $Y_2(\cdot) \in \Re^{1\times s}$ are regression matrices consisting of measurable quantities, and $\phi_1 \in \Re^{r\times 1}$, $\phi_2 \in \Re^{s\times 1}$ are constant unknown vectors. The reader is referred to Appendix E for explicit definitions of $Y_1(\cdot)$, $Y_2(\cdot)$, $\phi_1$ and $\phi_2$.

Remark 2 *Based on the definition of* $r_1(t)$ *and* $r_2(t)$ *given in (4.11) and (4.12), standard arguments [3] can be used to prove that: (i) if* $r_1(t)$, $r_2(t) \in \mathcal{L}_\infty$, *then* $e_1(t)$, $e_2(t)$, $\dot{e}_1(t)$, $\dot{e}_2(t) \in \mathcal{L}_\infty$, *and (ii) if* $r_1(t)$ *and* $r_2(t)$ *are asymptotically regulated, then* $e_1(t)$ *and* $e_2(t)$ *are asymptotically regulated.*

## Control Development

The first control objective requires the target following and the locked tracking error signals to approach zero asymptotically, while adapting for the system parameters that are assumed to be unknown. Further, the signals $\theta_1(t), \theta_2(t), \dot{\theta}_1(t), \dot{\theta}_2(t)$, $\tau_{sw}(t)$, $\tau_{road}(t) \in \mathcal{L}_\infty$ must be available for measurement.

### Control Formulation

Based on the subsequent stability analysis in next section and the structure of the open loop error system given in (4.15) and (4.16), the control inputs $T_1(t)$ and $T_2(t)$ are designed as

$$T_1 = k_1 r_1 + Y_1 \hat{\phi}_1 \tag{4.17}$$

$$T_2 = k_2 r_2 + Y_2 \hat{\phi}_2 \tag{4.18}$$

where $k_1, k_2 \in \Re$ are constant positive control gains, and $\hat{\phi}_1(t) \in \Re^{r\times 1}$, $\hat{\phi}_2(t) \in \Re^{s\times 1}$ are adaptive estimates for the unknown parameter matrices. The adaptive update laws are designed based on the subsequent stability analysis as

$$\dot{\hat{\phi}}_1 = \Gamma_1 Y_1^T r_1 \tag{4.19}$$

$$\dot{\hat{\phi}}_2 = \Gamma_2 Y_2^T r_2 \tag{4.20}$$

where $\Gamma_1 \in \Re^{r\times r}$, $\Gamma_2 \in \Re^{s\times s}$ are positive constant diagonal gain matrices.

Closed-Loop Error System Development

After substituting the control torques in (4.17) and (4.18) into the open-loop dynamics in (4.15) and (4.16), the closed-loop error system becomes

$$I_1 \dot{r}_1 = -k_1 r_1 + Y_1 \tilde{\phi}_1 \tag{4.21}$$

$$I_2 \dot{r}_2 = -k_2 r_2 + Y_2 \tilde{\phi}_2 \tag{4.22}$$

where the parameter estimation error signals, $\tilde{\phi}_1(t) \in \Re^{r\times 1}$, $\tilde{\phi}_2(t) \in \Re^{s\times 1}$ are defined as

$$\tilde{\phi}_1 = \phi_1 - \hat{\phi}_1 \tag{4.23}$$

$$\tilde{\phi}_2 = \phi_2 - \hat{\phi}_2 \tag{4.24}$$

Stability Analysis

Theorem 1 *Given the closed-loop system of (4.21) and (4.22), the tracking error signals defined in (4.9) and (4.10) are globally asymptotically regulated in the sense that*

$$\lim_{t\to\infty} e_1(t), e_2(t) = 0 \tag{4.25}$$

Proof: A non-negative, scalar function, denoted by $V(t) \in \Re$, is defined as

$$V = \frac{1}{2} I_1 r_1^2 + \frac{1}{2} I_2 r_2^2 + \frac{1}{2} \tilde{\phi}_1^T \Gamma_1^{-1} \tilde{\phi}_1 + \frac{1}{2} \tilde{\phi}_2^T \Gamma_2^{-1} \tilde{\phi}_2 \tag{4.26}$$

After taking the time derivative of (4.22) and making the appropriate substitutions from (4.17), (4.18), (4.19), and (4.20), the following expression is obtained

$$\begin{aligned}\dot{V} = r_1\left[-k_1 r_1 + Y_1\tilde{\phi}_1\right] + r_2\left[-k_2 r_2 + Y_2\tilde{\phi}_2\right] \\ -\tilde{\phi}_1^T\left[Y_1^T r_1\right] - \tilde{\phi}_2^T\left[Y_2^T r_2\right]\end{aligned} \tag{4.27}$$

where the fact that $\Gamma_1, \Gamma_2$ are constant diagonal gain matrices has been utilized along with the following equalities $\dot{\tilde{\phi}}_1 = -\dot{\hat{\phi}}_1$ and $\dot{\tilde{\phi}}_2 = -\dot{\hat{\phi}}_2$.

After canceling common terms, it is easy to see that we can upper bound $\dot{V}(t)$ as follows

$$\dot{V} \leq -k_1 r_1^2 - k_2 r_2^2 \tag{4.28}$$

From (4.28) and (4.26), it is straightforward to see that $r_1(t),\ r_2(t),\ \tilde{\phi}_1(t),\ \tilde{\phi}_2(t) \in \mathcal{L}_\infty$. After utilizing (4.19), (4.20), and Remark 2, we can conclude that $e_1(t),\ \dot{e}_1(t),\ e_2(t),\ \dot{e}_2(t),\ \hat{\phi}_1(t),\ \hat{\phi}_2(t) \in \mathcal{L}_\infty$. Using Remark 2, (4.13), (4.14) and their first derivatives, it is clear that $\theta_1(t),\ \theta_2(t),\ \dot{\theta}_1(t),\ \dot{\theta}_2(t) \in \mathcal{L}_\infty$. From the explicit definition for $Y_1(\cdot)$ given in Appendix A and using the fact that $\tau_{sw}(t),\ \tau_{road}(t) \in \mathcal{L}_\infty$, it is easy to see that $Y_1(\cdot) \in \mathcal{L}_\infty$. From (4.17), it is clear that the control torque $T_1(\cdot) \in \mathcal{L}_\infty$. Again, from the definition of $Y_2(\cdot)$ in Appendix A and from the above facts, $Y_2(\cdot) \in \mathcal{L}_\infty$. From (4.18), it is clear that $T_2(\cdot) \in \mathcal{L}_\infty$. Using standard signal chasing arguments, it can be shown that all the signals in the closed-loop system remain bounded. In particular, from (4.21) and (4.22), $\dot{r}_1(t), \dot{r}_2(t) \in \mathcal{L}_\infty$. After employing a corollary to Barbalat's Lemma [36], it is easy to show that

$$\lim_{t\to\infty} r_1(t),\ r_2(t) = 0$$

Finally, Remark 2 can be used to prove the result stated in (4.25).

# CHAPTER FIVE

# HAPTIC INTERFACE ADAPTIVE CONTROLLER SIMULATION RESULTS AND PROPOSED TEST CONFIGUARATION

## Numerical Simulation Results

Numerical simulations were performed to study the performance of the control algorithms developed in Chapter 4 by controlling a automotive steering system with an adaptive controller. The simulated vehicle steering system was assumed to have the dynamics described by (4.1) and (4.2). The nonlinear stiffness, damping and friction functions were chosen as

$$N_i\left(\theta_i,\dot{\theta}_i\right) = B_i\theta_i + K_i\theta_i + \mathrm{T}_{sci}\,\mathrm{sgn}\left(\dot{\theta}_i\right) \qquad i = 1,2 \tag{5.1}$$

The system parameters are listed in Table 5.1

Table 5.1 List of Simulation Parameters and Corresponding Values

| Variable | Value | Units | Variable | Value | Units |
|---|---|---|---|---|---|
| $I_1$ | $1.16\times10^{-2}$ | *kg-m²* | $B_{rack}$ | $1\times10^{4}$ | *N-m-sec/rad* |
| $B_1$ | 0.19 | *N-m-sec/rad* | $B_{sc}$ | 0.356 | *N-m-sec/rad* |
| $K_1$ | 0 | *N-m/rad* | $B_w$ | 200 | *N-m* |
| $\mathrm{T}_{sc1}$ | 0.1 | *N* | $F_{fr,rack}$ | 240.484 | *N* |
| $\alpha_1$ | 0.1 | - | $I_{sw}$ | $6.78\times10^{-5}$ | *kg-m²* |
| $I_2$ | 0.235 | *kg-m²* | $I_w$ | 0.356 | *kg-m²* |
| $B_2$ | 0.6 | *N-m-sec/rad* | $K_L$ | $48.816\times10^{3}$ | *N-m/rad* |
| $K_2$ | 1 | *N-m/rad* | $K_{sc}$ | 33.398 | *N-m/rad* |
| $\mathrm{T}_{sc2}$ | 0.5 | *N* | $K_T$ | 75.797 | *N-m/rad* |

Table 5.1 List of Simulation Parameters and Corresponding Values (Continued)

| | | | | | |
|---|---|---|---|---|---|
| $\alpha_2$ | 1 | - | $M_{rack}$ | 29.412 | $Kg$ |
| $M_{z3}$ | 0.3655 | - | $N_L$ | 0.11816 | - |
| $M_{z2}$ | 0.027 | - | $R_p$ | $8\times10^{-3}$ | $M$ |
| $M_{z1}$ | 62.467 | - | $\tau_{fr,sc}$ | 0.6 | $N$ |
| $M_{z0}$ | $7\times10^{-5}$ | - | $\tau_{fr,kp}$ | 0.5 | $N$ |

The power assist forces, dry friction, inherent friction and kingpin friction are expressed in the form of lookup tables are shown in Figures 5.1, 5.2, 5.3 and 5.4 respectively.

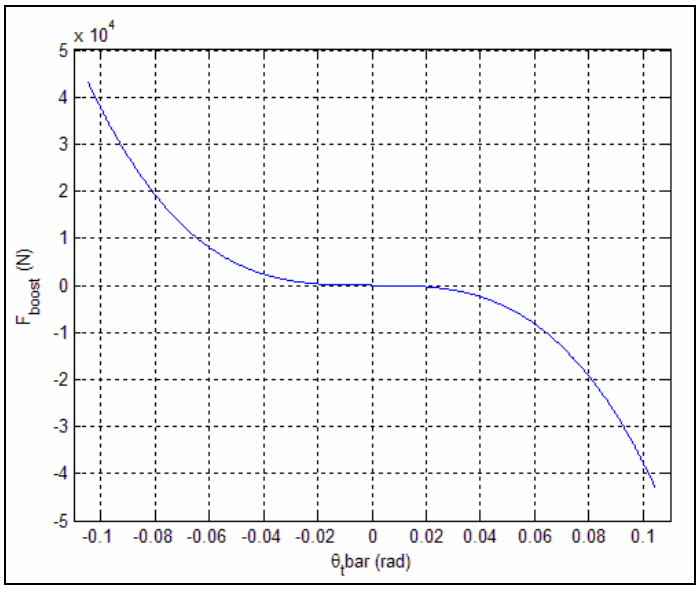


Figure 5.1. Boost Force $F_{boost}$ versus torsion bar displacement $\theta_{tbar}$

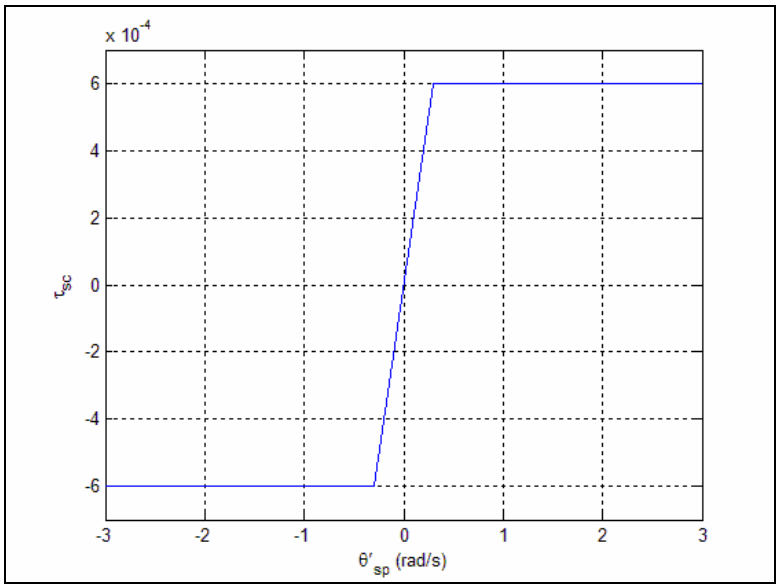


Figure 5.2 Dry Friction $\tau_{sc}$ versus spool valve angular velocity $\dot{\theta}_{sp}$

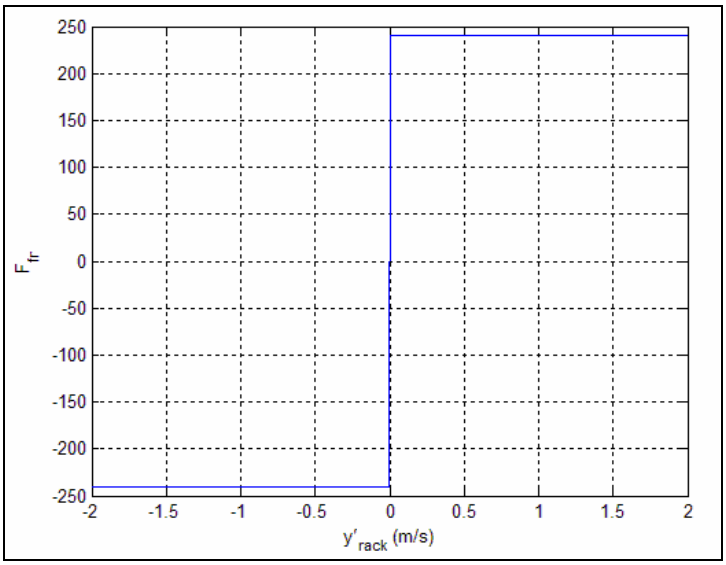


Figure 5.3 Inherent Friction $F_{fr,rack}$ versus rack velocity $\dot{y}_r$

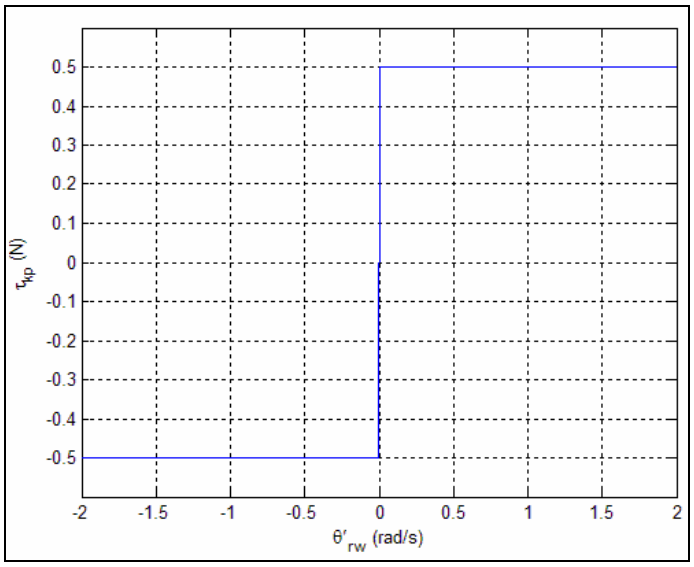


Figure 5.4. Kingpin friction $\tau_{fr,kp}$ versus rack velocity $\dot{\theta}_{rw}$

The reference trajectory was generated as previously discussed in Chapter 4. The parameters were chosen to model a commercial vehicle.

The reaction torque applied on the directional control assembly (due to the tire-road interface forces), was assumed to be related to the angular deflection of the road wheel angle of the reference model in the following manner

$$\tau_{fb} = M_{z3}\theta_{rw}^3 - M_{z2}\theta_{rw}^2 - M_{z1}\theta_{rw} + M_{z0}$$

The relationship data was obtained using the CarSim™ software package and curve fit using Microsoft Excel™. Numerical Simulations were performed for two driver input torque profiles: *Case 1:* $\tau_1(t) = 0.8\sin(5t)(1-\exp(-3t))$ which represents the input to perform a standard slalom maneuver; and *Case 2:* $\tau_2(t) = 0.9(1-\exp(-3t))$ which represents the input to follow a circular trajectory as shown in Figure 5.5.

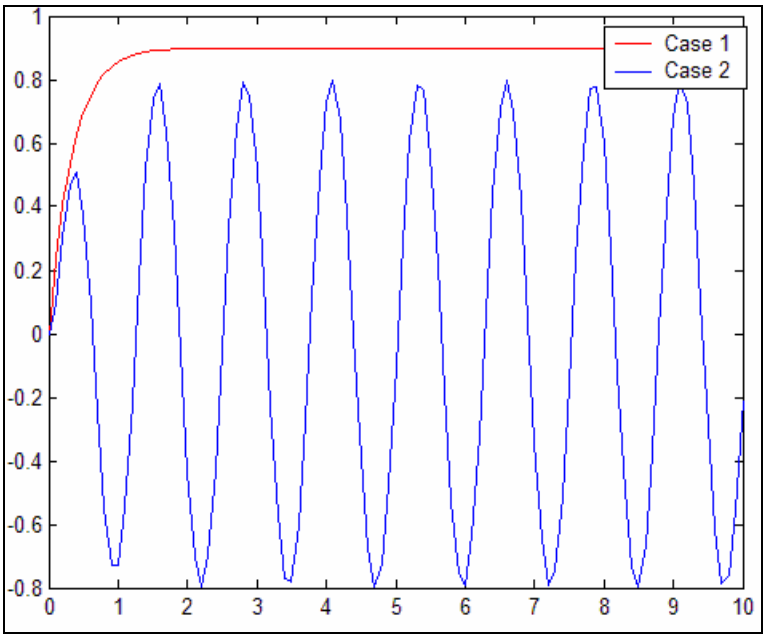


Figure 5.5 Driver input torque profiles

Case 1: Standard slalom maneuver

The set of simulations use a driver input torque profile that represents a standard slalom maneuver. All adaptive estimates were initialized to zero in this simulation. The driver experience and locked tracking errors, $e_1(t)$ and $e_2(t)$, are presented in Figure 5.6 and 5.7. The driver experience tracking error corresponds to the differences between the reference model and the primary subsystem of the haptic interface steer-by-wire system. As shown, the error $e_1(t)$ approaches zero after $t = 1\,\text{sec}$ which implies that the driver experiences the desired "feel" as specified by the reference model parameters (which as previously stated, correspond to a conventional hydraulic steering system of a commercial vehicle). The locked tracking error $e_2(t)$ also approaches zero, which demonstrates that the driver's steering

commands are followed by the directional control assembly. These two facts prove that the control algorithm achieves the two goals outlined in the control objective. All the parameter estimates were observed to approach constant values as shown in Figures 5.8 to 5.17. The corresponding motor control torques are displayed in Figure 5.18 and 5.19.

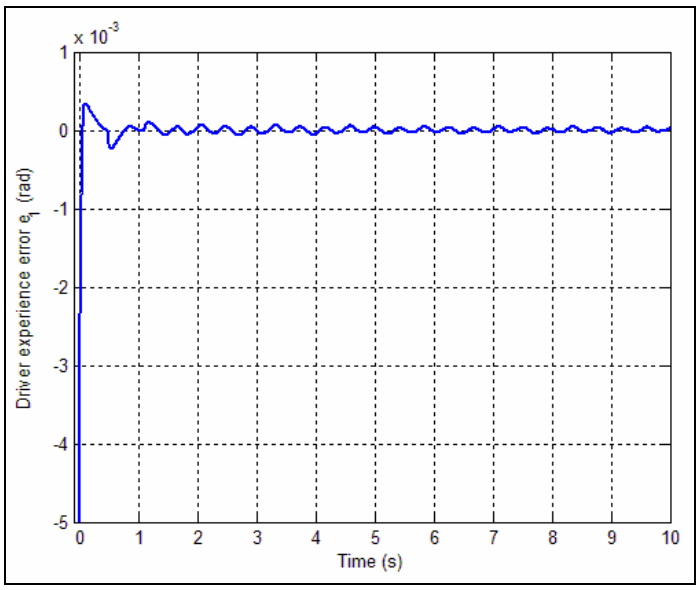


Figure 5.6 Driver experience error $e_1(t)$

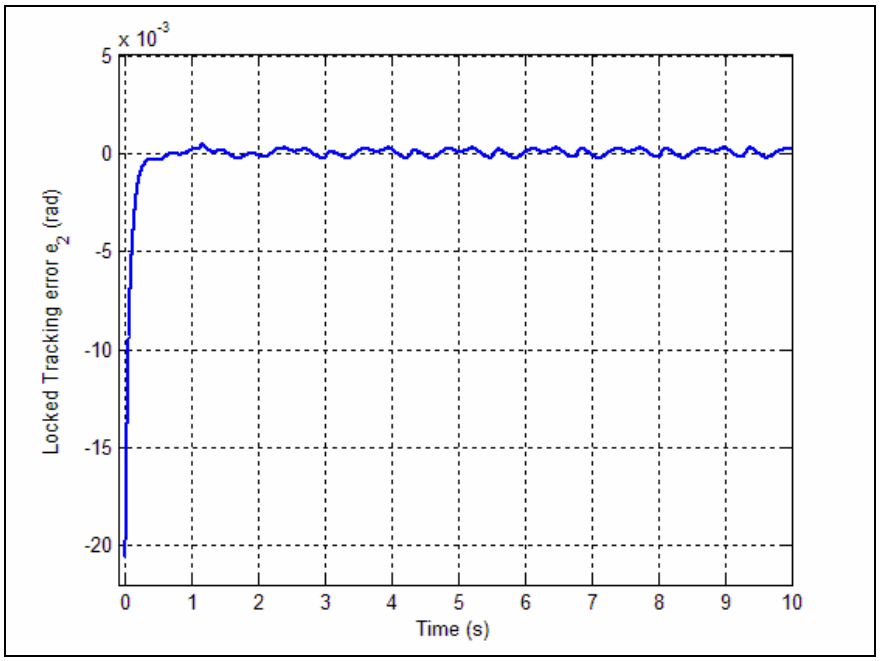


Figure 5.7 Locked tracking error $e_2(t)$

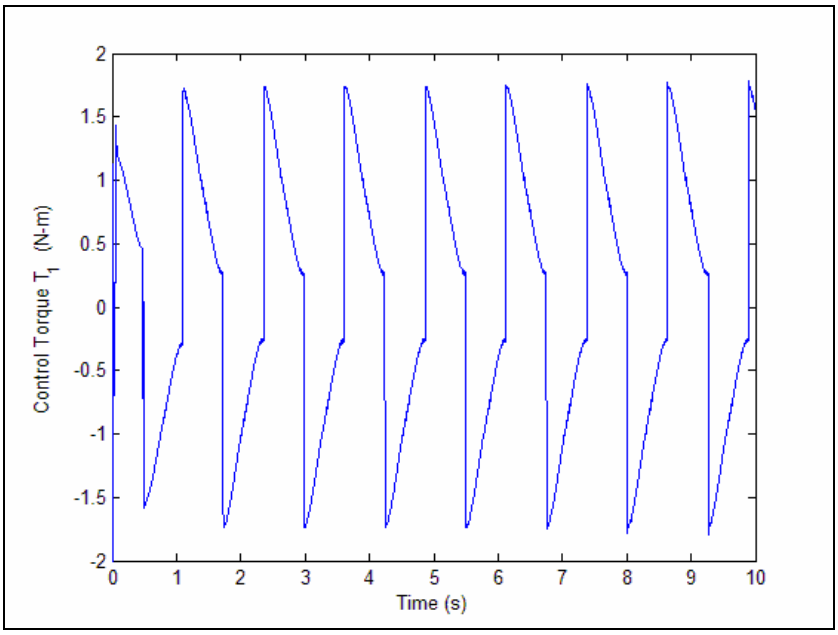


Figure 5.8 Control torque $T_1(t)$

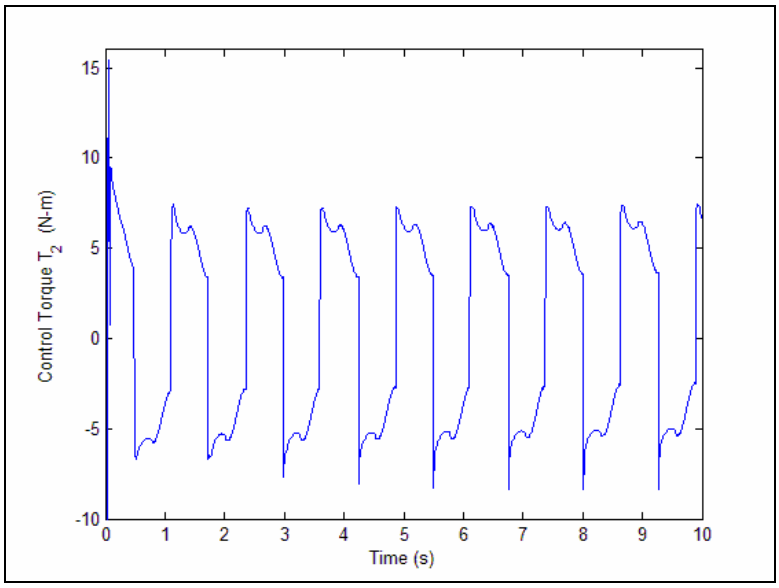


Figure 5.9 Control torque $T_2(t)$

Case 2: Circle Following maneuver

The set of simulations use a driver input torque profile that represents a standard slalom maneuver. All adaptive estimates were initialized to zero in this simulation. The driver experience and locked tracking errors, $e_1(t)$ and $e_2(t)$, are presented in Figure 5.10 and 5.11. The driver experience tracking error corresponds to the differences between the reference model and the primary subsystem of the haptic interface steer-by-wire system. As shown, the error $e_1(t)$ approaches zero after $t = 1\,\text{sec}$ which implies that the driver experiences the desired "feel" as specified by the reference model parameters (which as previously stated, correspond to a conventional hydraulic steering system of a commercial vehicle). The locked tracking error $e_2(t)$ also approaches zero, which demonstrates that the driver's steering commands are followed by the directional control assembly. These two facts

prove that the control algorithm achieves the two goals outlined in the control objective. All the parameter estimates were observed to approach constant values as shown in Appendix E. The corresponding motor control torques are displayed in Figure 5.12 and 5.13.

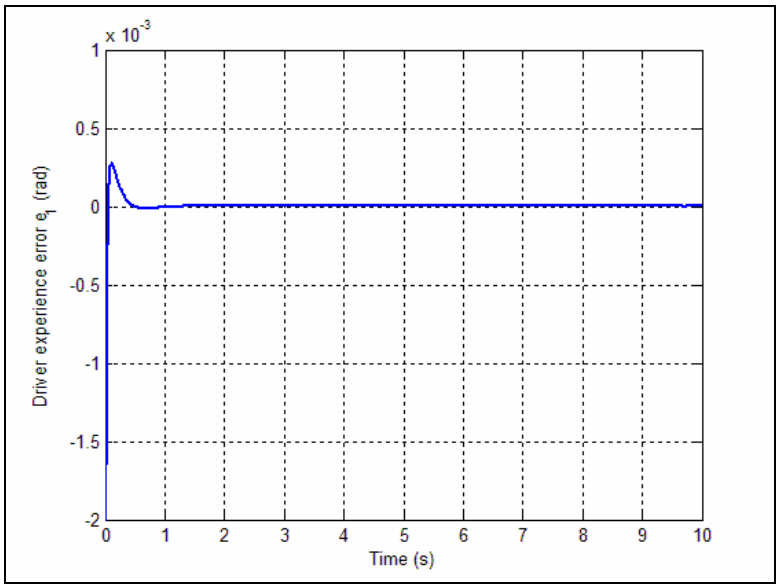


Figure 5.10 Driver Experience error $e_1(t)$

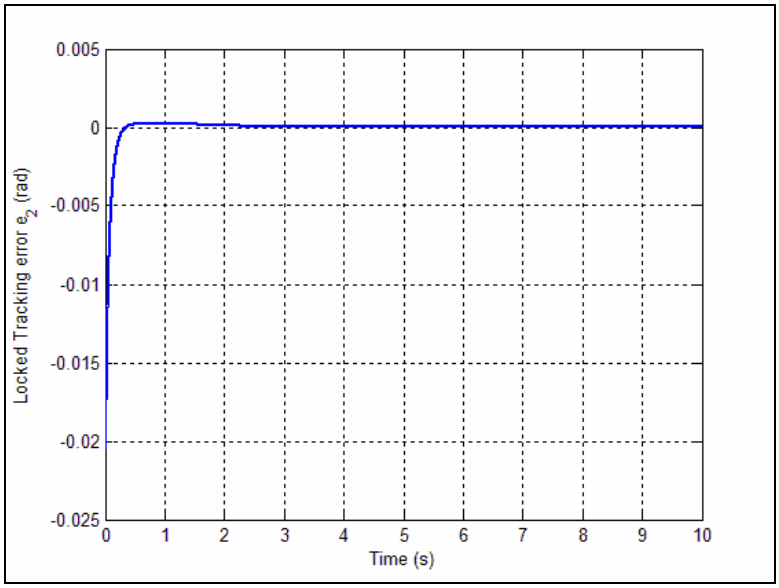


Figure 5.11 Locked Tracking error $e_2(t)$

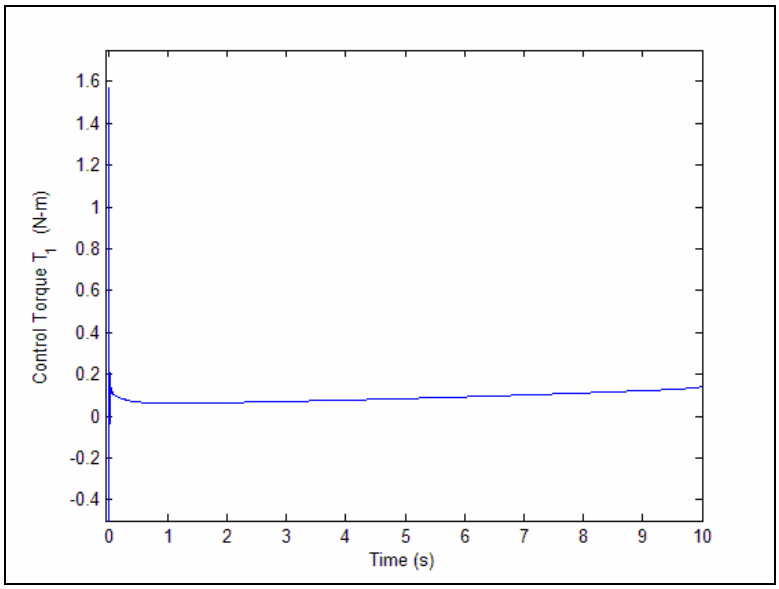


Figure 5.12 Control torque $T_1(t)$

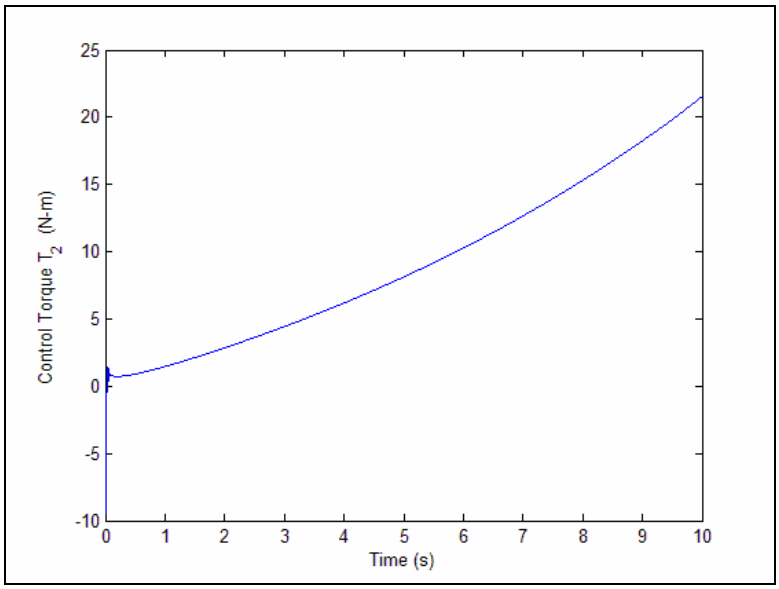


Figure 5.13 Control torque $T_2(t)$

## Proposed test configuration for a haptic interface in a steer-by-wire vehicle

The long-term goal is to utilize the designed adaptive controller for a haptic interface in an actual commercial vehicle. In this section an in-vehicle test configuration will be designed and some hardware criteria will stated. The steer-by-wire system consists of (i) a feedback subsystem: a motor and torque sensor at the steering wheel, (ii) a drive subsystem: a motor and torque sensor at the rack and pinion to control the road wheels, and (iii) a rapid control prototyping board. Installation of the steer-by-wire system in the vehicle will require that the some of the components in the hood of the commercial vehicle be removed or relocated. The factory standard hydraulic rack and pinion system will be replaced by an EPS rack and pinion. The air box fastened to the vehicle firewall will be removed to make room for the mounting of the EPS rack and pinion and drive motor. The steering column will be removed or shortened and replaced with the feedback subsystem. The feedback subsystem will be mounted on one of the two universal joints located inside the cabin of the car.

### Feedback subsystem

The purpose of the feedback system is to provide the feedback to the operator. The system consists of a torque sensor which measures the driver input torque $\tau_{sw}$, an incremental encoder to measure steering wheel angular displacement $\theta_1(t)$ and the feedback motor which is actuated by the control torque $T_1(t)$. The feedback motor must possess enough torque to adequately represent the aligning torque and friction felt at the road wheels to the driver. At the minimum, a motor with a max torque output of 5 NM would suffice. The feedback torque sensor must have enough resolution to detect relatively small changes

in driver input torque. To this end, the torque must be able to have a resolution of 0.1 NM at a maximum of 5 NM. The proposed feedback system hardware is listed in Table 5.2.

Table 5.2 Hardware present in the feedback subsystem

| Control Signals/Inputs | Part Name | Uses | Resolution |
|---|---|---|---|
| $\theta_1(t)$ | BEI Incremental encoder Model H25 | Measures Position of the feedback motor | 72,000 cycles per Shaft Turn |
| $\tau_{sw}(t)$ | Methode 2000 Square Drive Sensor | Measures driver- and tire-feedback torque | 5 NM at 0.1 NM resolution |
| $T_1(t)$ | NSK Megatorque System | Provides feedback to the driver | Provides 0-5 NM |

Drive subsystem

The purpose of the drive system is to ensure that the driver's steering angle commands are accurately followed by the road wheels. The system consists of a torque sensor which measures the reaction torque $\tau_{road}$, an incremental encoder to measure steering wheel angular displacement $\theta_2(t)$ and the feedback motor which is actuated by the control torque $T_2(t)$. The drive motor must be powerful enough to adequately turn the road wheels in high torque input maneuvers such as parking and a turn from rest. At the minimum, a motor must be capable of generating 70NM of torque. The reaction torque sensor must have enough resolution to detect relatively small changes in $\tau_{road}$. To this end, the torque must be able to have a resolution of 1 NM. The incremental encoder must possess enough resolution to detect small changes in the angular displacement of the vehicle directional control assembly. The proposed drive system hardware is listed in Table 5.3.

Table 5.3 Hardware present in the drive subsystem

| Control Signals/Inputs | Part Name | Uses | Resolution |
|---|---|---|---|
| $\theta_2(t)$ | BEI Incremental encoder Model H25 | Measures Position of the drive motor | 72,000 cycles per Shaft Turn |
| $\tau_{road}$ | Methode 2000 Square Drive Sensor | Measures driver- and tire-feedback torque | 70 NM at 1 NM resolution |
| $T_2(t)$ | EPS Motor | Actuates the vehicle directional control assembly | Provides 0-70 NM |

## Rapid control prototyping board

Rapid control prototyping is defined as evaluating the performance of a software controller in a real plant. In this case, the plant is the steer-by-wire steering system consisting of the feedback and drive subsystems; and the software controller is the model reference adaptive controller discussed in Chapter 4. A rapid control prototyping processor is therefore needed for real time calculations of control inputs. The processor must meet three criteria: (i) must be compatible with the sensors and motors previously discussed, (ii) must be reconfigurable, i.e. allow for on-the-fly parameter modification, (iii) must consume relatively low power. The proposed rapid control prototyping board is the dSPACE 1103 board which has the features: (i) PowerPC 750 GX 1GHz processor; (ii) 32MB local RAM and 96MB Global RAM of memory; (iii) 20 analog inputs; (iv) 8 analog outputs; (v) 32 digital I/Os; and (vi) 6 digital incremental encoder inputs.

## Power consumption

The amount of power that will be consumed by the necessary apparatus is issue that always a source of concern. In Table 5.4, an illustration of estimated power consumption by each component is listed.

Table 5.3 Theoretical power scheme for the steer-by-wire system

| Apparatus | Voltage (V) | Current (A) | Power (W) | Total Power (W) |
|---|---|---|---|---|
| Torque sensor (2 .ea) | 5 | 0.02 | 0.1 | 0.2 |
| Incremental Encoder (2 .ea) | 5 | 0.25 | 1.25 | 2.5 |
| Feedback motor | 12 | 12 | 144 | 144 |
| Drive motor | 12 | 30 | 360 | 360 |
| dSPACE 1103 board | - | - | 20 | 20 |
| **Total** | | | | 526.7 |

To satisfy the extra power requirement, a replacement alternator, which provides more power as well as a power inverter which converts battery power to usable AC power are proposed. These devices can be hardwired directly to the vehicle battery. The proposed inverter is the Vector VEC049 which has a maximum output power of 1,000W and has three standard outlets. It is of medium weight which increases the plausibility of using the device as an in-vehicle power source. A high-level block diagram of the entire steer-by-wire system architecture is illustrated in Figure 4.2 of the previous chapter.

# CHAPTER SIX

# CONCLUSION

The science of haptics is an emerging technology with several applications in areas of robotics, automotive systems and consumer technology. This science is explored in this thesis by focusing on two areas: robotics and automotive systems. The experiments performed involved using a robotic manipulator as an anisotropic device to aid patients with neuromuscular diseases with rehabilitation, as well as designing and controlling a haptic interface for a steer-by-wire vehicle using the model reference adaptive control (MRAC) approach.

In the first experiment, a path planner is developed based on an anisotropic force-velocity relationship that generates a bounded desired trajectory in the robot workspace given the interaction force at the end-effector while avoiding singularities, joint limits, and obstacles. The proposed path generator also ensures safety by maintaining the desired net flow of energy during the human robot interaction from the user toward the manipulator. A Lyapunov based control strategy is proposed to achieve semi-global asymptotically stable path following for the robot manipulator in the presence of uncertainty in the robot dynamics. The control algorithm also ensures sufficiently rapid error convergence at the end-effector such that the actual energy transfer profile follows the desired energy transfer profile. Simulation results using a 2-D planar elbow as well as a three degree-of-freedom revolute manipulator are provided to illustrate the performance of the proposed approach.

In the second experiment, the design of an adaptive, nonlinear tracking controller for a haptic interface in a steer-by-wire vehicle has been presented which ensures that: i) the

directional control assembly follows the driver commanded input, and ii) the dynamics of the driver input device follows that of a four degree-of-freedom target system. A complete stability analysis, using Lyapunov-based techniques, has been presented to demonstrate that the proposed control law guarantees global asymptotic regulation of the "locked tracking error" and the "driver experience tracking error". This in turn, infers that the driver experiences a steering feel determined by the choice of the target reference model.

# APPENDICES

## Appendix A
## Proof of Bound on $\tilde{N}$

We start by writing $\tilde{N}(t)$ from (2.31) and (2.33) as follows

$$\begin{aligned}\tilde{N} = &\left[\bar{M}(x) - \bar{M}(x_d)\right]\ddot{x}_d + \left[\dot{\bar{M}}(x,\dot{x}) - \dot{\bar{M}}(x_d,\dot{x}_d)\right]\ddot{x}_d \\ &+ \left[\dot{\bar{B}}(x,\dot{x},\ddot{x}) - \dot{\bar{B}}(x_d,\dot{x}_d,\ddot{x}_d)\right] + \bar{M}(x)(\dot{e}_1 + \dot{e}_2) + \dot{\bar{M}}(x,\dot{x})\left(\frac{1}{2}r - \ddot{e}_1\right) + e_2\end{aligned} \tag{2.49}$$

To simplify the notation, we define the following auxiliary functions

$$\begin{aligned}\Phi_{bf}(x,\dot{x},\ddot{x}) &\triangleq \dot{\bar{B}}(x,\dot{x},\ddot{x}) \\ \Phi_{mf}(x,\dot{x},\ddot{x}_d) &\triangleq \dot{\bar{M}}(x,\dot{x})\ddot{x}_d\end{aligned} \tag{2.50}$$

$$E = \bar{M}(\cdot)\ddot{e}_1 + \bar{M}(\cdot)\dot{e}_2 + e_2 + \dot{\bar{M}}(\cdot)\frac{1}{2}r - \dot{\bar{M}}(\cdot)\ddot{e}_1 \tag{2.51}$$

From (2.23)-(2.25), it is possible to write

$$\dot{e}_1 = e_2 - e_1 \quad \dot{e}_2 = r - e_2 \quad \ddot{e}_1 = r - 2e_2 + e_1$$

Given the definitions of (2.50) and (2.51), we can rewrite (2.49) by adding and subtracting a bevy of terms as follows

$$\begin{aligned}\tilde{N} = &\left[\bar{M}(x) - \bar{M}(x_d)\right]\ddot{x}_d + \left[\Phi_{mf}(x,\dot{x},\ddot{x}_d) - \Phi_{mf}(x_d,\dot{x},\ddot{x}_d)\right] \\ &+ \left[\Phi_{mf}(x_d,\dot{x},\ddot{x}_d) - \Phi_{mf}(x_d,\dot{x}_d,\ddot{x}_d)\right] + \left[\Phi_{bf}(x,\dot{x},\ddot{x}) - \Phi_{bf}(x_d,\dot{x},\ddot{x})\right] \\ &+ \left[\Phi_{bf}(x_d,\dot{x},\ddot{x}) - \Phi_{bf}(x_d,\dot{x}_d,\ddot{x})\right] + \left[\Phi_{bf}(x_d,\dot{x}_d,\ddot{x}) - \Phi_{bf}(x_d,\dot{x}_d,\ddot{x}_d)\right]\end{aligned} \tag{2.52}$$

Given Assumption 1, we can apply the Mean Value Theorem [8] to each bracketed term of (2.49) as follows

$$\begin{aligned}\tilde{N} = &\left.\frac{\partial \Phi_{M_f}\left(\sigma_1\right)}{\partial \sigma_1}\right|_{\sigma_1=\zeta_1} e_1 \dddot{x}_d + \left.\frac{\partial \Phi_{mf}\left(\sigma_2, \dot{x}, \ddot{x}_d\right)}{\partial \sigma_2}\right|_{\sigma_2=\zeta_2} e_1 \\ &+ \left.\frac{\partial \Phi_{mf}\left(x_d, \sigma_3, x_d\right)}{\partial \sigma_3}\right|_{\sigma_3=\zeta_3} \dot{e}_1 + \left.\frac{\partial \Phi_{bf}\left(\sigma_4, \dot{x}, \ddot{x}\right)}{\partial \sigma_4}\right|_{\sigma_4=\zeta_4} e_1 \\ &+ \left.\frac{\partial \Phi_{bf}\left(x_d, \sigma_5, \ddot{x}\right)}{\partial \sigma_5}\right|_{\sigma_5=\zeta_5} \dot{e}_1 + \left.\frac{\partial \Phi_{bf}\left(x_d, \dot{x}_d, \sigma_6\right)}{\partial \sigma_6}\right|_{\sigma_6=\zeta_6} \ddot{e}_1 + E\end{aligned} \tag{2.53}$$

where $\zeta_1(t), \zeta_2(t), \zeta_3(t) \in (x, x_d)$, $\zeta_3(t), \zeta_5(t) \in (\dot{x}, \dot{x}_d)$ while $\zeta_6(t) \in (\ddot{x}, \ddot{x}_d)$. From the preceding analysis, the right-hand side of (2.53) can be succinctly expressed as

$$\tilde{N} = \Phi_z \tag{2.54}$$

where $z(t) \in \mathbb{R}^{9\times 1}$ is the composite error vector that has previously been defined and $\Phi(x, \dot{x}, \ddot{x}, t) \in \mathbb{R}^{3\times 9}$ is the first-order differentiable system regressor. By virtue of its first-order differentiability, $\Phi(\cdot)$ can be upper-bounded as follows

$$\Phi(x, \dot{x}, \ddot{x}, t) \le \bar{\rho}(x, \dot{x}, \ddot{x}) \tag{2.55}$$

where $\bar{\rho}(\cdot)$ is a positive function non-decreasing in $x(t)$, $\dot{x}(t)$, and $\ddot{x}(t)$. Given Assumption 3, we can utilize (2.55) and the facts that

$$\begin{aligned} x &= x_d - e_1 \\ \dot{x} &= \dot{x}_d - e_2 + e_1 \\ \ddot{x} &= \ddot{x}_d - r + 2e_2 - e_1 \end{aligned}$$

in order to upper-bound $\tilde{N}(\cdot)$ as follows

$$N \le \rho(\|z\|)\|z\|$$

where $\rho(\|z\|)$ is some positive function non-decreasing in $\|z\|$.

## Appendix B
## Proof of Lemma

After substituting (2.25) into (2.38) and then integrating in time, we obtain

$$\begin{aligned}\int_{t_0}^{t} L(\tau)\,d\tau = \int_{t_0}^{t} e_2^T(\tau)\left(N_{1d}(\tau) - \beta_1 \operatorname{sgn}\left(e_2(\tau)\right)\right)d\tau + \int_{t_0}^{t} \frac{de_2^T(\tau)}{d\tau} N_{1d}(\tau)\,d\tau \\ - \beta_1 \int_{t_0}^{t} \frac{de_2^T}{d\tau} \operatorname{sgn}\left(e_2(\tau)\right)d\tau\end{aligned} \tag{2.56}$$

After integrating the second term on the right-hand side of (2.56) by parts, we obtain the following simplified expression

$$\begin{aligned}\int_{t_0}^{t} L(\tau)\,d\tau = \int_{t_0}^{t} e_2^T(\tau)\left(N_{1d}(\tau) - \frac{dN_{1d}(\tau)}{d\tau} - \beta_1 \operatorname{sgn}\left(e_2(\tau)\right)\right)d\tau \\ + e_2^T(t) N_{1d}(t) - e_2^T(t_0) N_{1d}(t_0) - \beta_1 \left\|e_2(t)\right\|_1 + \beta_1 \left\|e_2(t_0)\right\|_1\end{aligned} \tag{2.57}$$

We can now upper bound the right-hand side of (2.57) as follows

$$\begin{aligned}\int_{t_0}^{t} L(\tau)\,d\tau \le \int_{t_0}^{t} \left\|e_2(\tau)\right\|_1 \left(\left\|N_{1d}(\tau)\right\| + \left\|\frac{dN_{1d}(\tau)}{d\tau}\right\| - \beta_1\right)d\tau \\ + \left\|e_2(t)\right\|_1 \left(\left\|N_{1d}(t)\right\| - \beta_1\right) - e_2^T(t_0) N_{1d}(t_0) + \beta_1 \left\|e_2(t_0)\right\|_1\end{aligned} \tag{2.58}$$

From (2.58), it is easy to see that if $\beta_1$ is chosen according to (2.39), then (2.40) holds.

## Appendix C
## Proof of Theorem 1

Let us define two auxiliary functions $P_i(t) \in \Re$ as follows

$$P_i(t) \triangleq \zeta_{bi} - \int_{t_0}^{t} L_i(\tau) d\tau \geq 0 \; \forall i = 1,2 \tag{2.59}$$

where $\zeta_{bi}, L_i(t)$ have been previously defined in Lemmas 1 and 2. Based on the non-negativity of $P_i(t)$ above, one can define a nonnegative function $V_1(t)$ as follows

$$V_1 \triangleq \frac{1}{2} e_1^T e_1 + \frac{1}{2} e_2^T e_2 + \frac{1}{2} r^T \bar{M} r + P_1 + P_2 \tag{2.60}$$

After taking the time derivative of (2.60) and utilizing the definitions of (2.23-25) as well as the closed loop dynamics of (2.37), we can conveniently rearrange the terms to obtain the following expression for $\dot{V}_1(t)$

$$\begin{aligned} \dot{V}_1 = &-\|e_1\|^2 - \|e_2\|^2 - (k_s + 1)\|r\|^2 + e_1^T e_2 + r^T \tilde{N} - \beta_2 e_2^T \operatorname{sgn}(e_2) \\ &+ \left[ r^T \left( N_{1d} - \beta_1 \operatorname{sgn}(e_2) \right) - L_1 \right] - \left[ \dot{e}_2^T \beta_2 \operatorname{sgn}(e_2) + L_2 \right] \end{aligned} \tag{2.61}$$

where we have utilized the definition of (2.59). After utilizing the definitions of (2.38) and (2.42) to eliminate the bracketed terms in the above equality, we can utilize simple algebraic manipulations to obtain the following upper-bound for $\dot{V}_1(t)$

$$\dot{V}_1 \leq -\frac{1}{2}\|z\|^2 + \left[ \|r\| \rho(\|z\|) \|z\| - k_s \|r\|^2 \right] - \beta_2 \|e_2\|_1$$

where $z(t)$ is a composite error vector that has been defined previously in (2.32). Applying the nonlinear damping argument [10] to the bracketed term above, we obtain the following upper bound for $\dot{V}_1(t)$

$$\dot{V}_1 \leq -\frac{1}{2}\left[ 1 - \frac{\rho^2(\|z\|)}{2k_s} \right] \|z\|^2 - \beta_2 \|e_2\|_1 \tag{2.62}$$

From (2.62), it is possible to state that

$$\left.\begin{array}{l} \dot{V}_1 \leq -\alpha \left\| z \right\|^2 \\ \dot{V}_1 \leq -\beta_2 \left\| e_2 \right\|_1 \end{array}\right\} \text{for } k_s > \frac{1}{2}\rho^2\left(\left\| z \right\|\right) \tag{2.63}$$

where $\alpha \in \Re$ is some positive constant of analysis. We note here that it is possible to express the lower-bound on $k_s$ in terms of the initial conditions of the problem which has been referred to in literature as a semi-global stability result. We refer the interested reader to Appendix D for the details of such a procedure. Here onward, our analysis is valid in the region of attraction denoted by $\Omega_c$ in (2.67). From (2.63) and the analysis in Appendix C, it is easy to see that $z(t) \in \mathcal{L}_\infty \cap \mathcal{L}_2$ and $\lim_{t\to\infty} \left\| z \right\|^2 = 0$. From the previous assertions and the definitions of (2.24), (2.25), and (2.32), one readily obtains the result of (2.44).

## Appendix D
## Calculation of Region of Attraction

Following [27], we now define the region of attraction for the system. From (2.63), we obtain the following sufficient condition for the negative definiteness of $\dot{V}(t)$

$$\|z\| < \rho^{-1}\left(\sqrt{2k_s}\right) \tag{2.64}$$

Next, we define $\eta(t) = \left[ z^T(t) \quad \sqrt{P_1(t)} \quad \sqrt{P_2(t)} \right]^T \in \Re^{11}$ and a region $\Omega$ in state space as follows

$$\Omega = \left\{ \eta \in \mathbb{R}^{11} \middle| \|\eta\| < \rho^{-1}\left(\sqrt{2k_s}\right) \right\} \tag{2.65}$$

where the definition of $\eta(t)$ indicates that $\Omega$ is a subset of the space defined by (2.64). Based on Assumption 3 in Section 4, we define $\delta_1 \triangleq \frac{1}{2}\min\{1, \underline{m}\}$ and $\delta_2(x) \triangleq \frac{1}{2}\max\left\{\frac{1}{2}\overline{m}(x), 1\right\}$; thereby, (2.60) can be upper and lower bounded as

$$\xi_1(\eta) \le V \le \xi_2(\eta) \tag{2.66}$$

where $\xi_1(\eta) \triangleq \delta_1 \|\eta\|^2 \in \Re$ and $\xi_2(\eta) \triangleq \delta_2(x)\|\eta\|^2 \in \Re$. From the boundedness conditions above, we can further find an estimate for the region of attraction of the system as

$$\Omega = \left\{ \eta \in \Omega \middle| \xi_2(\eta) < \delta_1\left(\rho^{-1}\left(\sqrt{2k_s}\right)\right)^2 \right\} \tag{2.67}$$

Given (2.66) and (2.63), we can invoke Lemma 2 of [27] to state that

$$\|z\|^2 \to 0 \text{ as } t \to \infty \ \ \forall\ \eta(t_0) \in \Omega_c \tag{2.68}$$

From (2.67), we require

$$\xi_2(\eta(t)) < \delta_1\left(\rho^{-1}\left(\sqrt{2k_s}\right)\right)^2 \tag{2.69}$$

which implies that we can write (2.69) in terms of system initial conditions as follows

$$\left\|\eta\left(t_0\right)\right\| < \sqrt{\frac{\delta_1}{\delta_2\left(x\left(t_0\right)\right)}}\rho^{-1}\left(\sqrt{2k_s}\right) \tag{2.70}$$

where we have taken advantage of the fact that $V\left(t\right)$ is either decreasing or constant for all time. We can rewrite (2.70) in terms of an lower-bound on $k_s$ as follows

$$k_s > \frac{1}{2}\rho^2\left(\sqrt{\frac{\delta_2\left(x\left(t_0\right)\right)}{\delta_1}}\left\|n\left(t_0\right)\right\|\right) \tag{2.71}$$

Given the definition of $\eta\left(t\right)$, we can write

$$\begin{aligned}\left\|n\left(t_0\right)\right\| = \Big(&e_1^T\left(t_0\right)e_1\left(t_0\right) + e_2^T\left(t_0\right)e_2\left(t_0\right) + \left[\dot{e}_2\left(t_0\right) + e_2\left(t_0\right)\right]^T\left[\dot{e}_2\left(t_0\right) + e_2\left(t_0\right)\right] \\ &+ P_1\left(t_0\right) + P_2\left(t_0\right)\Big)^{\frac{1}{2}}\end{aligned} \tag{2.72}$$

where we have utilized the definitions of $z\left(t\right)$ and $r\left(t\right)$ from (2.32) and (2.25). From (2.41), (2.43), (2.23), and (2.29), we can obtain the following expression

$$\dot{e}_2\left(t_0\right) = \ddot{x}_d\left(t_0\right) + \dot{x}_d\left(t_0\right) - \dot{x}\left(t_0\right) + \bar{M}^{-1}\left(x\left(t_0\right)\right)\bar{B}\left(x\left(t_0\right), \dot{x}\left(t_0\right)\right)$$

After substituting the above expression into (2.72), we can finally express $\left\|\eta\left(t_0\right)\right\|$ in terms of system initial conditions as follows

$$\begin{aligned}\left\|\eta\left(t_0\right)\right\| = \Big(&e_1^T\left(t_0\right)e_1\left(t_0\right) + e_2^T\left(t_0\right)e_2\left(t_0\right) \\ &+ \left\|\ddot{x}_d\left(t_0\right) + \bar{M}^{-1}\left(x\left(t_0\right)\right)\bar{B}\left(x\left(t_0\right), \dot{x}\left(t_0\right)\right) + \dot{x}_d\left(t_0\right) - \dot{x}\left(t_0\right) + e2\left(t_0\right)\right\|^2 \\ &+ \beta_1\left\|e_2\left(t_0\right)\right\|_1 - e_2^T\left(t_0\right)N_{1d}\left(t_0\right) + \beta_2\left\|e_2\left(t_0\right)\right\|_1\Big)^{\frac{1}{2}}\end{aligned} \tag{2.73}$$

## Appendix E
## Explicit Parameter Definition

The explicit definition for $Y_1(\cdot)$, $Y_2(\cdot)$, $\phi_1(\cdot)$ and $\phi_2$ are given as follows

$$Y_1\left(\theta_1,\dot{\theta}_1,\tau_1,\ddot{\theta}_{d1},\dot{\theta}_{d1}\right)=\begin{bmatrix} Y_{N1} & -\tau_1 & \ddot{\theta}_{d1}+\mu_1\dot{e}_1 \end{bmatrix}$$

$$\phi_1=\begin{bmatrix}\phi_{N1} & \alpha_1 & I_1\end{bmatrix}^T$$

$$Y_1\left(\theta_1,\dot{\theta}_1,\theta_2,\dot{\theta}_2,\tau_1,\tau_2,T_1\right)=\begin{bmatrix}-Y_{N1} & \tau_1 & T_1 & Y_{N2} & -\tau_2 & \mu_2\dot{e}_2\end{bmatrix}$$

$$\phi_2=\begin{bmatrix}\frac{I_2}{I_1}\phi_{N1} & \frac{I_2}{I_1}\alpha_1 & \frac{I_2}{I_1} & \phi_{N2} & \alpha_2 & I_2\end{bmatrix}^T$$

where Remark 1 has been utilized.

Adaptive Estimates for Case 1: Standard slalom maneuver

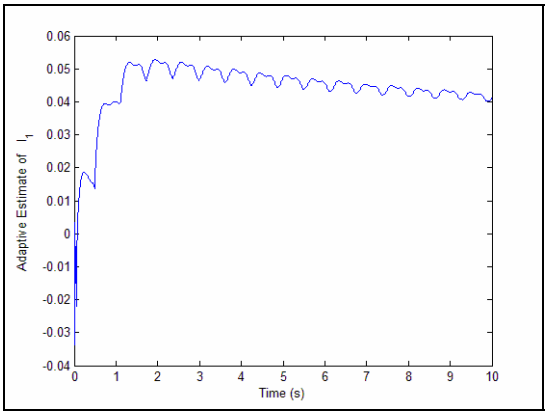


Figure E.1. Adaptive Estimate of $I_1$

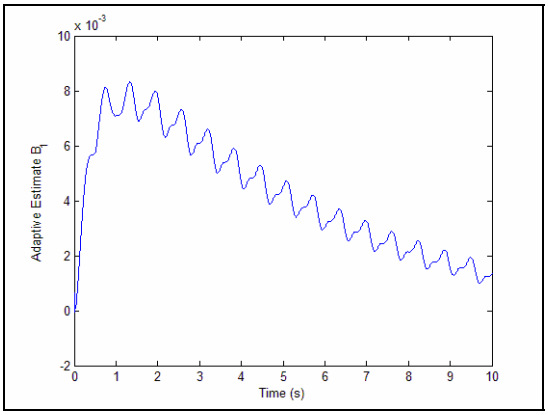


Figure E.2. Adaptive Estimate of $B_1$

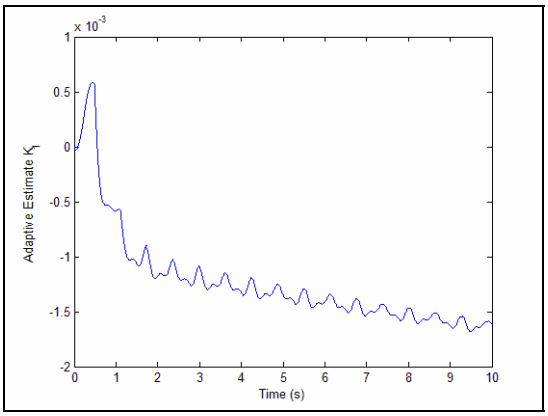


Figure E.3. Adaptive Estimate of $K_1$

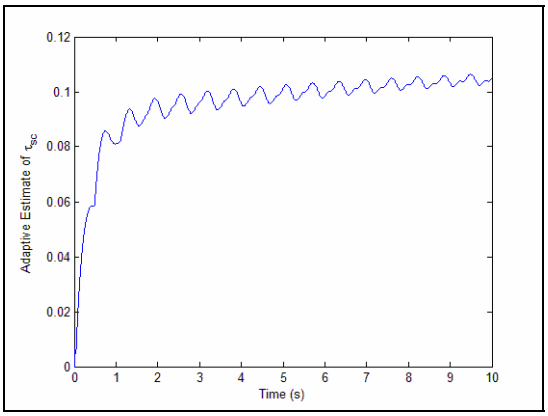


Figure E.4. Adaptive Estimate of $T_{sc}$

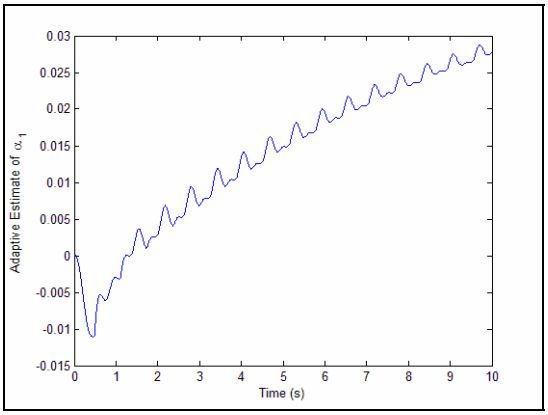


Figure E.5. Adaptive Estimate of $\alpha_1$

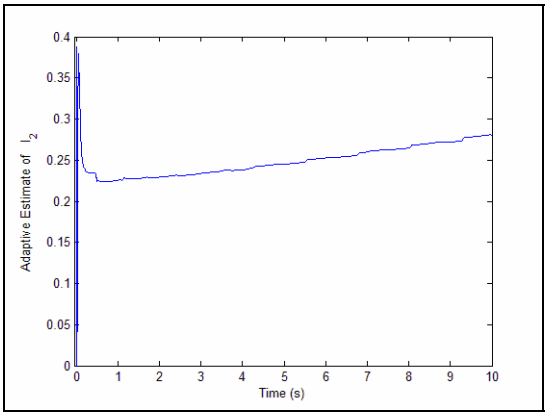


Figure E.6. Adaptive Estimate of $I_2$

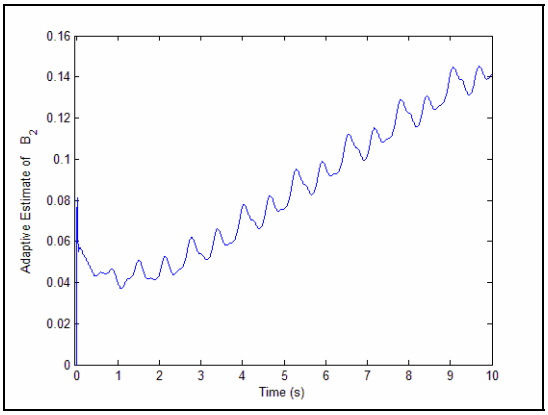


Figure E.7. Adaptive Estimate of $B_2$

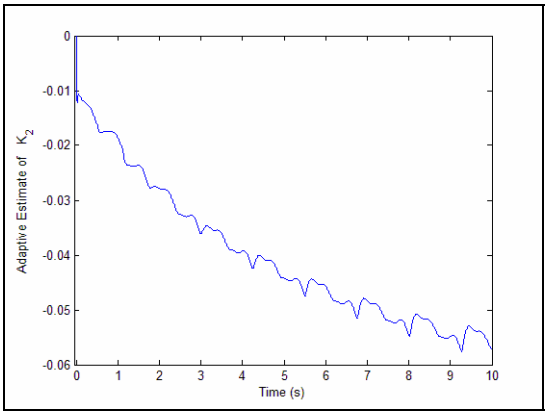


Figure E.8. Adaptive Estimate of $K_2$

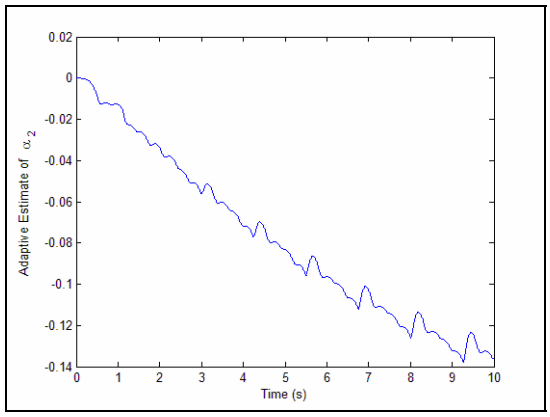


Figure E.9. Adaptive Estimate of $\alpha_2$

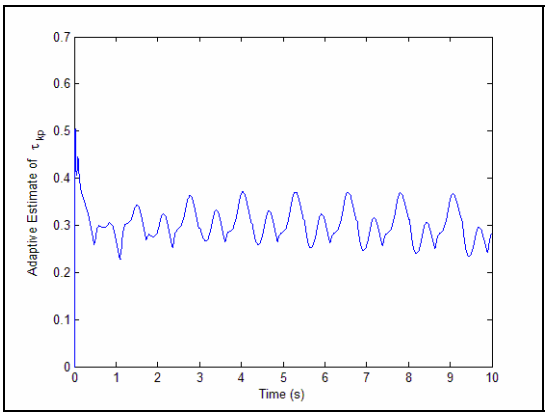


Figure E.10. Adaptive Estimate of $T_{kp}$

Adaptive Estimates for Case 2: Circle Following maneuver

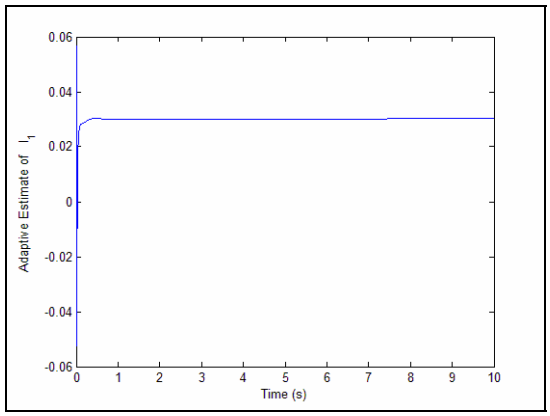


Figure E.11. Adaptive Estimate of $I_1$

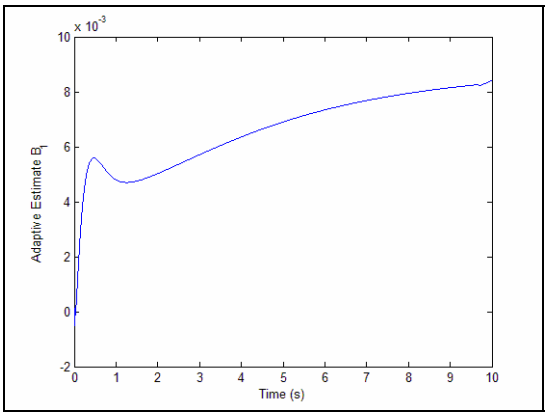


Figure E.12. Adaptive Estimate of $B_1$

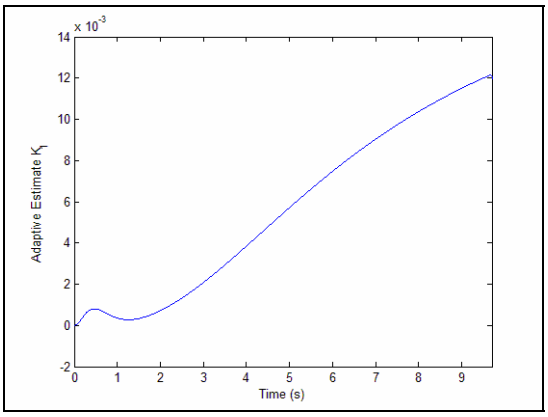


Figure E.13. Adaptive Estimate of $K_1$

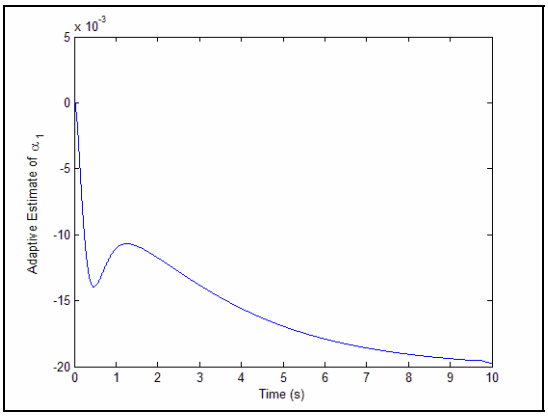


Figure E.14. Adaptive Estimate of $\alpha_1$

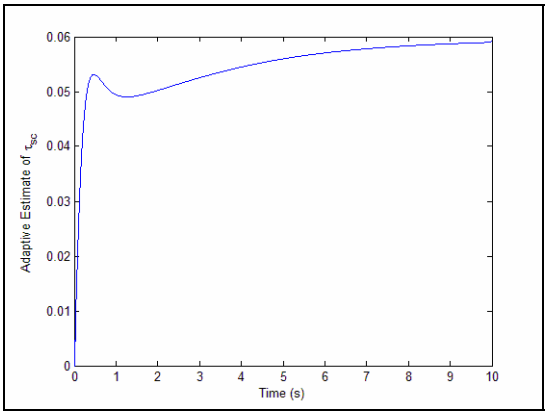


Figure E.15. Adaptive Estimate of $T_{sc}$

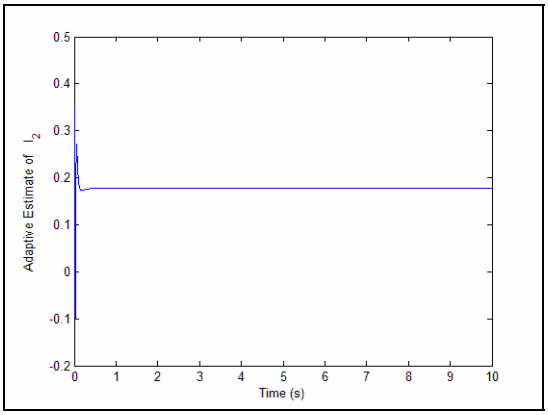


Figure E.16. Adaptive Estimate of $I_2$

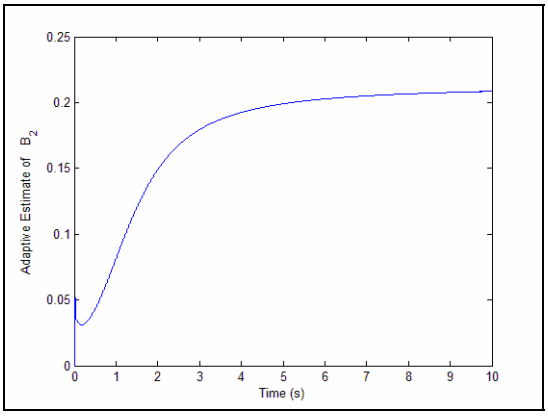


Figure E.17. Adaptive Estimate of $B_2$

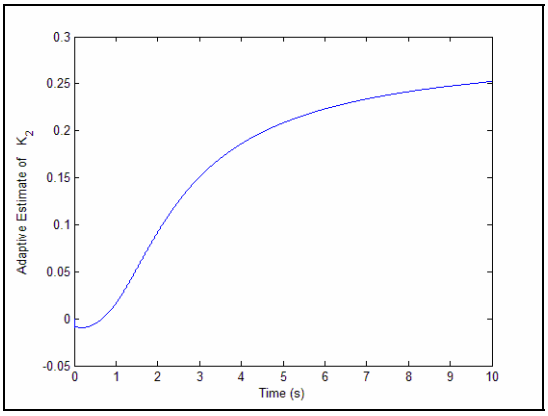


Figure E.18. Adaptive Estimate of $K_2$

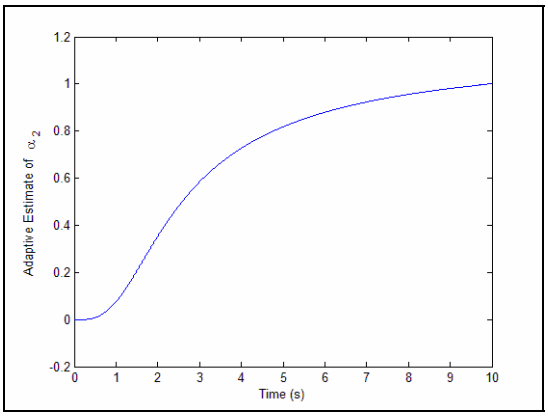


Figure E.19. Adaptive Estimate of $\alpha_2$

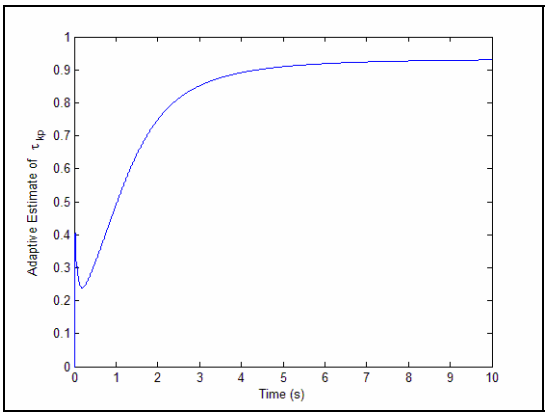


Figure E.20. Adaptive Estimate of $T_{kp}$

# Appendix F
# Rehabilitation 2-link Robot Simulation

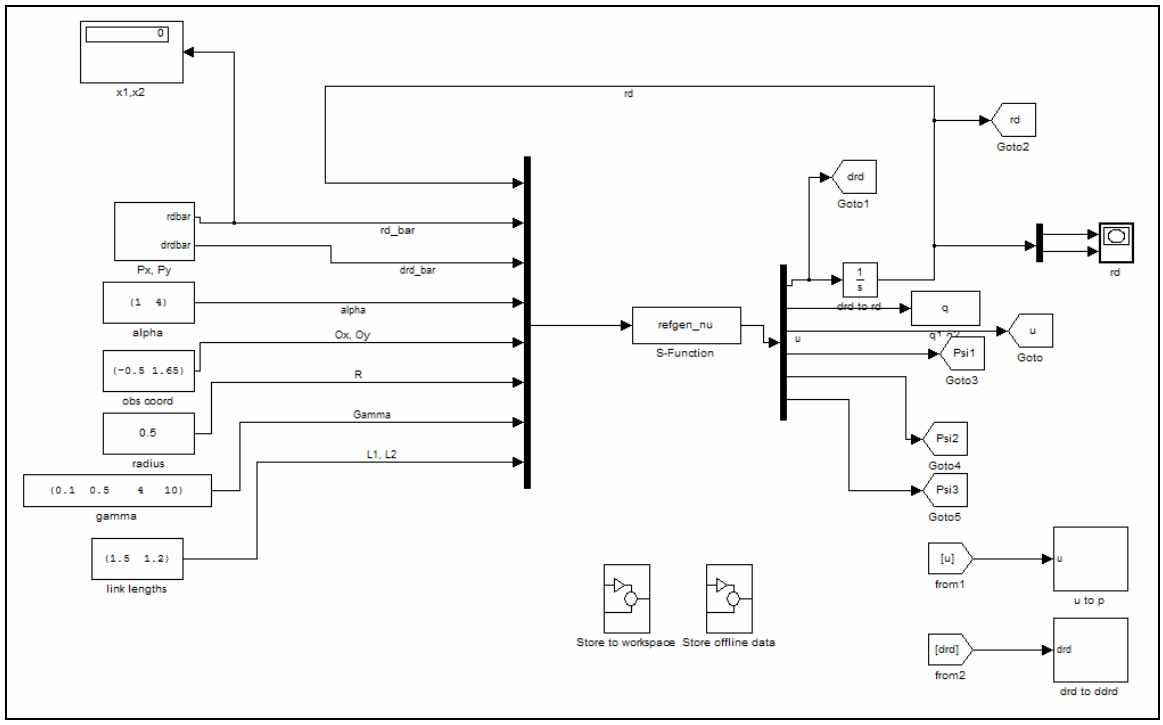


Figure F.1. Two-Link Elbow Manipulator Offline Trajectory Generator

## Offline trajectory generator program

*This program generates the desired trajectory offline*
function sys=mdlOutputs(t,x,u)

*Program Inputs*
rdx = u(1);
rdy = u(2);
px = u(3);
py = u(4);
dpx = u(5);
dpy = u(6);
alpha1= u(7);
alpha2= u(8);
Ox = u(9);

```
Oy    = u(10);
R     = u(11);
gamma1= u(12);
gamma2= u(13);
gamma3= u(14);
gamma4= u(15);
L1    = u(16);
L2    = u(17);

Calculate q1 and q2 using inverse kinematics
D=(rdx^2+rdy^2-L1^2-L2^2)/(2*L1*L2);
q2=atan2(sqrt(1-D^2),D);
q1=atan2(rdy,rdx)- atan2(L2*sqrt(1-D^2),(L1+L2*D));

Calculate Jacobian
J=[-L1*sin(q1)-L2*sin(q1+q2) -L2*sin(q1+q2);
   L1*cos(q1)+L2*cos(q1+q2)  L2*cos(q1+q2)];

ψ1 Measure
Y1=det(J*J');

Joint Limits
q1max=2*pi;
q2max=2*pi;

q1min=pi/16;
q2min=pi/16;

if q2>=0
   q2=q2;
   q2maxTerm= (1-q2/q2max);
   q2minTerm= (q2/q2min-1);
else
   q2=2*pi+q2;
   q2maxTerm= (1-q2/q2max);
   q2minTerm= (q2/q2min-1);
end

if q1>=0
   q1=q1;
   q1maxTerm= (1-q1/q1max);
   q1minTerm= (q1/q1min-1);
else
   q1=2*pi+q1;
   q1maxTerm= (1-q1/q1max);
   q1minTerm= (q1/q1min-1);
end
```

$\psi_2$ *Measure*

```
Y2= alpha1*q1maxTerm*q1minTerm * alpha2*q2maxTerm*q2minTerm;
```

*Cartesian Coordinates of end points*

```
f1=[L1*cos(q1);
   L1*sin(q1)];
f2=[rdx;
   rdy];
```

$\psi_3$ *Measure*

```
Y3_1=(f1-[Ox;Oy])'*(f1-[Ox;Oy])-R^2;
Y3_2=(f2-[Ox;Oy])'*(f2-[Ox;Oy])-R^2;
Y3=Y3_1*Y3_2;
```

*Potential Field Measure*

```
Y=gamma1*exp(-gamma2*Y1*Y2*Y3);
```

$\nabla\Psi$

```
dY1x= … (left out because of length; Calculated using Maple)
dY2x= … (left out because of length; Calculated using Maple)
dY3x=… (left out because of length; Calculated using Maple)
dY1y= … (left out because of length; Calculated using Maple)
dY2y= … (left out because of length; Calculated using Maple)
dY3y=… (left out because of length; Calculated using Maple)

dYx=-gamma1*gamma2*(dY1x*Y2*Y3+Y1*dY2x*Y3+Y1*Y2*dY3x)*exp(-
gamma2*Y1*Y2*Y3);
dYy=-gamma1*gamma2*(dY1y*Y2*Y3+Y1*dY2y*Y3+Y1*Y2*dY3y)*exp(-
gamma2*Y1*Y2*Y3);
```

*Filter*

```
drdx=-gamma3*[rdx-px]-gamma4*dYx+dpx;
drdy=-gamma3*[rdy-py]-gamma4*dYy+dpy;
```

*Calculate u vector*

```
ux=drdx/(drdx^2+drdy^2)^0.5;
uy=drdy/(drdx^2+drdy^2)^0.5;
```

*Outputs*

```
y(1)=drdx;
y(2)=drdy;
y(3)=q1;
y(4)=q2;
y(5)=ux;
y(6)=uy;
y(7)=Y1;
y(8)=Y2;
y(9)=Y3;
```

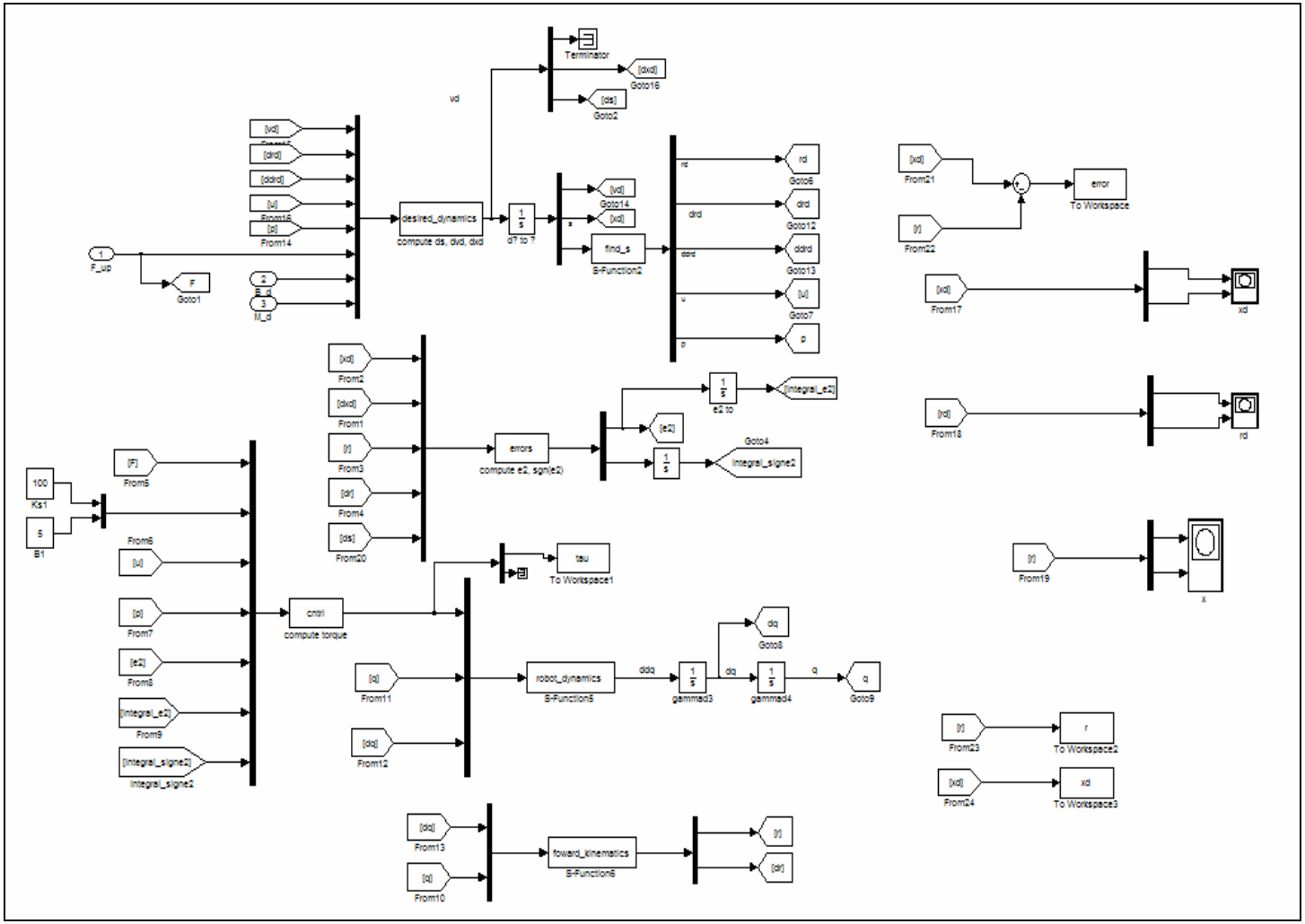


Figure F.2. Online Path Following with user applied torque

Online Path Following with user applied torque

Offline data binary search

*This program searches the offline data until s=t*

```
function sys=mdlOutputs(t,x,u)

Load vector containing offline data
load('vect.mat');

Input s
s=u(1);
s=s*1000;
s=ceil(s);
s=s/1000;
format short g;
i=1;
j=length(vec(1,:));

Conduct binary search algorithm
while((j-i)>1)
   k=round((i+j)/2);
   if s>=vec(1,k)
      i=k;
   else
      j=k;
   end
end

Output
y=vec(2:11,i);
```

Impedance Generator Program

*This program formulates the desired trajectory based on offline data*

```
function sys=mdlOutputs(t,x,u)

Program Inputs
vdx = u(1);
vdy = u(2);
drdx= u(3);
drdy= u(4);
ddrdx=u(5);
ddrdy=u(6);
uu=u(7:8);
p = u(9:10);
```

```
fu  = u(11);
fp  = u(12);
b1  = u(13);
b2  = u(14);
md1 = u(15);
md2 = u(16);


F=[fu;fp];
vd = [vdx;vdy];
Md=diag([md1,md2]);
B=diag([b1,b2]);

Calculate curvature and torsion
k=(drdx*ddrdy-drdy*ddrdx)/(drdx^2+drdy^2)^1.5;

w=[0 k;
   -k 0];

Calculate desired trajectory

ds=vdx/((drdx^2+drdy^2)^0.5);
dxd=vdx*uu+vdy*p;
dvd=inv(Md)*F-inv(Md)*B*vd;

Program Outputs
y(1:2)=dvd;
y(3:4)=dxd;
y(5)=ds;
sys = y(1:5);
```

Errors Program

*This program calculates the errors required for the generation of the control torque.*

```
function [sys,x0,str,ts] = errors(t,x,u,flag)
function sys=mdlOutputs(t,x,u)

Program Inputs
xd  = u(1:2);
dxd = u(3:4);
x = u(5:6);
dx = u(7:8);
ds = u(9);

e1 = xd - x;          error b/w desired and actual position
de1 = dxd - dx;       error b/w desired and actual velocity
e2 = e1 + de1;
```

```
sgne2 = sign(e2);

Program Outputs
y(1:2)= e2;
y(3:4) = sgne2;
sys = y(1:4);
```

Nonlinear compensator program

```
This program formulates the control torque
function sys=mdlOutputs(t,x,u)

Program inputs
fu = u(1);
fp = u(2);
Ks=u(3);
b=u(4);
uu = u(5:6);
p = u(7:8);
e2 = u(9:10);
inte2 = u(11:12);
isgne2 = u(13:14);

Force in Frenet F
F=[fu;fp];

Force in inertial I
fxy = [uu, p]*F;

Tau in inertial frame I
taux = Ks*e2 + Ks*inte2 + b*isgne2 - fxy;
y(1:2) = taux;
y(3:4) = fxy;
```

Forward kinematics program

```
This program calculates the forward kinematics of the robotic manipulator
function sys=mdlOutputs(t,x,u)

Program inputs
dq1=u(1);
dq2=u(2);
q1 =u(3);
q2 =u(4);

Link Parameters
l1=1.5; l2=1.2;
```

```
dq=[dq1;dq2];

End-effector position
x1 = l1*cos(q1) + l2*cos(q1+q2);
x2 = l1*sin(q1) + l2*sin(q1+q2);

End-effector velocity
dx1 = -l1*sin(q1)*dq1 - l2*(dq1+dq2)*sin(q1+q2);
dx2 =  l1*cos(q1)*dq1 + l2*(dq1+dq2)*cos(q1+q2);

Program outputs
y(1) = x1;
y(2) = x2;
y(3) = dx1;
y(4)=  dx2;
```

Robot dynamics program

```
This program calculates the dynamics of the robotic manipulator
function sys=mdlOutputs(t,x,u)

Program inputs
taux = u(1:2);
fxy = u(3:4);
q1 = u(5);
q2 = u(6);
dq1 = u(7);
dq2 = u(8);

Parameters
l1=1.5; l2=1.2;
m1=2.08; m2=0.168;

Inertia Matrix
c2=cos(q2);
s2=sin(q2);

m11 = (m1+m2)*l1^2 +m2*l2^2+ 2*m2*l1*l2*c2;
m12 = m2*l2^2 + m2*l1*l2*c2;
m21 = m12;
m22 = m2*l2^2;
M=[m11 m12;
   m21 m22];


Centripetal-Coriolis matrix
vmq = [-m2*l1*l2*(2*dq1*dq2+dq2^2)*s2;
```

```
      m2*l1*l2*dq1^2*s2            ];

j11 = -l1*sin(q1) - l2*sin(q1+q2);
j12 = -l2*sin(q1+q2);
j21 = l1*cos(q1) + l2*cos(q1+q2);
j22 = l2*cos(q1+q2);

J = [ j11 j12;
     j21 j22];

tau_q = J'*taux;
fu_q = J'*fxy;

Robot dynamics
ddq= inv(M)*(tau_q + fu_q-vmq);

Program outputs
y = ddq;
```

# Appendix G
## Rehabilitation 3-link Robot Simulation

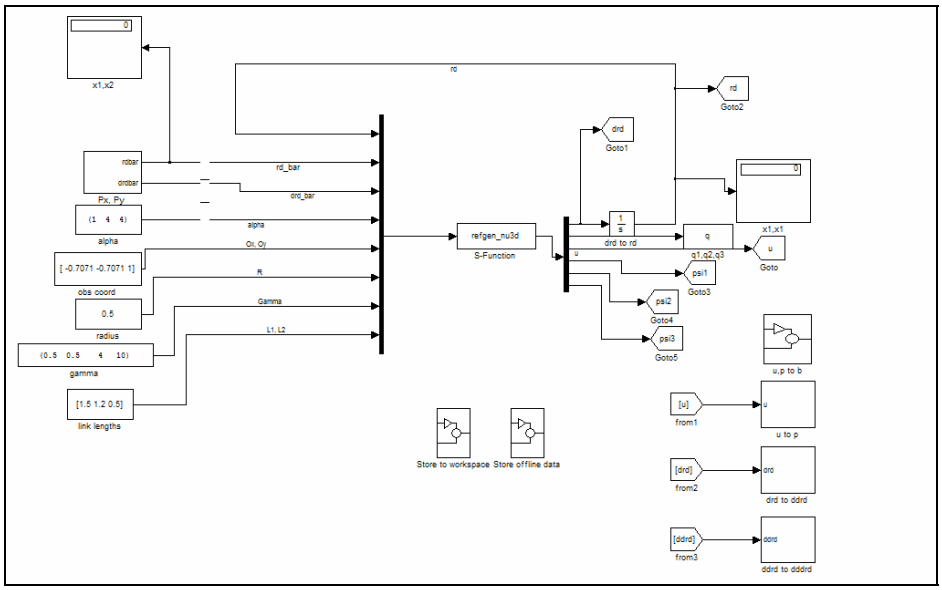


Figure G.1. Three-Link Elbow Manipulator Offline Trajectory Generator

Offline trajectory generator program

```
This program calculates the offline trajectory
function sys=mdlOutputs(t,x,u)
Program Inputs
rdx   = u(1);
rdy   = u(2);
rdz   = u(3);
px    = u(4);
py    = u(5);
pz    = u(6);
dpx   = u(7);
dpy   = u(8);
dpz   = u(9);
alpha1= u(10);
alpha2= u(11);
alpha3= u(12);
Ox    = u(13);
Oy    = u(14);
Oz    = u(15);
R     = u(16);
```

```
gamma1= u(17);
gamma2= u(18);
gamma3= u(19);
gamma4= u(20);
a1    = u(21);
a2    = u(22);
a3    = u(23);

Calculate q1, q2, q3 using inverse kinematics
q1=atan2(rdy,rdx);
D=(rdx^2+rdy^2+(rdz-a1)^2-a2^2-a3^2)/(2*a2*a3);
q3=atan2(sqrt(1-D^2),D);
q2=atan2((rdz-a1),sqrt(rdx^2+rdy^2))-atan2(a3*sin(q3),a2+a3*cos(q3));

Jacobian matrix
J = [-sin(q1) * (a3 * cos(q2 + q3) + a2 * cos(q2)), cos(q1) * (-a3 * sin(q2 + q3) - a2 * sin(q2)),
-cos(q1) * a3 * sin(q2 + q3);  cos(q1) * (a3 * cos(q2 + q3) + a2 * cos(q2)), sin(q1) * (-a3 *
sin(q2 + q3) - a2 * sin(q2)), -sin(q1) * a3 * sin(q2 + q3); 0,  a3 * cos(q2 + q3) + a2 * cos(q2),
a3 * cos(q2 + q3)];

PSI 1
Y1=det(J*J');

PSI 2
q1max=2*pi;
q2max=2*pi;
q3max=2*pi;


q1min=pi/16;
q2min=pi/16;
q3min=pi/16;


if q2>=0
   q2=q2;
   q2maxTerm= (1-q2/q2max);
   q2minTerm= (q2/q2min-1);
else
   q2=2*pi+q2;
   q2maxTerm= (1-q2/q2max);
   q2minTerm= (q2/q2min-1);
end

if q3>=0
   q3=q3;
   q3maxTerm= (1-q3/q3max);
   q3minTerm= (q3/q3min-1);
```

```
else
   q3=2*pi+q3;
   q3maxTerm= (1-q3/q3max);
   q3minTerm= (q3/q3min-1);
end

Y2= alpha2*q2maxTerm*q2minTerm *alpha3*q3maxTerm*q3minTerm;

PS1 3
f1=[0;
   0;
   a1];

f2=[a2*cos(q2)*cos(q1);
   a2*sin(q1)*cos(q2);
   a2*sin(q2)+a1];
f3=[rdx;
   rdy;
   rdz;];
Y3_1=(f1-[Ox;Oy;Oz])'*(f1-[Ox;Oy;Oz])-R^2;
Y3_2=(f2-[Ox;Oy;Oz])'*(f2-[Ox;Oy;Oz])-R^2;
Y3_3=(f3-[Ox;Oy;Oz])'*(f3-[Ox;Oy;Oz])-R^2;

Y3=Y3_1*Y3_2*Y3_3;

Potential Field Measure
Y=gamma1*exp(-gamma2*Y1*Y2*Y3);

∇Ψ
dY1x= … (left out because of length; Calculated using Maple)
dY2x= … (left out because of length; Calculated using Maple)
dY3x=… (left out because of length; Calculated using Maple)

dY1y= … (left out because of length; Calculated using Maple)
dY2y= … (left out because of length; Calculated using Maple)
dY3y=… (left out because of length; Calculated using Maple)

dY1z= … (left out because of length; Calculated using Maple)
dY2z= … (left out because of length; Calculated using Maple)
dY3z=… (left out because of length; Calculated using Maple)

dYx=-gamma1*gamma2*(dY1x*Y2*Y3 + Y1*dY2x*Y3 + Y1*Y2*dY3x)*exp(-
gamma2*Y1*Y2*Y3);
dYy=-gamma1*gamma2*(dY1y*Y2*Y3 + Y1*dY2y*Y3 + Y1*Y2*dY3y)*exp(-
gamma2*Y1*Y2*Y3);
dYz=-gamma1*gamma2*(dY1z*Y2*Y3 + Y1*dY2z*Y3 + Y1*Y2*dY3z)*exp(-
gamma2*Y1*Y2*Y3);
```

*Filter*
drdx=-gamma3*[rdx-px]-gamma4*dYx+dpx;
drdy=-gamma3*[rdy-py]-gamma4*dYy+dpy;
drdz=-gamma3*[rdz-pz]-gamma4*dYz+dpz;

*Calculate u vector*
ux=drdx/(drdx^2+drdy^2+drdz^2)^0.5;
uy=drdy/(drdx^2+drdy^2+drdz^2)^0.5;
uz=drdz/(drdx^2+drdy^2+drdz^2)^0.5;

*Outputs*
y(1)=drdx;
y(2)=drdy;
y(3)=drdz;
y(4)=q1;
y(5)=q2;
y(6)=q3;
y(7)=ux;
y(8)=uy;
y(9)=uz;
y(10)=Y1;
y(11)=Y2;
y(12)=Y3;

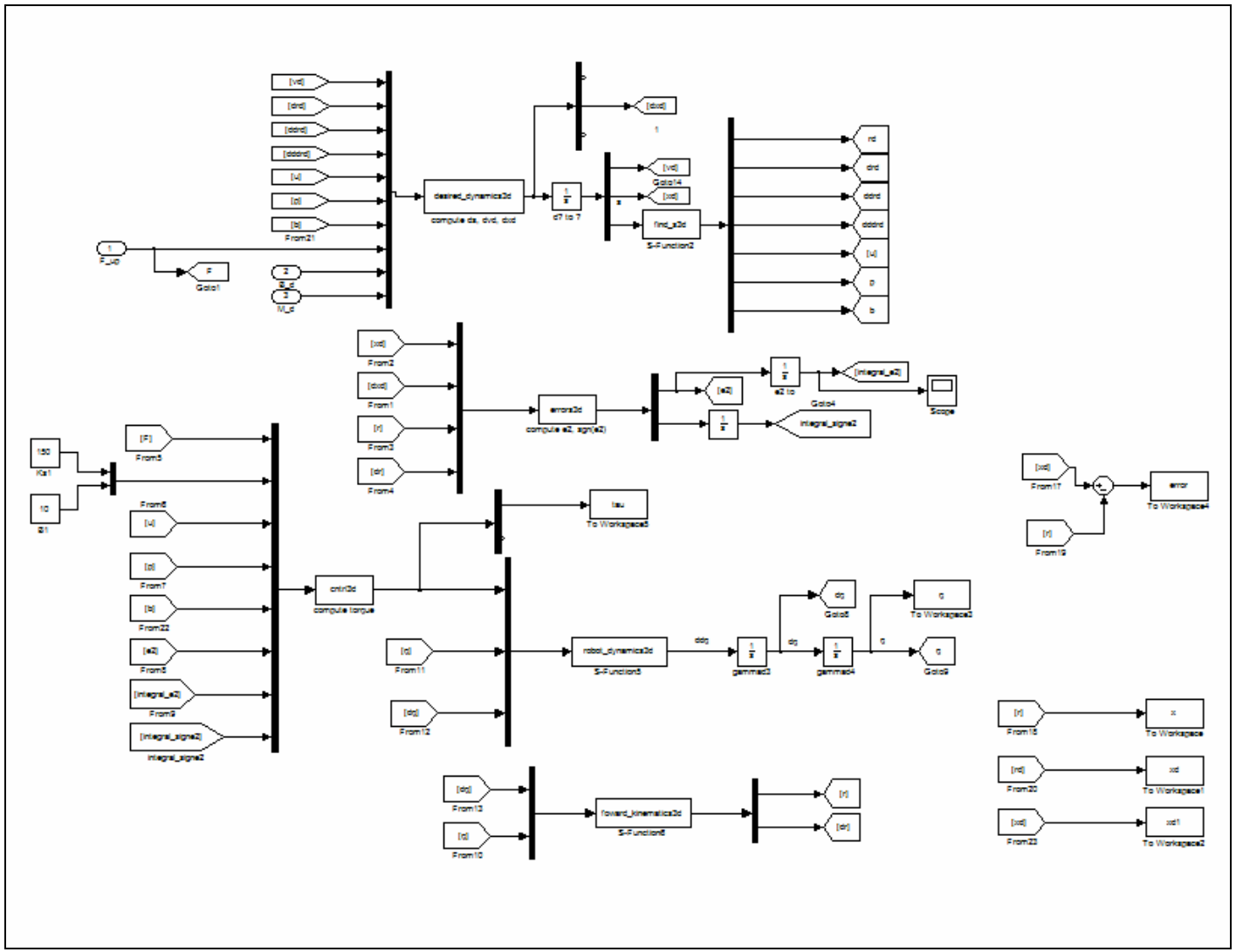


Figure G.2 Online Path Following with user applied torque

Online Path Following with user applied torque

Offline data binary search

*This program searches the offline data until s=t*

```
function sys=mdlOutputs(t,x,u)

Load offline data
load('vect.mat');

s=mod(u(1),2*pi);
s=s*1000;
s=ceil(s);
s=s/1000;
format short g;
i=1;
j=length(vec(1,:));

Conduct binary search algorithm
while((j-i)>1)
   k=round((i+j)/2);
   if s>=vec(1,k)
      i=k;
   else
      j=k;
   end
end

Outputs
y=vec(2:22,i);
```

Impedance Generator Program

*This program formulates the desired trajectory based on offline data*

```
function sys=mdlOutputs(t,x,u)

Program Inputs
vd1 = u(1);
vd2 = u(2);
vd3 = u(3);
drdx= u(4);
drdy= u(5);
drdz= u(6);
ddrdx=u(7);
ddrdy=u(8);
ddrdz=u(9);
```

```
dddrdx=u(10);
dddrdy=u(11);
dddrdz=u(12);
uu= u(13:15);
p = u(16:18);
b = u(19:21);
fu  = u(22);
fp  = u(23);
fb  = u(24);
b1  = u(25);
b2  = u(26);
b3  = u(27);
md1 = u(28);

F=[fu;fp;fb];
vd = [vd1;vd2;vd3];
B=diag([b1,b2,b3]);

dr=[drdx;drdy;drdz];
ddr=[ddrdx;ddrdy;ddrdz];
dddr=[dddrdx;dddrdy;dddrdz];

Calculate torsion and curvature
k=norm(cross(dr,ddr))/(norm(dr))^3;
T=(dot(dr,cross(ddr,dddr)))/(norm(k))^2;

w=[0 k 0;
   -k 0 T;
   0 -T 0];

Calculate desired trajectory
ds=(vd1)/norm(dr);
dxd=vd1*uu+vd2*p+vd3*b;
dvd=(F-B*vd)/md1-ds*w*vd;

Outputs
y(1:3)=dvd;
y(4:6)=dxd;
y(7)=ds;
sys = y(1:7);
```

Errors Program

*This program calculates the errors required for the generation of the control torque.*

```
function [sys,x0,str,ts] = errors(t,x,u,flag)
function sys=mdlOutputs(t,x,u)
```

```
Program Inputs
xd  = u(1:3);
dxd = u(4:6);
 x = u(7:9);
dx = u(10:12);

e1 = xd - x;          error b/w desired and actual position
de1 = dxd - dx;       error b/w desired and actual velocity
e2 = e1 + de1;
sgne2 = sign(e2);

Program outputs
y(1:3)= e2;
y(4:6) = sgne2;
```

Nonlinear compensator program

```
This program formulates the control torque
function sys=mdlOutputs(t,x,u)

Program inputs
fu = u(1);
fp = u(2);
fb = u(3);
Ks=u(4);
bb=u(5);
uu = u(6:8);
p = u(9:11);
b = u(12:14);
e2 = u(15:17);
inte2 = u(18:20);
isgne2 = u(21:23);

Force in Frenet F
F=[fu;fp;fb];
Force in inertial I
fxy = [uu, p, b]*F;

Tau in inertial I
taux = Ks*e2 + Ks*inte2 + bb*isgne2 - fxy;
y(1:3) = taux;
y(4:6) = fxy;
```

Forward kinematics program

```
This program calculates the forward kinematics
function sys=mdlOutputs(t,x,u)

Program inputs
dq1=u(1);
dq2=u(2);
dq3=u(3);
q1 =u(4);
q2 =u(5);
q3 =u(6);

Parameters
a1=1.5;
a2=1.2;
a3=0.5;

dq=[dq1;dq2;dq3];

End-effector position
x1=cos(q1)*(a3*cos(q2+q3)+a2*cos(q2));
x2=sin(q1)*(a3*cos(q2+q3)+a2*cos(q2));
x3=a3*sin(q2+q3)+a2*sin(q2)+a1;

J = [-sin(q1) * (a3 * cos(q2 + q3) + a2 * cos(q2)), cos(q1) * (-a3 * sin(q2 + q3) - a2 * sin(q2)),
-cos(q1) * a3 * sin(q2 + q3);  cos(q1) * (a3 * cos(q2 + q3) + a2 * cos(q2)), sin(q1) * (-a3 *
sin(q2 + q3) - a2 * sin(q2)), -sin(q1) * a3 * sin(q2 + q3);
     0, a3 * cos(q2 + q3) + a2 * cos(q2), a3 * cos(q2 + q3)];

End-effector velocity
dx=J*dq;

y(1) = x1;
y(2) = x2;
y(3) = x3;
y(4:6)=dx;
```

Robot dynamics program

```
This program calculates the dynamics of the robotic manipulator
function sys=mdlOutputs(t,x,u)

Program inputs
taux = u(1:3);
fxy = u(4:6);
q1 = u(7);
```

```
q2 = u(8);
q3 = u(9);
dq1 = u(10);
dq2 = u(11);
dq3 = u(12);

Parameters
a1=1.5;
a2=1.2;
a3=0.5;

I1=2.08;
m2=2.08;
m3=0.168;
g=9.8;

dq=[dq1;dq2;dq3];

Inertia Matrix
M = [m2 * a2 ^ 2 * cos((2 * q2)) / 0.2e1 + m2 * a2 ^ 2 / 0.2e1 + m3 * a3 ^ 2 * cos((2 * q2
+ 2 * q3)) / 0.2e1 + m3 * a3 ^ 2 / 0.2e1 + m3 * a3 * a2 * cos(q3) + m3 * a3 * a2 * cos((2 *
q2 + q3)) + m3 * a2 ^ 2 * cos((2 * q2)) / 0.2e1 + m3 * a2 ^ 2 / 0.2e1 + I1, 0, 0;
   0, m2 * a2 ^ 2 + 0.2e1 * m3 * a3 * a2 * cos(q3) + m3 * a2 ^ 2 + m3 * a3 ^ 2, m3 * a3 * a2
* cos(q3) + m3 * a3 ^ 2;
   0, m3 * a3 * a2 * cos(q3) + m3 * a3 ^ 2, m3 * a3 ^ 2;];

Centripetal-Coriolis matrix
C = [(-m2 * a2 ^ 2 * sin((2 * q2)) / 0.2e1 - m3 * a3 ^ 2 * sin((2 * q2 + 2 * q3)) / 0.2e1 - m3 *
a3 * a2 * sin((2 * q2 + q3)) - m3 * a2 ^ 2 * sin((2 * q2)) / 0.2e1) * dq2 + (-m3 * a3 ^ 2 *
sin((2 * q2 + 2 * q3)) / 0.2e1 - m3 * a3 * a2 * sin(q3) / 0.2e1 - m3 * a3 * a2 * sin((2 * q2 +
q3)) / 0.2e1) * dq3 (-m2 * a2 ^ 2 * sin((2 * q2)) / 0.2e1 - m3 * a3 ^ 2 * sin((2 * q2 + 2 * q3))
/ 0.2e1 - m3 * a3 * a2 * sin((2 * q2 + q3)) - m3 * a2 ^ 2 * sin((2 * q2)) / 0.2e1) * dq1 (-m3 *
a3 ^ 2 * sin((2 * q2 + 2 * q3)) / 0.2e1 - m3 * a3 * a2 * sin(q3) / 0.2e1 - m3 * a3 * a2 * sin((2
* q2 + q3)) / 0.2e1) * dq1;
   (m2 * a2 ^ 2 * sin((2 * q2)) / 0.2e1 + m3 * a3 ^ 2 * sin((2 * q2 + 2 * q3)) / 0.2e1 + m3 *
a3 * a2 * sin((2 * q2 + q3)) + m3 * a2 ^ 2 * sin((2 * q2)) / 0.2e1) * dq1 -m3 * a3 * a2 *
sin(q3) * dq3 -m3 * a3 * a2 * sin(q3) * dq2 - m3 * a3 * a2 * sin(q3) * dq3;
   (m3 * a3 ^ 2 * sin((2 * q2 + 2 * q3)) / 0.2e1 + m3 * a3 * a2 * sin(q3) / 0.2e1 + m3 * a3 *
a2 * sin((2 * q2 + q3)) / 0.2e1) * dq1 m3 * a3 * a2 * sin(q3) * dq2 0;];

Gravity effects
G= [0;
   a2 * cos(q2) * g * m2;
   m3 * g * a3 * cos(q2 + q3);];

J = [-sin(q1) * (a3 * cos(q2 + q3) + a2 * cos(q2)), cos(q1) * (-a3 * sin(q2 + q3) - a2 * sin(q2)),
-cos(q1) * a3 * sin(q2 + q3); cos(q1) * (a3 * cos(q2 + q3) + a2 * cos(q2)), sin(q1) * (-a3 *
```

```
sin(q2 + q3) - a2 * sin(q2)), -sin(q1) * a3 * sin(q2 + q3);  0, a3 * cos(q2 + q3) + a2 * cos(q2),
a3 * cos(q2 + q3)];

tau_q = J'*taux;
fu_q = J'*fxy;
```

*Robot Dynamics*

```
ddq= inv(M)*(tau_q + fu_q-C*dq-G);
```

*Outputs*

```
y = ddq;
```

# Appendix H
## Haptic interface adaptive controller simulation

### Reference Model Generator

Reference Model Generator program

```
function sys=mdlOutputs(t,x,u)
Program Inputs
Rack
yr=u(1);
dyr=u(2);
Road Wheel
Or=u(3);
dOr=u(4);
Spool Valve
Osp=u(5);
Steering wheel
Osw=u(6);
dOsw=u(7);
Measurements
Tsw=u(8);
Mz=u(9);
Fboost=u(10);
Tsc=u(11);
Ffr=u(12);
Tkp=u(13);
tire_spin=u(14);

Variables
Brack=10000;
Bsc=0.356;
Bw=200;
Isw=6.78E-5;
Iw=.356;
Iwy=.005;
Kl=48.816E3;
Ksc=33.898;
Kt=75.797;
Mrack=29.412;
Nl=0.11816;
Rp=8.00E-3;
Tv=0.0;

Reference Model
ddOr=1/Iw*(Kl*(yr/Nl-Or)-Bw*dOr-Mz-Tkp-Iwy*tire_spin^2*Or);
Ot=Osp-yr/Rp;
```

```
ddyr=1/Mrack*(Kt/Rp*(Ot)-Brack*dyr-2*Kl/Nl*(yr/Nl-Or)-Ffr-Fboost);
dOsp=dOsw+1/Bsc*(Ksc*(Osw-Osp)-Kt*(Ot));
ddOsw=1/Isw*(Tsw-Ksc*(Osw-Osp)-Bsc*(dOsw-dOsp)+Tsc+Tv*sign(Osp)*Osp^2);
```

*Outputs*

```
y(1)=ddyr;
y(2)=ddOr;
y(3)=dOsp;
y(4)=ddOsw;
y(5)=Ot;
```

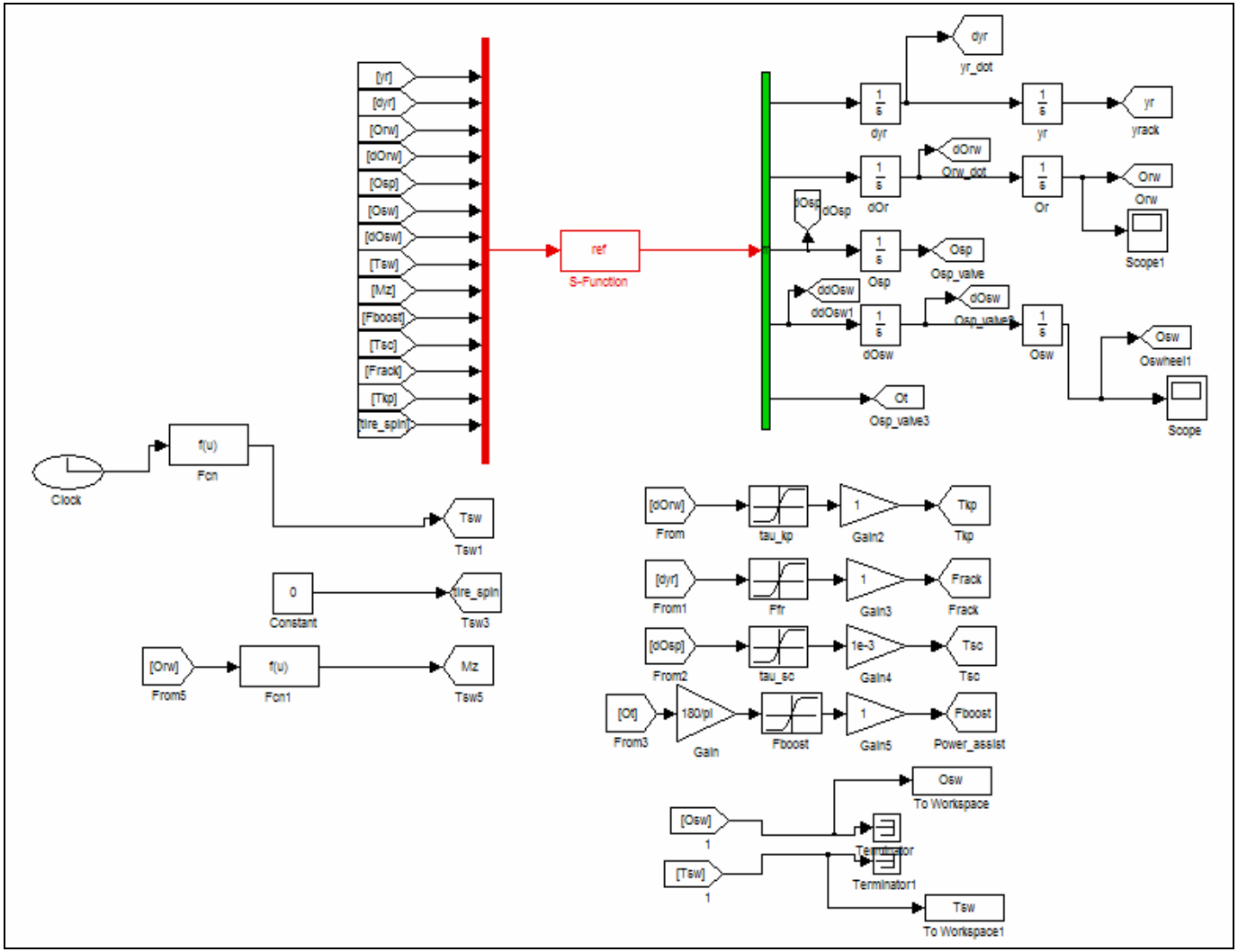


Figure H.1 Reference Model

Feedback subsystem

Feedback dynamics

*This program models the primary subsystem dynamics*
function sys=mdlOutputs(t,x,u)

*Inputs*
dO1=u(1);
O1 =u(2);
tau1=u(3);
T1=u(4);
*Parameters*
I1=1.16e-0;
B1=1.9e-0;
K1=0;
a1=1e-1;
Tsc=0;

*Feedback plant dynamics*
ddO1=1/I1*(a1*tau1+T1-B1*dO1-K1*O1-Tsc*sign(dO1));

*Plant output*
y(1)=ddO1;

Adaptive Estimator

*This program estimates the unknown feedback plant parameters*
function sys=mdlOutputs(t,x,u)

*Inputs*
dO1 = u(1);
O1 = u(2);
tau1= u(3);
ddOsw=u(4);
de1 = u(5);
e1 = u(6);
l1 = u(7);
l2 = u(8);
l3 = u(9);
l4 = u(10);
l5 = u(11);
miu = u(12);

*Adaptive estimator*
r1=de1+miu*e1;

```
lambda=diag([l1,l2,l3,l4,l5]);
dphi1=lambda*[dO1;O1;sign(dO1);-tau1;ddOsw+miu*de1]*r1;
Program outputs
y=dphi1;
```

Control torque formulator

```
This program calculates the feedback control torque
function sys=mdlOutputs(t,x,u)

Inputs
dO1 = u(1);
O1  = u(2);
tau1= u(3);
ddOsw=u(4);
de1 = u(5);
e1  = u(6);
phi1_bar=u(7:11);
miu=u(12);
k1=u(13);

Calculate control torque
r1=de1+miu*e1;
Y1=[dO1,O1,sign(dO1),-tau1,ddOsw+miu*de1];
T1=k1*r1+Y1*phi1_bar;

Program output
y=T1;
```

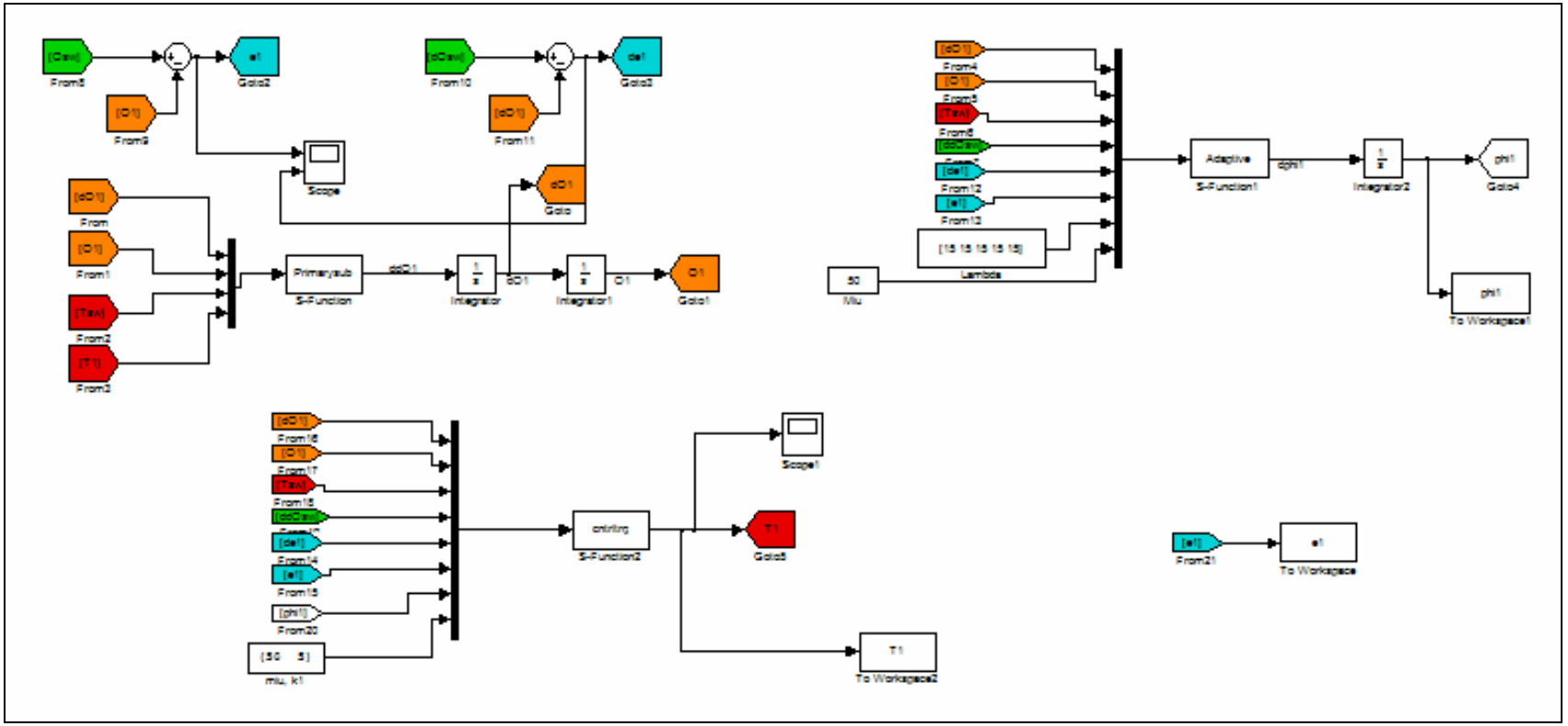


Figure H.2 Feedback subsystem

Drive dynamics

*This program models the secondary subsystem dynamics*
function sys=mdlOutputs(t,x,u)

*Inputs*
dO2=u(1);
O2 =u(2);
tau2=u(3);
T2=u(4);

*Parameters*
I2=2.35e-1;
B2=0.6;
K2=1;
a2=1;
Tkp=0.5;

*Drive subsystem plant dynamics*
ddO2=1/I2*(a2*tau2+T2-B2*dO2-K2*O2-Tkp*sign(dO2));

*Program output*
y(1)=ddO2;

Adaptive Estimator

*This program estimates the unknown drive plant parameters*
function sys=mdlOutputs(t,x,u)

*Inputs*
dO1 = u(1);
O1 = u(2);
dO2 = u(3);
O2 = u(4);
tau1= u(5);
tau2= u(6);
T1 = u(7);
de2 = u(8);
e2 = u(9);
l1 = u(10);
miu2 = u(11);

*Adaptive estimator*
r2=de2+miu2*e2;
lambda=diag([l1,l1,l1,l1,l1,l1,l1,l1,l1,l1,l1]);
dphi2=lambda*[-dO1;-O1;-sign(dO1);tau1;T1;dO2;O2;sign(dO2);-tau2;miu2*de2]*r2;
*Program outputs*

```
y=dphi2;
```

Control torque formulator

*This program calculates the drive control torque*

```
function sys=mdlOutputs(t,x,u)

Inputs
dO1 = u(1);
O1  = u(2);
dO2 = u(3);
O2  = u(4);
tau1= u(5);
tau2= u(6);
T1  = u(7);
de2 = u(8);
e2  = u(9);
phi2_bar=u(10:19);
miu2 = u(20);
k2=u(21);

Calculate control torque
r2=de2+miu2*e2;
Y2=[-dO1;-O1;-sign(dO1);tau1;T1;dO2;O2;sign(dO2);-tau2;miu2*de2]';
T2=k2*r2+Y2*phi2_bar;

Program outputs
y=T2;
```

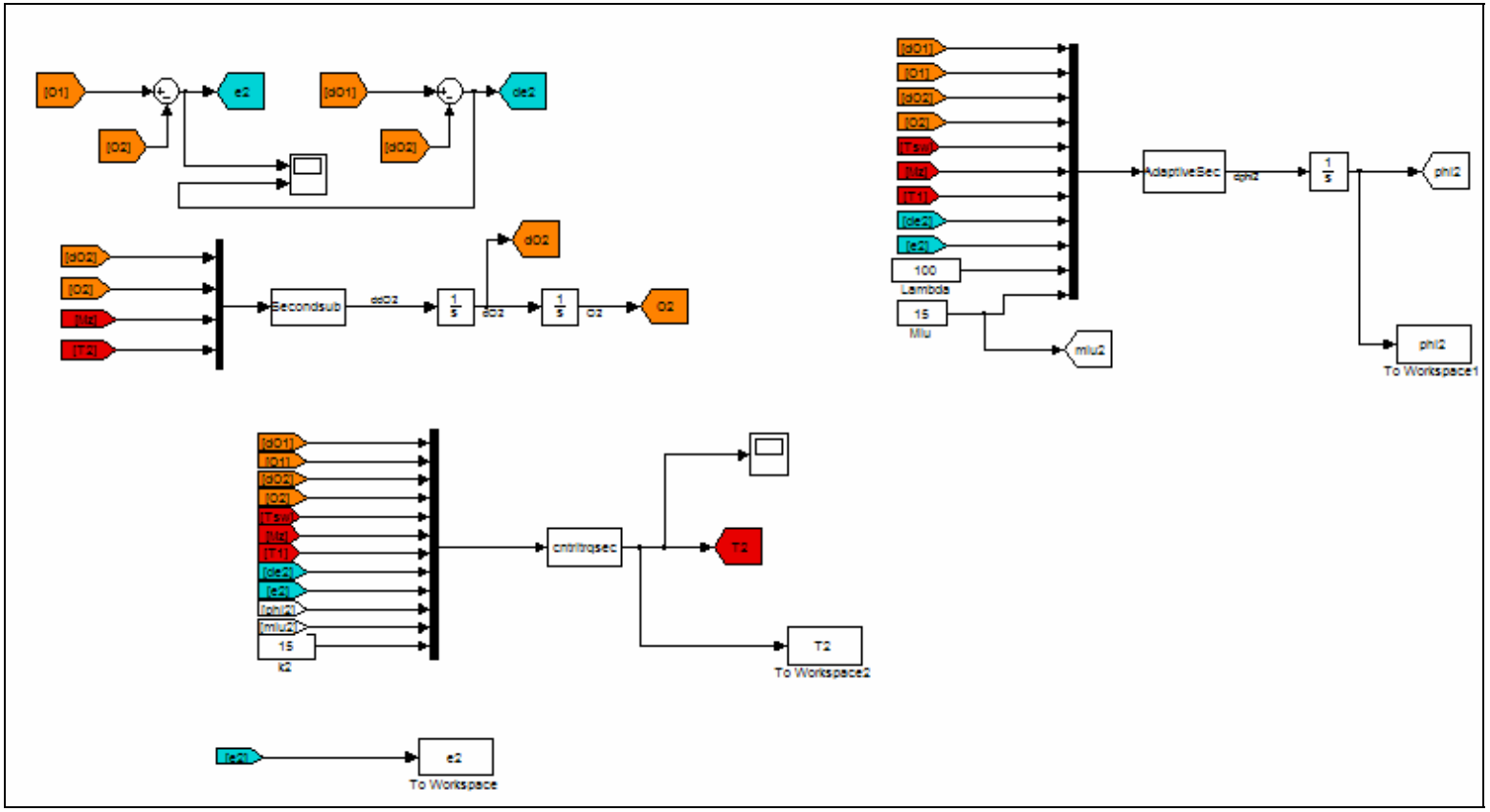


Figure H.3 Drive subsystem

REFERENCES


[1] R. J. Anderson and M.W. Spong, "Bilateral control of teleoperators with time delay", IEEE Trans. Automat. Contr., vol. 34, pp. 494-501, May 1989.

[2] G. T. Carter, "Rehabilitation Management in Neuromuscular Disease", J. Neuro. Rehab., 11:69-80, 1997.

[3] D. Dawson, J. Hu, and T. Burg, Nonlinear Control of Electric Machinery, Marcel Dekker, 1998.

[4] C. A. Desoer, and M. Vidyasagar, "Feedback Systems: Input-output Properties", Academic Press, 1975.

[5] J. Engelberger, Robotics in Service, Cambridge, Massachussets, The MIT Press, 1989.

[6] B. R. Gillespie, C. Hasser, and P. Tang, "Cancellation of Feed-through Dynamics Using a Force-Reflecting Joystick" proceedings of the ASME International Mechanical Engineering Conference and Exposition, Nashville, TN, Vol. 67, pp.319-329, November 1999.

[7] S. T. Godley, T. J. Triggs, and B. N. Fildes, "Driving Simulator Validation for Speed Research", Accident Analysis and Prevention, Vol. 34, Issue 5, pp.589-600, September 2002.

[8] A. Gray, "Drawing Space Curves with Assigned Curvature", in Modern Differential Geometry of Curves and Surfaces with Mathematica, 2nd ed. Boca Raton, FL: CRC Press, pp. 222-224, 1997.

[9] B. Hannaford, "A Design Framework for Teleoperators with Kinesthetic Feedback", IEEE Trans. Robot. Automat., vol. 5, pp. 426434, August 1989.18

[10] N. Hogan, "Impedance control: An approach to manipulation, part I - theory, part II - implementation, part III - applications", Journal of Dynamics, Systems, Measurement, and Control, pp. 107:124, 1985.

[11] K. Huh, C. Seo, J. Kim, and D. Hong, D "Active Steering Control Based on the Estimated Tire Forces", Proceedings of the American Control Conference, San Diego, CA, pp. 729-733, June 1999.

[12] H. Jeffreys and B. Jeffreys, Methods of Mathematical Physics, Cambridge University Press, 3rd edition, Feb. 2000.

[13] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", The International Journal of Robotics Research, vol. 5, no. 1, 1986.

[14] P. Kokotovic, "The Joy of Feedback: Nonlinear and Adaptive", IEEE Control Syst. Mag., Vol.12, pp. 177-185, June 1992.

[15] E. Kreyszig, Advanced Engineering Mathematics, Wiley Eastern Limited, Sept. 1990.

[16] H.H. Kwee, et al., "The MANUS Wheelchair-Borne Manipulator: System Review and First Results", Proc. IARP Workshop on Domestic and Medical & Healthcare Robotics, Newcastle, 1989.

[17] K.F. Laurin-Kovitz, J.E. Colgate, S.D.R. Carnes, "Design of Components for Programmable Passive Impedance", Proc. IEEE International Conference on Robotics and Automation, Sacramento, CA, pp. 1476-1481, April 1991.

[18] D. A. Lawrence, "Stability and transparency in bilateral teleoperation", IEEE Trans. Robot. Automat., vol. 9, pp. 624-637, Oct. 1993.

[19] D. Lee and P.Y. Li, "Passive Bilateral Feedforward Control of Linear Dynamically Similar Teleoperated Manipulators", IEEE Trans. Robot. Automat., vol. 19, no. 3., pp. 443-456, June 2003.

[20] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, Control of Robot Manipulators, Macmillan Publishing Company, 1993.

[21] P.Y. Li and R. Horowitz, "Control of Smart Exercise Machines - Part I: Problem Formulation and Non-adaptive Control", IEEE Trans. Mechatronics, vol. 2, no. 4, pp. 237-247, December 1997.

[22] P.Y. Li and R. Horowitz, "Control of Smart Exercise Machines - Part II: Self Optimizing Control", IEEE Trans. Mechatronics, vol. 2, no. 4, pp. 247-248, December 1997.

[23] A. Liu, and S. Chang, "Force Feedback in a Stationary Driving Simulator", proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Vol. 2, Vancouver, Canada, pp.1711-1716, 1995.

[24] M.Van der Loos, S. Michalowski, and L. Leifer, "Design of an Omnidirectional Mobile Robot as a Manipulation Aid for the Severely Disabled", Interactive Robotic Aids, World Rehabilitation Fund Monograph #37 (R. Foulds ed.), New York, 1986.

[25] R.M. Mahoney, "The Raptor Wheelchair Robot System", Integration of Assistive Technology in the Information Age (M. Mokhtari, ed.), IOS, Netherlands, pp.135-141, 2001.

[26] U.B. Mandhatta, "Investigation of operator feedback in human-vehicle haptic interfaces for steer-by-wire transportation systems", Master's Thesis,

Department of Mechanical Engineering, Clemson University, Clemson, SC, 2005.

[27] V. Mills, J. Wagner, and D. Dawson, "Nonlinear Modeling and Analysis of Steering Systems for Hybrid Vehicles", proceedings of ASME International Mechanical Engineering Congress and Exposition, New York, NY, pp. 571-578, November 2001.

[28] V. Mills and J. Wagner, "Behavioral Modeling and Analysis of Hybrid Vehicle Steering Systems", proceedings of Institution of Mechanical Engineers, Part D, Journal of Automobile Engineering, Vol. 217, no. 5, pp. 349-361, 2003.

[29] J. W. Post, and E. H. Law, "Modeling, Simulation and Testing of Automobile Power Steering Systems for the Evaluation of On-Center Handling", SAE Paper No. 960178, 1996.

[30] Z. Qu, and J. Xu , "Model Based Learning Controls and their Comparisons using Lyapunov Direct Method" Asian Journal of Control, Special Issue on Learning Control, Vol. 4, No.1, pp. 99-110, March 2002.

[31] M.S. de Queiroz, D.M. Dawson, S. Nagarkatti, F. Zhang, Lyapunov-Based Control of Mechanical Systems, Cambridge: Birkhäuser, 2000.

[32] E. Rimon and D. E. Koditschek, "Exact Robot Navigation Using Artificial Potential Functions", IEEE Transactions on Robotics and Automation, Vol. 8, No. 5, pp. 501-518, (1992).

[33] E. J. Rossetter, J. P. Switkes, and J. C. Gerdes, "A Gentle Nudge Towards Safety: Experimental Validation of the Potential Field Driver Assistance System", proceedings of the American Control Conference, Denver, CO, pp. 3744-3749, June 2003.

[34] J. Ryu and H. Kim, "Virtual Environment for Developing Electronic Power Steering and Steer-by-Wire Systems", proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1374-1379, Kyongju, Korea, 1999.

[35] P. Setlur, D.M. Dawson, J. Wagner, and Y. Fang, "Nonlinear tracking Controller Design for Steer-by-wire systems", Proceedings of the American Control Conference, Anchorage, AK, pp. 280-285, May 2002.

[36] P. Setlur, D. Dawson, J. Chen, and J. Wagner, "A Nonlinear Tracking Controller for a Haptic Interface Steer-by-Wire Systems", Proc. of the IEEE Conference on Decision and Control, Las Vegas, NV, pp. 3112-3117, December 2002.

[37] J.J. Slotine and W. Li, Applied Nonlinear Control, New York: Prentice Hall, 1991.

[38] M. W. Spong and M. Vidyasagar, Robot Dynamics and Control, New York: John Wiley and Sons, Inc., 1989.

[39] J.T. Wen, D. Popa, G. Montemayor, P.L. Liu, "Human Assisted Impedance Control of Overhead Cranes," 2001 Conference on Control Applications, Mexico City, Mexico, Sept. 2001.

[40] B. Xian, M.S. de Queiroz, and D.M. Dawson, "A Continuous Control Mechanism for Uncertain Nonlinear Systems", Optimal Control, Stabilization, and Nonsmooth Analysis, Lecture Notes in Control and Information Sciences, Heidelberg, Germany: Springer-Verlag, to appear, 2004.

[41] Y. Yokokohji and T. Yoshikawa, "Bilateral Control of Master-Slave Manipulators for Ideal Kinesthetic Coupling - Formulation and Experiment, IEEE Trans. Robot. Automat. , vol. 10, pp. 605-620, Oct. 1994.

[42] T. Yoshikawa, "Analysis and control of robot manipulators with redundancy", In M. Brady and E. R. Paul, editors, Robotics Research: The First International Symposium, pages 735—747. MIT Press, 1984.

[43] X. Zhang, A. Behal, D.M. Dawson, and J. Chen, "A Novel Passive Path Following Controller for a Rehabilitation Robot", Proceedings of the IEEE Conference on Decision and Control, Paradise Island, Bahamas, pp. 5374-5379, December 2004.