Robust Adaptive Control Modeling of Human Arm Movements subject to Altered Gravity and Mechanical Loads

by

Michail Tryfonidis

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements of the degree of

Doctor of Philosophy

at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Author ....
Department of Aeronautics and Astronautics
May 19, 1999

Accepted by ....
Dava J. Newman
Associate Professor
Committee Chair

Accepted by ....
James K. Kuchar
Assistant Professor

Accepted by ....
Chi-Sang Poon
Principal Research Scientist
Thesis Supervisor

Accepted by ....
Professor Jaime Peraire
Department of Aeronautics and Astronautics

Robust Adaptive Control Modeling of Human Arm Movements subject to Altered Gravity and Mechanical Loads

by

Michail Tryfonidis

Submitted to the Department of Aeronautics and Astronautics
on May 24, 1999, in partial fulfillment of the
requirements of the degree of
Doctor of Philosophy

Abstract

It has been observed that during orbital spaceflight the absence of gravitation related sensory inputs causes incongruence between the expected and the actual sensory feedback resulting from voluntary movements. This incongruence results in a reinterpretation or neglect of gravity-induced sensory input signals. Over time, new internal models develop, gradually compensating for the loss of spatial reference.

The study of adaptation of goal-directed movements is the main focus of this thesis. The hypothesis is that during the adaptive learning process the neural connections behave in ways that can be described by an adaptive control method. The investigation presented in this thesis includes two different sets of experiments. A series of dart throwing experiments took place onboard the space station Mir. Experiments also took place at the Biomechanics lab at MIT, where the subjects performed a series of continuous trajectory tracking movements while a planar robotic manipulandum exerted external torques on the subjects' moving arms. The experimental hypothesis for both experiments is that during the first few trials the subjects will perform poorly trying to follow a prescribed trajectory, or trying to hit a target. A theoretical framework is developed that is a modification of the sliding control method used in robotics. The new control framework is an attempt to explain the adaptive behavior of the subjects. Numerical simulations of the proposed framework are compared with experimental results and predictions from competitive models.

The proposed control methodology extends the results of the sliding mode theory to human motor control. The resulting adaptive control model of the motor system is robust to external dynamics, even those of negative gain, uses only position and velocity feedback, and achieves bounded steady-state error without explicit knowledge of the system's nonlinearities. In addition, the experimental and modeling results demonstrate that visuomotor learning is important not only for error correction through internal model adaptation on ground or in microgravity, but also for the minimization of the total mean-square error in the presence of random variability. Thus human intelligent decision displays certain attributes that seem to conform to Bayesian statistical games.

Thesis Supervisor: Chi-Sang Poon

Title: Research Scientist

Acknowledgments

This is the section of my thesis that actually means the most to me because I have the opportunity to thank all those kind souls who stood by me for short or long time for the past five years. My doctoral degree marathon was the longest, most arduous, most undefined, and least funded task I have ever taken on. Thank God it is over.

I would like to especially thank David Clark for supporting me tremendously during the past four and a half years. (Dave, this is not just for the car rides, and the dinners, which I also appreciated a lot!)

Thanks for the unceasing prayers mom! I have read in a news magazine that doctors explore use of prayer to fight disease. Apparently, they should also do a study on the power of YOUR prayer on green-card and mortgage applications and doctoral thesis work.

Professor Markey, Professor Newman, Dr. McGee and Dr. Zhou, thank you for the reference letters that landed me a chance in this country. Professor Markey, thank you also for trusting me with the TA for your class, despite some "not-so-positive" remarks you heard about me. Being a TA in your class refreshed my knowledge of classical control theory, supported me financially for a term, and opened to me the door for more teaching, which I enjoyed tremendously. You are a gentleman and a scholar! I hope your back gets better.

Professor and Mrs. (Dr.) Essigmann your kind words have meant so much to me. You showed genuine concern about my wellbeing, and despite your extremely busy schedule you found the time and energy to listen to my "whining". John, your fruit smoothies are incredible!

Charlie Boppe, Professor Crawley, and Professor Rowell, working with you was not only a lot of fun and educational, but also crucial to my agenda to take over the world. Just kidding. It is a fact, however, that without your financial support I might have had to abandon my Ph.D. program, leave the US, and return to Greece.

NH4 students, living with you for over three years was of immense value to me. I hope you guys enjoyed the Fall hiking trips to the New Hampshire mountains, the dinners and the study breaks. I sure did. Please stay in touch.

Rich Domenico, you have been my friend for the past seven years. This must have been hard on you, and for that I thank you. I also want to thank you for being patient with me, sensitive, and a good listener when I was feeling down.

You brave souls, Rich, Mike, Jennifer, Karen, Jesse, Yee, Cindy, Danny, Mike, Tom, Christine, Mark and Dora, thank you for sparing a valuable hour of your young lives to be subjects in one or two of my experiments. Your presence made this thesis possible. Thanks!

Finally, I would like to thank the faculty members of my committee for helping along the way with suggestions and constructive comments.

Adios.

Table Of Contents

List Of Figures

List Of Tables

Nomenclature

a_1 The position gain in the generation of the horizontal disturbance force during the manipulandum experiment.

a_2 The velocity gain in the generation of the horizontal disturbance force during the manipulandum experiment.

a_i The length of the common normal in the Denavit-Hartenberg notation.

b_1 The position gain in the generation of the vertical disturbance force during the manipulandum experiment.

b_2 The velocity gain in the generation of the vertical disturbance force during the manipulandum experiment.

C The Coriolis and centripetal matrix of the entire armature.

d_i The distance between the origin O_{i-1} and the point H_i in the Denavit-Hartenberg notation.

E_K The kinetic energy.

E_P The potential energy.

$\underline{g}$ The acceleration of gravity with respect to the base coordinate frame.

$G(s)$ Transfer function notation.

$G_{cl}(s)$ Closed-loop transfer function.

H The inertia matrix of the entire armature.

K_D The positive definite matrix that represents the feedback gain in the MRAC for robotic systems.

L_2 The space of bounded-energy signals.

L The Lagrangian energy.

m_i The mass of the ith link.

J The manipulator Jacobian.

J_i The moment of inertia of the ith link.

$\underline{Q}$ The generalized force vector.

s The sliding mode variable or the Laplace transform variable.

$\underline{s}$ The sliding mode variable vector.

$\underline{x}$ Generic state vector.

$\underline{u}$ Generic control vector.

$\underline{v}_e$ The translational velocity of the end-effector.

z_i The ith zero of a PID compensator.

α_i The angle between the joint axis i and the z_i axis in the right hand sense in the Denavit-Hartenberg notation.

Γ The positive definite matrix that represents the adaptive gain in the MRAC for robotic systems.

$\underline{\theta}$ The angular position vector indicating the states of the entire armature.

$\underline{\tilde{\theta}}$ The angular position tracking error vector.

$\underline{\theta}_d$ The desired angular position vector.

θ_i The angle between the x_{i-1} axis and the common normal H_iO_i measured about the z_{i-1} axis in the right hand sense in the Denavit-Hartenberg notation.

$\underline{\theta}_r$ The “reference” angular position vector.

Λ The Hurwitz matrix used in the transformation from the tracking error vector to the sliding variable vector.

λ The positive number that represents the pole of the filter transformation from the tracking error to the sliding variable.

σ Standard deviation.

$\underline{\tau}$ The generalized torque vector or the vector of joint torques for the arm.

$\underline{\omega}_e$ The angular velocity of the end-effector.

1. Introduction and Review of Recent Literature

1.1. Hypothesis and Background

It has been observed that during orbital spaceflight the absence of gravitation-related sensory inputs causes an incongruence between the expected and the actual sensory feedback resulting from voluntary movements. This incongruence creates a corresponding sensory error for certain central adaptive processes involved in the mapping of the external environment into some "internal models", resulting in a reinterpretation or neglect of gravity-induced sensory input signals (Paloski, 1998). Over time, new internal models develop, gradually compensating for the loss of spatial reference. Eventually, the CNS no longer seeks gravitational sensory inputs for use in estimating spatial orientation. Postural internal model adaptation in spaceflight is very rapid but measurable (Collins et al., 1995; Grigorova and Kornilova, 1996; Lackner and DiZio, 1996; Paloski, 1998; Reschke et al., 1984).

In addition to influencing postural adaptation, microgravity also introduces multiple effects on motor control and visuomotor coordination of movement. Previous research on motor adaptation in space has been limited mostly to experiments involving either target pointing or simple trajectory tracking by the hand (Berger et al., 1997; Watt, 1997; Watt et al., 1986; Young et al., 1986). In the present investigation, we performed the more complex classic "dart-throwing" experiment for visuomotor adaptation (Thach et al., 1992) as a means to discern significant sensorimotor adaptation in microgravity.

The question that is raised is: "What are the methods through which the Central Nervous System (CNS) adapts to the presence of new dynamics, such as altered gravity?"

The study of adaptation of goal-directed movements is the main focus of this thesis. The hypothesis is that during an adaptive learning process the neural connections behave in ways that can be described by an "*adaptive control*" methodology. Part of this thesis includes two different sets of experiments. First, in the Summer of 1996 and the Spring of 1997, a series of microgravity dart throwing experiments took place onboard the Russian space station Mir (МИР). Second, in the Spring of 1999, a series of ground-based experiments took place at the Biomechanics lab at MIT, where the subjects performed continuous trajectory tracking movements, while a planar robotic manipulandum exerted external torques on the subjects' moving arms. The experimental hypothesis for both experiments is that during the first few trials the subjects will perform poorly trying to follow a prescribed trajectory, or trying to hit a target, due to the modified external dynamics.

1.2. Purpose and Procedure

The external torques induced by the robotic manipulandum on the ground and also the lack of gravity on the space station, introduce new arm dynamics. The absence of gravitational forces acting on the subjects' arms and the induced mechanical torques during the tracking movements could cause degraded performance. However, the subjects' performance is expected to improve as they adjust to their new arm dynamics.

The data from the space station supplements the ground data, since only a limited number of trials were performed, and three dimensional video footage (which is necessary for position, velocity, and acceleration calculations) was not available. According to the hypothesis, initially, the throwing performance of the astronauts/cosmonauts is expected to be worse in microgravity than on the ground, but that their performance will improve as they perform the task several times. This hypothesis seems quite reasonable, since throughout our lives our CNS has acquired a very good representation of the kinematic and dynamic behavior of our arms in the presence of the external loading of gravity. The motor control learning of the CNS, just described, is a trial and error process. For example, toddlers trying to learn how to stand up or walk are formulating their motor control programs. Similarly, when astronauts are going from 1G to microgravity or to artificial G environments the actual musculoskeletal system dynamics change. The operating point of a previously well behaved non-linear system changes. However, the controller (CNS) sends to the plant (arm) an input (neural activation) that is appropriate only for the plant dynamics in 1G. This input can not guarantee accurate tracking (the arm following the proper trajectory for target hit) under the new conditions. Therefore, the CNS has to estimate the new internal model in order to bring the new system dynamics to an optimal stable operating point.

This thesis is an attempt to model the way the human CNS achieves desired trajectory tracking when performing an arm-motor task in altered gravity. The proposed model is compared to several other human-arm control models in terms of error minimization, convergence time, stability and robustness. Further, the proposed model is compared to experimental data.

As a first step toward a quantitative analysis of motor learning, I tried to test a functional model of the adaptive motor control system. The proposed model is chosen because it is a simple representation of the motor learning system that incorporates an internal model. Further, the model includes the motor system, the mechanical environment, the internal model, and the controller.

The *motor system* includes the musculoskeletal system for force and torque generation and part of the primary motor cortex that drives it. Its dynamical behavior is dependent on passive system dynamics, which is given by the some of inertial, Coriolis, centripetal, and frictional forces.

The *environment* subsystem is the mechanical load to the motor system resulting from an external force field. For goal-directed movements in the horizontal plane the force field is negligible. In the experimental procedures, an artificial force field is generated by means of a robot arm to produce varying mechanical perturbations to the arm movement. The performance of the motor learning system in adaptively countering the external perturbation can be studied systematically by carefully choosing the types of force fields that are presented to the subjects' arms.

The motor system and its mechanical environment are coupled together to form the system dynamics that the neural controller has to learn to guide the human arm accurately along the desired trajectory. A command to the motor system may result in different movement trajectories depending on the environment force field. Such perturbations in movement trajectories are corrected through proprioceptive and tactile feedback, which convey information about the position and speed of movement. However, the excessive time delays (40-60 ms; Evarts, 1981) and the limited gains, which are inherent in such long sensory feedback loops, make it impossible to achieve completely accurate trajectory tracking under the environmental perturbations.

The model in this thesis predicts trajectories, in terms of joint positions, velocities and accelerations. These are the same variables that the experimental setup recorded. The ground experiment itself has been designed to put the proposed model to the test. Predictions of the proposed model as well as of some competitive models are compared with the experimental findings. For instance, trying to track a fast oscillatory trajectory in the presence of strong external forces provides a more accurate and complete understanding of the control strategies that the CNS employs than other experimental setups that include pointing tasks (Dizio and Lackner, 1995; Shadmehr and Mussa-Ivaldi, 1994). Although both pointing tasks and trajectory tracking tasks appear similar at first, they are quite different. Many models perform well and their behavior looks similar for transitions from one steady state condition to another (pointing tasks), while they behave very differently during cyclic inputs.

1.3. Thesis Road Map

The work in Chapter 2 includes basic concepts in dynamics and coordinate system transformation and an introduction of a particular system of rules for choosing coordinate systems for each link of a general robotic armature. Fundamental background work on the coordinate transformation within this set of coordinate systems is provided, and important conclusions are presented. Finally, the coordinate frames are defined (according to the set rules) for all the links of the robotic model of the arm. The dynamical analysis of this particular set of coordinates frames is presented. The results of the dynamical analysis are necessary for the design of a controller to control a multiple-link arm, which comes in Chapter 3.

The organization of the Chapter 3 includes an introduction and general literature review on the areas of biological systems, intelligent controllers and neural adaptive control. An indepth review of fundamental work (Slotine and Li, 1991) in the area of sliding mode control and nonlinear adaptive control is presented. This work forms the basis of the proposed control model, which is shown last. The rigorous proof of the robust stability of the proposed control model on any robotic arm (and therefore the seven-link arm representing the human arm) is presented, and simulation examples presented to show the performance of the proposed model.

Chapter 4 includes an experimental study on continuous trajectory tracking under external dynamics, caused by a robotic manipulandum. Chapter 4 describes the comparison of experimental results during continuous trajectory tracking with the model predictions. Finally, Chapter 5 will concentrate more the throwing experiment on board Mir and draw conclusions on the effect of gravity in the selections of a throwing strategy .

1.4. Review of Relevant Literature

There have been quite a few papers published in the area of arm motion modeling. Of special interest are those papers that directly or indirectly deal with the adaptive nature of arm movements in the presence of an external force field. The works summarized here are divided into three distinct areas: 1) Experimental evidence 2) Physiology of the CNS, and 3) Adaptive learning control of movement.

1.4.1. The Experimental Evidence

During the 41-G space shuttle mission, two payload specialist astronauts took part in six Canadian medical experiments designed to measure how the human nervous system adapts to weightlessness, and how this might contribute to space motion sickness (Watt et al., 1985; Watt et al., 1986). Fight data was compared to pre-flight baseline data to examine adaptation to microgravity and also to post-flight data to examined re-adaptation to 1G. Pronounced proprioceptive illusions were experienced, especially, immediately post-flight. Also, the ability to judge limb position was degraded, and estimates of the location of familiar targets were grossly distorted in the absence of vision. Awareness of arm and target position was tested by putting the subjects through an active pointing task. Both subjects were able to point quite accurately at the different targets pre-flight after a minimal amount of training and practice. While, initially, both subjects demonstrated very poor accuracy on-orbit, one subject adapted to the new microgravity environment and improved his task performance during the course of the flight.

While Watt et al., clearly demonstrated the effect of the change in the gravitational environment onto proprioception, the fact that proprioception closely affects motor control is shown by Rothwell et al., 1982. More recently, Bock et al., 1992, investigated whether the accuracy of aimed arm movements deteriorates in simulated altered gravity. The authors attribute the discrepancies between limb position sense in simulated gravity to degraded proprioceptive information about arm position. The results of the experiment indicated that the response variability was larger in the reduced gravity simulation than in normal 1G conditions, and that the joint position sense was systematically biased in simulated microgravity in the absence of visual feedback. Finally, when visual feedback was provided between trials, systematic errors were not significantly different in simulated microgravity from normal gravity, and the increase of variable error was less pronounced.

Shadmehr and Mussa-Ivaldi, 1994, investigated how the CNS learns to control movements in different dynamical conditions, and how this learned behavior is represented in the CNS. In particular, they considered the task of making reaching movements in the presence of externally imposed forces from a mechanical environment. This environment was a force field produced by a robot manipulandum, and the subjects made reaching movements while holding the end-effector of this manipulandum. Since the force field significantly changed the dynamics of the task, subjects' initial movements in the force field were grossly distorted compared to their movements in free space. However, with practice, hand trajectories in the force field converged to a path very similar to that observed in free space. The motor adaptation also indicated that for reaching movements, there was a kinematic plan independent of dynamical conditions.

Reaching movements made in a rotating room generate Coriolis forces that are directly proportional to the cross product of the room's angular velocity and the human's arm linear velocity. Such Coriolis forces are inertial forces not involving mechanical contact with the arm. Dizio and Lackner, 1995, measured the trajectories of arm movements made in darkness to a visual target that was extinguished at the onset of each reach. Pre-rotation subjects pointed with both the right and left arms in alternating sets of eight movements. During rotation at 10 rpm, the subjects reached only with the right arm. Post-rotation, the subjects pointed with the left and right arms, starting with the left, in alternating sets of eight movements. The initial reaching movements during rotation of the right arm were highly deviated both in movement path and endpoint relative to the pre-rotation reaches of the right arm. With additional movements, subjects rapidly regained straight movement paths and accurate endpoints despite the absence of visual or tactile feedback about reaching accuracy.

1.4.2. Physiology of the CNS

While there is a very extensive literature on the physiology of the CNS, this section summarizes some of the more recent and relevant papers. Soechting and Flanders, 1992, made insightful observations about internal representation of frames of reference, vectors, and coordinate systems. In their study the authors were able to identify coordinate systems for three motor tasks. In all three motor tasks one of the coordinate axes was defined by the gravitational vertical. Another coordinate was defined by a sagittal horizontal axis. A neural representation in a coordinate system generally defines a best direction along which activity is maximal. Neural activity decreases by an amount proportional to the cosine of the angle for inputs or outputs oriented along directions other than best directions. Vestibulospinal neurons appear to provide a vector-code of rotation in an earth-fixed frame of reference, of either the head or the trunk. In most instances, the vectors of each of the afferent inputs (semicircular canals, otoliths, and neck afferents) are in approximate alignment.

Kalaska and Crammond, 1992, addressed two questions: 1) What are the planning steps between the decision to make a reaching movement and its initiation? and 2) How are those steps implemented in the CNS? The authors described neurophysiological studies that reveal neuronal correlates of several putative hierarchical planning stages for reaching movements, distributed across three cortical areas. The different combinations of response properties in each area imply that each cortical area has primary, but not exclusive, responsibility for certain representations or transformations. Each of these representations may be involved to varying degrees in planning several aspects of movement. Thus, the findings do not support an absolute segregation of response properties within a strict serial hierarchy of areas but are more consistent with a heterarchy of nested interconnected cortical populations. In a heterarchy, there is no fixed chain of command or direction of information flow; instead, the flow of information is flexible and dependent upon the context of the event, allowing different movement attributes or cortical areas to take priority under different behavioral circumstances.

Graziano et al., 1994, took the coding of three-dimensional space one step further. Their experiments with primates revealed that in primates, the premotor cortex is involved in the sensory guidance of movement, supporting the general principle that space is encoded in different brain structures for different behavioral functions. This view of multiplicity of spatial structures and coordinate systems contrasts with commonly held views that all of visual space is encoded by one master coordinate system, probably centered on the point between the eyes and located in the posterior parietal cortex.

During a serial reaction time test, subjects developed implicit knowledge of the test sequence that was reflected by diminishing response times (Pascual-Leone et al., 1994). Motor cortical mapping with transcranial magnetic stimulation revealed that the cortical output maps to the muscles involved in the task became progressively larger until explicit knowledge was achieved, after which they returned to their baseline topography. The development of explicit knowledge resulted in a continued improvement in response time because of a change in the subject's strategy (anticipation instead of reaction), while there was a rapid return of the cortical motor outputs to their baseline topography. This return of the maps suggests that as a motor sequence is explicitly learned the contribution of the motor cortex is attenuated and other brain structures assume more active roles in the execution of the task. The consistent coupling between changes in type of knowledge and cortical motor outputs strongly suggests that the latter reflect a sustained cortical modulation associated with motor learning. The rapid time course of this modulation is most compatible with unmasking of previously existing connections, perhaps as a result of decreased inhibition or increased synaptic efficacy in existing neural circuits. Such flexible short-term modulation seems important in the acquisition of new skills and could lead to structural changes in the intracortical and subcortical networks as the skill becomes learned and automatic.

Functional imaging of the brain demonstrates that within 6 hours after completion of practice, while performance remains unchanged, the brain engages new regions to perform the task; there is a shift from prefrontal regions of the cortex to the premotor, posterior parietal, and cerebellar cortex structures. This shift is specific to recall of an established motor skill and suggests that with the passage of time, there is a change in the neural representation of the internal model and that this change may underlie its increased functional stability (Shadmehr and Holcomb, 1997).

More recently, investigators studied at the cellular level the role of the CNS onto motor learning performance. Welsh et al., 1995, studied the dynamic organization of motor control within the olivocerebellar system. They found that a burst of action potentials known as the complex spike were recorded simultaneously from 26-29 purkinje cells with as many microelectrodes individually placed into a 1.7-2.8 mm^2 area of the cerebellum. Crosscorrelation of the population complex spike activity to the behavior revealed that olivocerebellar activity was highly rhythmic and significantly time related to the movement ($\rho < 0.05$). The results show that olivocerebellar control of movement derives from populations of olivary neurons operating as a distributed system whose collective activity is rhythmic and temporally related to specific parameters of movement. Although olivary neurons fire infrequently during sequences of movement, they do so at highly defined moments, indicating that olivary control of movement is not encoded by single neurons in the frequency domain, but rather in the temporal domain across neuronal populations. In fact, the contribution of single olivary neurons to the population activity is actually diminished during movement, because the increase in the size of active population exceeds the degree to which single neurons increase their firing. The authors proposed that the inferior olive is a mosaic of motor representation in which individual muscles are represented by discrete neuronal ensembles embedded within a large electrically coupled network.

McCormick, 1995, reported that the inferior olive may be using a population code (i.e., the ensemble of activity in this structure may be the way in which it contributes to the coordination of movement). Ojakangas and Ebner, 1994, also studied the complex spike activity in relationship to the changes in arm movement velocity during a motor learning experiment. Two primate subjects had to move a hand-cursor from one point to another. The gain in the hand-cursor was changed and the animals had to adjust to the new gain. In many of the purkinje cells studied complex spike activity increased for a period of 200-300 ms starting just before movement onset.

Summarizing the literature on the physiology of the CNS, there is clear evidence that the CNS is greatly involved in the coordination of movement. The CNS, not only forms coordinate systems and frames of reference during the execution of motor tasks, but also is actively involved in the execution of movement (complex spike activity of the inferior olivary). More importantly, the CNS is also involved in the adaptive learning process of a motor task by creating new neural connection when new tasks are learned.

1.4.3. Control Modeling of the Human Arm

In recent years, there have been numerous attempts to develop an accurate control model of the adaptive learning behavior of the CNS during the execution of arm movements. The next sections give a summary of the competitive control models and present the caveats that are inherent with each of the approaches.

Bizzi et al., 1992, provided a summary of the literature on the equilibrium-point hypothesis, which is one of the most recent attempts to model the musculoskeletal system as a mass-spring-damper mechanical system. A key feature of the equilibrium-point hypothesis is that muscles have spring-like behavior in the sense that the force they generate is a function of their length and neural activation level. The force-length relationship in individual muscle fibers was studied by Gordon et al., 1966. Feldman suggested that the CNS may execute a movement by generating signals that change the relative activation of agonist and antagonist muscles (Feldman and Levin, 1995). A movement occurs as a result of changes in the equilibrium point of the system. Higher centers of the brain guide joint movement by a gradual shift in equilibrium states, while maintaining proper joint stiffness (as predicted by the equilibrium-point hypothesis). However, according to Gottlieb (Gottlieb, 1996), equilibrium-point control models do not explain “how to control the equilibrium point to move different external loads, since this gets into the inverse dynamics problem”. Also, the results of a series of experiments by Gomi and Kawato argue against the hypothesis that the brain sends as a motor command only an equilibrium point trajectory similar to the actual trajectory (Gomi and Kawato, 1996).

Kawato et al., 1987, proposed a hierarchical neural-network model for control and learning of voluntary movement. In their model the association cortex sends the desired motor pattern, x_d, expressed in body coordinates, to the motor cortex where the joint torque, u, is calculated. The musculoskeletal system interacts with the environment to realize some kind of motor pattern, x, which, in general, does not coincide with x_d. The actual motor pattern (x) and its derivative ($\dot{x}$) (e.g., muscle length and its derivative) are measured by proprioceptors, and they are sent back to the motor cortex. Theoretically the feedback loop could close using the error signal $x_d - x$. However, long time delays, as well as small gains limit controllable speeds of motions. Consequently, one must first execute the movement very slowly because otherwise the control system becomes unstable. Alternatively, the spinocerebellum predicts the output motor pattern x^*, using x, u and an internal learned model. The error $x_d - x^*$ is transmitted back to the motor cortex. This feedback loop has a much shorter time delay than the one in which the actual error ($x_d - x$) is measured. In higher primates (especially humans) the cerebrocerebellum monitors both x_d and u, but it does not receive any information about x. Kawato et al. postulated that in the cerebrocerebellum, a neural network slowly develops the appropriate inverse-dynamics model that can compute the motor command (i.e., joint torque, u^*), directly from x_d. Figure 1 shows the hierarchical neural network model proposed by Kawato. In this control approach the internal dynamics model and the internal inverse-dynamics model can be realized by a parallel-distributed-processing neural network with heterosynaptic plasticity. The basic organization of such a neural network is the same as the LMS (Least Mean Square) adaptive filter of Widrow et al., 1976, and the adaptive filter model of Fujita (Fujita, 1982b; Fujita, 1982a).

Kawato, in a consecutive paper, presented a computational scheme for acquiring the inverse model through learning (Kawato and Gomi, 1992). In the *feedback-error-learning* approach, a summation of the feedback motor command and the feedforward command, generated by the inverse model, is fed to the controlled object and the feedback controller transforms the trajectory error into the motor command error (Figure 2). The inverse model is trained during motor control using the feedback motor command as the error signal.

However, the feedback-error-learning approach achieves trajectory tracking by inverting the forward dynamics model. When training is complete the inverse dynamics cancel the forward model. This control method has two very serious disadvantages. First, if the forward model has any non-minimum phase zeros, the inverse model will have poles in the right half plane (RHP), making the inverse dynamics unstable. Second, the forward model behaves like a low pass filter, making the inverse model a high pass filter, which may be prone to noise amplification. Further, the proposed Kawato et al. model represents a purely empirical model, with no rigorous proof of stability, convergence or robustness (except by numerical simulation on a case by case basis). Therefore, there is no guarantee that the system will perform well under severe control system conditions.

Gerdes and Happee, 1994, studied single joint goal-directed movements, and initially visual feedback was omitted. A non-linear third-order lumped parameter muscle model was adopted from Winters and Stark (Winters and Stark, 1985; Winters and Stark, 1988). The model parameters were chosen to match anteflexion/retroflexion in the shoulder (Happee, 1992). Eight states were defined: one for the arm position (ϕ (rad)), one for the arm velocity ($\dot{\phi}$ (rad/s)), and three for each muscle's neural excitation, active state, and length of the contractile element. Gerdes and Happee took a step in the right direction by using muscle physiological data in their control model. However, the linear Kalman estimators work for Linear Time Invariant (LTI) systems. Moreover, the modeling was only performed for a single-degree-of-freedom movement of a single joint without any disturbances present.

Clearly, a great number of investigators have focused on the control strategies that the CNS employs during the control of voluntary arm movements. However, the modeling attempts have concentrated on simple step responses (i.e., going from an initial zero-velocity state to a final zero-velocity state). Such models are hard to prove since any controller with a simple integrator can perform exact tracking for these final conditions. A more appropriate test for these control models is their performance during continuous trajectory tracking such as the task used in the second series of experiments described in detail in Chapter 4. The dynamical and control model presented in this thesis does not include any muscle dynamics. However, a Hill-type muscle model was simulated and is shown in Appendix A. Incorporation of the muscle model into the control of the arm dynamics is beyond the scope of this thesis. However, the analysis on that model is provided since incorporating the muscle dynamics would be a natural next step of this work.

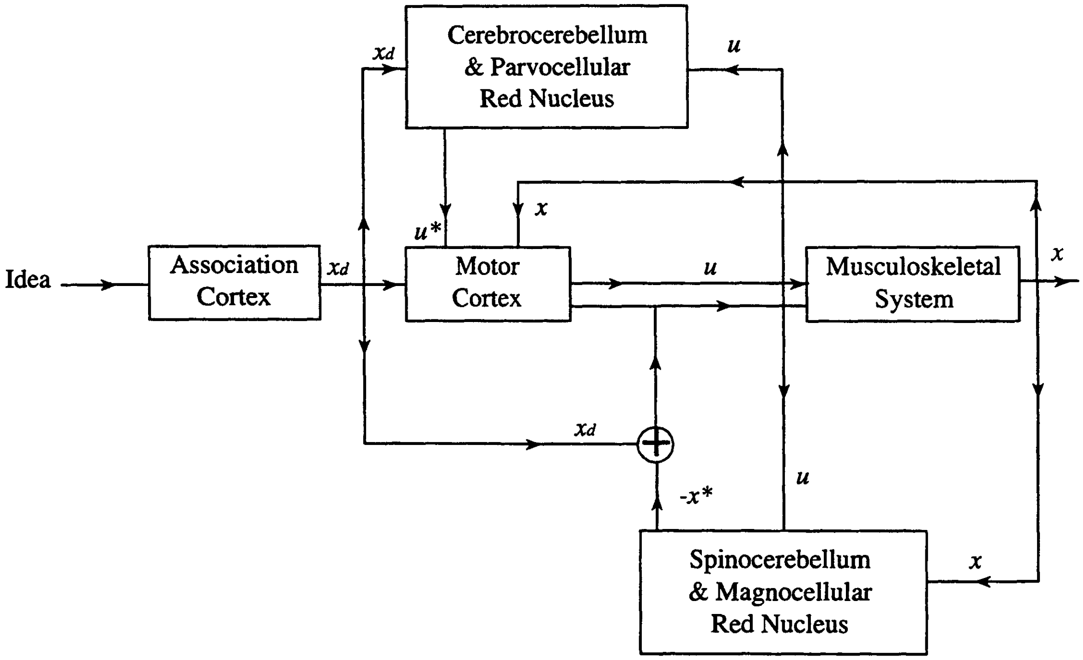


Figure 1.1 Feedback-Error-Learning block diagram. After training is complete the actual trajectory is the same as the desired trajectory (Kawato et al., 1987).

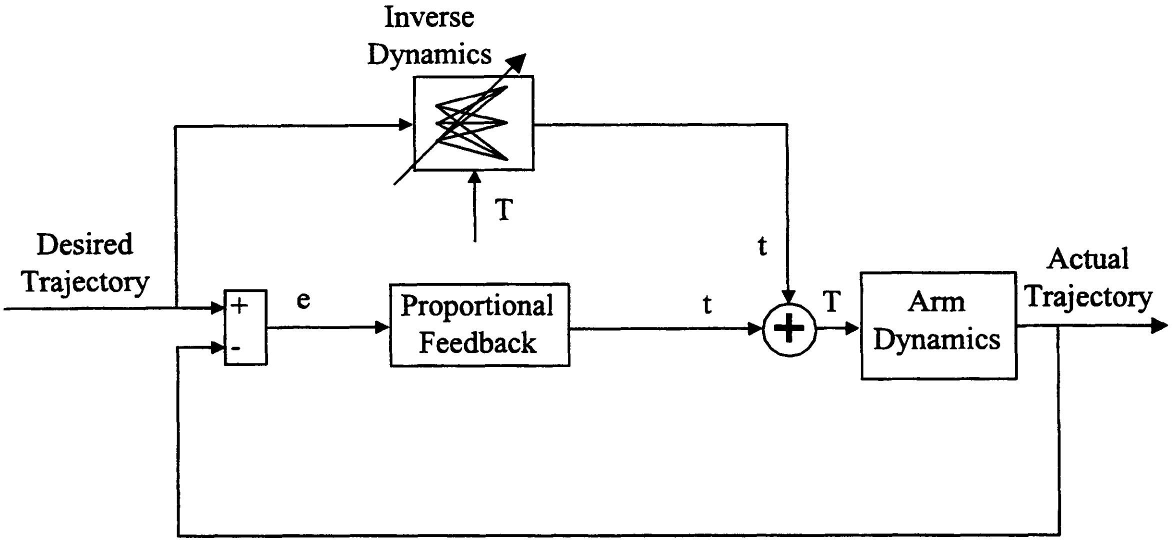


Figure 1.2 A hierarchical neural network model for control and learning of voluntary movement (Kawato and Gomi, 1992).

2. Dynamical Modeling of the Human Arm

This research is unique in the fact that the three-dimensional dynamics of the human arm are highly nonlinear, time-varying and present quite a challenge when one attempts to control them, especially under the feedback restrictions of biological control. Since performance of specific tasks is achieved through the movement of the human arm, kinematics is an essential tool in manipulator design and control. Fundamental work in the area of formulating the manipulator kinematics and dynamics has been done by several researchers in the past (Asada and Slotine, 1986; Corke, 1996) and application of those results in this analysis is quite straightforward.

The work in this chapter is broken down into three major parts. The first part includes basic concepts in dynamics and coordinate system transformation. The second part is the introduction of a particular system of rules for choosing coordinate systems for each link of a general robotic armature. Fundamental background work on the coordinate transformation within this set of coordinate systems is provided, and important conclusions are presented. The fundamental results that come out of the choice of coordinate frames are crucial for the control theory development that is presented in Chapter 3. Finally, in the third part the coordinate frames are defined (according to the set rules) for all the links of the robotic model of the arm. The dynamical analysis of this particular set of coordinates frames is presented. The results of the analysis are necessary for the design of a controller to control a seven-link arm according to the description of the proposed control system (Chapter 3).

2.1. Modeling Preliminaries

2.1.1. Modeling Assumptions

Most of the background on dynamical modeling that is shown in this chapter is work developed by Asada and Slotine, 1986. Before attempting to develop a control method for human arm movements, it is necessary to obtain an accurate forward dynamics model. This model will not be used by itself in the calculation of the forward dynamics in the simulations of Chapter 3. It will be used to derive the "linearized" dynamics of the system to derive the control law of the proposed model (Chapter 3). For the sake of simplicity, the human arm is modeled as a three-link robotic manipulator. Thus, the human arm dynamics is modeled as a collection of three massive rigid segments (upper arm, forearm and hand) interconnected at three joints (shoulder, elbow and wrist). Each segment has a longitudinal axis of geometric and mass symmetry that passes through the joints and contains the center of gravity. For each segment it is necessary to know, either by direct measurement or by estimation, the mass, the location the center of mass, and the six independent components of the 9 element, symmetric, mass-center inertia matrix. Expressed in terms of local system components these six independent inertia matrix elements include the three moments of inertia and the three products of inertia. Finally, the model of the human arm that was chosen has seven degrees of freedom. For reasons of convenience each joint consists of two or three pin joints of zero length - one for each degree of freedom.

2.1.2. Coordinate Transformations

The dynamical analysis of the robotic equivalent of the human arm model is based on the development by Asada and Slotine (1986). Let P be an arbitrary point in space, as shown in Figure 2.1. The coordinates of point P with reference to the fixed frame *O-xyz* are:

$$\underline{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

The position of point *P* can also be represented with reference to the coordinate frame fixed to the rigid body, $O' - x_b y_b z_b$, by:

$$\underline{x}^b = \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

The superscript ^{b} indicates that the vector is defined with reference to the body coordinate frame. The position and orientation of the rigid body are represented by the 3×1 vector $\underline{x}_0$. It is easy to see from Figure 2.1 that the point *P* can be reached through points *O'*, *A* and *B*. This can be represented by:

$$\underline{OP} = \underline{OO'} + \underline{O'A} + \underline{AB} + \underline{BP} \text{ or } \underline{x} = \underline{x}_0 + u_n + v_t + w_b, \qquad \textbf{eq. 2.1}$$

or $$\underline{x} = \underline{x}_0 + R\underline{x}^b$$

where $R = [\underline{n}, \ \underline{t}, \ \underline{b}]$. The vectors $\underline{n}$, $\underline{t}$, and $\underline{b}$ are in fixed coordinates. The equations above provide the desired coordinate transformation from the body coordinates $\underline{x}^b$ to the fixed coordinates $\underline{x}$. By multiplying both sides to the previous equations by the transpose R^T of matrix *R* one can retrieve the inverse transformation as well.

2.1.3. Euler Angles

Asada and Slotine, 1986, continue by defining the Euler angles. Consider the three rotations of frame $O - xyz$ shown in Figure 2.2. First, the coordinate frame is rotated about the *z* axis by an angle *ϕ*. Secondly, the new coordinate frame $O - x'y'z$ is rotated about the x' axis by an angle *θ*. Finally, the newest frame $O - x'y''z''$ is then rotated about the z'' axis by an angle *ψ*. The resultant coordinate frame $O' - x_b y_b z_b$ is shown in Figure 2.2c. The three angles *ϕ*, *θ*, and *ψ* are referred to as *Euler angles* and they uniquely determine the orientation of the coordinate frame. However, there are an infinite number of Euler angles that correspond to each orientation.

The three coordinate transformations shown in Figure 2.2 happen in the following order:

$$\underline{x} = R_z(\phi)\underline{x}' \qquad R_z(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\underline{x}' = R_{x'}(\theta)\underline{x}'' \qquad R_{x'}(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$\underline{x}'' = R_{z''}(\psi)\underline{x}^{\mathrm{b}} \quad R_{z''}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Putting it all together:

$$\underline{x} = R(\phi,\theta,\psi)\underline{x}^b = R_z(\phi)R_{x'}(\theta)R_{z''}(\psi)\underline{x}^b \qquad \textbf{eq. 2.2}$$

In eq. 2.2 the first term of the right-hand side represents the translational transformation, while the second term represents the rotational transformation.

Let, $\underline{X} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$ and $X^b = \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix}$, and the 4×4 matrix $A = \begin{bmatrix} R & \underline{x}_0 \\ O & 1 \end{bmatrix}$

Then, eq. 2.1 can be written as

$$\underline{X} = A\underline{X}^b \qquad \textbf{eq. 2.3}$$

Now, consider n consecutive transformations from frame n back to frame 0. Let A_i^{i-1} be the 4×4 matrix associated with the transformation from frame i to frame i-1. Then a position vector $\underline{X}^n$ in frame n is transformed to $\underline{X}^0$ in frame 0 by $\underline{X}^0 = A_1^0 A_2^1 \cdots A_n^{n-1}\underline{X}^n$. Thus, the consecutive transformations can be described by a single term.

2.2. Kinematic Modeling of Manipulator Arms

As mentioned above, the human arm will be modeled as a robotic manipulator with rigid links to make the analysis manageable. A manipulator arm is basically a series of rigid bodies in a kinematic structure. The human arm, like other robotic arms in the industry form an *open kinematic chain.* Each link of the open kinematic chain can be numbered in series from 0 to *n*. The base link, which is basically fixed to the ground, is numbered 0 for convenience. The most distal link, on the other hand, is numbered *n*.

In order to represent the position and orientation of the end-effector, a coordinate frame $O_n - x_n y_n z_n$ was attached to the last link. The location of that coordinate frame is described with reference to another frame $O_0 - x_0 y_0 z_0$, fixed to the ground. The position and orientation of frame $O_i - x_i y_i z_i$ relative to the previous frame $O_{i-1} - x_{i-1} y_{i-1} z_{i-1}$ is described by using the 4×4 matrix of the homogeneous transformation between these two frames. The end-effector position and orientation is then obtained by consecutive homogeneous transformations from the last frame back to the base frame.

2.2.1. The Denavit-Hartenberg Notation

The *Denavit-Hartenberg* (DH) notation (Corke, 1996; Asada and Slotine, 1986) is a systematic method of describing the kinematics of the human arm and more generally of any robotic manipulator. The method is based on the 4×4 matrix representation of rigid body position and orientation, which was described in the previous paragraphs. There are two differing DH methodologies each of which allows some freedom in the actual coordinate frame attachment. We chose what is called the "standard" form, which is the most widely used form in the literature (Corke, 1996). Also, the robotics toolbox of Matlab (Mathworks Inc., Natick, MA), which we use in the simulations for the calculations of the forward dynamics uses this standard notation.

Figure 2.3 shows a given pair of adjacent links and their associated joints. The line $H_i O_i$ is the common normal to joint axes *i* and *i*+1. The relationship between the two links is described by the relative position and orientation of the two coordinate frames attached to the two links.

In the Denavit-Hartenberg notation the origin of the *i*th coordinate frame O_i is located at the intersection of joint axis *i*+1 and the common normal between joint axes *i and i*+1. Note that the frame of link *i* is at joint *i*+1 rather than at joint *i*. The x_i axis is directed along the extension line of the common normal, while the z_i axis is along the joint axis i+1. Finally, the y_i is chosen such that the resultant frame $O_i - x_i y_i z_i$ forms a right-hand coordinate system.

The relative location of the two frames can be completely determined by the following four parameters:

a_i the length of the common normal.
d_i the distance between the origin $O_{i\text{-}1}$ and the point H_i.
α_i the angle between the joint axis *i* and the z_i axis in the right hand sense.
θ_i the angle between the x_{i-1} axis and the common normal H_iO_i measured about the z_{i-1} axis in the right hand sense.

Figure 2.4 gives the coordinate transformation from $\underline{X}^i$ to $\underline{X}'$ as

$$\underline{X}' = A_i^{\text{int}} \underline{X}^i \quad \text{where } A_i^{\text{int}} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\underline{X}^{i-1} = A_{\text{int}}^{i-1} \underline{X}' \quad \text{where } A_i^{\text{int}} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_i \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Combining the two transformations one can solve for the total transformation from $\underline{X}^i$ to $\underline{X}^{i\text{-}1}$. The resultant transformation is

$$\underline{X}^{i-1} = A_i^{i-1} \underline{X}^i$$

where

$$A_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \textbf{eq. 2.4}$$

For the n-th link the origin of the coordinate frame can be chosen at any convenient point of the end-effector. The orientation of the last coordinate frame must be such that the x_n axis intercepts the last joint axis at a right angle, while the angle α_n is arbitrary. For the base link the origin of the coordinate frame can be chosen at an arbitrary point on the joint axis 1, while the z_0 axis must be parallel to the joint axis.

2.2.2. Infinitesimal Rotations

Infinitesimal rotations or time derivatives of orientations are substantially different from finite angle rotations and orientations. The analysis from Asada and Slotine (1986), begins by writing the 3×3 rotation matrix representing infinitesimal rotation $d\phi_x$ about the x axis:

$$R_x(d\phi_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(d\phi_x) & -\sin(d\phi_x) \\ 0 & \sin(d\phi_x) & \cos(d\phi_x) \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -d\phi_x \\ 0 & d\phi_x & 1 \end{bmatrix} \qquad \text{eq. 2.5}$$

The result of consecutive rotations about the x and y axes is given by:

$$R_x(d\phi_x)R_y(d\phi_y) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -d\phi_x \\ 0 & d\phi_x & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & d\phi_y \\ 0 & 1 & 0 \\ -d\phi_y & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d\phi_y \\ 0 & 1 & -d\phi_x \\ -d\phi_y & d\phi_x & 1 \end{bmatrix}$$

Notice that the infinitesimal rotations are *commutative*. Adding a third rotation is trivial,

$$R_x(d\phi_x)R_y(d\phi_y)R_z(d\phi_z) = \begin{bmatrix} 1 & -d\phi_x & d\phi_y \\ d\phi_x & 1 & -d\phi_x \\ -d\phi_y & d\phi_x & 1 \end{bmatrix}. \qquad \text{eq. 2.6}$$

Further, it is easy to see that when higher order terms are neglected, two consecutive infinitesimal rotations yield:

$$R(d\phi_x, d\phi_y, d\phi_z)R(d\phi_x', d\phi_y', d\phi_z') = \begin{bmatrix} 1 & (d\phi_z + d\phi_z') & -(d\phi_y + d\phi_y') \\ -(d\phi_z + d\phi_z') & 1 & (d\phi_x + d\phi_x') \\ (d\phi_y + d\phi_y') & -(d\phi_x + d\phi_x') & 1 \end{bmatrix}$$

$$= R(d\phi_x + d\phi', d\phi_y + d\phi_y', d\phi_z + d\phi_z') \qquad \text{eq. 2.7}$$

Therefore, infinitesimal rotations are also *additive*. Thus, infinitesimal rotations can be treated like vectors, unlike finite rotations which are not commutative.

2.2.3. The Manipulator Jacobian

For the n degree-of-freedom manipulator arm, we denote the infinitesimal end-effector translation by a three-dimensional vector $d\underline{x}_e$, and the infinitesimal end-effector rotation by a three-dimensional vector $d\underline{\phi}_e$ (Asada and Slotine, 1986). Both vectors are referenced to the base coordinate frame $O_0 - x_0 y_0 z_0$. Remember, that for any joint angle configuration the position vector $\underline{x}_e = \begin{bmatrix} x & y & z \end{bmatrix}^T$ is a function of all joint angles.

$$\underline{x}_e = \begin{bmatrix} x(\theta_1 \ \cdots \ \theta_7) & y(\theta_1 \ \cdots \ \theta_7) & z(\theta_1 \ \cdots \ \theta_7) \end{bmatrix}^T \qquad \textbf{eq. 2.8}$$

The infinitesimal change of $\underline{x}_e$ can be found by using partial derivatives and the chain rule of differentiation.

$$d\underline{x}_e = \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial x(\theta_1 \ \cdots \ \theta_n)}{\partial \theta_1} & \frac{\partial x(\theta_1 \ \cdots \ \theta_n)}{\partial \theta_2} & \cdots & \frac{\partial x(\theta_1 \ \cdots \ \theta_n)}{\partial \theta_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial z(\theta_1 \ \cdots \ \theta_n)}{\partial \theta_1} & \frac{\partial z(\theta_1 \ \cdots \ \theta_n)}{\partial \theta_2} & \cdots & \frac{\partial z(\theta_1 \ \cdots \ \theta_n)}{\partial \theta_n} \end{bmatrix}}_{J_L} \begin{bmatrix} d\theta_1 \\ d\theta_2 \\ \vdots \\ d\theta_n \end{bmatrix}$$

Now, let

$$d\underline{p} = \begin{bmatrix} d\underline{x}_e \\ d\underline{\phi}_e \end{bmatrix}$$

By dividing both sides by dt the end-effector velocity and angular velocity can be written as a linear function of joint velocities using the manipulator Jacobian (J).

$$\underline{\dot{p}} = J\underline{\dot{\theta}}$$

where $\underline{\dot{\theta}} = \begin{bmatrix} \dot{\theta}_1 & \cdots & \dot{\theta}_n \end{bmatrix}^T$ is the $n \times 1$ joint velocity vector.

Let us partition the Jacobian matrix so that:

$$J = \left[\begin{array}{c:c:c:c} \underline{J}_{L1} & \underline{J}_{L2} & \cdots & \underline{J}_{Ln} \\ & & & \\ \underline{J}_{A1} & \underline{J}_{A2} & \cdots & \underline{J}_{An} \end{array}\right] \qquad \textbf{eq. 2.9}$$

Then,

$$\underline{v}_e = \underline{J}_{L1}\dot{\theta}_1 + \cdots + \underline{J}_{Ln}\dot{\theta}_n \tag{eq. 2.10}$$

Similarly, the angular velocity of the end-effector can be expressed in the following manner:

$$\underline{\omega}_e = \underline{J}_{Ai}\dot{\theta}_i + \cdots + \underline{J}_{An}\dot{\theta}_n \tag{eq. 2.11}$$

Since the model that will be used in this analysis includes only *revolute* joints (no prismatic joints), the composite of distal links rotates altogether at an angular velocity

$$\underline{\omega}_e = \underline{b}_0\dot{\theta}_1 + \underline{b}_1\dot{\theta}_2 + \cdots + \underline{b}_{n-1}\dot{\theta}_n \tag{eq. 2.12}$$

where $\underline{b}_{n-1}$ is the unit vector pointing along the direction of the joint axis i-1.

Also, since every angular velocity component contributes one term to the linear velocity vector, the resultant linear velocity is

$$\underline{v}_e = \underline{\omega}_1 \times \underline{r}_{0,e} + \cdots + \underline{\omega}_n \times \underline{r}_{n-1,e} = \left(\underline{b}_0 \times \underline{r}_0\right)\dot{\theta}_1 + \cdots + \left(\underline{b}_{n-1} \times \underline{r}_{n-1}\right)\dot{\theta}_n \tag{eq. 2.13}$$

From the preceding discussion, it is clear that the i^{th} column of the Jacobian is

$$\begin{bmatrix} \underline{J}_{Li} \\ \underline{J}_{Ai} \end{bmatrix} = \begin{bmatrix} \underline{b}_{i-1} \times \underline{r}_{i-1,e} \\ \underline{b}_{i-1} \end{bmatrix} \tag{eq. 2.14}$$

As we will see in the next section this relationship is very important in the energy analysis of any robotic system.

2.2.4. Lagrangian Dynamics of Manipulator Arms

Slotine and Li (1991), have utilized the dynamic developments of the Denavit-Hartenberg notation and applied them in an energy conservation formulation to prove a property critical to the control of any robotic system. For the i^{th} link of the manipulator model of the human arm the kinetic energy is

$$E_{K_i} = \frac{1}{2} m_i \underline{v}_{ci}^T \underline{v}_{ci} + \frac{1}{2} \underline{\omega}_i^T I_i \underline{\omega}_i \tag{eq. 2.15}$$

where m_i is the mass of the link and I_i is the 3×3 inertia tensor at the centroid expressed in the base coordinates. Since energy is additive, the total kinetic energy is

$$E_K = \sum_{i=1}^{N} E_{K_i}$$

where *N* is the total number of links.

Using the results of eq. 2.14 and eq. 2.15, and remembering that in the present study all joints are assumed to be revolute, and that we use the Denavit-Hartenberg notation,

$$E_K = \frac{1}{2}\sum_{i=1}^{N}\left(m_i \underline{\dot{\theta}}^T J_L^{(i)T} J_L^{(i)} \underline{\dot{\theta}} + \underline{\dot{\theta}}^T J_A^{(i)T} I_i J_A^{(i)} \underline{\dot{\theta}}\right) = \frac{1}{2}\underline{\dot{\theta}}^T H \underline{\dot{\theta}} \qquad \textbf{eq. 2.16}$$

where $H \in \Re^{n \times n}$ and

$$H = \sum_{i=1}^{N}\left(m_i J_L^{(i)T} J_L^{(i)} + J_A^{(i)T} I_i J_A^{(i)}\right) \qquad \textbf{eq. 2.17}$$

Notice that the elements of H (H_{ij}) are functions of the angular joint positions and that H is symmetric (because it is the sum of symmetric matrices).

The formulation of Lagrange's equations of motion utilizes generalized forces. The generalized forces account for all forces and moments acting on the arm except for all conservative forces (gravity). In the present study, assume that the only forces acting on the arm linkage besides gravity are the forces from the joint actuators (muscle for the human arm) and denote the generalized force vector $\underline{Q}^T = \underline{\tau}^T = \begin{bmatrix}\tau_1 & \cdots & \tau_N\end{bmatrix}$, where τ_i is the muscle torque exerted on the *i* i^{th} joint.

Finally, form classical mechanics it is known that

$$Work = \Delta E_K + \Delta E_P$$

Δ represents the change, and $E_P = \sum_{i=1}^{N} m_i \underline{g}^T \underline{r}_{0,ci}$ is the potential energy of the system. The vector $\underline{g}$ is the 3×1 acceleration of gravity with respect to the base coordinate frame. Notice that the gravitational potential energy is a function of the joint angular position vector alone, *i.e.* $E_P = E_P(\underline{\theta})$.

The system's Lagrangian is a measure of the power (energy rate) input to the arm manipulator:

$$\frac{dL}{dt} = \frac{dWork}{dt} = \frac{dE_K}{dt} - \frac{dE_P}{dt} \quad \text{or} \quad \frac{d}{dt}\left(\underline{\theta}^T \underline{\tau}\right) = \frac{1}{2}\frac{d}{dt}\left(\underline{\dot{\theta}}^T H(\underline{\theta})\underline{\dot{\theta}}\right) + \frac{dE_P(\underline{\theta})}{dt}$$

Solving Lagrange's equation,

$$\underline{\tau} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\underline{\theta}}}\right) - \frac{\partial L}{\partial \underline{\theta}} \tag*{\textbf{eq. 2.18}}$$

which can be rewritten:

$$\underline{\tau} = H(\underline{\theta})\ddot{\underline{\theta}} + \underbrace{\left(\dot{H}\dot{\underline{\theta}} - \frac{1}{2}\dot{\underline{\theta}}^{T} \frac{\partial H(\underline{\theta})}{\partial \underline{\theta}}\dot{\underline{\theta}}\right)}_{b(\underline{\theta},\dot{\underline{\theta}}) = C(\underline{\theta},\dot{\underline{\theta}})\dot{\underline{\theta}}} + G(\underline{\theta}) \tag*{\textbf{eq. 2.19}}$$

The elements of matrix C are:

$$C_{ij} = \frac{1}{2}\dot{H}_{ij} + \frac{1}{2}\sum_{k=1}^{N}\dot{\theta}_k\left(\frac{\partial H_{ik}}{\partial \theta_j} - \frac{\partial H_{jk}}{\partial \theta_i}\right) \tag*{\textbf{eq. 2.20}}$$

Note that

$$\left(\dot{H} - 2C\right)_{ij} = -\sum_{k=1}^{N}\dot{\theta}_k\left(\frac{\partial H_{ik}}{\partial \theta_j} - \frac{\partial H_{jk}}{\partial \theta_i}\right) = -\left(\dot{H} - 2C\right)_{ji} \tag*{\textbf{eq. 2.21}}$$

This means that $\dot{H} - 2C$ is skew-symmetric, which is a very important fact for the control modeling of the arm linkage. The relationship $\dot{H} = 2C$ is derived using the Lagrangian for the conservation of energy, therefore the equality is a manifestation of the concept of the conservation of energy on the robotic system. While such a conservation of energy is true for all robotic system, the relationship between $\dot{H}$ and $2C$ can only be seen if the coordinate frames are chosen in a certain manner (Denavit-Hartenberg notation).

2.3. A Link Model of the Human Arm

In the previous paragraphs we saw that the rules of the Denavit-Hartenberg notation and the Lagrangian formulation provide closed-form dynamic equations for the arm linkage. The choice of Denavit-Hartenberg coordinate frames for the dynamical modeling of the human arm has been used in the past often because of the symmetrical properties of the resulting inertia tensor matrix (Raikova, 1992). The arm model shown in Figure 2.5 has been used in the literature in the past to simplify the analysis of the dynamics (Martin et al., 1989). The spherical joints of two- or three-degree-of-freedom joints at the shoulder, elbow and wrist have been replaced by two or three one-degree-of-freedom links of zero length.

Despite the existence of the Denavit-Hartenberg rules for choosing the coordinate frames, the process of doing so is complex and time consuming. Further, although many people have defined these coordinate frames for their particular robotic configuration, most literature provides little reference to it since defining the coordinate frames is only the set up of the more complicated task of solving the forward dynamics. While the application of the Denavit-Hartenberg set of rules for arm model is a tedious and complicated process, it was absolutely necessary for proceeding with the controller design.

Figure 2.6 is an illustration of a man using his left arm to perform a three-dimensional movement. The schematics of Figure 2.7 show the resulting coordinate systems on each link that are consistent to the Denavit-Hartenberg notation. Notice that the coordinate systems put the arm in a very unnatural position when the joint angles are 0°. The joint angles were adjusted accordingly to bring the arm to perform a more "natural" movement such as the one shown in Figure 2.7. Also, keep in mind that the first link represents a rotating room. The four parameters, which are necessary in the Denavit-Hartenberg notation, are given in Table 2.1. The values of the angles α and θ in Table 2.1 are calculated in the following manner: $\alpha_i = \angle\left(z_{i-1} \rightarrow z_i\right)\Big|_{x_i}$ and $\theta_i = \angle\left(x_{i-1} \rightarrow x_i\right)\Big|_{z_{i-1}}$ (Corke, 1996). The notation for α_i means: "α_i is

Link	a (m)	α (rad)	d (m)	θ (rad)
1	$0.30 = L_1$	π/2	0.00	0
2	0.00	π/2	0.00	0
3	0.00	−π/2	0.00	0
4	0.00	π/2	$0.25 = L_4$	0
5	0.00	−π/2	$0.00 = L_6$	0
6	0.00	π/2	0.25	0
7	0.00	π/2	0.00	0
8	$0.05 = L_8$	0	0.00	0

Table 2.1 The four Denavit-Hartenberg notation parameters for each link of the human arm linkage. See Figure 2.7 for a graphical representation of these parameters.

the angle around the x_i axis from the axis z_{i-1} to the axis z_i". Let us do one example by applying this rule for link 3 ($i=3$). Look at Figure 2.7 b and c. The angle between the axes z_2 (which is pointing into the page) and z_3 (which is pointing vertically downwards) using the right-hand rule is $-\frac{\pi}{2}$. The negative sign indicates that the right-hand rule points toward the negative x_3 direction. The first link represents the distance between the center of the rotating room and the center of the shoulder joint.

The next step is to derive to closed-form equations of dynamics for the armature that was described above. First, using eq. 2.4 and the values of Table 2.1 the equations of the positions of the centroids C_1, $\cdots$ $,C_7$ can be easily written as functions of the transitional matrices A_i^{i-1}. Using the appropriate values for each link the necessary matrices are calculated.

$$A_1^0=\begin{bmatrix}\cos(\theta_1) & 0 & \sin(\theta_1) & L_1\cos(\theta_1)\\ \sin(\theta_1) & 0 & -\cos(\theta_1) & L_1\sin(\theta_1)\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \qquad A_2^1=\begin{bmatrix}\cos(\theta_2) & 0 & \sin(\theta_2) & 0\\ \sin(\theta_2) & 0 & -\cos(\theta_2) & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$A_3^2=\begin{bmatrix}\cos(\theta_3) & 0 & -\sin(\theta_3) & 0\\ \sin(\theta_3) & 0 & \cos(\theta_3) & 0\\ 0 & -1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \qquad A_4^3=\begin{bmatrix}\cos(\theta_4) & 0 & \sin(\theta_4) & 0\\ \sin(\theta_4) & 0 & -\cos(\theta_4) & 0\\ 0 & 1 & 0 & L_4\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$A_5^4=\begin{bmatrix}\cos(\theta_5) & 0 & -\sin(\theta_5) & 0\\ \sin(\theta_5) & 0 & \cos(\theta_5) & 0\\ 0 & -1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \qquad A_6^5=\begin{bmatrix}\cos(\theta_6) & 0 & \sin(\theta_6) & 0\\ \sin(\theta_6) & 0 & -\cos(\theta_6) & 0\\ 0 & 1 & 0 & L_6\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$A_7^6=\begin{bmatrix}\cos(\theta_7) & 0 & \sin(\theta_7) & 0\\ \sin(\theta_7) & 0 & -\cos(\theta_7) & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \qquad A_8^7=\begin{bmatrix}\cos(\theta_8) & -\sin(\theta_8) & 0 & L_8\cos(\theta_8)\\ \sin(\theta_8) & \cos(\theta_8) & 0 & L_8\sin(\theta_8)\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

Then the total transformation matrices from the i^{th} coordinate system to the fixed coordinate system are given by

$$A_1^0 = \begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & L_1\cos(\theta_1) \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & L_1\sin(\theta_1) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^0 = \begin{bmatrix} \cos(\theta_1)\cos(\theta_2) & \sin(\theta_1) & \cos(\theta_1)\sin(\theta_2) & L_1\cos(\theta_1) \\ \sin(\theta_1)\cos(\theta_2) & -\cos(\theta_1) & \sin(\theta_1)\sin(\theta_2) & L_1\sin(\theta_1) \\ \sin(\theta_2) & 0 & -\cos(\theta_2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_1^0 A_2^1$$

$$A_3^0 = \begin{bmatrix} \begin{array}{l}\cos(\theta_1)\cos(\theta_2)\cos(\theta_3) \\ +\sin(\theta_1)\sin(\theta_3)\end{array} & -\cos(\theta_1)\sin(\theta_2) & \begin{array}{l}-\cos(\theta_1)\cos(\theta_2)\sin(\theta_3) \\ +\sin(\theta_1)\cos(\theta_3)\end{array} & L_1\cos(\theta_1) \\ \begin{array}{l}\sin(\theta_1)\cos(\theta_2)\cos(\theta_3) \\ -\cos(\theta_1)\sin(\theta_3)\end{array} & -\sin(\theta_1)\sin(\theta_2) & \begin{array}{l}-\sin(\theta_1)\cos(\theta_2)\sin(\theta_3) \\ -\cos(\theta_1)\cos(\theta_3)\end{array} & L_1\sin(\theta_1) \\ \sin(\theta_2)\cos(\theta_3) & +\cos(\theta_2) & -\sin(\theta_2)\sin(\theta_3) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_3^2 A_2^0$$

$$A_4^0 = \begin{bmatrix} \begin{array}{l}\cos(\theta_1)\cos(\theta_2)\cos(\theta_3) \\ \cos(\theta_4) \\ +\sin(\theta_1)\sin(\theta_3)\cos(\theta_4) \\ -\cos(\theta_1)\sin(\theta_2)\sin(\theta_4)\end{array} & \begin{array}{l}-\cos(\theta_1)\cos(\theta_2) \\ \sin(\theta_3) \\ +\sin(\theta_1)\cos(\theta_3)\end{array} & \begin{array}{l}\cos(\theta_1)\cos(\theta_2)\cos(\theta_3) \\ \sin(\theta_4) \\ +\sin(\theta_1)\sin(\theta_3)\sin(\theta_4) \\ +\cos(\theta_1)\sin(\theta_2)\cos(\theta_4)\end{array} & \begin{array}{l}-L_4\cos(\theta_1)\cos(\theta_2) \\ \sin(\theta_3) \\ +L_4\sin(\theta_1)\cos(\theta_3) \\ +L_1\cos(\theta_1)\end{array} \\ \begin{array}{l}\sin(\theta_1)\cos(\theta_2)\cos(\theta_3) \\ \cos(\theta_4) \\ -\cos(\theta_1)\sin(\theta_3)\cos(\theta_4) \\ -\sin(\theta_1)\sin(\theta_2)\sin(\theta_4)\end{array} & \begin{array}{l}-\sin(\theta_1)\cos(\theta_2) \\ \sin(\theta_3) \\ -\cos(\theta_1)\cos(\theta_3)\end{array} & \begin{array}{l}\sin(\theta_1)\cos(\theta_2)\cos(\theta_3) \\ \sin(\theta_4) \\ -\cos(\theta_1)\sin(\theta_3)\sin(\theta_4) \\ +\sin(\theta_1)\sin(\theta_2)\cos(\theta_4)\end{array} & \begin{array}{l}-L_4\sin(\theta_1)\cos(\theta_2) \\ \sin(\theta_3) \\ -L_4\cos(\theta_1)\cos(\theta_3) \\ +L_1\sin(\theta_1)\end{array} \\ \begin{array}{l}\sin(\theta_2)\cos(\theta_3)\cos(\theta_4) \\ +\cos(\theta_2)\sin(\theta_4)\end{array} & -\sin(\theta_2)\sin(\theta_3) & \begin{array}{l}\sin(\theta_2)\cos(\theta_3)\sin(\theta_4) \\ -\cos(\theta_2)\cos(\theta_4)\end{array} & -L_4\sin(\theta_2)\sin(\theta_3) \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_4^3 A_3^0$$

Because of the fact that each transitional matrix element is a very lengthy expression, the rest of the matrices are not shown here but they can be easily calculated using the Maple code given in Appendix B. For notational convenience we can break down the transitional matrices in an element-by-element form

$$A_K^0 = \begin{bmatrix} [A_{0K}]_{11} & [A_{0K}]_{12} & [A_{0K}]_{13} & [A_{0K}]_{14} \\ [A_{0K}]_{21} & [A_{0K}]_{22} & [A_{0K}]_{23} & [A_{0K}]_{24} \\ [A_{0K}]_{31} & [A_{0K}]_{32} & [A_{0K}]_{33} & [A_{0K}]_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The position vector of the center of gravity (c.o.g.) of the *i*th link with respect to the co-ordinate system fixed to the ground is $\underline{r}_{CK}^{0} = A_{K}^{0}\underline{r}_{CK}^{K}$. Therefore,

$$\underbrace{\begin{bmatrix} x_{C1}^0 \\ y_{C1}^0 \\ z_{C1}^0 \\ 1 \end{bmatrix}}_{\underline{r}_{C1}^0} = \underbrace{\begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0.3\cos(\theta_1) \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0.3\sin(\theta_1) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{A_1^0} \underbrace{\begin{bmatrix} -L_{C1}^1 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{\underline{r}_{C1}^1} = \begin{bmatrix} (L_1 - L_{C1}^1)\cos(\theta_1) \\ (L_1 - L_{C1}^1)\sin(\theta_1) \\ 0 \\ 1 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} x_{C2}^0 \\ y_{C2}^0 \\ z_{C2}^0 \\ 1 \end{bmatrix}}_{\underline{r}_{C2}^0} = A_2^0 \underbrace{\begin{bmatrix} 0 \\ -L_{C2}^2 \\ 0 \\ 1 \end{bmatrix}}_{\underline{r}_{C2}^2} = \begin{bmatrix} -L_{C2}^2\sin(\theta_1) + L_1\cos(\theta_1) \\ L_{C2}^2\cos(\theta_1) + L_1\sin(\theta_1) \\ 0 \\ 1 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} x_{C3}^0 \\ y_{C3}^0 \\ z_{C3}^0 \\ 1 \end{bmatrix}}_{\underline{r}_{C3}^0} = A_3^0 \underbrace{\begin{bmatrix} 0 \\ 0 \\ L_{C3}^3 \\ 1 \end{bmatrix}}_{\underline{r}_{C3}^3} = \begin{bmatrix} -L_{C3}^3\cos(\theta_1)\cos(\theta_2)\sin(\theta_3) + L_{C3}^3\sin(\theta_1)\cos(\theta_3) + L_1\cos(\theta_1) \\ -L_{C3}^3\sin(\theta_1)\cos(\theta_2)\sin(\theta_3) - L_{C3}^3\cos(\theta_1)\cos(\theta_3) + L_1\sin(\theta_1) \\ -L_{C3}^3\sin(\theta_2)\sin(\theta_3) \\ 1 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} x_{C4}^0 \\ y_{C4}^0 \\ z_{C4}^0 \\ 1 \end{bmatrix}}_{\underline{r}_{C4}^0} = A_4^0 \underbrace{\begin{bmatrix} 0 \\ -l_{C4}^4 \\ 0 \\ 1 \end{bmatrix}}_{\underline{r}_{C4}^4} = \begin{bmatrix} [A_{04}]_{14} - l_{C4}^4[A_{04}]_{12} \\ [A_{04}]_{24} - l_{C4}^4[A_{04}]_{22} \\ [A_{04}]_{34} - l_{C4}^4[A_{04}]_{32} \\ 1 \end{bmatrix} \qquad \underbrace{\begin{bmatrix} x_{C5}^0 \\ y_{C5}^0 \\ z_{C5}^0 \\ 1 \end{bmatrix}}_{\underline{r}_{C5}^0} = A_5^0 \underbrace{\begin{bmatrix} 0 \\ 0 \\ l_{C5}^5 \\ 1 \end{bmatrix}}_{\underline{r}_{C5}^5} = \begin{bmatrix} [A_{05}]_{14} + l_{C5}^5[A_{05}]_{13} \\ [A_{05}]_{24} + l_{C5}^5[A_{05}]_{23} \\ [A_{05}]_{34} + l_{C5}^5[A_{05}]_{33} \\ 1 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} x_{C6}^0 \\ y_{C6}^0 \\ z_{C6}^0 \\ 1 \end{bmatrix}}_{\underline{r}_{C6}^0} = A_6^0 \underbrace{\begin{bmatrix} 0 \\ -l_{C6}^6 \\ 0 \\ 1 \end{bmatrix}}_{\underline{r}_{C6}^6} = \begin{bmatrix} [A_{06}]_{14} - l_{C6}^6[A_{06}]_{12} \\ [A_{06}]_{24} - l_{C6}^6[A_{06}]_{22} \\ [A_{06}]_{34} - l_{C6}^6[A_{06}]_{32} \\ 1 \end{bmatrix} \qquad \underbrace{\begin{bmatrix} x_{C7}^0 \\ y_{C7}^0 \\ z_{C7}^0 \\ 1 \end{bmatrix}}_{\underline{r}_{C7}^0} = A_7^0 \underbrace{\begin{bmatrix} -l_{C7}^7 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{\underline{r}_{C7}^7} = \begin{bmatrix} [A_{07}]_{14} - l_{C7}^7[A_{07}]_{11} \\ [A_{07}]_{24} - l_{C7}^7[A_{07}]_{21} \\ [A_{07}]_{34} - l_{C7}^7[A_{07}]_{31} \\ 1 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} x_{C8}^{0} \\ y_{C8}^{0} \\ z_{C8}^{0} \\ 1 \end{bmatrix}}_{\underline{r}_{C8}^{0}} = A_8^0 \underbrace{\begin{bmatrix} -l_{C8}^{8} \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{\underline{r}_{C8}^{8}} = \begin{bmatrix} [A_{08}]_{14} - l_{C8}^{8}[A_{08}]_{11} \\ [A_{08}]_{24} - l_{C8}^{8}[A_{08}]_{21} \\ [A_{08}]_{34} - l_{C8}^{8}[A_{08}]_{31} \\ 1 \end{bmatrix}$$

The lengths of several c.o.g are 0 (or $L_{C1} = L_1$ for the case of the first link). Thus,

$$\underline{r}_{C1}^{0} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \underline{r}_{C2}^{0} = \begin{bmatrix} L_1 \cos(\theta_1) \\ L_1 \sin(\theta_1) \\ 0 \\ 1 \end{bmatrix}, \quad \underline{r}_{C3}^{0} = \begin{bmatrix} L_1 \cos(\theta_1) \\ L_1 \sin(\theta_1) \\ 0 \\ 1 \end{bmatrix}, \quad \underline{r}_{C4}^{0} = \begin{bmatrix} [A_{04}]_{14} - l_{C4}^{4}[A_{04}]_{12} \\ [A_{04}]_{24} - l_{C4}^{4}[A_{04}]_{22} \\ [A_{04}]_{34} - l_{C4}^{4}[A_{04}]_{32} \\ 1 \end{bmatrix}$$

$$\underline{r}_{C5}^{0} = \begin{bmatrix} [A_{05}]_{14} \\ [A_{05}]_{24} \\ [A_{05}]_{34} \\ 1 \end{bmatrix}, \quad \underline{r}_{C6}^{0} = \begin{bmatrix} [A_{06}]_{14} - l_{C6}^{6}[A_{06}]_{12} \\ [A_{06}]_{24} - l_{C6}^{6}[A_{06}]_{22} \\ [A_{06}]_{34} - l_{C6}^{6}[A_{06}]_{32} \\ 1 \end{bmatrix}, \quad \underline{r}_{C7}^{0} = \begin{bmatrix} [A_{07}]_{14} \\ [A_{07}]_{24} \\ [A_{07}]_{34} \\ 1 \end{bmatrix}, \quad \underline{r}_{C8}^{0} = \begin{bmatrix} [A_{08}]_{14} \\ [A_{08}]_{24} \\ [A_{08}]_{34} \\ 1 \end{bmatrix}$$

Notice, that $\underline{r}_{C1}^{0}$ remains stationary, since the center of gravity of the rotating room (link 1) is located exacly on the rotational axis, and it concides with the origin of the fixed coordinate system. To get the velocity of the c.o.g. of each link with respect to the fixed coordinate frame simply differentiate the position vectors.

$$\underline{v}_{C1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \underline{v}_{C2} = \begin{bmatrix} -L_1 \sin(\theta_1)\dot{\theta}_1 \\ L_1 \cos(\theta_1)\dot{\theta}_1 \\ 0 \end{bmatrix}, \quad \underline{v}_{C3} = \begin{bmatrix} -L_1 \sin(\theta_1)\dot{\theta}_1 \\ L_1 \cos(\theta_1)\dot{\theta}_1 \\ 0 \end{bmatrix}$$

For the rest of the velocity vectors the compact notation is used since the expressions are very lengthy.

$$\underline{v}_{C4} = \begin{bmatrix} \frac{\partial\left\{[A_{04}]_{14} - l_{C4}^{4}[A_{04}]_{12}\right\}}{\partial\theta_1} & \frac{\partial\left\{[A_{04}]_{14} - l_{C4}^{4}[A_{04}]_{12}\right\}}{\partial\theta_2} & \frac{\partial\left\{[A_{04}]_{14} - l_{C4}^{4}[A_{04}]_{12}\right\}}{\partial\theta_3} & \frac{\partial\left\{[A_{04}]_{14} - l_{C4}^{4}[A_{04}]_{12}\right\}}{\partial\theta_4} \\ \frac{\partial\left\{[A_{04}]_{24} - l_{C4}^{4}[A_{04}]_{22}\right\}}{\partial\theta_1} & \frac{\partial\left\{[A_{04}]_{24} - l_{C4}^{4}[A_{04}]_{22}\right\}}{\partial\theta_2} & \frac{\partial\left\{[A_{04}]_{24} - l_{C4}^{4}[A_{04}]_{22}\right\}}{\partial\theta_3} & \frac{\partial\left\{[A_{04}]_{24} - l_{C4}^{4}[A_{04}]_{22}\right\}}{\partial\theta_4} \\ \frac{\partial\left\{[A_{04}]_{34} - l_{C4}^{4}[A_{04}]_{32}\right\}}{\partial\theta_1} & \frac{\partial\left\{[A_{04}]_{34} - l_{C4}^{4}[A_{04}]_{32}\right\}}{\partial\theta_2} & \frac{\partial\left\{[A_{04}]_{34} - l_{C4}^{4}[A_{04}]_{32}\right\}}{\partial\theta_3} & \frac{\partial\left\{[A_{04}]_{24} - l_{C4}^{4}[A_{04}]_{22}\right\}}{\partial\theta_4} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix}$$

$$\underline{v}_{C5} = \begin{bmatrix} \frac{\partial\{[A_{05}]_{14}\}}{\partial\theta_1} & \frac{\partial\{[A_{05}]_{14}\}}{\partial\theta_2} & \frac{\partial\{[A_{05}]_{14}\}}{\partial\theta_3} & \frac{\partial\{[A_{05}]_{14}\}}{\partial\theta_4} & \frac{\partial\{[A_{05}]_{14}\}}{\partial\theta_5} \\ \frac{\partial\{[A_{05}]_{24}\}}{\partial\theta_1} & \frac{\partial\{[A_{05}]_{24}\}}{\partial\theta_2} & \frac{\partial\{[A_{05}]_{24}\}}{\partial\theta_3} & \frac{\partial\{[A_{05}]_{24}\}}{\partial\theta_4} & \frac{\partial\{[A_{05}]_{24}\}}{\partial\theta_5} \\ \frac{\partial\{[A_{05}]_{34}\}}{\partial\theta_1} & \frac{\partial\{[A_{05}]_{34}\}}{\partial\theta_2} & \frac{\partial\{[A_{05}]_{34}\}}{\partial\theta_3} & \frac{\partial\{[A_{05}]_{34}\}}{\partial\theta_4} & \frac{\partial\{[A_{05}]_{34}\}}{\partial\theta_5} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_5 \end{bmatrix}$$

$$\underline{v}_{C6} = \begin{bmatrix} \frac{\partial\{[A_{06}]_{14} - L^6_{C6}[A_{06}]_{12}\}}{\partial\theta_1} & \cdots & \frac{\partial\{[A_{06}]_{14} - L^6_{C6}[A_{06}]_{12}\}}{\partial\theta_6} \\ \frac{\partial\{[A_{06}]_{24} - L^6_{C6}[A_{06}]_{22}\}}{\partial\theta_1} & \cdots & \frac{\partial\{[A_{06}]_{24} - L^6_{C6}[A_{06}]_{22}\}}{\partial\theta_6} \\ \frac{\partial\{[A_{06}]_{34} - L^6_{C6}[A_{06}]_{32}\}}{\partial\theta_1} & \cdots & \frac{\partial\{[A_{06}]_{34} - L^6_{C6}[A_{06}]_{32}\}}{\partial\theta_6} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_6 \end{bmatrix}$$

$$\underline{v}_{C7} = \begin{bmatrix} \frac{\partial\{[A_{07}]_{14}\}}{\partial\theta_1} & \cdots & \frac{\partial\{[A_{07}]_{14}\}}{\partial\theta_7} \\ \frac{\partial\{[A_{07}]_{24}\}}{\partial\theta_1} & \cdots & \frac{\partial\{[A_{07}]_{24}\}}{\partial\theta_7} \\ \frac{\partial\{[A_{07}]_{34}\}}{\partial\theta_1} & \cdots & \frac{\partial\{[A_{07}]_{34}\}}{\partial\theta_7} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_7 \end{bmatrix} \qquad \underline{v}_{C8} = \begin{bmatrix} \frac{\partial\{[A_{08}]_{14}\}}{\partial\theta_1} & \cdots & \frac{\partial\{[A_{08}]_{14}\}}{\partial\theta_8} \\ \frac{\partial\{[A_{08}]_{24}\}}{\partial\theta_1} & \cdots & \frac{\partial\{[A_{08}]_{24}\}}{\partial\theta_8} \\ \frac{\partial\{[A_{08}]_{34}\}}{\partial\theta_1} & \cdots & \frac{\partial\{[A_{08}]_{34}\}}{\partial\theta_8} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_8 \end{bmatrix}$$

From eq. 2.10 and eq. 2.11,

$$J_L^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$J_L^{(2)} = \begin{bmatrix} -L_1\sin(\theta_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ L_1\cos(\theta_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$J_L^{(3)} = \begin{bmatrix} -L_1\sin(\theta_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ L_1\cos(\theta_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$J_L^{(4)} = \begin{bmatrix} \frac{\partial\left\{\left[A_{04}\right]_{14} - l_{C4}^{4}\left[A_{04}\right]_{12}\right\}}{\partial\theta_1} & \cdots & \frac{\partial\left\{\left[A_{04}\right]_{14} - l_{C4}^{4}\left[A_{04}\right]_{12}\right\}}{\partial\theta_4} & 0 & 0 & 0 & 0 \\ \frac{\partial\left\{\left[A_{04}\right]_{24} - l_{C4}^{4}\left[A_{04}\right]_{22}\right\}}{\partial\theta_1} & \cdots & \frac{\partial\left\{\left[A_{04}\right]_{24} - l_{C4}^{4}\left[A_{04}\right]_{22}\right\}}{\partial\theta_4} & 0 & 0 & 0 & 0 \\ \frac{\partial\left\{\left[A_{04}\right]_{34} - l_{C4}^{4}\left[A_{04}\right]_{32}\right\}}{\partial\theta_1} & \cdots & \frac{\partial\left\{\left[A_{04}\right]_{24} - l_{C4}^{4}\left[A_{04}\right]_{22}\right\}}{\partial\theta_4} & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\vdots$$

$$J_L^{(8)} = \begin{bmatrix} \frac{\partial\left\{\left[A_{08}\right]_{14}\right\}}{\partial\theta_1} & \cdots & \frac{\partial\left\{\left[A_{08}\right]_{14}\right\}}{\partial\theta_8} \\ \frac{\partial\left\{\left[A_{08}\right]_{24}\right\}}{\partial\theta_1} & \cdots & \frac{\partial\left\{\left[A_{08}\right]_{24}\right\}}{\partial\theta_8} \\ \frac{\partial\left\{\left[A_{08}\right]_{34}\right\}}{\partial\theta_1} & \cdots & \frac{\partial\left\{\left[A_{08}\right]_{34}\right\}}{\partial\theta_8} \end{bmatrix}$$

Similarly, according to eq. 2.14, the angular velocity part of the Jacobian matrix can be calculated. This is quite simple after the derivation of the transformations matrices, because of the particular construction of the coordinate system that obey the Denavit-Hartenberg rules. There are two important things to remember during this calculation: *a*) b_{i-1} are the coordiantes of a unit vector along the (i-1) joint in the 0th coordinate system, and *b*) The ith coordinate system is fixed to the ith link. The calculation based on the quantities defined above is given by the following matrix multiplication:

$$J_A^i = \sum_{j=1}^{i} \left\{ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} A_{j-1}^{0} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & \cdots & 0 & \underbrace{1}_{1\le j\le 8} & 0 & \cdots & 0 \end{bmatrix} \right\} \qquad \textbf{eq. 2.22}$$

where, $A_0^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ (directly from eq. 2.4.)

The reason that the transitional matrices can be used directly in finding the unit vector along the i-1 joint is intuitively obvious from the defintition of the transitional matrix A_{i-1}^{0} and the fact that by construction the unit vector along the i-1 joint has cordinates $\begin{bmatrix}0 & 0 & 1\end{bmatrix}$ in the coordinate system atteched to that same joint. By simple inspection of the dynamics model of Figure 2.7,

$$J_A^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$J_A^{(2)} = \begin{bmatrix} 0 & \sin(\theta_1) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\cos(\theta_1) & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and so on. The calculation then, of the H matrix (the manipulator inertia tensor) immediately follows from eq. 2.17. The inertia matix is a 8×8 matrix, whose explicit form is several pages long and therefore omitted here. However, to partially test the approach, the H matrix of the system that describes the dynamics of just a two link armature moving in the horizontal plane is given next. The form of the matrix ($H_{2\times2}$) resulted by just setting $\theta_1 = \theta_2 = \theta_4 = \theta_6 = \theta_7 = \theta_8 = 0$. The resulting matrix has the same elements to similar matrices derived in numerous other publications specifically for the planar case (Slotine and Li, 1991). Of course, the matrix shown here is eight-dimensional because of the extra degrees of freedom (which are being constrained). The constrains that were imposed are shown in Table 2.2.

$$H_{2\times2} = \begin{bmatrix} H_{11} & 0 & H_{13} & 0 & H_{15} & 0 & 0 & 0 \\ 0 & H_{22} & 0 & H_{24} & 0 & H_{26} & 0 & 0 \\ H_{31} & 0 & H_{33} & 0 & H_{35} & 0 & 0 & 0 \\ 0 & H_{42} & 0 & H_{44} & 0 & H_{46} & 0 & 0 \\ H_{51} & 0 & H_{53} & 0 & H_{55} & 0 & 0 & 0 \\ 0 & H_{62} & 0 & H_{64} & 0 & H_{66} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad \textbf{eq. 2.23}$$

where,

$$H_{33} = m_4(L_4 - L_{C4})^2 + I_{4z} + m_6\left[L_4^2 + L_{C6}^2 - 2L_4L_{C6}\cos(\theta_5)\right] + I_{6z}$$
$$H_{35} = H_{53} = m_6\left[L_{C6}^2 - L_4L_{C6}\cos(\theta_5)\right] + I_{6z}$$
$$H_{55} = m_6L_{C6}^2 + I_{6z}$$

Then, the contribution of the inertia tensor matrix to the torques of joints 3 and 5 are:

$$\begin{bmatrix} \tau_3 \\ \tau_5 \end{bmatrix} = \begin{bmatrix} H_{33} & H_{35} \\ H_{53} & H_{55} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_3 \\ \ddot{\theta}_5 \end{bmatrix}$$

Further, for the same two-link, two-dimensional system, eq. 2.20 gives that:

$$C_{2\times 2} = \begin{bmatrix} & & & & & & & \\ & & & & & & & \\ & & C_{33} = \left[m_6 L_4 L_{C6} \cos(\theta_5)\right]\dot{\theta}_5 & & C_{35} = \left[m_6 L_4 L_{C6} \cos(\theta_5)\right](\dot{\theta}_3 + \dot{\theta}_5) & & & \\ & & & & & & & \\ & & C_{53} = -\left[m_6 L_4 L_{C6} \cos(\theta_5)\right]\dot{\theta}_3 & & C_{55} = 0 & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{bmatrix}$$

Link Number	Mass (m_i)	Length (L_i)	c.o.g. Location (L_{Ci})	Inertia
1	0	0	0	$I_X=I_Y=I_Z=0$
2	0	0	0	$I_X=I_Y=I_Z=0$
3	0	0	0	$I_X=I_Y=I_Z=0$
4	m_4	L_4	L_{C4}	I_X, I_Y, I_Z
5	0	0	0	$I_X=I_Y=I_Z=0$
6	m_4	L_4	L_{C4}	I_X, I_Y, I_Z
7	0	0	0	$I_X=I_Y=I_Z=0$
8	0	0	0	$I_X=I_Y=I_Z=0$

Table 2.2 The constrained parameters that reduce the inertia matrix of the eight-dimensional system to the corresponding matrix of just a planar two-link armature.

2.4. Summary of the human arm dynamic modeling

This chapter on the dynamics of the human arm derives the equations of dynamics for the arm linkage using a particular rule for the choice of coordinate systems (Denavit-Hartenberg notation). Although, Raikova, 1992, has produced a similar model, complete with muscles, details of the analysis and the resulting dynamical matrices were missing, which are important for control. There are Matlab (Mathworks Inc., Natick, MA) toolboxes that calculate the forward dynamics of a robotic system provided the user provides the parameters of Table 2.1. Those parameters of evaluated using the Denavit-Hartenberg rules. Therefore, the results of dynamical analysis of the arm were not used in solving the forward dynamics. More importantly, they are necessary for the control of the arm as we will see in Chapter 3.

The choice of the coordinate frames for each link, as well as the equations of the dynamics were shown detailed. The dynamical model for the human arm has seven degrees of freedom at the arm joints. An eighth degree of freedom is included in the dynamical modeling to represent the dynamics induced on the arm when humans (or robotic arms) execute desired trajectory tracking tasks in rotating environments, such as artificial gravity.

Through the use of the Denavit-Hartenberg rules, the equations of dynamics for the human arm linkage can be written in a compact form: $H(\underline{\theta})\underline{\ddot{\theta}} + C(\underline{\theta},\underline{\dot{\theta}})\underline{\dot{\theta}} + \underline{g}(\underline{\theta}) = \underline{\tau}$. $H(\underline{\theta})$ is the symmetric, positive definite inertia tensor, $C(\underline{\theta},\underline{\dot{\theta}})\underline{\dot{\theta}}$ is the vector of centripetal and Coriolis forces, and $\dot{H} - 2C$ is skew-symmetric at all times and joint angle positions and velocities. Despite the existence of the Denavit-Hartenberg rules, the creation of the coordinate frames of Figure 2.7 and the choice of the parameters of Table 2.1 were two of the most challenging contributions to this thesis. The reason for such difficulty is the lack of any references in the literature, since such a development is considered to be only a first step of the more complex problem of solving the forward arm dynamics.

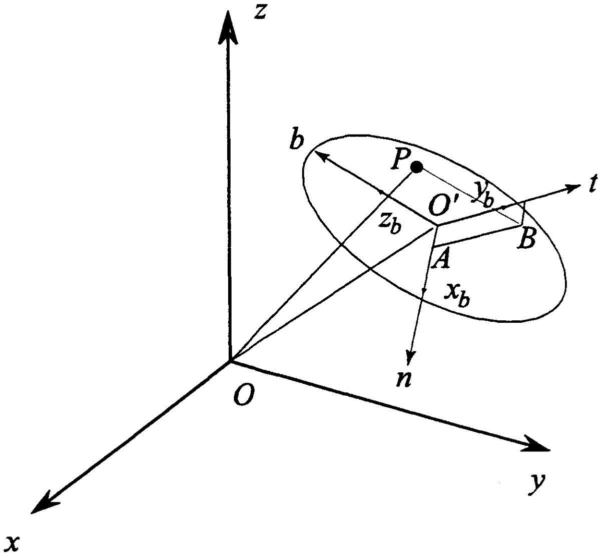


Figure 2.1 Coordinate system transformations (Asada and Slotine, 1986). The coordinates of point P are shown with reference to the fixed frame $O - xyz$. The position of point P can also be represented with reference to the coordinate frame fixed to the rigid body, $O' - x_b y_b z_b$.

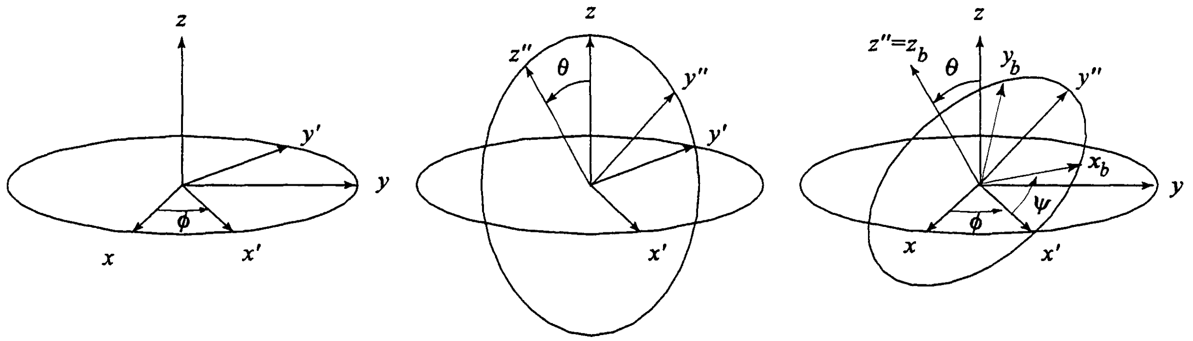


Figure 2.2 The three consecutive rotations used to define the Euler angles (Asada and Slotine, 1986). First, the coordinate frame is rotated about the z axis by an angle ϕ. Secondly, the new coordinate frame $O - x'y'z$ is rotated about the x' axis by an angle θ. Finally, the newest frame $O - x'y''z''$ is then rotated about the z'' axis by an angle ψ. The three angles ϕ, θ, and ψ are referred to as *Euler angles* and they uniquely determine the orientation of the coordinate frame.

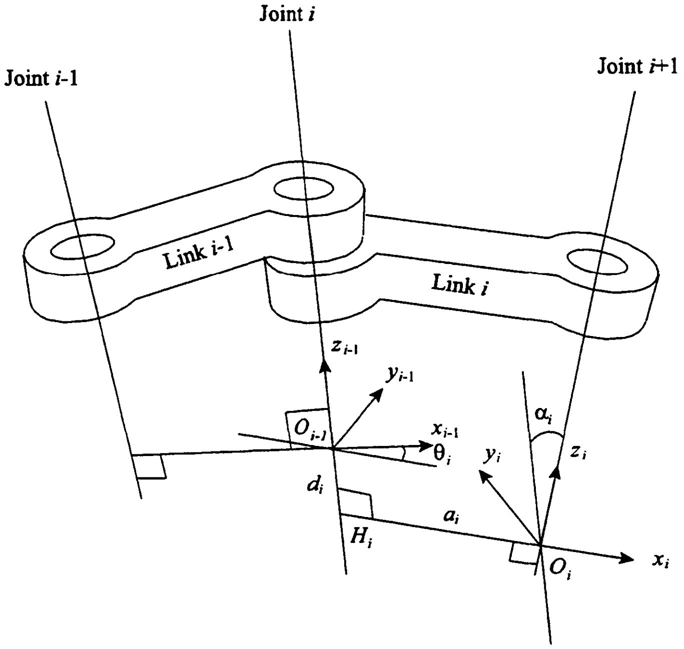


Figure 2.3 The Denavit-Hartenberg notation (Asada and Slotine, 1986). The line H_iO_i is the common normal to joint axes i and i+1. The relationship between the two links is described by the relative position and orientation of the two coordinate frames attached to the two links.

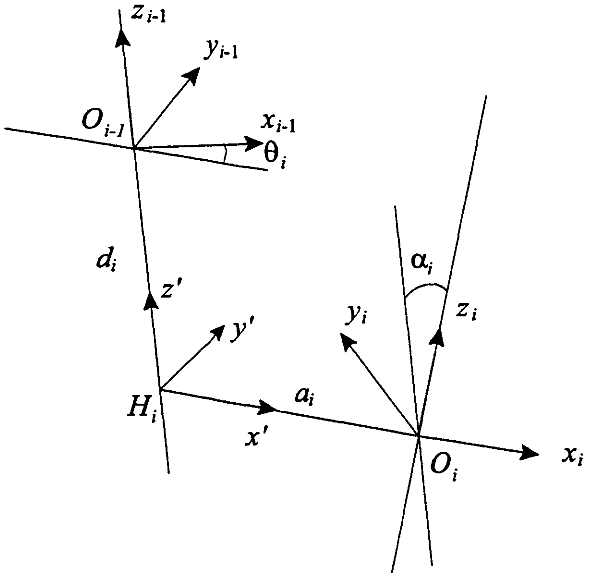


Figure 2.4 The relationship between adjacent coordinate frames in the Denavit-Hartenberg notation (Asada and Slotine, 1986). The origin of the i^{th} coordinate frame O_i is located at the intersection of joint axis i+1 and the common normal between joint axes *i and* i+1. Note that the frame of link *i* is at joint i+1 rather than at joint *i*. The x_i axis is directed along the extension line of the common normal, while the z_i axis is along the joint axis i+1. Finally, the y_i is chosen such that the resultant frame $O_i - x_i y_i z_i$ forms a right-hand coordinate system.

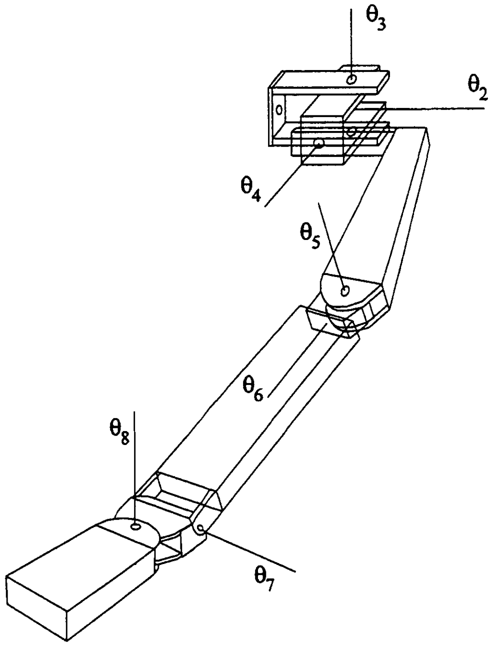


Figure 2.5 A dynamical model is developed for studying the motion and control of a human arm. The figure shows the breakdown of the human arm into simple one-dimensional joint links. (Martin et al., 1989)

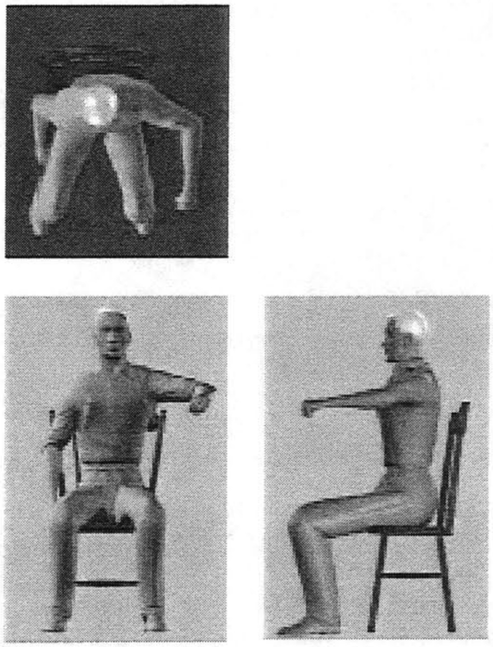


Figure 2.6 A computer generated representation of a man while applying motor control to his left arm. (Top-Front-Side view) This is the posture for which the Denavit-Hartenberg coordinate frames are defined.

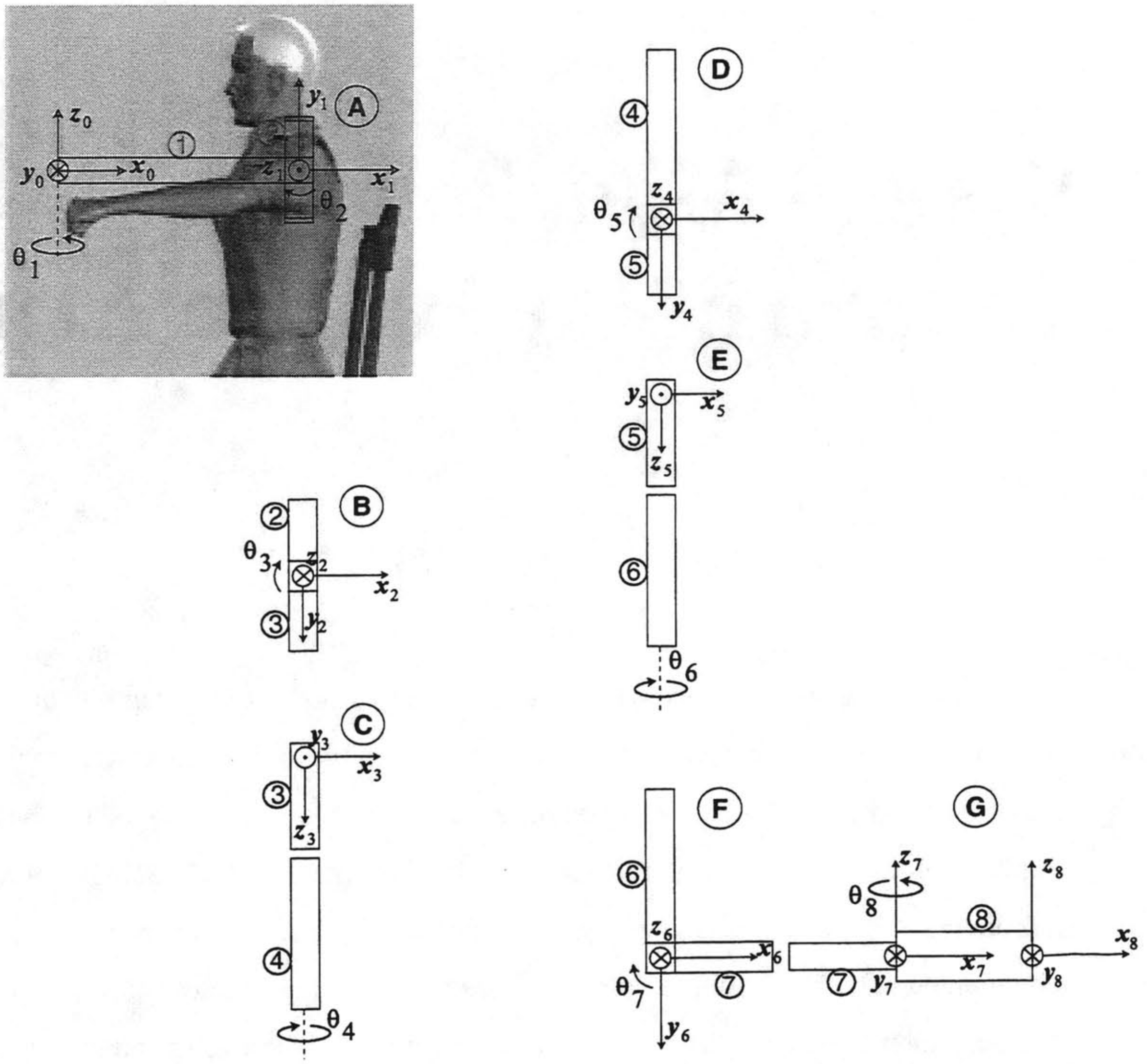


Figure 2.7 The coordinate systems on each link that are consistent to the Denavit-Hartenberg notation. Notice that the coordinate systems put the arm in a very unnatural position when the joint angles are 0°. The joint angles were adjusted accordingly to bring the arm to perform a more "natural" movement. The first link represents a rotating room. The values of the angles α and θ are calculated in the following manner: $\alpha = \angle\left(z_{i-1} \to z_i\right)\Big|_{x_i}$ and $\theta = \angle\left(x_{i-1} \to x_i\right)\Big|_{z_{i-1}}$.

3. System Modeling and Control of the Human Arm

Feedback control is well understood for large classes of nonlinear systems with single inputs or uncoupled states. However, general multi-input-multi-output (MIMO) nonlinear systems feedback control and especially perfomance and robustness issues are still topics for further research. Furthermore, since it is clear that the dynamical and control modeling of robotic manipulators and human limbs are very similar in principle, this chapter reflects the association by looking at both control problems in parallel and drawing conclussions.

The organization of this chapter includes an introduction and general literature review on the areas of biological systems, intelligent controllers and neural adaptive control. An indepth review of fundamental work (Slotine and Li, 1991) in the area of sliding mode control and nonlinear adaptive control is presented. This work forms the basis of the proposed control model, which is shown last. At first, for the sake of simplicity, the proposed model is presented for the case of a planar two-link robotic arm. Elements of the proof follow for that case without any rigor. Finally, the rigorous proof of the robust stability of the proposed control model on any robotic arm (and therefore the seven-link arm representing the human arm) is presented.

3.1. Introduction and Literature Review

The robotics industry has reached a plateau with the successful introduction of robots into automotive manufacturing. There are many other areas where robotics usage is at its infancy. Chief areas for future (and current) uses of robotic manipulators are in hazardous environments, medical applications using multifingered hands, and microbotics. In space exploration, robotic manipulators find use in a variety of applications. Currently, there are plans for building a flight telerobotic servicer that has two cooperating robot arms for repairing satellites and other coordinated activities on board a self-propelled platform. Completely autonomous robots can aid tremendously the exploration of ocean floors and other unstructured environments for which it is unsafe to rely on human remote control.

3.1.1. Biological Systems and Intelligent Controllers

Intelligent control, inspired by biological and artificial intelligence principles, has increased the understanding of controlling complex processes without precise mathematical model of the controlled process. The human Central Nervous System (CNS) demonstrates a great ability to control human arm movement in multiple environments by recognizing which environment is currently in existence and adjusting to it appropriately. In other words, the human CNS performs in an “intelligent” manner. An important prerequisite for an intelligent controller is the ability to adapt rapidly to any unknown but constant (or slowly changing) operating environment. The adaptive capability of an intelligent robotic controller enables it to move the arm to arbitrary targets without any knowledge of the arm’s kinematics, while automatically compensating for perturbations such as target movements, wheel slippage, or changes in the arm’s dynamics. An intelligent controller also adapts to long-term perturbations, enabling the arm to compensate for statistically significant changes in the plant.

Through customized applications, such as autonomous vehicles, intelligent control has demonstrated that it is a step in the right direction. However, intelligent control has yet to provide a complete solution to the problem of integrated manufacturing systems via intelligent reconfiguration of the robotic systems. Today, the state-of-the-art of intelligent robotic control is used in simple remote robotic manipulation. Movie studios use *animatronics* to give life to inanimate robots and make them behave like living creatures. The python snake in the movie "Anaconda" was a one-hundred-degree manipulator requiring 20 PentiumPro PCs to maneuver smoothly. Yet, despite all the computer power available, today's robotic manipulators require very complex feedback and immense computing power to complete simple manipulation in unknown or unstructured environments.

Since the CNS outperforms artificial intelligent controllers, it would be appealing to try to understand those mechanisms and implement them in the design of more efficient controllers. The proposed project looks into the adaptive learning capacity of the CNS during human arm manipulation when the environment becomes unpredictable.

3.1.2. Previous Approaches to Neural Adaptive Control

Recently, several intelligent control strategies for robotic manipulation have been proposed. Some of these strategies have their roots in the area of computer science and artificial intelligence, and were initially developed in solving games and puzzles, such as checkers. Many heuristic search and dynamic programming algorithms are off-line procedures (i.e., cannot be implemented in real time) for designing optimal control policies, and they also require an accurate model of the system (Barto et al., 1995; Sutton, 1990). Further, although approaches such as Learning-Real-Time A* (Korf, 1990) (a kind of receding horizon control algorithm) can be implemented on-line, they are not adaptive in that they assume the existence of an accurate model. Finally, adaptive real-time dynamic programming (Q-learning), a form of reinforcement learning, requires an immense amount of computational power, especially as the number of states increases ("curse of dimensionality"), and it becomes unusable in the case of systems with unobservable states or noisy measurements (Barto et al., 1995). However, noisy state measurements are inevitable in electromechanical systems such as robotic manipulators, especially when they operate in unstructured and unknown environments.

Recent proposed schemes to use multiple models along with an adaptive control scheme require guessing or explicit knowledge of the environment (Ciliz and Narendra, 1996; Narendra and Balakrishnan, 1997; Narendra and Mukhopadhyay, 1997), which is not a realistic proposition. In addition, proposed schemes for the use of neural networks and back-propagation are extremely slow to adapt and very expensive computationally. Also, they require off-line training, and perform poorly outside the region for which they have been trained (Jordan et al., 1994; Santiago and Werbos, 1994; Werbos and Xiaozhong, 1996). Wolpert et al., 1995), used a Kalman filter on experimental data to demonstrate the existence of an internal model, but their results assume a very simple linear model. Some models (Miller et al., 1987) even tried the use neural networks to "learn" a look-up table (Cerebellar Model Arithmetic Computer, CMAC) of the inverse dynamics. However, CMAC control approaches are slow and require very large amounts of memory for good control performance; it is very difficult to evaluate system performance or to prove system stability analytically (Miller et al., 1987). Besides, attempting to control a dynamic system by inverting the plant dynamics and attempting to cancel those dynamics is not a sound control approach especially for non-minimum phase systems. The inverse model of a dynamic system may posses certain high-pass filtering characteristics that are prone to noise amplification. Finally, the Adaptive Resonance Theory (ART), as used in the AVITE (Adaptive Vector Integration to the Endpoint) model (Gaudiano et al., 1997), functions as a mechanism for feedforward control, and therefore, cannot provide on-line compensation for error in performance and quick adjustments to changing conditions.

3.2. Control Methods for Systems with Parametric Uncertainties

The basis of the theoretical work for the Lyapunov stability of the proposed framework was introduced by Slotine (Slotine and Li, 1991). The initial stability arguments were developed for nonlinear control of robotic systems whose dynamics are completely known. Clearly, such an approach might not be feasible in extreme environments were the nature of the nonlinearities of the environment are unknown yet a certain level of trajectory tracking performance is required. It is known that the CNS achieves high levels of accuracy when it controls a redundant system such as the human arm during complex human movements. The CNS clearly does not have explicit knowledge of the environmental nonlinearities, and its only sensory feedback is through vision and proprioception.

Despite the caveats of Slotine's control method for the control of biological systems, his theoretical development forms the basis of the analysis presented towards the end of this chapter. The proposed model is an attempt to present an alternative control method to the model reference adaptive control for robotic system. The proposed alternative appropriate to describe human motor control since it requires no knowledge of the nonlinear dynamics of the arm or the environment.

3.2.1. Sliding Mode Control

Modeling inaccuracies can have strong adverse effects on the control of nonlinear systems. A simple approach to robust control is the so-called sliding control methodology (Slotine and Li, 1991). Intuitively, it is based on the notion that it is much easier to control 1st order systems. Accordingly a notational simplification is introduced, which allows n^{th} order problems to be replaced by equivalent 1st order representations. Although in theory sliding control can achieve high performance, this comes with very high control activity that may excite unmodeled dynamics in the system. This can lead to a modified version of sliding control that can compromize between very high performance and low control activity.

Since the problem of the control of a human arm, even for planar movements, has multiple inputs (joint torques) and multiple output (joint angles), consider a nonlinear multi-input system of the form

$$\theta_i^{(n_i)} = f_i(\underline{\theta}) + \sum_{j=1}^{m} b_{ij}(\underline{\theta})\tau_j \qquad i = 1,\ldots,m \qquad j = 1,\ldots,m \qquad \textbf{eq. 3.1}$$

where the vector $\underline{\tau}$ of components τ_j is the control input vector, and the state $\underline{\theta}$ is composed of the angles θ_i and their first (n_i-1) derivatives. Such systems are called square systems since they have as many control inputs τ_j as outputs θ_i. For trajectory tracking the state $\underline{\theta}$ has to track a desired time-varying trajectory $\underline{\theta}_d$ in the presence of parametric uncertainties.

We define, B as the input matrix of components b_{ij}, which is invertible over the whole state-space. Further, assume that $\hat{B}$ be an invertible matrix that is continuously dependent on parametric uncertainty such that $\hat{B} = B$ in the absence of parametric uncertainty. The functions

$f_i(\underline{\theta})$ (generally nonlinear) are not exactly known, but the extent of the imprecisions on $f_i(\underline{\theta})$ are assumed to be upper bounded by a continuous function of $\underline{\theta}$. Similarly, only the signs of the control gains $b_{ij}(\underline{\theta})$ are known, but they are also upper bounded by a continuous function of $\underline{\theta}$.

At this point we just set up the system dynamics of controlling the multiple link amrature using sliding mode control. The following paragraphs will analyze the system's stability and performance trying to develop a control strategy that is both robust and can be implemented by the human CNS during motor control.

3.2.2. Sliding mode state space

To analyze the arm system using sliding mode control, it is necessary to look at each state (angle) individually before putting it all together as the state vector. This section is adopted directly from Slotine and Li, 1991, and has been modified appropriately to accommodate for MIMO systems. Let $\tilde{\theta}_i = \theta_i - \theta_{d_i}$ be the tracking error in the state variable θ_i, and let

$$\underline{\tilde{\theta}}_i = \underline{\theta}_i - \underline{\theta}_{d_i} = \begin{bmatrix} \tilde{\theta}_i & \dot{\tilde{\theta}}_i & \cdots & \tilde{\theta}_i^{(n-1)} \end{bmatrix}$$

be the tracking error vector. Furthermore, define a time varying surface $S_i(t)$ in the state-space $\Re^{(n)}$ by the scalar equation $s_i(\underline{\theta},t) = 0$, where

$$s_i(\underline{\theta}_i,t) = \left(\frac{d}{dt} + \lambda_i\right)^{n-1} \tilde{\theta}_i \qquad \textbf{eq. 3.2}$$

and λ_i are strictly positive constants. To evaluate $\left(\frac{d}{dt} + \lambda_i\right)^{n-1}$ use the binomial expansion equation.

Given the initial condition, $\underline{\theta}_{d_i}(0) = \underline{\theta}_i(0)$, the problem of tracking $\underline{\theta}_i \equiv \underline{\theta}_{d_i}$ is equivalent to that remaining on the surface $S_i(t)$ for all $t > 0$. Indeed, $s_i \equiv 0$ represents a linear differential equation whose unique solution is $\underline{\tilde{\theta}}_i \equiv 0$.

Furthermore, bounds on s_i (i.e., $\forall t \geq 0, |s_i(t)| \leq \Phi$) can be directly translated into bounds on the tracking error vector $\underline{\tilde{\theta}}_i$, and therefore the scalar s_i represent a true measure of tracking performance. By definition, the tracking error $\widetilde{\theta}_i$ is obtained from s_i through a sequence of first-order lowpass filters. Let y_1 be the output of the first filter. We have

$$y_1 = \int_0^t e^{-\lambda_i(t-T)} s_i(T) dT$$

From $|s_i(t)| \leq \Phi$ we get

$$|y_1| \leq \Phi \int_0^t e^{-\lambda_1(t-T)} dT = \frac{\Phi}{\lambda_1}\left(1 - e^{-\lambda_1 t}\right) \leq \Phi / \lambda_1$$

The same reasoning can be applied to the second filter and so on. Then,

$$\left|\widetilde{\theta}_i(t)\right| \leq \Phi / \lambda_i^{n-1} = \varepsilon$$

Similarly, $\widetilde{\theta}_i^{(k)}$ can be thought of as being obtained through a sequence of (*n*-*j*-1) such filters. Therefore,

$$\left|\widetilde{\theta}_i^{(j)}\right| \leq (2\lambda_i)^j \varepsilon$$

Assuming zero initial conditions for the error, we have

$$\forall t \geq 0, |s_i(t)| \leq \Phi \qquad \Rightarrow \qquad \forall t \geq 0, \left|\widetilde{\theta}_i^{(j)}(t)\right| \leq (2\lambda_i)^j \varepsilon, \; j = 1, \ldots, n-1 \qquad \textbf{eq. 3.3}$$

where $\varepsilon = \Phi / \lambda_i^{n-1}$.

The simplified first-order problem of keeping s_i at zero can now be achieved by choosing the control law τ_i such that outside of $S_i(t)$ the squared distance to the surface $S_i(t)$ decreases along all system trajectories. Thus it constrains trajectories to point towards the surface $S_i(t)$.

$$\frac{1}{2}\frac{d}{dt} s_i^2 \leq -\eta_i |s_i| \qquad \textbf{eq. 3.4}$$

where η_i is a positive constant. In addition, once the system trajectories reach the surface they remain on the surface, which means that the "sliding condition" makes the surface $S_i(t)$ an invariant set.

Finally, satisfying eq. 3.4 guarantees that if the zero initial condition is not satisfied, the surface $S_i(t)$ will still be reached in a finite time smaller than $|s_i(t=0)|/\eta_i$. Further, eq. 3.4 implies that once on the surface, the tracking error goes assymptotically to zero with a time constant $(n-1)/\lambda_i$ (Slotine and Li, 1991).

The idea behind sliding mode control is to formulate well-behaved functions of the tracking errors s_i, and then select the feedback control law τ_i, such that s_i^2 remains a Lyapunov-like function of the closed loop system in the presence of unmodeled dynamics and environmental disturbances.

Now consider the i^{th} differential equation of the MIMO system of eq. 3.1.

$$\theta_i^{(n)} = f_i(\underline{\theta}) + \sum_{j=1}^{m} b_{ij}(\underline{\theta})\tau_j$$

The dynamics f_i (possibly nonlinear or time-varying) is not exactly known, but estimated as $\hat{f}_i$. The estimation error of f_i is assumed to be bounded by some known function $F_i = F_i(\underline{\theta}, \underline{\dot{\theta}})$.

$$|\hat{f}_i - f_i| \le F_i$$

In order to achieve tracking, define the sliding surface $s_i = 0$

$$s_i = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{\theta}_i = \tilde{\theta}_i^{(n-1)} + \cdots + \lambda^{n-1}\tilde{\theta}_i$$

Then,

$$\dot{s}_i = \tilde{\theta}_i^{(n)} + \cdots + \lambda^{n-1}\dot{\tilde{\theta}}_i = \theta_i^{(n)} - \theta_{di}^{(n)} + \cdots + \lambda^{n-1}\dot{\tilde{\theta}}_i$$

$$= f_i(\underline{\theta}) + \sum_{j=1}^{m} b_{ij}(\underline{\theta})\tau_j - \theta_{di}^{(n)} + \cdots + \lambda^{n-1}\dot{\tilde{\theta}}_i$$

Notice that the best approximation $\hat{\tau}_i$ of a continuous control law that would achieve $\dot{s}_i = 0$ (or $s_i = 0$) is

$$\hat{\tau}_i = -\hat{f}_i(\underline{\theta}) + \theta_{di}^{(n)} - \cdots - \lambda^{n-1}\dot{\tilde{\theta}}.$$

If we use the approximate control instead of the ideal sliding mode control, then

$$\dot{s}_i = \hat{f}_i - f_i$$

However, the sign of the resulting $\dot{s}_i$ is not known. If $\hat{\tau}_i = -\hat{f}_i(\underline{\theta}) + \theta_{d\,i}^{(n)} - \cdots - \lambda^{n-1}\dot{\tilde{\theta}} - k\,\mathrm{sgn}(s_i)$, then the control law satisfies eq. 3.4.

$$\frac{1}{2}\frac{d}{dt}s_i^2 = \dot{s}_i s_i = \left(\hat{f}_i - f_i - k\,\mathrm{sgn}(s_i)\right)s_i = \left(\hat{f}_i - f_i\right)s_i - k|s_i| \le -\eta|s_i|$$

if $k = F_i + \eta$.

Although sliding control provides great flexibity for robust stability and tracking in the presence of unmodeled dynamics and parametric uncertainties, it has one major drawback for use in the motor cotrol of human movements: *The sliding mode control law is discontinuous (because of the sign function in the control law), and highly unlikely to be implemented in a biological system.*

3.2.3. Continuous alternatives to sliding mode control

The resulting control law to satisfy the sliding condition results in a discontinuity across the hyperplane $s_i = 0$ (the sign function in the control law). However, s_i is not known to infinite precision, neither is the switching of the control instantaneously. These conditions can lead to chattering that can lead to excitation of high frequency unmodeled dynamics. One way to avoid such undesirable behavior and at the same time make the control law continuous is to define a thin boundary layer around the sliding surface $S_i(t)$ (Slotine and Li, 1991)

$$B_i(t) = \left\{\ \underline{\theta}_i, |s_i(\underline{\theta}_i, t)| \le \Phi\ \right\} \qquad \Phi > 0$$

Outside $B_i(t)$, we choose the control law $\hat{\tau}_i$ as before, which guarantees that the boundary layer is attractive and therefore invariant (i.e., all trajectories starting inside $B_i(t)$ remain inside $B_i(t)$ for all $t \ge 0$. We then interpolate $\hat{\tau}_i$ inside $B_i(t)$ replacing the discontinuous term $\mathrm{sgn}(s_i)$ by s_i/Φ. This leads to tracking to within a guaranteed precision $\varepsilon_i = \Phi/\lambda_i^{n-1}$ (instead of "perfect" tracking), and more generally it guarantees that for all trajectories starting inside $B_i(t)$ remain inside the boundary layer.

$$\forall t \ge 0, \left|\tilde{\theta}_i^{(j)}(t)\right| \le (2\lambda_i)^j\,\varepsilon, \qquad j = 1, \ldots, n-1$$

3.2.4. Summary of Sliding Mode control for MIMO systems

In order to extend the results of the previous paragraphs to the entire MIMO system described in section 3.2.1, we need to write the uncertainies on $\underline{f}$ in additive form and the uncertainties on the input matrix B in mutiplicative form

$$\left|\hat{f}_i - f_i\right| \leq F_i \qquad i = 1,\ldots,m$$

$$B = (I+\Delta)\hat{B} \qquad \left|\Delta_{ij}\right| \leq D_{ij} \qquad i = 1,\ldots,m \qquad j = 1,\ldots,m$$

where I is the $n \times n$ identity matrix.

The components of the vector $\underline{s}$ can be rewritten for notational simplicity.

$$s_i = \left(\frac{d}{dt}+\lambda\right)^{n-1}\tilde{\theta}_i = \theta_i^{(n-1)} - \theta_{ri}^{(n-1)}$$

This defines the vector $\underline{\theta}_r^{(n-1)}$ of components $\theta_{ri}^{(n-1)}$, which can be computed directly from $\underline{\theta}$ and $\underline{\theta}_d$

Extending the results from the earlier paragraphs the appropriate control law takes on the form

$$\underline{\tau} = \hat{B}^{-1}\left(\underline{\theta}_r^{(n-1)} - \hat{\underline{f}} - \underline{k}\,\mathrm{sgn}(\underline{s})\right)$$

In the same manner as in section 3.2.2, we can write

$$\dot{s}_i = \hat{f}_i - f_i + \sum_{j=1}^{n}\Delta_{ij}\left(\theta_{ri}^{(n-1)} - \hat{f}_j\right) - \sum_{j\neq i}\Delta_{ij}k_j\,\mathrm{sgn}(s_j) - (1+\Delta_{ii})k_i\,\mathrm{sgn}(s_i)$$

The sliding condition is verified if

$$(1-D_{ii})k_i \geq F_i + \sum_{j=1}^{n}D_{ij}\left|\theta_{ri}^{(n-1)} - \hat{f}_j\right| - \sum_{j\neq i}D_{ij}k_j + \eta_i \qquad i = 1,\ldots,m$$

and in particular if the vector $\underline{k}$ is chosen such that

$$(1-D_{ii})k_i + \sum_{j\neq i}D_{ij}k_j \geq F_i + \sum_{j=1}^{n}D_{ij}\left|\theta_{ri}^{(n-1)} - \hat{f}_j\right| + \eta_i \qquad i = 1,\ldots,m \qquad \textbf{eq. 3.5}$$

The above expression represents a set of m equations and m unknown switching gains k_i. The Frobenius-Perron theorem assures that there exists a unique solution $\underline{k}$ to this set of equations

(Slotine, 1991).

3.2.5. Model Reference Adaptive Control

While sliding control presents a viable option for control of nonlinear systems with unmodeled dynamics and parametric uncertainties, it does not provide a means of "learning" the model parameters as time passes. A system controlled by a sliding controller performs the same way during the first or the millionth cycle of a repetitive task. It makes more sense to find a way to estimate the parameters of the system while trying to achieve tracking.

Model Reference Adaptive Control (MRAC) is the method most engineering applications use. In general a MRAC system contains four parts: a plant containing the unknown parameters, a reference model for compactly specifying the desired output of the control system, a feedback control law containing adjustable parameters, and an adaptation mechanism for updating the adjustable parameters (see Figure 3.1).

The plant is assumed to have a known structure, although the parameters are unknown. For linear plants this means that the number of poles and zeros is known, but their location is not. For nonlinear plants this means that the form of the dynamics is known but some of the parameters are not. In the case of the human motor control we will denote the plant to be the arm dynamics.

A reference model is used to specify the ideal response of the adaptive control system to the external command. Intuitively, it provides the ideal plant response, which the adaptation mechanism seeks when adjusting the parameters. In engineering problems the choice of the reference model has to satisfy two requirements. It should reflect the performance specifications of the control task, such as rise time, settling time, overshoot, etc. And this ideal behavior should be achievable for the adaptive control system. In the case of the human motor control we will denote the reference model to be the external altered dynamics and the desired trajectory.

The controller is usually parameterized by a number of adjustable parameters (implying that one may obtain a family of controllers by assigning various values to the adjustable parameters). The controller should have perfect tracking capacity in order to allow the possibility of tracking convergence. That is, when the plant parameters are exactly known, the corresponding controller parameters should make the plant output identical to that of the reference model. In the case of the human motor control we will denote the controller to be the subject's central nervous system.

The adaptation mechanism is used to adjust the parameters in the control law. In MRAC systems, the adaptation law searches for parameters such that the response of the plant under adaptive control becomes the same as that of the reference model.

3.2.6. Single-Input-Single-Output MRAC for Linear Systems with Full-State Feedback

Slotine's summary of MRAC theory is presented here to help understand further the use of the sliding variables in robotic control. First, consider the case when the full state is measurable. The n^{th} order linear system in companion form can be written,

$$a_n\theta^{(n)} + a_{n-1}\theta^{(n-1)} + \cdots + a_0\theta = \tau ,$$

where the states $\theta,\ \dot{\theta},\ \dots\ ,\theta^{(n-1)}$ are all measurable. We assume that the coefficient vector $\underline{a} = [a_n\ \dots\ a_1\ \ a_0]^T$ is unknown, but that the sign of a_n is assumed to be known. The objective of the control system is to make y closely track the response of a stable reference model,

$$a_n\theta_m^{(n)} + a_{n-1}\theta_m^{(n-1)} + \cdots + a_0\theta_m = r(t)$$

with $r(t)$ being a bounded reference signal. To that end, let,

$$\theta_m^{(n)} - \beta_{n-1}e^{(n-1)} - \cdots - \beta_0 e = z(t)$$

with $\beta_1\ \dots\ \beta_n$ be positive constants such that $s^n + \beta_{n-1}s^{n-1} + \cdots + \beta_0$ is Hurwitz (i.e. all it roots have negative real parts). This can be rewritten as

$$u = \hat{a}_n z + \hat{a}_{n-1}\theta^{(n-1)} + \cdots \hat{a}_0\theta = \underline{v}^T(t)\underline{\hat{a}}(t)$$

where $\underline{v}(t) = [z(t)\ \ \theta^{(n-1)}\ \ \dots\ \ \dot{\theta}\ \ \theta]^T$ and $\underline{\hat{a}}(t) = [\hat{a}_n\ \ \hat{a}_{n-1}\ \dots\ \hat{a}_1\ \ \hat{a}_0]^T$. The vector $\underline{v}(t)$ represents quantities that are completely known, while $\underline{\hat{a}}(t)$ denotes the estimated parameter vector. Then, the tracking error satisfies the closed-loop dynamics

$$a_n[e^{(n)}\ \ \beta_{n-1}e^{(n-1)}\ \dots\ \beta_0 e] = \underline{v}(t)\underline{\tilde{a}}(t) = \underline{v}(t)[\underline{\hat{a}}(t) - \underline{a}(t)]$$

To choose the parameter adaptation law, let us rewrite the closed-loop error dynamics in a state space form,

$$\dot{\underline{x}} = A\underline{x} + \underline{b}\left[(1/a_n)\underline{v}^T \underline{\tilde{a}}\right]$$

$$e = \underline{cx}$$

where,

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -\beta_0 & -\beta_1 & -\beta_2 & \cdots & -\beta_{n-1} \end{bmatrix}, \ \underline{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \text{and } \underline{c} = [1 \ 0 \cdots 0 \ 0].$$

The resulting control law ($u = \underline{v}^T \underline{\hat{a}}$) is complete by defining the adaptation law

$$\dot{\underline{\hat{a}}} = -\Gamma \underline{v} \underline{b}^T P \underline{x},$$

where both Γ and P are symmetric positive definite constant matrices.

While single-input-single-output MRAC provides system parameter estimation and tracking at the same time, it is clearly very limiting to apply directly to the control of human arm movements for several reasons. First, at least in principle, the unknown parameters need to be constant and not time-varying. Second, the theory as shown above is true only for linear systems. Finally, the complex dynamics of a human arm cannot be decomposed into individual components corresponding to separate joints because the dynamics for all degrees of freedom are coupled with each other.

3.2.7. Adaptive control and Nonlinear Systems (The SISO case)

There exists very little theory for the adaptive control of nonlinear systems. However, adaptive control theory has been successfully extended for a very special class of nonlinear systems. This special class of nonlinear systems can be linearly parameterized, their full state is measurable, and nonlinearities can be cancelled stably (i.e., without unstable hidden modes or dynamics) by the control input if the parameters are known.

Slotine's analysis continues by considering an n^{th} order nonlinear systems in the companion form

$$\theta^{(n)} + \sum_{i=1}^{n} a_i f_i(\underline{x}, t) = b\tau \qquad \textbf{eq. 3.6}$$

where $\underline{x} = \begin{bmatrix}\theta & \dot{\theta} & \dots & \theta^{(n-1)}\end{bmatrix}^T$ is the state vector, f_i are known nonlinear functions of the state and time, and the parameters a_i and b are unknown constants. Assume that the state is measured and the sign of b is known. For notational convenience rewrite eq. 3.6 as

$$h\theta^{(n)} + \sum_{i=1}^{n} c_i f_i(\underline{x}, t) = \tau$$

where $h = 1/b$ and $c_i = a_i/b$.

Similarly to the development of the sliding mode control define a combined error

$$s = e^{(n-1)} + \lambda_{n-2}e^{(n-2)} + \dots + \lambda_0 e = \Delta\left(\frac{d}{dt}\right)e$$

where e is the output tracking error and $\Delta\left(\frac{d}{dt}\right) = \frac{d^{(n-1)}}{dt^{(n-1)}} + \lambda_{n-2}\frac{d^{(n-2)}}{dt^{(n-2)}} + \dots + \lambda_0$. Note that s can be rewritten as

$$s = \theta^{(n-1)} - \theta_r^{(n-1)}$$

where

$$\theta_r^{(n-1)} = \theta_d^{(n-1)} - \lambda_{n-2}e^{(n-2)} - \dots - \lambda_0 e$$

Choose the following control law:

$$\tau = \hat{h}\theta_r^{(n)} + \sum_{i=1}^{n} \hat{c}_i f_i(\underline{x}, t) - ks$$

where k has the same sign as h, and

$$\dot{\hat{h}} = -\gamma\,\mathrm{sgn}(h)s\theta_r^{(n)}$$

$$\dot{\hat{c}}_i = -\gamma\,\mathrm{sgn}(h)sf_i$$

The adaptive control law can be checked for stability using a Lyapunov function candidate. For this control law the Lyapunov function

$$V = |h|s^2 + \frac{1}{\gamma}\left(\tilde{h}^2 + \sum_{i=1}^{n} \tilde{c}_i^2\right) \geq 0$$

Then,

$$\dot V = 2|h|s\dot s + \frac{1}{\gamma}\left(2\tilde h\dot{\tilde h} + 2\sum_{i=1}^{n}\tilde c_i\dot{\tilde c}_i\right)$$

$$= 2|h|s\dot s + \frac{1}{\gamma}\left(2\tilde h\dot{\hat h} + 2\sum_{i=1}^{n}\tilde c_i\dot{\hat c}_i\right) = 2s\left(|h|\left(\theta^{(n)} - \theta_r^{(n)}\right) - \operatorname{sgn}(h)\theta_r^{(n)}\tilde h - \sum_{i=1}^{n}\tilde c_i \operatorname{sgn}(h) f_i\right)$$

$$= 2s\operatorname{sgn}(h)\left(h\theta^{(n)} - \sum_{i=1}^{n}\tilde c_i f_i - h\theta_r^{(n)} - \tilde h\,\theta_r^{(n)}\right)$$

$$= -2s\operatorname{sgn}(h)\left(-h\theta^{(n)} - \sum_{i=1}^{n}c_i f_i + \sum_{i=1}^{n}\hat c_i f_i + \hat h\theta_r^{(n)}\right)$$

$$= -2s\operatorname{sgn}(h)\left(-\tau + \left(\sum_{i=1}^{n}\hat c_i f_i + \hat h\theta_r^{(n)} - ks\right) + ks\right) = -2s^2\operatorname{sgn}(h)k$$

$$= -2|k|s^2 \le 0$$

Therefore, the global tracking convergence can be easily shown.

Notice, that this approach provides parameter estimation, tracking and at the same time has the robustness characteristics of the sliding mode control to unknown but bounded disturbances.

3.2.8. Multi-Input-Multi-Output (MIMO) MRAC

The need of a control method that handles MIMO systems is clear. Both the human and the robot arm have more than one joint moving at the same time. The controller has to apply control torques to each and every one of them at the same time. Further, the joint angles, velocities and accelerations are coupled together, and therefore, they could not be controlled individually using decoupled dynamics.

To extend the results from the MRAC-nonlinear-SISO systems, consider the system of a robotic arm in a MIMO configuration (Figure 3.2). For the general MIMO system robust stability cannot be shown. However, for systems that can be linearly parameterized robust stability can be extended from the SISO nonlinear case. In Chapter 2 we derived the dynamics for the general arm model.

$$H(\underline{\theta})\underline{\ddot{\theta}} + C(\underline{\theta},\underline{\dot{\theta}})\underline{\dot{\theta}} + \underline{g}(\underline{\theta}) = \underline{\tau}$$

Remember, $H(\underline{\theta})$ is an $n \times n$ (7×7 for the case of the human arm model shown in Figure 3.2) inertia matrix, which is symmetric and positive definite, $C(\underline{\theta},\underline{\dot{\theta}})\underline{\dot{\theta}}$ is an *n*-vector of centripetal and Coriolis torques, and $\underline{g}(\underline{\theta})$ is an *n*-vector of gravitational torques. Define,

$$\underline{s} = \underline{\dot{\tilde{\theta}}} + \Lambda\underline{\tilde{\theta}}$$

or better yet,

$$\underline{s} = \underline{\dot{\theta}} - \underline{\dot{\theta}}_r$$

where

$$\underline{\dot{\theta}}_r = \underline{\dot{\theta}}_d + \Lambda\underline{\tilde{\theta}}$$

Let the Lyapunov function candidate be

$$V(t) = \frac{1}{2}\underline{s}^T H \underline{s} \geq 0 \qquad \textbf{eq. 3.7}$$

Then,

$$\dot{V}(t) = \frac{1}{2}\underline{s}^T \dot{H}\underline{s} + \underline{s}^T\left(H\underline{\ddot{\theta}} - H\underline{\ddot{\theta}}_r\right)$$

$$= \frac{1}{2}\underline{s}^T \dot{H}\underline{s} + \underline{s}^T\left(\underline{\tau} - C\underline{\dot{\theta}} - \underline{g} - H\underline{\ddot{\theta}}_r\right)$$

$$= \frac{1}{2}\underline{s}^T \dot{H}\underline{s} + \underline{s}^T\left(\underline{\tau} - C(\underline{s} + \underline{\dot{\theta}}_r) - \underline{g} - H\underline{\ddot{\theta}}_r\right)$$

Previously, we proved that $\dot{H} = 2C$ (skew-symmetry property; page 39). Therefore,

$$\dot{V}(t) = \underline{s}^T\left(\underline{\tau} - H\underline{\ddot{\theta}}_r - C\underline{\dot{\theta}}_r - \underline{g}\right)$$

Similarly to the development in earlier sections

$$\underline{\tau} = \underline{\hat{\tau}} - \underline{k\,\mathrm{sgn}(\underline{s})}$$

where the vector $\underline{k\,\mathrm{sgn}(\underline{s})}$ has components $k_i\,\mathrm{sgn}(s_i)$. Also,

$$\underline{\hat{\tau}} = \hat{H}\underline{\ddot{\theta}}_r + \hat{C}\underline{\dot{\theta}}_r + \underline{\hat{g}}$$

Then,

$$\dot{V}(t) = \underline{s}^T\left(\tilde{H}(\underline{\theta})\ddot{\underline{\theta}}_r + \tilde{C}(\underline{\theta},\dot{\underline{\theta}})\dot{\underline{\theta}}_r - \tilde{\underline{g}}(\underline{\theta})\right) - \sum_1^n k_i|s_i|$$

Given bounds on the modeling errors we see that choosing the components k_i of the vector $\underline{k\,\mathrm{sgn}(s)}$ such that

$$k_i \geq \left| \left[\tilde{H}(\underline{\theta})\ddot{\underline{\theta}}_r + \tilde{C}(\underline{\theta},\dot{\underline{\theta}})\dot{\underline{\theta}}_r - \tilde{\underline{g}}(\underline{\theta})\right]_i \right| + \eta_i$$

allows the system to satisfy the sliding condition

$$\dot{V}(t) \leq -\sum_1^n \eta_i|s_i| \leq 0.$$

Assuming a bounded desired trajectory, let us define $\tilde{\underline{a}} = \hat{\underline{a}} - \underline{a}$ as the parameter estimation error. In this expression $\underline{a}$ is a constant vector of unknown parameters describing the manipulator's mass properties, and $\hat{\underline{a}}$ its estimate. Then, instead of using the Lyapunov function of eq. 3.7, consider

$$V(t) = \frac{1}{2}\left[\underline{s}^T H \underline{s} + \tilde{\underline{a}}^T \Gamma^{-1} \tilde{\underline{a}}\right] \geq 0$$

Γ is a symmetric positive definite matrix. Using similar arguments as previously we get

$$\dot{V}(t) = \underline{s}^T\left(\underline{\tau} - H\ddot{\underline{\theta}}_r - C\dot{\underline{\theta}}_r - \underline{g}\right) + \dot{\hat{\underline{a}}}^T \Gamma^{-1} \tilde{\underline{a}}$$

At this point the choice of the Denavit-Hartenberg notation (introduced in Chapter 2) allows the manipulator dynamics to have a very special property, namely, that given a proper definition of the unknown parameter $\underline{a}$ vector describing the manipulator's mass properties, the terms $H(\underline{\theta})$, $C(\underline{\theta},\dot{\underline{\theta}})$, and $\underline{g}(\underline{\theta})$ depend linearly on $\underline{a}$. Thus,

$$H(\underline{\theta})\ddot{\underline{\theta}}_r + C(\underline{\theta},\dot{\underline{\theta}})\dot{\underline{\theta}}_r - \underline{g}(\underline{\theta}) = Y(\underline{\theta},\dot{\underline{\theta}},\dot{\underline{\theta}}_r,\ddot{\underline{\theta}}_r)\underline{a}$$

In a similar process, the control law is chosen as

$$\underline{\tau} = Y\hat{\underline{a}} - K_D\underline{s}$$

and the adaptation law as

$$\dot{\hat{\underline{a}}} = -\Gamma Y^T \underline{s}$$

Then,

$$\dot{V}(t) = -\underline{s}^T K_D \underline{s} \leq 0$$

Using Barbalat's lemma we see that stability and tracking are guaranteed by the above controller.

3.3. The Proposed Control Model

The proposed approach to control robot manipulators is to design a dynamic control law directly in the task space which allows on-line path modification based on sensory feedback data. The suggested controller, unlike the sliding mode controller, does not require explicit knowledge of the environment or its nonlinearities. Rather, it uses simple position and velocity feedback, and it achieves satisfactory tracking performance (at steady state) as fast as the sliding mode controller (measured by the time to reach steady-state performance). The general proof for the n-link model will be given first and then a case study will be presented for the 2-link model.

Before we proceed with the proof of the robust stability of the proposed controller for the n-link robotic model, it is necessary to introduce some definitions and theorems on robustness and stability of any system.

3.3.1. Definitions and Background theory on Robustness

DEFINITION (ZAMES, 1966): A system with input $\underline{x}$ and output $(\underline{Hx})$ is positive if

$$\forall t > 0,\ \underline{x}^T \cdot (\underline{Hx}) \geq 0$$

DEFINITION (Wyatt et al., 1980; Wyatt et al., 1981): A system with input $\underline{x}$ and output $(\underline{Hx})$ is passive if

$$\int_0^\infty \underline{x}^T \cdot (\underline{Hx}) dt \geq 0$$

In the definitions above both the input $\underline{x}$ and output $(\underline{Hx})$ are L_2 (i.e., bounded energy). The definition of passivity is somewhat less restrictive.

THEOREM 1 (ZAMES, 1966): If the open loop dynamics of a system can be factored into two passive relations one of which is strongly positive and has finite gain, then the closed loop is bounded.

A passive element is one that over infinite time absorbs energy, but, in general, from one instant to the next it may absorb or generate energy. Similarly, a positive element absorbs (dissipates) energy at each instant of time even when time tends to infinity. Finally, when a system is bounded it means that it generates finite amounts at all times, provided of course that the input is bounded at all times. We shall now state the general problem, propose a solution, and provide rigorous proof of the validity of the proposed solution.

3.3.2. Robust Stability of the Proposed Controller for the n-link Robotic Arm

Consider the n-link robotic arm. Appropriate choice of coordinate systems (Denavit-Hartenberg rule) guarantees that this systems dynamics can be written in the form $H(\underline{\theta})\underline{\ddot{\theta}} + C(\underline{\theta},\underline{\dot{\theta}})\underline{\dot{\theta}} = \underline{\tau}$, where $\underline{\theta}$ is the state vector, $H(\underline{\theta})$ is positive definite for all values of $\underline{\theta}$, and $\dot{H} - 2C = 0$ (from the conservation of energy). The effect of gravity is ignored for convenience, but since gravity is a conservative force the results will be true even when gravity is present. A control law is sought that provides tracking for the robotic arm. The controller has to provide robust stability while keeping the error bounded. Furthermore, the controller is constrained to only use positional and velocity feedback and lacks any explicit knowledge of the nonlinear nature of the system dynamics.

We propose a variation (extension) to the MRAC control law for robotic systems. The proposed compensator that has a control law $\underline{\tau} = \overline{Y}\underline{\hat{a}} - K_D\underline{\dot{\tilde{\theta}}}$ and an adaptation rule $\dot{\tilde{a}} = \dot{\hat{a}} = -\Gamma\overline{Y}^T\underline{\dot{\tilde{\theta}}}$ can provide robust stability to this system. K_D and Γ are positive definite matrices. The form of $\overline{Y}$ is such that it satisfies the condition $\overline{H}\underline{\ddot{\theta}}_r = \overline{Y}\underline{a}$, where $\overline{H}$ is a positive definite constant matrix, and $\underline{a}$ has no dependence on the inputs or the states, and it only depends on the mass properties of the robotic arm. $\underline{s} = \underline{\dot{\theta}} - \underline{\dot{\theta}}_r$ where $\underline{\dot{\theta}}_r = \underline{\dot{\theta}}_d + \Lambda\underline{\tilde{\theta}}$, and $\underline{\tilde{\theta}} = \underline{\theta} - \underline{\theta}_d$. $\underline{\theta}_d$ is the vector of the desired values of the system states. $\underline{\dot{\theta}}_r$ is a functional with no physical interpretation, and it is defined solely to give the sliding variable the appearance of an error quantity ($\underline{s} = \underline{\dot{\theta}} - \underline{\dot{\theta}}_r$). The control law that results by replacing $\underline{\dot{\tilde{\theta}}}$ with $\underline{s}$ is equivalent since $\dot{\tilde{\underline{\theta}}}$ and $\underline{s}$ are directly related. The theorem and its proof are stated for the case of $\dot{\tilde{\underline{\theta}}}$ because the quantity $\dot{\tilde{\underline{\theta}}}$ has a physical meaning unlike the sliding variable $\underline{s}$.

THEOREM

Consider the BIBO stable n-link robotic model whose dynamics can be written in the form $H(\underline{\theta})\ddot{\underline{\theta}} + C(\underline{\theta}, \dot{\underline{\theta}})\dot{\underline{\theta}} = \underline{\tau}$ *(using the Denavit-Hartenberg rules), where* $\underline{\theta}$ *is the joint position vector,* $H(\underline{\theta})$ *is positive definite for all values of* $\underline{\theta}$*, and* $\dot{H} - 2C = 0$. *Then a compensator with a control law* $\underline{\tau} = \overline{Y}\underline{\hat{a}} - K_D\dot{\underline{\theta}}$ *and an adaptation rule* $\dot{\tilde{a}} = \dot{\hat{a}} = -\Gamma\overline{Y}^T\dot{\underline{\theta}}$ *can provide robust stability to this system.* K_D *and* Γ *are any positive definite matrices. The form of* $\overline{Y}$ *is such that it satisfies the condition* $\overline{H}\ddot{\underline{\theta}}_r = \overline{Y}\underline{a}$, *where* $\overline{H}$ *is a positive definite constant matrix, and* $\underline{a}$ *has no dependence on the inputs or the states, and it only depends on the mass properties of the robotic arm.* $\underline{s} = \dot{\underline{\theta}} - \dot{\underline{\theta}}_r$ *where* $\dot{\underline{\theta}}_r = \dot{\underline{\theta}}_d + \Lambda\tilde{\underline{\theta}}$, *and* $\tilde{\underline{\theta}} = \underline{\theta} - \underline{\theta}_d$. $\underline{\theta}_d$ *is the vector of the desired values of the system states.*

PROOF

Consider again the manipulator dynamics.

$\underline{\tau} = H(\underline{\theta})\ddot{\underline{\theta}} + C(\underline{\theta}, \dot{\underline{\theta}})\dot{\underline{\theta}}$ (for the sake of convenience I will not carry the dependence of H and C to the states any further)

Then,

$$\int_0^\infty \left(\dot{\underline{\theta}}^T \cdot \underline{\tau}\right)dt =$$

$$\int_0^\infty \left(\dot{\underline{\theta}}^T H\ddot{\underline{\theta}} + \dot{\underline{\theta}}^T C\dot{\underline{\theta}}\right)dt =$$

$$\int_0^\infty \left(\dot{\underline{\theta}}^T H\ddot{\underline{\theta}} + \frac{1}{2}\dot{\underline{\theta}}^T \dot{H}\dot{\underline{\theta}}\right)dt = \qquad (\dot{H} - 2C = 0)$$

$$\int_0^\infty \left(\frac{1}{2}\dot{\underline{\theta}}^T H\ddot{\underline{\theta}} + \frac{1}{2}\ddot{\underline{\theta}}^T H\dot{\underline{\theta}} + \frac{1}{2}\dot{\underline{\theta}}^T \dot{H}\dot{\underline{\theta}}\right)dt =$$

$$\frac{1}{2}\int_0^\infty \frac{d}{dt}\left(\dot{\underline{\theta}}^T H\dot{\underline{\theta}}\right)dt =$$

$$\frac{1}{2}\dot{\underline{\theta}}^T H\dot{\underline{\theta}} > 0$$

This implies that the mapping $\underline{\tau} \to \dot{\underline{\theta}}$ is passive. The control law is chosen to be $\underline{\tau} = \overline{Y}\hat{\underline{a}} - K_D\dot{\tilde{\underline{\theta}}}$, where K_D is some positive definite matrix, and $\overline{Y}$ is such that it satisfies the condition $\overline{H}\ddot{\underline{\theta}}_r = \overline{Y}\underline{a}$. Further, the adaptation rule is $\dot{\hat{\underline{a}}} = -\Gamma\overline{Y}^T\dot{\tilde{\underline{\theta}}}$.

Figure 3.3a explains graphically the relationships between the control law, the adaptation rule and the arm dynamics. The control input to the arm is the joint torque $\underline{\tau}$ which results to the arm angular states according to the arm dynamics ($\underline{\tau} = H\ddot{\underline{\theta}} + C\dot{\underline{\theta}}$). Using the adaptation rule $\dot{\tilde{\underline{\theta}}}$ is related to one of the components ($\overline{Y}\hat{\underline{a}}$) of the control joint torque $\underline{\tau}$. The second component of the control torque ($K_D\dot{\tilde{\underline{\theta}}}$) is also related to $\dot{\tilde{\underline{\theta}}}$. The two components of the control input can be lumped together into one block in the forward path leaving the arm dynamics as a separate block in the feedback path. Figure 3.3b gives a representation of the compact block representation.

We would like to show that the forward path (between $\dot{\tilde{\underline{\theta}}}$ and $K_D\dot{\tilde{\underline{\theta}}} + \overline{Y}\hat{\underline{a}}$) is passive. From the adaptation rule

$$\dot{\hat{\underline{a}}} = -\Gamma\overline{Y}^T\dot{\tilde{\underline{\theta}}} \quad (\Gamma : \text{positive definite matrix}) \qquad \Rightarrow$$

$$\Gamma^{-1}\dot{\hat{\underline{a}}} = -\overline{Y}^T\dot{\tilde{\underline{\theta}}} \qquad \Rightarrow$$

$$\dot{\hat{\underline{a}}}^T\Gamma^{-1} = -\dot{\tilde{\underline{\theta}}}^T\overline{Y} \qquad \Rightarrow$$

$$\dot{\hat{\underline{a}}}^T\Gamma^{-1}\hat{\underline{a}} = -\dot{\tilde{\underline{\theta}}}^T\overline{Y}\hat{\underline{a}} \qquad \Rightarrow$$

$$\frac{1}{2}\frac{d}{dt}\left[\hat{\underline{a}}^T\Gamma^{-1}\hat{\underline{a}}\right] = -\dot{\tilde{\underline{\theta}}}^T\overline{Y}\hat{\underline{a}}$$

Thus,

$$\int_0^\infty \left(\underline{\dot{\tilde{\theta}}}^T \cdot \left(K_D \underline{\dot{\tilde{\theta}}} + \overline{Y} \underline{\hat{a}} \right) \right) dt =$$

$$\int_0^\infty \left(\underline{\dot{\tilde{\theta}}}^T K_D \underline{\dot{\tilde{\theta}}} \right) dt - \frac{1}{2} \int_0^\infty \frac{d}{dt} \left(\underline{\hat{a}}^T \Gamma^{-1} \underline{\hat{a}} \right) dt =$$

$$\int_0^\infty \left(\underline{\dot{\tilde{\theta}}}^T K_D \underline{\dot{\tilde{\theta}}} \right) dt + \frac{1}{2} \underline{\hat{a}}^T \Gamma^{-1} \underline{\hat{a}} > 0, \qquad \textbf{eq. 3.8}$$

K_D is positive definite, and therefore, the forward loop is passive.

$\underline{\dot{\tilde{\theta}}}$ belongs to L_2 (because when we study passivity/positivity the input signals are assumed to be L_2), and $\overline{Y}$ has elements that are functions of $\underline{\dot{\tilde{\theta}}}$ and $\underline{\ddot{\theta}}_d$. Thus, as long as $\underline{\ddot{\theta}}_d$ is continuous and differentiable and L_2, then $\Gamma \overline{Y}^T \underline{\dot{\tilde{\theta}}}$ is also in L_2 (i.e., $\int_0^\infty \underline{\dot{\tilde{\theta}}}^T \overline{Y} \Gamma \Gamma \overline{Y}^T \underline{\dot{\tilde{\theta}}}$ is finite).

The parameter estimates ($\underline{\hat{a}}$) are always finite since they are the integrated output of the stable system $\underline{\dot{\hat{a}}} = -\Gamma \overline{Y}^T \underline{\dot{\tilde{\theta}}}$ (i.e., $\underline{\hat{a}} = \int_0^\infty \Gamma \overline{Y}^T \underline{\dot{\tilde{\theta}}}$ for which $\left| \underline{\hat{a}} \right| = \left| \int_0^\infty \Gamma \overline{Y}^T \underline{\dot{\tilde{\theta}}} \right| < \int_0^\infty \underline{\dot{\tilde{\theta}}}^T \overline{Y} \Gamma \Gamma \overline{Y}^T \underline{\dot{\tilde{\theta}}}$ $= A \in \Re$; $A < \infty$). Thus, the integral in eq. 3.8 is not only positive but also finite (because when we study passivity/positivity the input signals are assumed to be L_2). This implies that the integral $\int_0^\infty \left(\underline{\dot{\tilde{\theta}}}^T \cdot \overline{Y} \underline{\hat{a}} \right) dt$ is finite when the integrand is negative (or positive).

In mathematical notation,

$$-\infty < \int_0^\infty \left(\underline{\dot{\tilde{\theta}}}^T \cdot \overline{Y} \underline{\hat{a}} \right) dt = K < 0 \qquad \text{when } \overline{Y} \underline{\hat{a}} < 0$$

There are two cases for this to happen.

i. $-\infty < K < \min\left(\underline{\dot{\tilde{\theta}}}^T \cdot \overline{Y} \underline{\hat{a}} \right) < 0$ (i.e. $\min\left(\underline{\dot{\tilde{\theta}}}^T \cdot \overline{Y} \underline{\hat{a}} \right)$ is finite)

It is clear that when K_D is chosen to be large enough then $\dot{\tilde{\underline{\theta}}}^T K_D \dot{\tilde{\underline{\theta}}} > \left|\dot{\tilde{\underline{\theta}}}^T \overline{Y}\hat{\underline{a}}\right|$ for all cases when $\overline{Y}\hat{\underline{a}}$ is negative. Therefore, *the forward loop is strictly positive*. The inequality above means that the energy generated in the forward loop at each instant is finite at all times, and this is a fact regardless of the value of K_D.

ii. $\min\left(\dot{\tilde{\underline{\theta}}}^T \cdot \overline{Y}\hat{\underline{a}}\right) = \infty$ (i.e. $\min\left(\dot{\tilde{\underline{\theta}}}^T \cdot \overline{Y}\hat{\underline{a}}\right)$ is an infinite impulse; in fact it is equal to a finite number ($\hat{\underline{a}}$) times an infinite impulse squared ($\dot{\tilde{\underline{\theta}}}^T \cdot \overline{Y}$))

It is clear that when K_D is chosen to be large enough then $\dot{\tilde{\underline{\theta}}}^T K_D \dot{\tilde{\underline{\theta}}} > \left|\dot{\tilde{\underline{\theta}}}^T \overline{Y}\hat{\underline{a}}\right|$ for all cases when $\overline{Y}\hat{\underline{a}}$ is negative. Therefore, *the forward loop is strictly positive*.

The passivity of the feedback block was shown at the beginning of the proof. Theorem 1 implies that a compensator with a control law $\underline{\tau} = \overline{Y}\hat{\underline{a}} - K_D \dot{\tilde{\underline{\theta}}}$ and an adaptation rule $\dot{\tilde{a}} = \dot{\hat{a}} = -\Gamma \overline{Y}^T \dot{\tilde{\underline{\theta}}}$ can robustly control the complete nonlinear system.

The analysis of the stability, robustness and performance of such system is based on the basic assumption that the desired trajectory is "well behaved" (i.e. it does not have discontinuous velocity profile). In the proposed control method, it is necessary that the desired trajectory $\underline{\theta}_d(t)$ be known. However, some or all of the parameters of the system dynamics may be unknown. The adaptive control-design problem is to derive a control law for the actuator torques, and an estimation law for the unknown parameters, such that the manipulator output $\underline{\theta}(t)$ closely tracks the desired states.

From the proof it appears that the stability of the closed-loop system is for large values of K_D. This is a result of the very conservative constraints of Theorem 1. In reality, passivity of the forward and the feedback loops along with the fact that the generated energy of the forward loop is finite at all times is enough to show boundedness of the closed loop.

Clearly, the control law and the adaptation rule are not unique since the choices for $\overline{H}$, $\overline{Y}$, K_D and Γ are not unique either. The proposed control model has certain advantages over the classical sliding mode control for robotic systems. The most important of the advantages is the fact that the controller only needs linear combinations of position and velocity feedback to robustly control the system. Classical sliding mode requires exact knowledge of the form of the nonlinearities. On the downside however, the proposed model reaches bounded, non-zero steady state error, while sliding mode guarantees zero error asymptotically.

3.3.3. The proposed control model for the two-link robotic arm paradigm

I will demonstrate the principle arguments of the stability and robustness by going through the example of the two-link model.

Consider the same two-link manipulator model, whose dynamics are

$$\underbrace{\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix}=\begin{bmatrix}H_{11} & H_{12}\\ H_{21} & H_{22}\end{bmatrix}\begin{bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{bmatrix}+\begin{bmatrix}-h\dot{\theta}_2 & -h\left(\dot{\theta}_1+\dot{\theta}_2\right)\\ h\dot{\theta}_1 & 0\end{bmatrix}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix} + \begin{bmatrix}m_2g\left(L_{c2}\cos(\theta_1+\theta_2)+L_1\cos(\theta_1)\right)+m_1gL_{c1}\cos(\theta_1)\\ m_2gL_{c2}\cos(\theta_2)\end{bmatrix}}_{P} \qquad \textbf{eq. 3.9}$$

where

$$H_{11}=a_1+2a_3\cos(\theta_2) \qquad H_{12}=H_{21}=a_2+a_3\cos(\theta_2)$$

$$H_{22}=a_2 \qquad h=a_3\sin(\theta_2)$$

with

$$a_1=I_1+m_1L_{c1}^2+I_2+m_2L_{c2}^2+m_2L_1^2 \qquad a_2=I_2+m_2L_{c2}^2 \qquad a_3=m_2L_{c2}L_1$$

$$a_4=m_2gL_{c2} \qquad a_5=m_2gL_1 \qquad a_6=m_1gL_{c1}$$

For this system it is easy to show that

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix}=\begin{bmatrix}\ddot{\theta}_1 & \ddot{\theta}_2 & \left(2\ddot{\theta}_1+\ddot{\theta}_2\right)\cos(\theta_2)-\left(2\dot{\theta}_1\dot{\theta}_2+\dot{\theta}_2^2\right)\sin(\theta_2) & \cos(\theta_1+\theta_2) & \cos(\theta_1) & \cos(\theta_1)\\ 0 & \ddot{\theta}_1+\ddot{\theta}_2 & \ddot{\theta}_1\cos(\theta_2)+\dot{\theta}_1^2\sin(\theta_2) & \cos(\theta_1+\theta_2) & 0 & 0\end{bmatrix}\begin{bmatrix}a_1\\ a_2\\ a_3\\ a_4\\ a_5\\ a_6\end{bmatrix}$$

The resulting controller matrices are:

$$Y=\left[\begin{array}{c:c:c:c:c:c}\ddot{\theta}_{1r} & \ddot{\theta}_{2r} & \left(2\ddot{\theta}_{1r}+\ddot{\theta}_{2r}\right)\cos(\theta_2)-\left(\dot{\theta}_{1r}\dot{\theta}_2+\dot{\theta}_{2r}\dot{\theta}_2+\dot{\theta}_{2r}\dot{\theta}_1\right)\sin(\theta_2) & \cos(\theta_1+\theta_2) & \cos(\theta_2) & \cos(\theta_2)\\ \hdashline 0 & \ddot{\theta}_{1r}+\ddot{\theta}_{2r} & \ddot{\theta}_{1r}\cos(\theta_2)+\dot{\theta}_{1r}\dot{\theta}_1\sin(\theta_2) & \cos(\theta_1+\theta_2) & 0 & 0\end{array}\right]$$

and

$$\underline{a}=\begin{bmatrix}a_1 & a_2 & a_3 & a_4 & a_5 & a_6\end{bmatrix}^T$$

For the nonlinear system dynamics of the two-link model, the sliding-mode control law guarantees robust stability.

Notice that the dynamics of the two-link system (P) can be rewritten as the sum of two separate systems in the following form

$$P=\overline{P}+\delta P\Rightarrow$$

$$\underline{\tau}=\overline{\underline{\tau}}+\underline{\tau}_{\delta p}\Rightarrow$$

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix}=\begin{bmatrix}\overline{\tau}_1\\ \overline{\tau}_2\end{bmatrix}+\begin{bmatrix}\delta\tau_1\\ \delta\tau_2\end{bmatrix}=$$

$$\underbrace{\begin{bmatrix}a_1+2a_3 & a_2+a_3\\ a_2+a_3 & a_2\end{bmatrix}\begin{bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{bmatrix}}_{\overline{P}}+\underbrace{\left(a_3\left(\cos(\theta_2)-1\right)\begin{bmatrix}2 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{bmatrix}+a_3\sin(\theta_2)\begin{bmatrix}-\dot{\theta}_2 & -\left(\dot{\theta}_1+\dot{\theta}_2\right)\\ \dot{\theta}_1 & 0\end{bmatrix}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix}\right)}_{\delta P}\quad\textbf{eq. 3.10}$$

Figure 3.4 shows this arrangement in block diagram form. In this new control arrangement we would like for the "linearized" controller (with output $\overline{\underline{\tau}}$) to be able to robustly control the nonlinear system P. The dynamics of the nonlinear system P, which is also given by eq.

3.9, can be rewritten in terms of the dynamics of eq. 3.10. Figure 3.4 shows this rearrangement. Figure 3.5 shows the dynamics of the controller and the nonlinear system. Now, the controller is controlling the "linearized" dynamics, while the nonlinear terms are lumped as additive disturbances.

The proposed controller structure uses only a linearized version of the nonlinear matrix Y. Let the linearized version of Y be $\overline{Y}$.

$$\overline{Y} = \begin{bmatrix} \ddot{\theta}_{1r} & \ddot{\theta}_{2r} & 2\ddot{\theta}_{1r} + \ddot{\theta}_{2r} & 1 & 1 & 1 \\ 0 & \ddot{\theta}_{1r} + \ddot{\theta}_{2r} & \ddot{\theta}_{1r} & 1 & 0 & 0 \end{bmatrix}$$

To get $\overline{Y}$, set $\theta_1 = 0$ and $\theta_2 = 0$, which results in $\overline{H} = \begin{bmatrix} a_1 + 2a_3 & a_2 + a_3 \\ a_2 + a_3 & a_2 \end{bmatrix}$, and $\overline{C} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$ (exactly like $\overline{P}$). Notice that the decomposition of P into $\overline{P}$ and δP is not unique. In fact, while θ_2 has to be zero for the controller to consist of linear terms, there are infinite number of such decompositions for depending on the value of θ_1.

Classical sliding-mode theory the controller with a control law $\overline{\underline{\tau}} = \overline{Y}\underline{\hat{a}} - K_D\underline{\dot{\tilde{\theta}}}$ and an adaptation rule $\dot{\tilde{a}} = \dot{\hat{a}} = -\Gamma\overline{Y}^T\underline{\dot{\tilde{\theta}}}$ provides robust control to the following system dynamics:

$$\begin{bmatrix} a_1 + 2a_3 & a_2 + a_3 \\ a_2 + a_3 & a_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} a_4 + a_5 + a_6 \\ a_4 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \qquad \textbf{eq. 3.11}$$

The reduced linearized system fits the generalized form for robotic system dynamics ($\overline{H}\underline{\ddot{\theta}} + \overline{C}\underline{\dot{\theta}} + \underline{g} = \underline{\tau}$, with $\frac{d}{dt}\left(\overline{H}\right) - 2\overline{C} = 0$ and $\overline{H}\underline{\ddot{\theta}}_r + \overline{C}\underline{\dot{\theta}}_r + \underline{g} = \overline{Y}\underline{a}$), which is guaranteed to be robust through sliding mode theory.

In order to compare the performance of the proposed model to the classical sliding mode control for robotic systems numerical simulations were run for the same two-link system using the same controller gains (K_D and Γ). Figure 3.6 shows the results of the sliding mode controller. The following control gains were used:

$$K_d = \begin{bmatrix} 70 & 0 \\ 0 & 70 \end{bmatrix},\ \Lambda = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix},\ \text{and } \Gamma = \begin{bmatrix} 0.6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.04 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.03 & 0 & 0 & 0 \\ 0 & 0 & 0 & 300 & 0 & 0 \\ 0 & 0 & 0 & 0 & 40 & 0 \\ 0 & 0 & 0 & 0 & 0 & 225 \end{bmatrix}.$$

The performance of the proposed controller in this two-link arm paradigm is shown in Figure 3.7. The gains used in this case are exactly the same as those used classical sliding mode controller. Clearly, for the same gains the lack of the knowledge of the nonlinearities translates in lost of asymptotic tracking. However, the error is always bounded to within a boundary layer. Further, the proposed control law allows reduction of this boundary layer to whatever size is desirable by increasing accordingly the feedback gain K_D. Figure 3.8 demonstrates exactly that. By increasing K_D by a factor of 100 the boundary layer has been driven to very close to zero.

3.3.4. Example: Seven-Input-Seven-Output robotic arm system

What follows is an example of the proposed nonlinear control method to demonstrate numerically the stability of the proposed model for a system with more than two links. The system is a seven-link robotic arm tracking a movement in the three dimensional space during two distinct condition. In the first situation there are no external dynamics influencing the arm, while in the second the environment of the system is also in motion (rotation) exerting Coriolis effects on the dynamics of the arm. The joint angles are numbered to correspond to Figure 3.2.

Figures 3.10 and 3.11 are direct applications of the proposed control model to a seven-link manipulator arm similar to the seven-link model of the human arm developed in Chapter 2. For the simulation the following control gains were used:

$$K_d = 10^5 \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix},\ \Lambda = 20 \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix},\ \text{and } \Gamma = 10^4 \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}.$$

K_d and Λ are 7×7 while Γ is 64×64.

Clearly, the proposed model manages to control the multiple-link robotic system while keeping the tracking error bounded even in the presence of external disturbances (i.e. Coriolis and centripetal forces). Further, the proposed controller does not require explicit knowledge of the system's nonlinearities. Rather, it uses only position and velocity feedback to achieve trajectory tracking.

3.4. Summary of the control modeling of the human arm

This chapter described the theoretical contribution of this thesis. A new control method was proposed for the control of robotic arms. The control law and the adaptation rule are based on sliding mode control of robotic systems. However, the preexisting control methods require complete knowledge of the nonlinear robotic dynamics and very complex feedback. The proposed scheme was motivated by the fact that the CNS achieves good tracking performance, while it remains robust in the presence of external unknown dynamics. The CNS achieves satisfactory performance using just position and velocity feedback, and without any explicit knowledge of the nonlinear nature of the dynamics. The robust stability of the proposed model was proven rigorously and it was demonstrated numerically through Matlab (Mathworks Inc., Natick, MA) simulations.

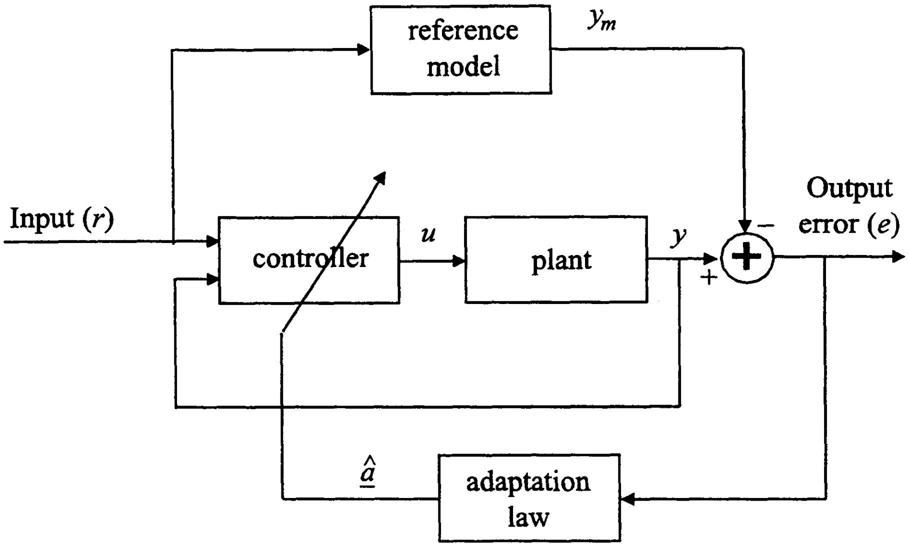


Figure 3.1 The block diagram gives a description of a Model Reference Adaptive Control (MRAC) method for compensation.

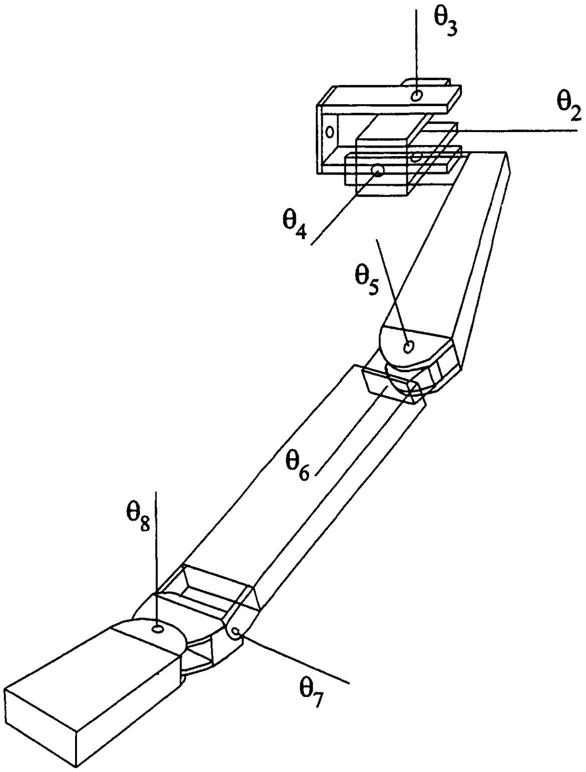


Figure 3.2 Breakdown of a human or robotic model arm into simple one-dimensional joint links.

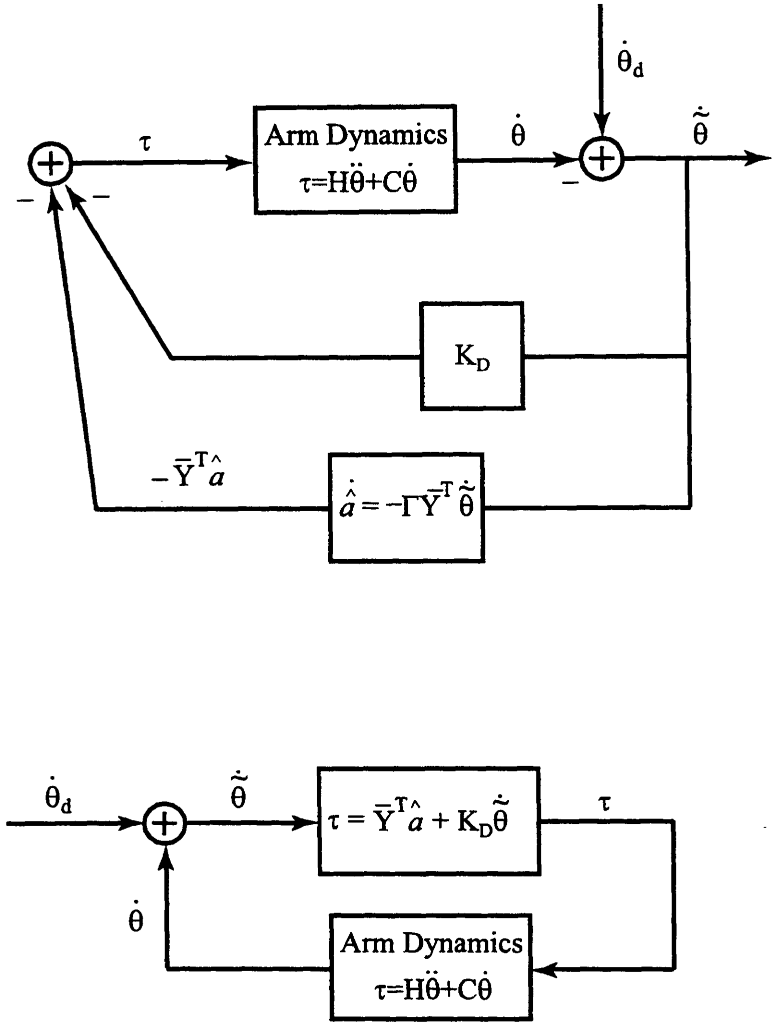

$\dot{\theta}_d$
τ
Arm Dynamics
$\tau = H\ddot{\theta} + C\dot{\theta}$
$\dot{\theta}$
$\dot{\tilde{\theta}}$
$-$
$-$
$-$
K_D
$-\bar{Y}^T\hat{a}$
$\dot{\hat{a}} = -\Gamma\bar{Y}^T\dot{\tilde{\theta}}$
$\dot{\theta}_d$
$\dot{\tilde{\theta}}$
$\tau = \bar{Y}^T\hat{a} + K_D\dot{\tilde{\theta}}$
τ
$\dot{\theta}$
Arm Dynamics
$\tau = H\ddot{\theta} + C\dot{\theta}$

Figure 3.3 Passivity interpretation of the adaptive robot controller.

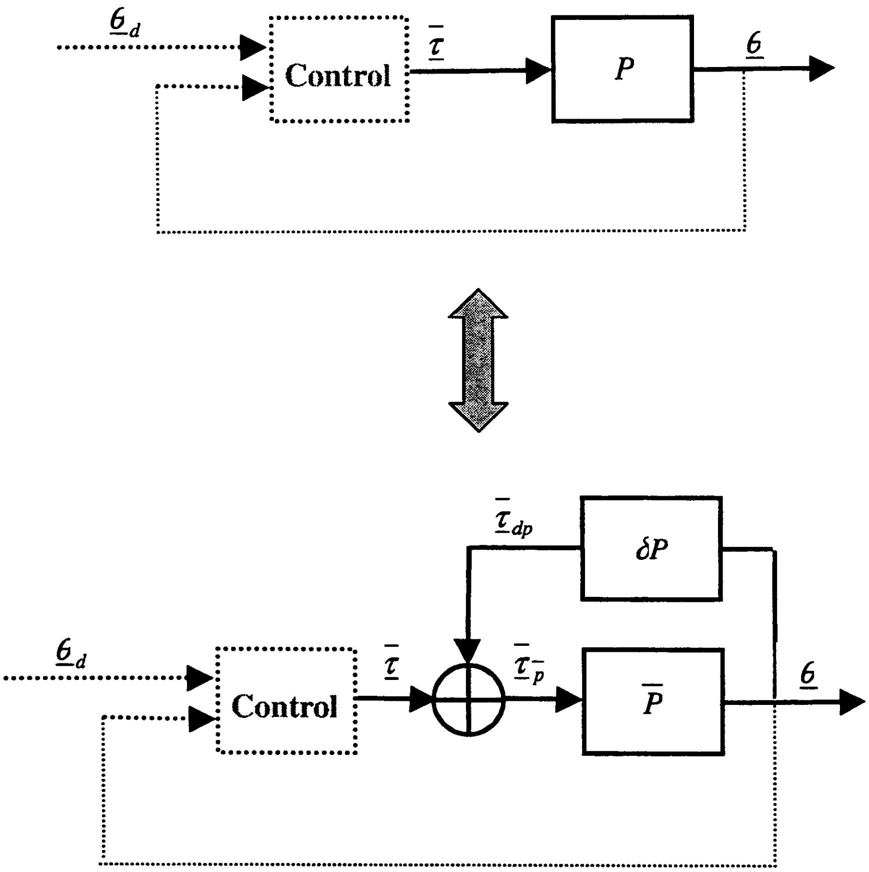


Figure 3.4 The proposed adaptive scheme for trajectory tracking control of a manipulator in an unknown or unstructured environment. The nonlinearities can be modeled as an additive block (represented by δP) on a linearized system model.

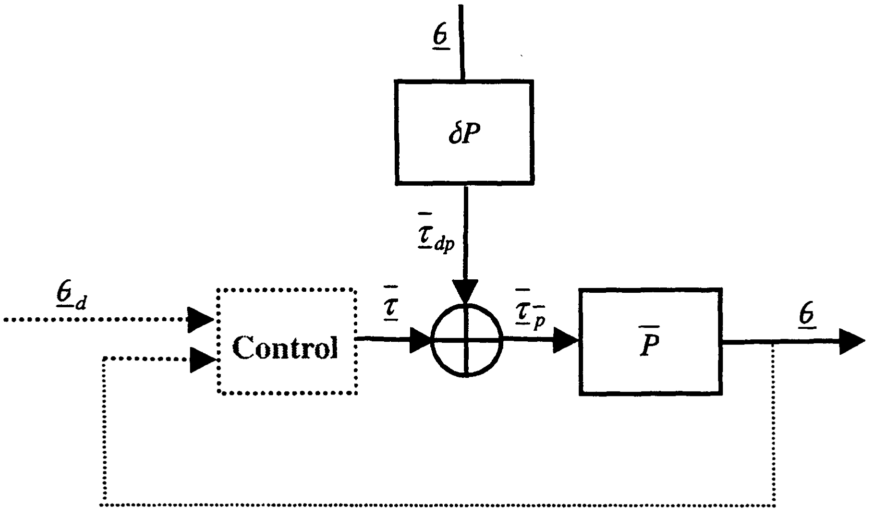


Figure 3.5 The proposed adaptive scheme for trajectory tracking control of a manipulator in an unknown or unstructured environment. The nonlinearities can be modeled as an additive disturbances (represented by δP) on a linearized system model.

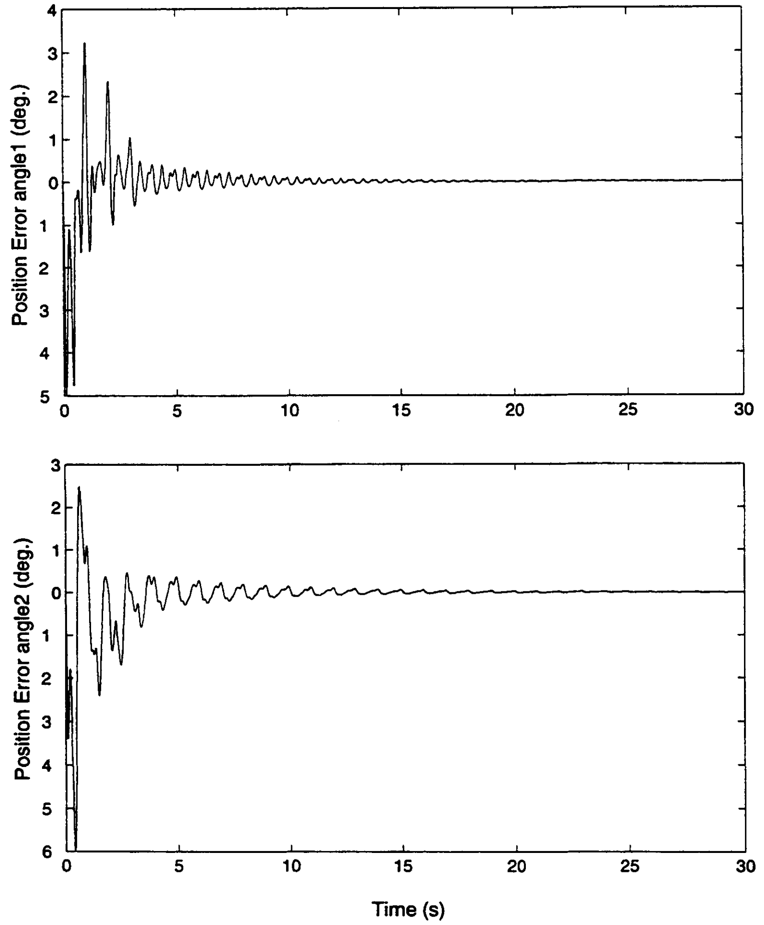


Figure 3.6 Graph of position errors versus time for the trajectory-tracking simulation of the planar dynamics of the seven-degree-of-freedom arm. Since the movement is planar only the reduced (two-dimensional) dynamics were used. The desired trajectories for the two joint angles were: Desired Angle 1 = $\left(1-\cos(2\pi t)\right)\frac{180}{\pi}$ **and Desired Angle 2 =** $2\left(1-\cos(2\pi t)\right)\frac{180}{\pi}$**.**

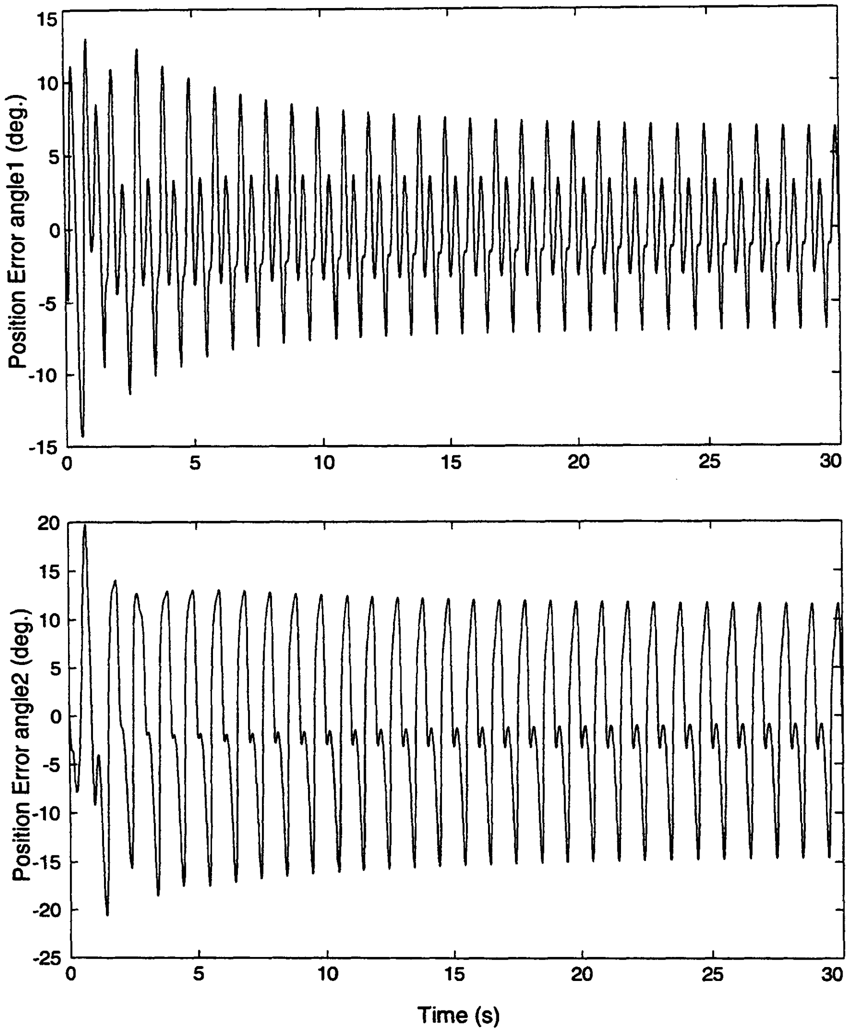


Figure 3.7 Graph of the position tracking errors versus time for the proposed adaptive scheme for control of a manipulator. The arm is executing a planar two-dimensional movement. The desired trajectories in degrees for the two joint angles were: Desired Angle 1 = $(1-\cos(2\pi t))\frac{180}{\pi}$ and Desired Angle 2 = $2(1-\cos(2\pi t))\frac{180}{\pi}$.

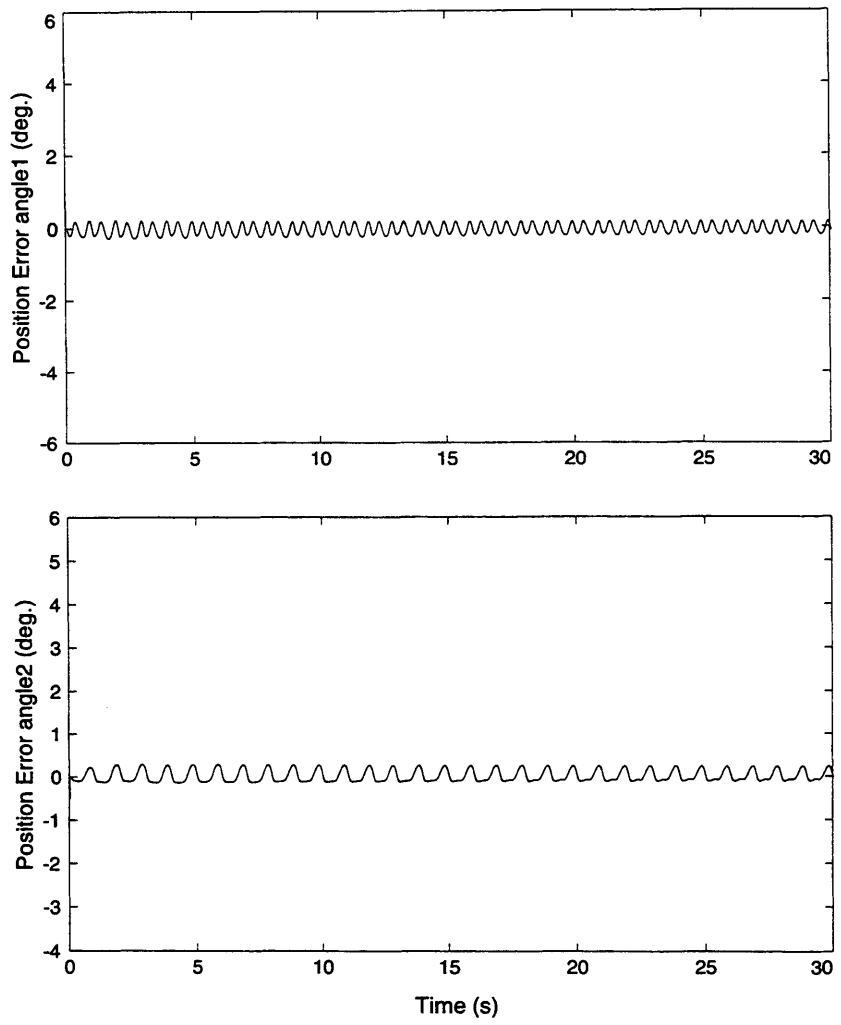


Figure 3.8 Graph of the position tracking errors versus time for the proposed adaptive scheme for control of a manipulator. The arm is executing a planar two-dimensional movement. In this simulation, the proportional and adaptation gains (K_D and Γ) are 200 times larger than that in Figure 3.7. The desired trajectories for the two joint angles were: Desired Angle 1 = $(1-\cos(2\pi t))\frac{180}{\pi}$ and Desired Angle 2 = $2(1-\cos(2\pi t))\frac{180}{\pi}$.

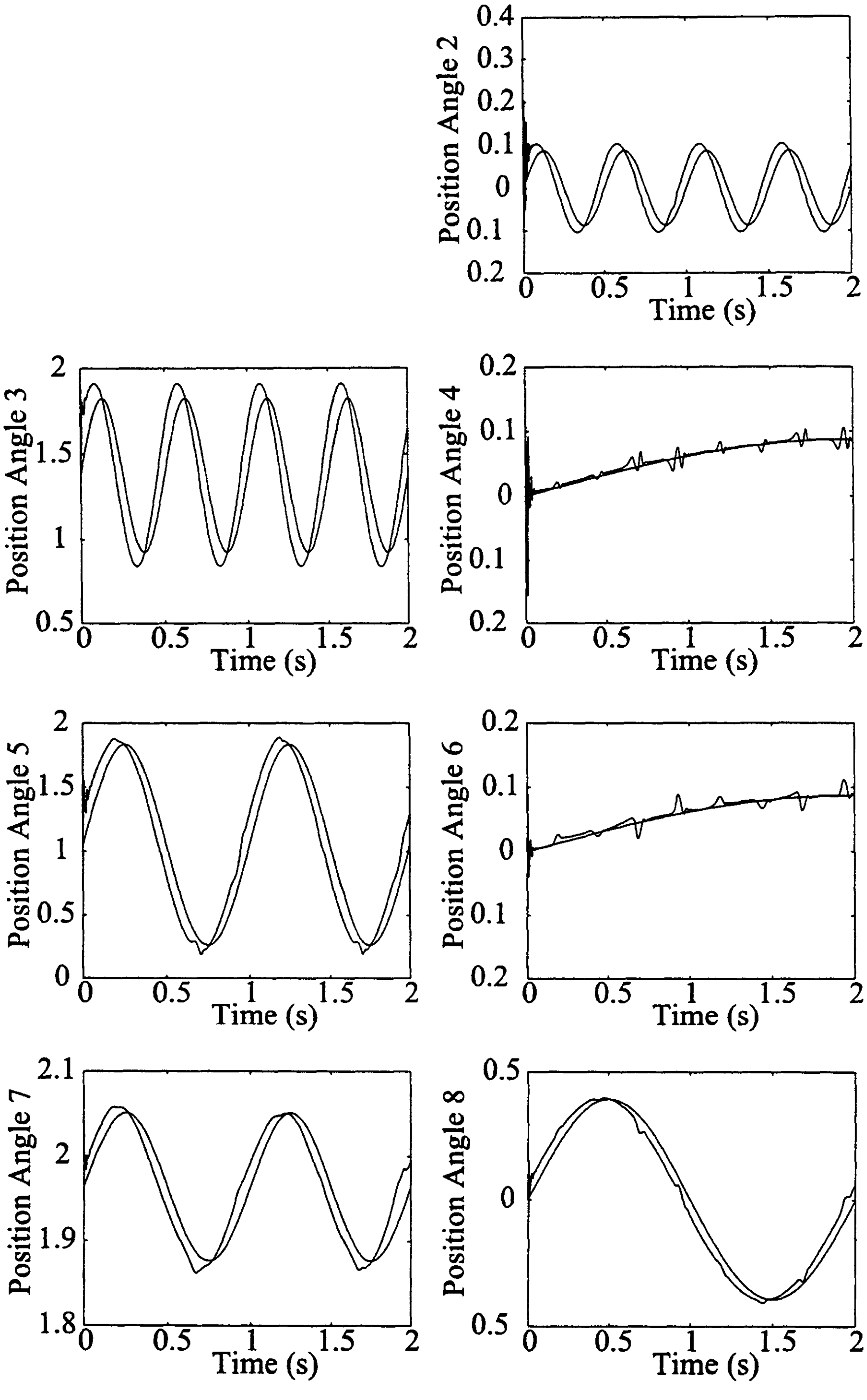


Figure 3.9 Position errors versus time of the proposed adaptive scheme for trajectory tracking control. The seven-link manipulator is executing a three-dimensional movement. There are no external influences except from gravity.

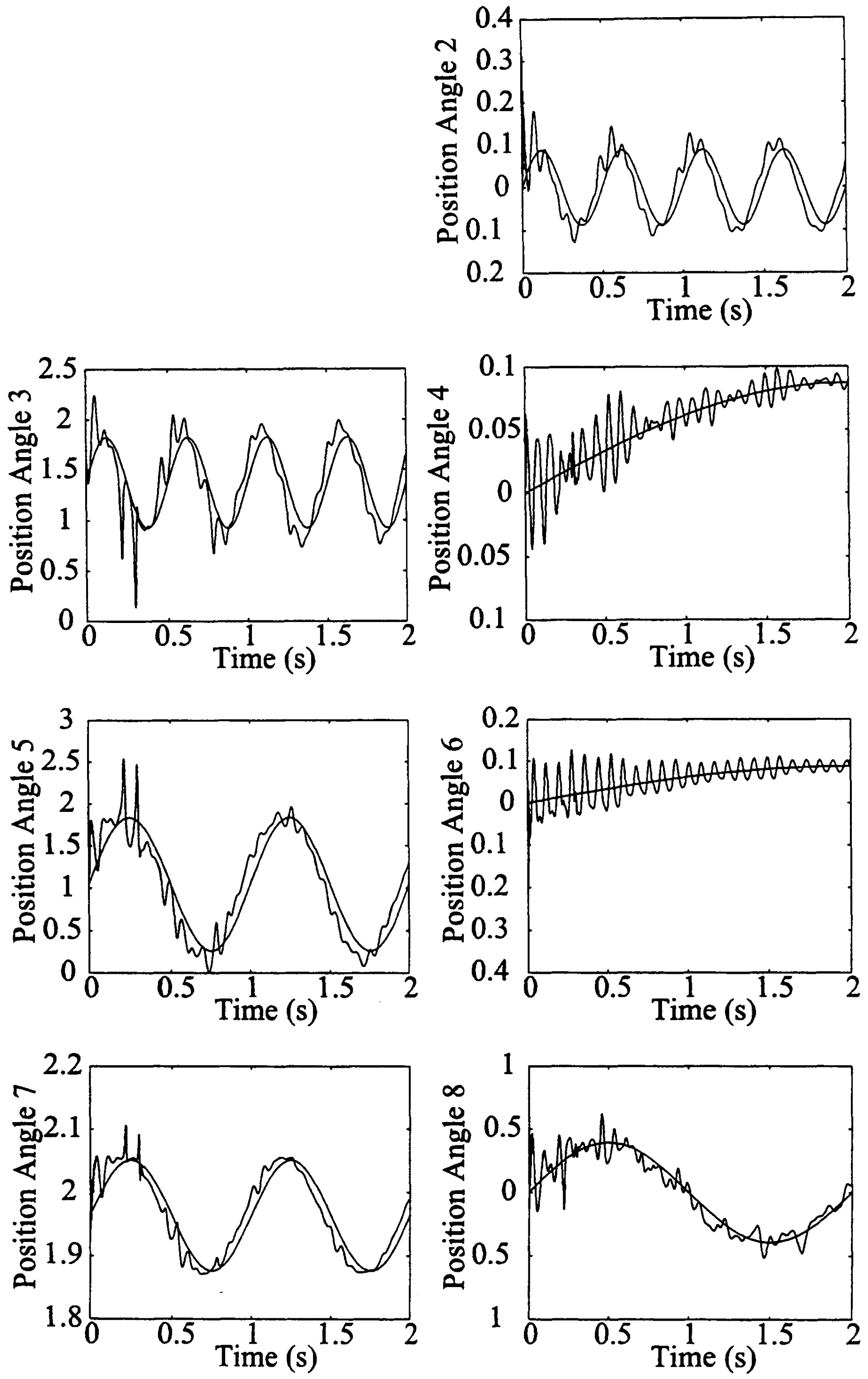


Figure 3.10 Position errors versus time of the proposed adaptive scheme for trajectory tracking control. The seven-link manipulator is executing a three-dimensional movement. The environment is also assumed to be in motion (rotation), and thus, the are external forces (Coriolis) in addition to gravity.

4. Comparisons between Experimental and Simulation Results

The experimental result section is the last part of the analysis presented in this thesis. In this and the following chapter the experimental results are presented, they are compared with simulation results or theoretical predictions, and conclusions are drawn. It was mentioned in the introduction that the experimental part is twofold. The first part includes a throwing task experiment on board the Russian space station Mir (Мир), while the second a continuous trajectory tracking task under an external disturbance, caused by a robotic manipulandum. This chapter will describe the comparison of experimental results during continuous trajectory tracking with the model predictions, while the following chapter will concentrate more the throwing experiment on board Mir.

The continuous tracking experiments are a unique contribution to the understanding of human motor control. The present study looks into the behavior of the motor control system in the closed loop. Most previous work has concentrated in looking at the motor control system behavior as in an open loop by looking simply tasks that imitate reaching movements (Bizzi et al., 1992; Bizzi et al., 1991; Dizio and Lackner, 1995; Shadmehr and Mussa-Ivaldi, 1994). This types of experiments, while valuable in showing certain adaptation trends, they were only looking at a limiting description of the problem. The limitations of those experiments was the fact that the input to the motor controller was a step, and that the general duration of each task was very short. Clearly such an approach is limited to the amount of insight it can provide, and this was one of the main drivers for the design of this series of experiments.

The trajectory tracking experiments are unique for another reason. Among competitive closed loop models of the motor system some do not compare their predictions to experimental data (Kawato and Gomi, 1992), while other studies are limited to looking only into the undisturbed behavior of the tracking response (Hanneton et al., 1997) making impossible to recognize differences between models.

4.1. Reaching Experiment in a Rotating Room

Before looking at the continuous tracking experiments it is important to see how does the proposed adaptive method fit with the experimental results produced by the reaching movements under the influence of an external force field. Coriolis forces generated by movements in a rotating environment can serve as a way of studying the influence of transient perturbations on limb movement control without providing mechanical contact with the limb. Coriolis forces are proportional to limb velocity (Coriolis force $\underline{F}_{cor} = -2m(\underline{\omega} \times \underline{v})$, where m is the mass of the arm (or any other object), $\underline{\omega}$ is the angular velocity vector of the rotating environment in radians, and $\underline{v}$ is the linear velocity vector of the arm). This means that there is not a Coriolis force acting at the very beginning or at the very end of a limb movement, but during motion a Coriolis force orthogonal and proportional to $\underline{v}$ is generated. Such forces are inertial ones that act without physical contact. Local contact, if present, necessarily activates a variety of somatosensory and mechanoreceptors in the limb that potentially provide information about the nature of the perturbation. The use of Coriolis forces to perturb limb movements avoids this and permites the study of how natural, unencumbered movements respond to transient perturbations of trajectory. Recently, investigators demonstrated that, when subjects, who are in a fully enclosed room rotating at 10 rpm, reach out to touch 9 targets, the subjects show substantial movement curvature and miss the desired target position by a large amount (Dizio and Lackner, 1995). The deviations were generated in the direction of the Coriolis force during the reaching movements. Within ~10 reaches, subjects adapt almost completely and move in straight paths to the target location (Lackner and Dizio, 1994). Adaptation occurs even when subjects were denied both sight of their movements and cutaneous texture cues about the target (a light-emitting diode embedded from below in a smooth Plexiglas sheet). When rotation ceases, subjects again make reaching errors with the adapted arm. Endpoints and movement paths are initially deviated in the direction opposite the Coriolis forces that had been present during rotation, and then re-adaptation restores a straight accurate reach to the target. Figure 4.1 was adopted from the paper published by Dizio and Lackner, 1995 and it illustrates the initial effects Coriolis forces have on movement endpoints and trajectories as well as the pattern of adaptation and aftereffects.

The end of chapter 3 shows that the proposed control model has no difficulty controlling Coriolis forces, even during three dimensional movements. It would be useful further to see in a qualitative way how the performance of the proposed system compares with experimental data also under the influence of Coriolis forces.

4.1.1. Methods and Results (Adopted from Dizio and Lackner, 1995)

The experiment was divided into pre-rotation, perrotation, and post-rotation components. The pre-rotation period was used to measure baseline pointing accuracy for the two arms. It involved 24 reaching movements to the target with each arm in alternate sets of 8 movements starting with the right arm. The perrotation period involved 80 reaching movements, in sets of 8, using only the right arm. The post-rotation period involved 24 movements of each arm in alternating sets of 8 movements beginning with the left arm. Eight subjects followed the sequence just described. On completion of the pre-rotation reaches, the rotating room was accelerated to a constant velocity of 60°/s (1/6 rev/s) counterclockwise. Two minutes were allowed to elapse at constant velocity before the subject began reaching movements so that the horizontal semicircular canals would be back at their resting discharge levels. Because the room was rotating counterclockwise, the Coriolis forces generated during reaching movements were rightward with respect to the subject.

Subjects made nearly straight reaches toward the target position with both the right and the left arms. Right-handed reaches were convex to the left and left-handed ones to the right. Neither movement distance nor peak velocity changed significantly throughout the experiment.

On their first perrotation reach, *every subject showed an increase of movement curvature*, and a deviation of movement endpoint, in the direction of the Coriolis force acting during the movement. These deviations from the pre-rotation baseline values of the right arm were highly significant. With subsequent reaches *subjects rapidly adapted such that their reaches straightened out and came closer to the target*. The curvature of the last set of eight movements during rotation did not differ from the pre-rotation baseline curvature values. The endpoints of the last eight perrotary movements were significantly less deviated than the endpoint of the initial one but remained significantly to the right of the pre-rotation baseline. This pattern indicates that the right arm had achieved complete adaptation to curvature induced by Coriolis forces and ~70% adaptation to induced endpoint errors. Figure 4.1 shows a graphical representation of the setup as well as the subject behavior during the execution of the pointing task inside the rotating room.

4.1.2. Adaptive Control Simulations

The experimental data obtained by Dizio were compared with the proposed control model, which was discussed extensively in the last chapter. Figure 4.2 shows the experimental data from the rotating facility, and Figure 4.3 demonstrates the simulation results. There are a number of differences between the simulations and the experimental data. First and foremost, the experimental protocol called for a pointing task. While the subjects had to extend their arm to reach the target, only the initial and final positions of the hand were important. The points in between were important but not in the sense of trajectory tracking. On the contrary, the simulation of the proposed control model is based on continuous trajectory tracking, which means that a desired trajectory was assumed throughout the duration of the movement. Second, the simulation model did not included any muscle modeling information, neither did it include any time delays or friction losses. Finally, unlike the human subjects the simulation has to be continuous in time (because of programming restrictions). In the simulation the movement is continuous and smooth. It is clear from the experimental results (Figure 4.2) that human subjects move their arms slower at the beginning of the movement and finish quite abruptly. The reason for such behavior is quite likely the time delay between the onset of the visual queue and the initiation of the reaching movement. On the contrary, the simulation includes no time delays and, therefore, the velocity profile of the simulated movement is symmetric at the beginning and at the end of the task.

Finally, while during the pointing task the final state is fixed, in the simulation, the "final"/desired state changes continuously.

4.1.3. Problem Formulation and Control Modeling and Result Discussion

The control model is defined in joint angles and therefore the desired trajectory had to be converted from a straight line on the horizontal plane into joint angles. To guarantee the desired trajectory path is smooth, the desired velocity profile along the straight path was chosen to be a sinusoid. The time (t) profile of the distance y was a sinusoid with an amplitude of 5cm. The velocity profile is "smooth" (i.e. continuously differentiable), and for the sake of simplicity the upper arm and the forearm are considered to be of the same length L. The desired trajectory kinematics was defined via the joint angles to be:
for the shoulder angle (in degrees):

$$\theta_1 = 15.562 - 10.815\cos(2t),\ \dot{\theta}_1 = 21.63\sin(2t), \text{ and } \dot{\theta}_1 = 43.26\cos(2t)$$

and for the elbow (in degrees):

$$\theta_2 = 93.25 - 13.93\cos(2t),\ \dot{\theta}_2 = 27.86\sin(2t), \text{ and } \ddot{\theta}_2 = 55.72\cos(2t)\cdot$$

The adaptive controller developed in Chapter 3 can be adjusted appropriately to match the behavior of the subjects. The simulation results depend significantly on the gains of the control model of the motor system. The adaptation gain Γ of the controller affects directly the number of trials the subjects need to do before their curvature in the Coriolis environment drops to the pre-rotational levels. Correspondingly, the error feedback gain of the controller K_D influences the size (both initial and final) of the trajectory curvature. Large values in Γ imply fast adaptation, while large values in K_D indicate small curvature in the trajectory. The simulation Matlab (Mathworks Inc., Natick, MA) code which was used to produce the results of Figure 4.3a is given in Appendix D. The simulation time was 200 seconds, and the gains were:

$$K_d = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \ \Lambda = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix}, \text{ and } \Gamma = \begin{bmatrix} 0.03 & 0 & 0 & 0 & 0 \\ 0 & 0.05 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0.03 & 0 \\ 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}.$$

It is important to remember that these gains are not special or unique. They are only chosen for the sake of the demonstration.

Because the desired trajectory was not a straight line to start with the adapted model did not approach a straight line but rather a line close to the desired trajectory. If the desired trajectory had had a slight curvature in the direction opposing to the Coriolis force, the final trajectory would have approached a straight line much more closely. For the simulation in Figure 4.3b the desired trajectory had a curvature in the direction opposing to the Coriolis force. Further, differences in the exact details of the actual and simulated paths are most likely due to the oversimplification of the full-arm model by just a dynamics model (of unknown physical parameters) as well as the fact that the desired velocity profile is completely unknown.

Summarizing, it is important to mention that the proposed model controls Coriolis disturbances effectively, and its performance agrees qualitatively with the experimental data. This means that the proposed controller manages to reduce the curvature of the actual path as the number of repetitions increases. It is also important to emphasize that matching the behavior of the model to the experimental data does not reject other competitive models. For instance a PID controller that has somehow the ability to adjust its gain could potentially increase its gain incrementally to high enough values so that the tracking error would be too small. Another instance were a PID with an adaptive mechanism might work is if the system could somehow incrementally adjust its desired (reference) trajectory. If the desired path is the mirror image of the actual path when the reference input is the straight line, then the final trajectory will match the straight line more closely (see Figure 4.3c). This is also very similar to one way to make the proposed model predictions match the experimental data, and it is a behavior that it is quite possible during the execution of a reaching task. Even in these two special situations that a PID variant can explain the observed data, the existence of an adaptation scheme is imperative. A constant gain PID controller could have never shown incremental improvement observed during the experiments.

4.2. Continuous Trajectory Tracking

As a first step towards a quantitative analysis of motor learning, I tried to test a functional model of the adaptive motor control system. The proposed model is chosen because it is a simple representation of the motor learning system that incorporates an internal model, it conforms with the adaptive control ideas that were presented in the previous chapter. Further, the model includes the motor system, the mechanical environment, the internal model, and the controller.

The *motor system* includes the musculoskeletal system for force and torque generation and part of the primary motor cortex that drives it. Its dynamical behavior is dependent on passive system dynamics, which is given by the some of inertial, Coriolis, centripetal, and frictional forces.

The *environment* subsystem is the mechanical load to the motor system resulting from an external force field. For goal-directed movements in the horizontal plane the force field is negligible. In the experimental procedures, an artificial force field is generated by means of a robot arm to produce varying mechanical perturbations to the arm movement. The performance of the motor learning system in adaptively countering the external perturbation can be studied systematically by carefully choosing the types of force fields that are presented to the subjects' arms.

The motor system and its mechanical environment are coupled together to form the system dynamics that the neural controller has to learn to guide the human arm accurately along the desired trajectory. A command to the motor system may result in different movement trajectories depending on the environment force field. Such perturbations in movement trajectories are corrected through proprioceptive and tactile feedback, which convey information about the position and speed of movement. However, the excessive time delays (40-60 ms; Evarts, 1981) and the limited gains, which are inherent in such long sensory feedback loops, make it impossible to achieve completely accurate trajectory tracking under the environmental perturbations.

The *internal model* is a neural replicate of the inverse dynamics of the motor system and its environment. At any instant of time, the internal model represents the cumulative experience of the motor learning system in predicting the input-output relationship of the motor-environment complex. This experience is acquired by gradual adaptation to changes in the environment.

Such adaptation requires the existence of neural integrators capable of accumulating and memorizing inputs. In the past, the visuomotor system has been associated with such neural integrators (Robinson, 1989; Shen, 1989).

To learn the input-output relationship, the internal model must receive as an input an efference copy of the motor command, and its output (reference trajectory) must be compared with the actual trajectory. The error signal, which is the difference between the reference and the actual movement trajectory, is then used to update the internal (reference) model until the error vanishes, and the internal model closely replicates the motor-environment complex.

In visually guided movements, the error signal is derived at least in part from visual feedback (Jeannerod, 1990). Accordingly, we assume that the error signal in our model is mediated by visual feedback (or other secondary sensory feedback) expressed in body coordinates.

The adaptation of the internal model with respect to the motor-environment complex is analogous to MIMO MRAC and sliding mode control. Here, the motor and environment blocks represent the actual and desired trajectories. The controller of the system is a neural controller. It produces the motor command that drives the motor system. The controller includes the internal model of the inverse dynamics, and it must be adaptively adjusted so that its predicted output accurately tracks the desired input. Note that the internal model is actually part of the controller, which is continuously updated. The motor command is augmented if the predicted movement differs from the desired trajectory. The underlying transformation is equivalent to an inversion of the arm kinematics (Kawato et al., 1987).

4.2.1. Experimental setup

For this experimental study a setup similar to the one shown in Figure 4.4 has been used. The special robotic manipulator has been used extensively in other studies (Shadmehr and

Mussa-Ivaldi, 1994). The manipulator has two torque motors that induce various dynamic force fields on the subjects arm during execution of tracking tasks. The manipulator is planar with two degrees of freedom, it weighs little, and its joints are have very little friction. It also includes a six-axis force-torque transducer (Lord F/T sensor) mounted on its end-effector (the handle). Two low-inertia, DC torque motors (PMI Corp., model JR16M4CH), mounted on the base of the robot, are connected independently to each joint via a parallelogram configuration. Position and velocity measurements are made using two optical encoders (Teledyne Gurley) and two tachometers (PMI). The optical encoders and the tachometers are mounted on the axes of the mechanical joints. The apparatus includes a video display monitor mounted directly above the base of the robot, approximately at the eye level of the subjects. The monitor is used to display the position of the robot's handle as well as the target for the tracking movement.

Subjects were seated on a chair bolted onto an adjustable positioning mechanism and were instructed to grip the handle of manipulandum with their strong arm. Their shoulders were restrained by a harness belt, and their right upper arms were supported in the horizontal plane by a vertical suspension system.

The experimental subjects were six young adult men and women (age range: 20-25) with no known neurological or physiological deficits. To minimize the influence of prior learning on motor performance different subjects were used in each group, and the order of experimental sessions for each subject was randomized. Preliminary studies have shown that learned motor behavior under a given force field may be abolished by immediate exposure to another force field. In addition, that long-term memory of a newly learned motor behavior is formed only if it is not interrupted by other learning pattern for an extended period of time (Shadmehr and Holcomb, 1997). Accordingly, all experimental subjects completed all experimental sessions on the same day.

4.2.2. Motivation and design of the experiment

The design of the experimental protocols was motivated by predictions of the sliding-mode control theory and classical control theory. The experimental protocol was designed as to give evidence for or against the proposed control model. The experimental results were to be compared with the predictions of the proposed control model and of those of a PID controller suggested by other investigators (Darlot et al., 1996). Often, the main reason investigators are eager to embrace PID control is the fact that it can explain experimental data from reaching tasks. However, this is not surprising at all, since reaching movements are like step inputs to a type zero plant (external forces) and a type one compensator (PID). A PID (and any other compensator of type one) can take the steady-state error to a step input to zero. There is biological base for the integrator structure of a PID controller since neural integrators have been found in several instances in the brain (Arnold and Robinson, 1997; Chi-Sang Poon et al., 1999; Shen, 1989). Finally, although a PID controller requires acceleration signals, which has not been verify biologically (through proprioception), it has been theorized that such feedback might be available through vision (Kawato et al., 1987).

A more challenging task would be to achieve good tracking-error performance in the presence of a sinusoidal input. For sinusoidal inputs the steady-state error of a PID controller is not zero, but rather it remains bounded. An important purpose of the experimental protocol is to validate one of the two hypotheses. i. *Human motor control of movement during continuous trajectory tracking uses a control strategy similar to the proposed method*, and ii. *Human motor control utilizes a control scheme similar to a PID controller (constant gain PID)*. Accordingly, the following experimental protocols were used to test specific hypotheses concerning the structure and function of the motor learning system.

Most physical tasks we encounter in everyday life involve moving of objects that are dynamically stable. Examples include human locomotion (moving our own body), pushing a stroller, swinging a racquet, etc. Once set in motion, these objects will usually come to a stop after a while if not further moved. We have been exposed to these kinds of stable objects most of our lives and have learned how to handle them skillfully. Indeed, we ordinarily expect that all physical objects are stable.

The stability margin of a dynamical system is measured by its time constant (or the largest time constant if there are more than one). A system is highly stable if it has a small time constant, and is less stable otherwise. One can readily learn to manipulate a highly stable object easily, but it may take much longer to adapt to a novel, less stable object. For example, a naive subject who has no problem walking around on solid ground may need some training in order to roller-skate. Quite possibly it would take even longer for him/her to learn how to ice-skate. In each instance, the time constant of the moving object is increased with the reduction of friction to movement (i.e. the time constant is proportional to the inertia-friction ratio). This observation leads to the following hypothesis on motor learning: "*The time needed to learn a novel motor task is proportional to the intrinsic time constant of the task*".

The observation above leads to an even more interesting question: "*what would happen if the task not only is less stable, but actually becomes unstable?*" That is to say, that the time constant is not only large, but also becomes negative. (Imagine what it is like when a skater begins to skid on the rink and lose balance, not only is there absolutely no friction to stop the skidding, but there is an active force to pull him/her down further).

However, the human brain seems to be able to do quite well even . For example, consider the motor skill of balancing an inverted pendulum. The system is intrinsically unstable because of unstable equilibrium, but a circus performer can still master the skill after some intense practice. Therefore, our second hypothesis is: "*It takes even longer to learn an unstable motor task than a stable one, though it is still possible to train the system to the new dynamics*".

Finally, although the hypotheses mentioned above are true for both control schemes (proposed model and constant-gain PID), the detailed examination of the experimental data will indicate which of the two control models more accurately describes the behavior of the human CNS. The distinctions in performance between the two competitive control schemes are described in more detailed in parallel with the protocol.

4.2.3. Experimental protocol

Subjects were asked to move the manipulandum on a horizontal *x-y* plane, as described in section 4.2.1. For simplicity and without loss of generality, *the force fields in the x and y directions are assumed to be decoupled* and are given by:

$$f_x = -a_1 x - a_2 \dot{x}$$

$$f_y = -b_1 y - b_2 \dot{y}$$

In the equations above a_1 and b_1 are the translational coefficients and a_2 and b_2 are frictional coefficients. Further, the assumption of decoupled dynamics is not a restriction because a mixed force field can always be decoupled by a suitable choice of coordinate systems.

The subjects had to track a known circular trajectory of radius R with a constant angular velocity ω. A disturbance force was imposed by the manipulandum during the tracking. The disturbance had distinct characteristics in the x and y directions (see Table 4.1). Each session had a distinct force field and it lasted 2.5 minutes (150 sec) for the stable force fields and 5 minutes (300 sec) for the unstable force fields. The lengths of the sessions were chosen to be several times longer than the time constants of the slowest dynamics. This would allow the error dynamics to reach an asymptotic behavior. The sampling speed was 10 Hz. which is much higher than twice the maximum frequency that the humans can generate (0.3 Hz) (Hanneton et al., 1997). Finally, the time constants and the gains of the disturbance dynamics were chosen such that $F_{\max} \leq 30\,\mathrm{N}$ at all times. The 30N limit was imposed by the mechanical capabilities of the motors of the manipulandum.

$\Big(X = R\cos(\omega t) \Rightarrow V_x = -\omega R\sin(\omega t) \Rightarrow F_D = a_1\cos(\omega t) - K\sin(\omega t)$, where $K = \omega\frac{a_2}{a_1}$. Then,

$$F_D = \frac{a_1 R}{\cos(\arctan(K))}\cos(\omega t + \arctan(K)), \text{ and } F_{\max} = \frac{a_1 R}{\cos\left(\arctan\left(\frac{\omega a_2}{a_1}\right)\right)}\Bigg).$$

The higher the ratio a_2/a_1 (time constant) and the higher the angular velocity ω, the higher $F_{\max}$ will be.

When the external force of the manipulator act on the arm the arm dynamics become insignificant. This is an accurate assumption since the end point of the manipulandum is moving at a very small constant angular trajectory and its mass and inertia are also very small. Therefore, the time constants of the external disturbances become the time constants of the system ($\tau_x = a_2/a_1$ and $\tau_y = b_2/b_1$). The arm and disturbance block of Figure 4.5 can then be transformed into the linear blocks of Figure 4.6. Notice that there is an inversion as well as a sign change to the transfer function when replacing the arm disturbance combo with the linear block of the disturbances. The inversion is due to the fact that, after the replacement of the dynamic blocks, the new block has to reflect the transfer functions from F to x and y, which are the inverses of the transfer functions from x and y to F. The sign change reflects Newton's 3rd law, or in other words, the fact that we are trying to capture the effect of the forces of the arm on the manipulandum, which are the exact opposite of the manipulandum forces on the arm. Newton's law holds since, as mentioned earlier, the system has very small velocities, and even smaller accelerations, and it is, therefore, for all practical purposes inertial.

Subjects performed visually guided tracking movements in nine experimental sessions with varying time constants and gains as shown in Table 4.1. The tasks were performed in two uninterrupted larger sessions of about 20 minutes each. Each larger session consisted of several sessions that had distinct external force fields. The changes in the external dynamics were done while tracking and without any interruption of the task. The time constants were altered by changing the translational and/or frictional coefficients.

In each experimental session, the subjects were trained to adapt to the force field by continuously moving the manipulandum to track a rotating visual target projected on the computer screen. The tracking error was continuously recorded and stored in the computer.

Force Field : Stable, +ve gain, Static			Stable,	Force Field : +ve gain,	Dynamic
Session 1	Session 2	Session 3	Session 4	Session 5	Session 6
$f_x = 10\,\mathrm{N}$ $\omega = 1\,\mathrm{rad/s}$	$f_x = 0\,\mathrm{N}$ $\omega = 1\,\mathrm{rad/s}$	$f_x = 20\,\mathrm{N}$ $\omega = 1\,\mathrm{rad/s}$	$f_x = -27\dot{x} - 81x$ $\tau_x = 1/3\,\mathrm{s}$ $\omega = 1\,\mathrm{rad/s}$	$f_x = -54\dot{x} - 162x$ $\tau_x = 1/3\,\mathrm{s}$ $\omega = 1\,\mathrm{rad/s}$	$f_x = -54\dot{x} - 162x$ $\tau_x = 1/3\,\mathrm{s}$ $\omega = 0.7\,\mathrm{rad/s}$
$f_y = 0\,\mathrm{N}$ $\omega = 1\,\mathrm{rad/s}$	$f_y = 10\,\mathrm{N}$ $\omega = 1\,\mathrm{rad/s}$	$f_y = 0\,\mathrm{N}$ $\omega = 1\,\mathrm{rad/s}$	$f_y = -10\dot{x} - 30x$ $\tau_y = 1/3\,\mathrm{s}$ $\omega = 1\,\mathrm{rad/s}$	$f_y = -7.5\dot{x} - 7.5x$ $\tau_y = 1\,\mathrm{s}$ $\omega = 1\,\mathrm{rad/s}$	$f_y = -7.5\dot{x} - 7.5x$ $\tau_y = 1\,\mathrm{s}$ $\omega = 0.7\,\mathrm{rad/s}$

Force Field : Unstable, +ve gain, Dynamic		Force Field : Stable, -ve gain, Dynamic
Session 7	Session 8	Session 9
$f_x = -27\dot{x} + 81x$ $\tau_x = -1/3\,\mathrm{s}$ $\omega = 1\,\mathrm{rad/s}$	$f_x = -54\dot{x} + 162x$ $\tau_x = -1/3\,\mathrm{s}$ $\omega = 1\,\mathrm{rad/s}$	$f_x = 27\dot{x} + 81x$ $\tau_x = 1/3\,\mathrm{s}$ $\omega = 1\,\mathrm{rad/s}$
$f_y = -10\dot{x} + 30x$ $\tau_y = -1/3\,\mathrm{s}$ $\omega = 1\,\mathrm{rad/s}$	$f_y = -7.5\dot{x} + 7.5x$ $\tau_y = -1\,\mathrm{s}$ $\omega = 1\,\mathrm{rad/s}$	$f_y = 10\dot{x} + 30x$ $\tau_y = 1/3\,\mathrm{s}$ $\omega = 1\,\mathrm{rad/s}$

Table 4.1 Characteristic time constants of the external force field induced by the manipulandum.

4.2.4. Data Analysis

In each experimental session, the subjects were trained to adapt to the force field by continuously moving the manipulandum to track a rotating visual target projected on the computer screen. The actual trajectory was continuously recorded and was stored on the computer. The time needed by the subjects to learn to track the target to within a given tolerance level was noted during a test run. The lengths of the rest of the experimental sessions were based on the results of the test run.

The decoupling of the x and y force vectors makes it convenient to study the effects of changing system time constants on learning speed. Let $e_x(t)$ and $e_y(t)$ be the tracking errors in the x and y directions at any time t, and $\|e_x(t)\|_n$ and $\|e_y(t)\|_n$ be the corresponding magnitudes averaged over the n^{th} revolution (where $\|f\|_n = \frac{1}{2\pi}\int f^2 dt \approx \frac{1}{2\pi}\sum_i f_i^2 0.1 = \frac{0.1}{2\pi}\sum_i f_i^2$, which is the mean of f^2 during the n^{th} revolution of the circular movement). With $\tau_x = \tau_y$, the subjects should be able to track equally well in the x and y directions so that one would expect that the statistics of $e_x(t)$ and $e_y(t)$ would be the same for all t. With $\tau_x \neq \tau_y$, there would be a divergence in the learning speed in the x and y directions.

If $\|e_x(t)\|$ and $\|e_y(t)\|$ at steady state remain the same for dynamics with the same time constants but different disturbance gains, it will indicate that the motor system uses an adaptive mechanism much like the proposed method instead of a PID control scheme. To understand this, consider that the desired trajectory in one of the directions (say x) is $A\sin(t)$. Then the output of the PID controlled system that is tracking the trajectory very closely would be $A\sin(t+\phi_1)$ for a control gain K_1. Let us also assume that the ratio of gains in the disturbance is $a_1/a_2 = 3$ (picked from the experimental protocols).

PID controller + System: $K_1 \frac{(s+z_1)(s+z_2)}{s} \frac{1/a_2}{(s+a_1/a_2)}$

Then,

Closed loop transfer function: $G_{cl}(s)=\dfrac{(s+z_1)(s+z_2)\dfrac{K_1}{a_2}}{s(s+a_1/a_2)+\dfrac{K_1}{a_2}(s+z_1)(s+z_2)}$

The magnitude of the closed loop transfer function at the tracking frequency of $\omega = 1$ rad/s is:

$$\|G_{cl}(s)\| = \left|\frac{K_1}{a_2}\right| \frac{\sqrt{(1+z_1^2)(1+z_2^2)}}{\sqrt{\left(3+\dfrac{K_1}{a_2}(z_1+z_2)^2\right)^2+\left(\dfrac{K_1}{a_2}z_1z_2-\dfrac{K_1}{a_2}-1\right)^2}}$$

If the amplitude of the actual trajectory is the same as the desired (i.e. good tracking performance) then, $\|G_{cl}(s)\| \approx 1$. Assuming a constant gain PID controller $\|G_{cl}(s)\|$ is a function of the controller's zeros (z_1 and z_2) and the gain $\left|\dfrac{K_1}{a_2}\right|$. Figure 4.9 shows the dependence of $\|G_{cl}(s)\|$ with respect to the PID controller's open-loop zeros. The plot shows the dependence for three distinct open-loop gains ($\left|\dfrac{K_1}{a_2}\right|$). It is true that every triplet $(\|G\|,\ z_1,\ z_2)$ is one-to-one with a gain $\left|\dfrac{K_1}{a_2}\right|$. However, for high gains, which are needed to bring $\|G_{cl}(s)\| \approx 1$, the surfaces of Figure 4-9 come very close together even if the gains are increased by five times. This means that trying to distinguish whether the motor controller is a constant-gain PID controller based on the changes of the closed loop gain as a function of the changes to the open loop gain would be very difficult especially if the data are noisy.

4.2.5. Results and Discussion

The presentation of the raw experimental data is given Figure 4.10. In this figure are shown the plotted error data (mean squared errors per circular trajectory) for six subjects in all nine protocols. Notice that the data are extremely noisy and they have quite large variances, which makes impossible to locate these point on a figure like the one shown in Figure 4.9.

What follows is a pooling of the data from all subjects. Figure 4.11 shows the mean and standard error lines for all x and y data shown in Figure 4.10. Notice that the pooled data from protocols one through eight behave in a similar manner. Therefore the data of Figure 4.11 were further pooled to combine all eight protocols (separately for the x and y errors). The results are shown in Figure 4.12 (mean and standard error lines). Finally, the data of Figure 4.12 were pooled together and the result is shown in Figure 4.13 (mean and standard deviation lines).

To prove the hypothesis that the proposed control model gives a better description of the motor control system than a PID controller would, it is enough to look at the data from protocol 9 (stable dynamics with a negative gain). As discussed earlier, no constant-gain PID controller that is able to control the dynamics of protocols 1-8 (positive gain controller) would have been able to stabilize negative gain dynamics. However, the data show clearly that subjects are able to control the dynamics and keep the error to within a finite range (boundary layer). What is even more convincing is the fact that one can see the effect of the adaptation in the gradual reduction of the tracking error. Because the adaptation and feedback gains vary from subject to subject, Figure 4.14 shows the size of the x-error separately for all subjects. A constant-gain PID could not possibly explain the adaptive correction in the tracking error. The same figure also includes estimates of the adaptation time constant, which are estimated to be $\frac{1}{4}$ the time it takes to reach a steady state tracking error performance.

Simulations of the behavior of the proposed model and the PID controller during a negative-gain disturbance are shown in Figures 4.15 and 4.16. Again we see that the proposed controller achieves bounded asymptotic error while improving its performance over time (reducing the error).

In general, protocols 1-8 were easier for the subjects. This was true especially for the stable force fields (protocols 1-6). In those cases the subjects reached steady-state performance levels almost immediately after the initiation of the force field. If there were any further adaptation it was so small that was lost in the noise. The unstable dynamics (protocols 7 and 8) are definitely controllable. They are also the only other set that a small adaptation can be seen, since subjects need a little more time to learn the force field dynamics.

Closing, it is important to remember that the fact that the disturbances are linear makes it easier to predict the behavior of this system under various control strategies. This is of particular interest for comparing the proposed control method with a PID controller. This is especially true for the very last session in the protocol, which addresses the capability of the human CNS to control (and learn) a stable, negative-gain force field, after it has successfully controlled stable and unstable, positive-gain force fields. If the CNS is successful in this task, it rules out completely the constant-gain-PID hypothesis, since a PID controller could not have stabilized a negative-gain system. Figure 4.7 shows the root locus of a generic unstable system controlled by a PID controller indicating that PID controllers can stabilize unstable (positive-gain) systems. However, in Figure 4.8 is shown that while positive-gain, stable systems are stable at any gain of the PID controller, negative-gain, stable (and unstable) plants are impossible to stabilize at any gain. It is true that a simple logical operator could have switched the sign of the gain of the PID controller, however, such a change represents a form of adaptation that would involve recognizing the sign of the disturbance at each instance.

The experiments showed that it takes very short time to learn both stable and unstable positive-gain motor tasks. The negative gain disturbances, though admittedly the most difficult to control, are still controllable. What comes as the greatest surprise in this study of continuous control of movement is that human subjects, astronauts and on the ground alike, adapt to the new dynamics within a matter of minutes.

While adaptation to new external dynamics is fast, there also seems to be happening simultaneously with an "adaptation" to the execution of the motor task. The following chapter will look into the effect of neuro-adaptive optimization during the throwing task. After seeing the fast adaptation rates during trajectory tracking, it comes to no surprise anymore that the results of the throwing task showed no detectable adaptation. Nevertheless, the next chapter explores ways of how an optimization model can lead to an "adaptation". Not by changing gains but rather by adopting a totally different throwing strategy.

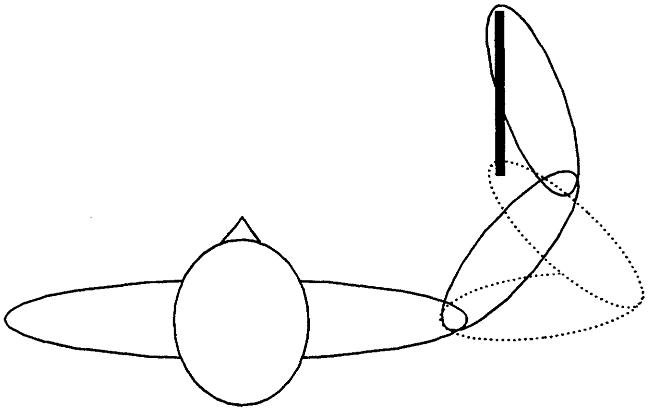


Figure 4.1 A graphical representation of the pointing task in the rotational room (Top View). The desired trajectory is assumed to be a straight line, which is continuously tracked. The dash-line representation of the arm corresponds to the starting position of the subject's arm, while the continuous line describes the final state. The Coriolis forces are in effect throughout the movement.

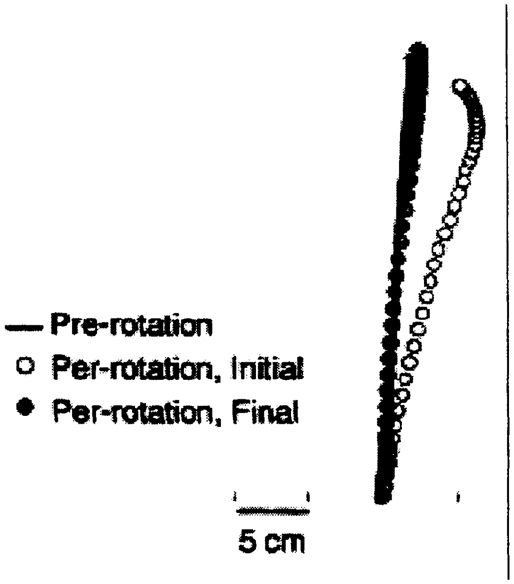


Figure 4.2 Experimental data indicating the adaptation of subjects to Coriolis forces (Adopted from Dizio and Lackner, 1995). Notice that when fully adapted all subjects move on a straight line. This is an indication that their desired (maybe even optimal) trajectory is a straight line trajectory. In addition, the important feature of the first reach during rotation is the size of the curvature (i.e., deviation from the straight line path connecting the initial and final points). This curvature slowly disappears as the subjects adapt to the Coriolis environment.

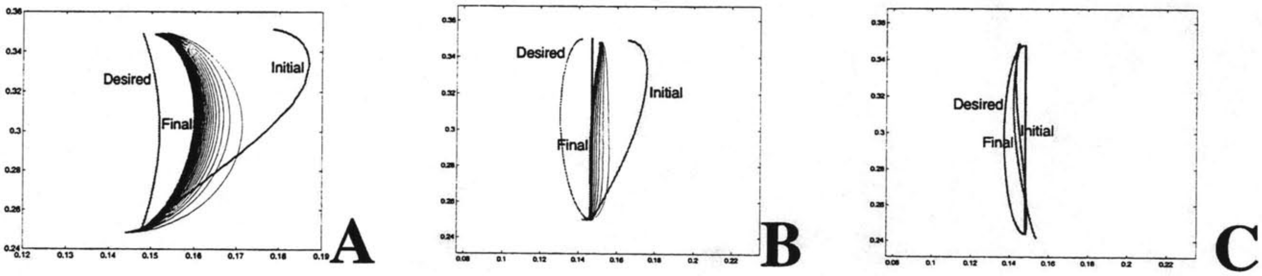


Figure 4.3 Plot of the results of the simulations of a two-link armature moving in the horizontal plane, while the environment is rotating (i.e., Coriolis forces are present). The proposed model shows adaptation and improvement in the performance over time. Notice the reduction in the curvature of the actual path between the initial and the final trials. Frame B shows the proposed controller performing when the desired trajectory opposes the Coriolis forces. The performance of the PID controller for the same desired trajectory as in frame B is shown in frame C.

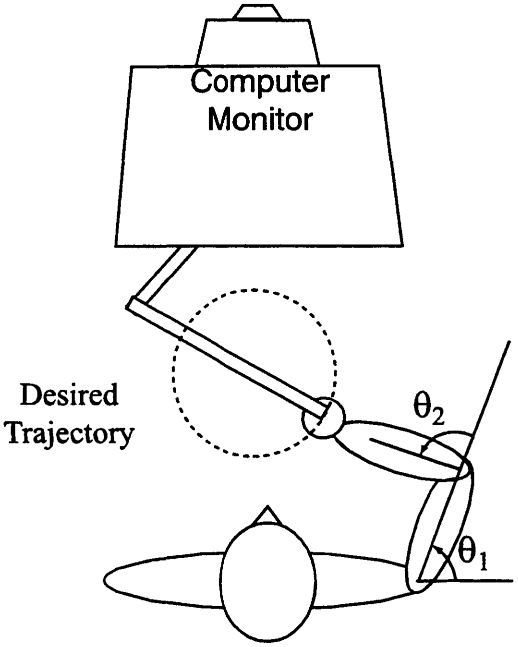


Figure 4.4 A graphical representation of the continuous trajectory tracking experiment using the robotic manipulandum. The subject is sitting in front of a computer monitor holding the manipulandum. The desired trajectory appears as a moving cursor on the monitor. The cursor that indicates the desired trajectory is tracing a circle of constant radius and is moving at a constant angular velocity. The location of the manipulandum also appears on the monitor as a cursor of different color.

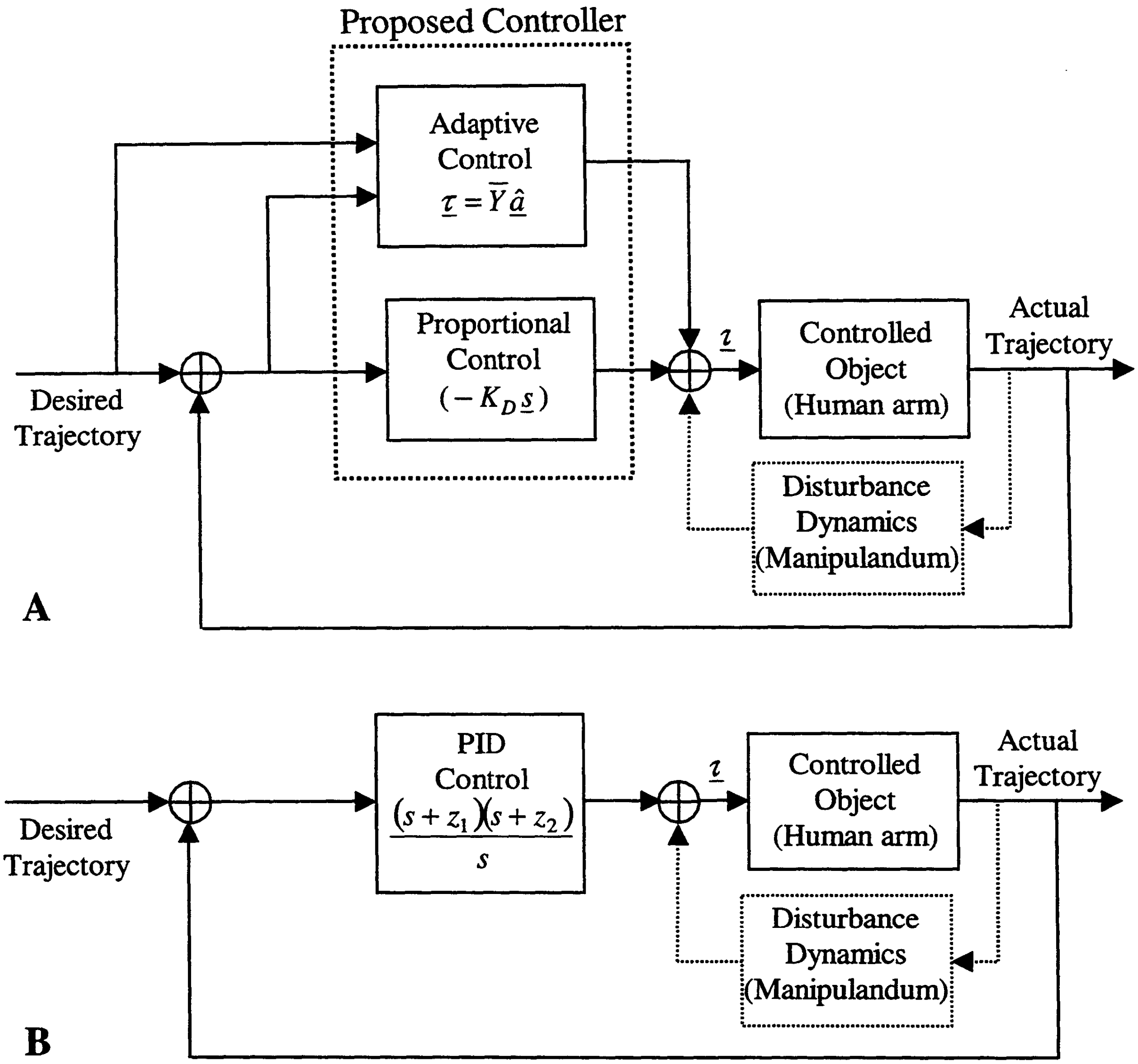

Proposed Controller
Adaptive Control
$\underline{\tau} = \overline{Y}\underline{\hat{a}}$
Proportional Control
$(-K_D\underline{s})$
$\underline{\tau}$
Controlled Object (Human arm)
Actual Trajectory
Desired Trajectory
Disturbance Dynamics (Manipulandum)
A
PID Control
$\frac{(s+z_1)(s+z_2)}{s}$
$\underline{\tau}$
Controlled Object (Human arm)
Actual Trajectory
Desired Trajectory
Disturbance Dynamics (Manipulandum)
B

Figure 4.5 Block diagram representations of two competitive control models. They include the arm dynamics and the manipulator external dynamics. A) The proposed control model consists of two parts: the proportional feedback part and the adaptive part (as described in Chapter 3).

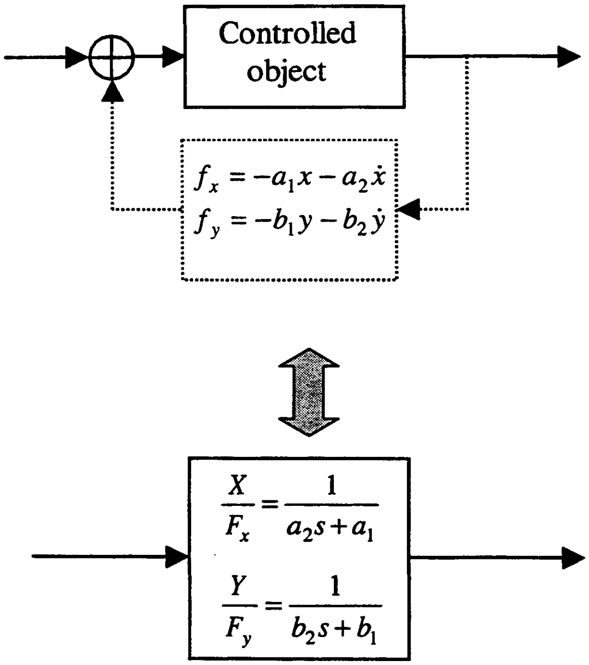

Controlled object
$f_x = -a_1 x - a_2 \dot{x}$
$f_y = -b_1 y - b_2 \dot{y}$
$\frac{X}{F_x} = \frac{1}{a_2 s + a_1}$
$\frac{Y}{F_y} = \frac{1}{b_2 s + b_1}$

Figure 4.6 Two equivalent block representations. Under the influence of a strong external force field the combined dynamics of the arm and the disturbance can be solely described by the dynamics of the disturbance.

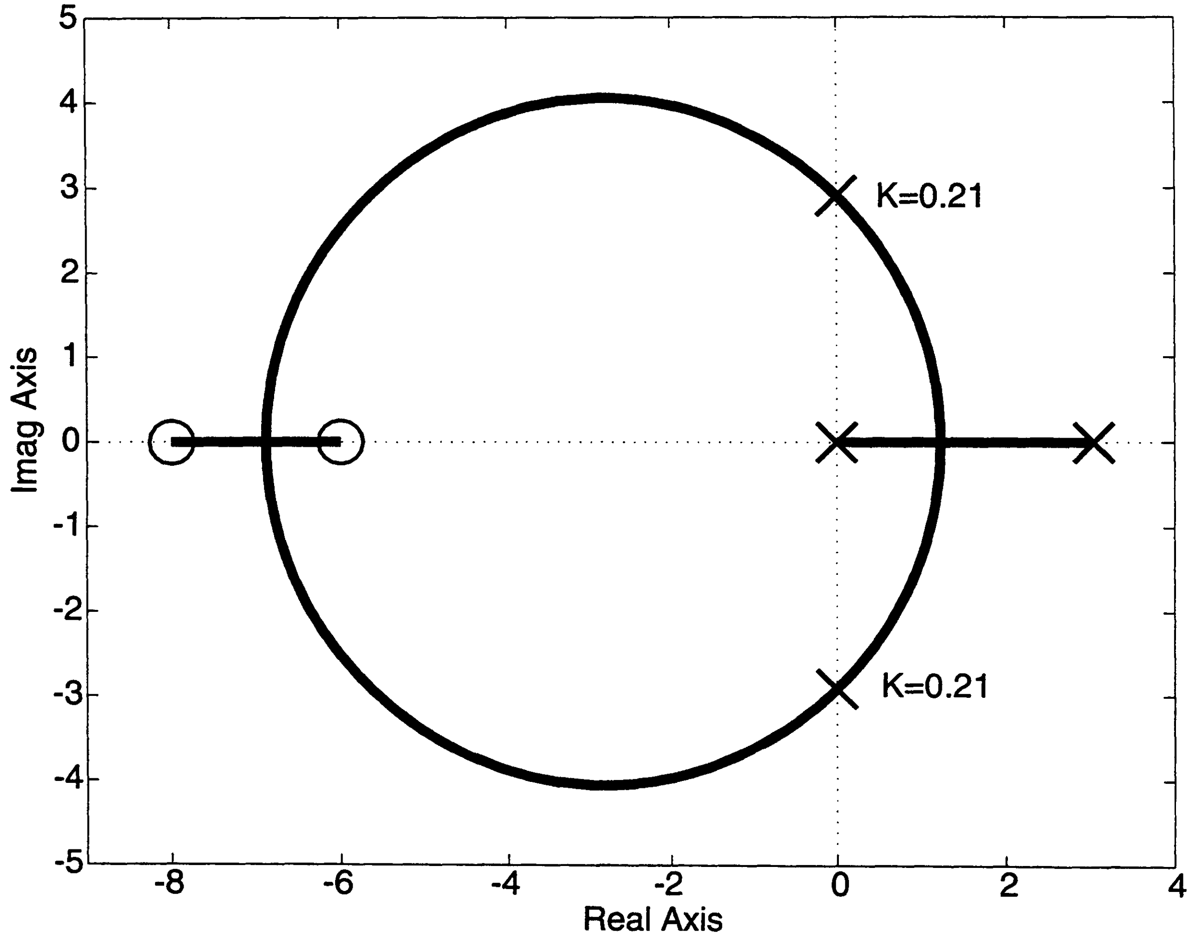


Figure 4.7 Root locus of a PID controller and an open-loop unstable plant (with positive gains).

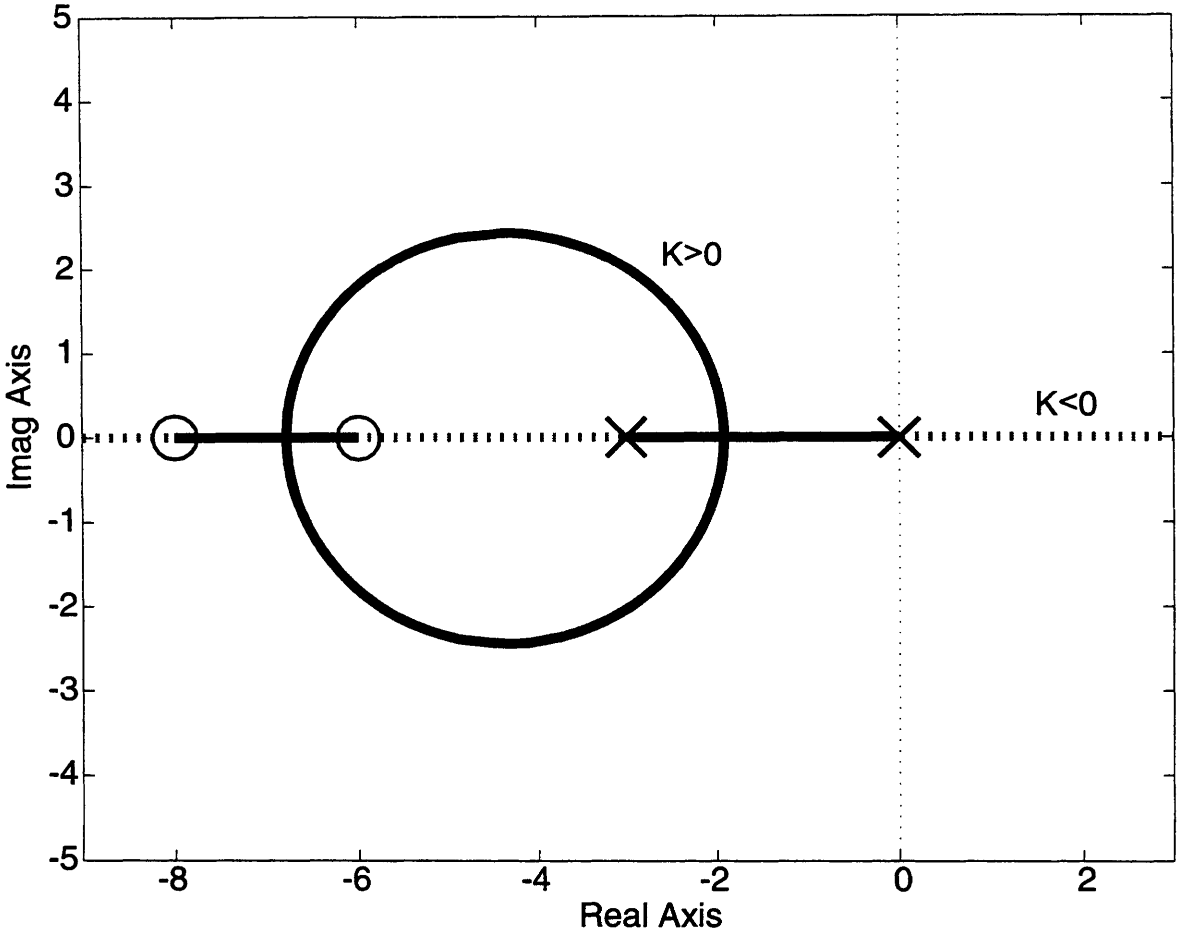


Figure 4.8 Root locus of a PID controller and an open-loop stable plant (with both positive and negative gains).

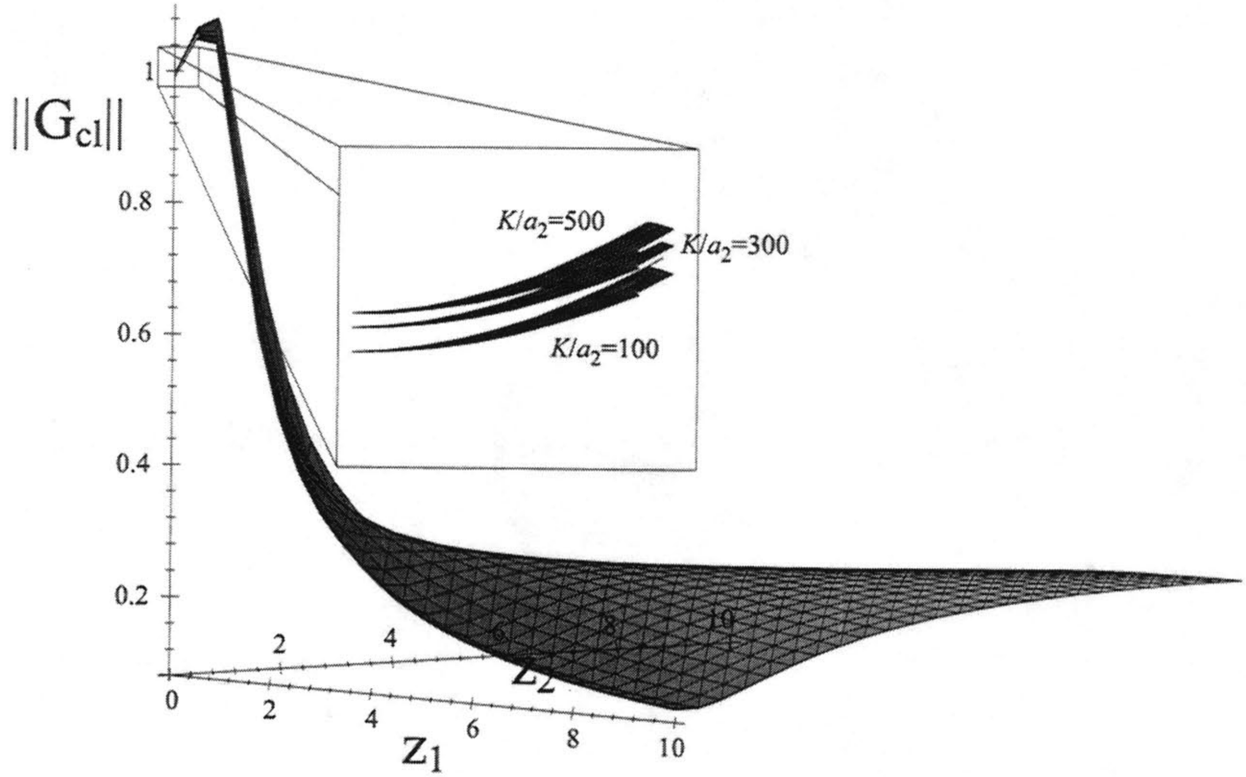


Figure 4.9 Plot demonstrating the dependence of the closed-loop gain of the motor system as a function of the PID controller's open-loop zeros. The surfaces correspond to distinct open-loop gains.

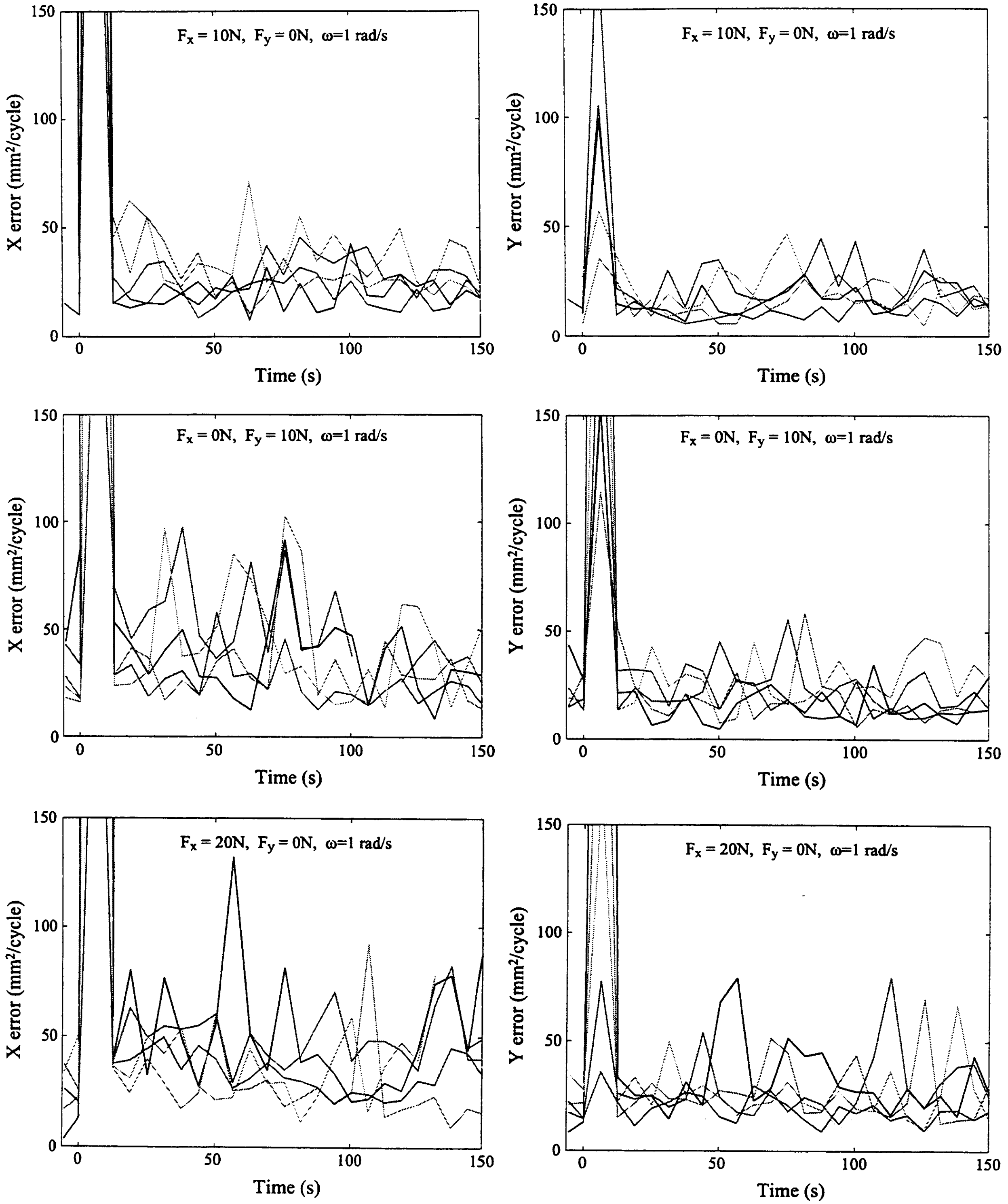


Figure 4.10 The plots include the integrated values of X and Y tracking error per cycle of the desired trajectory versus time. The data is from 6 subjects. Protocols 1 through 3 are shown here. The rest of the data is shown in Appendix D.

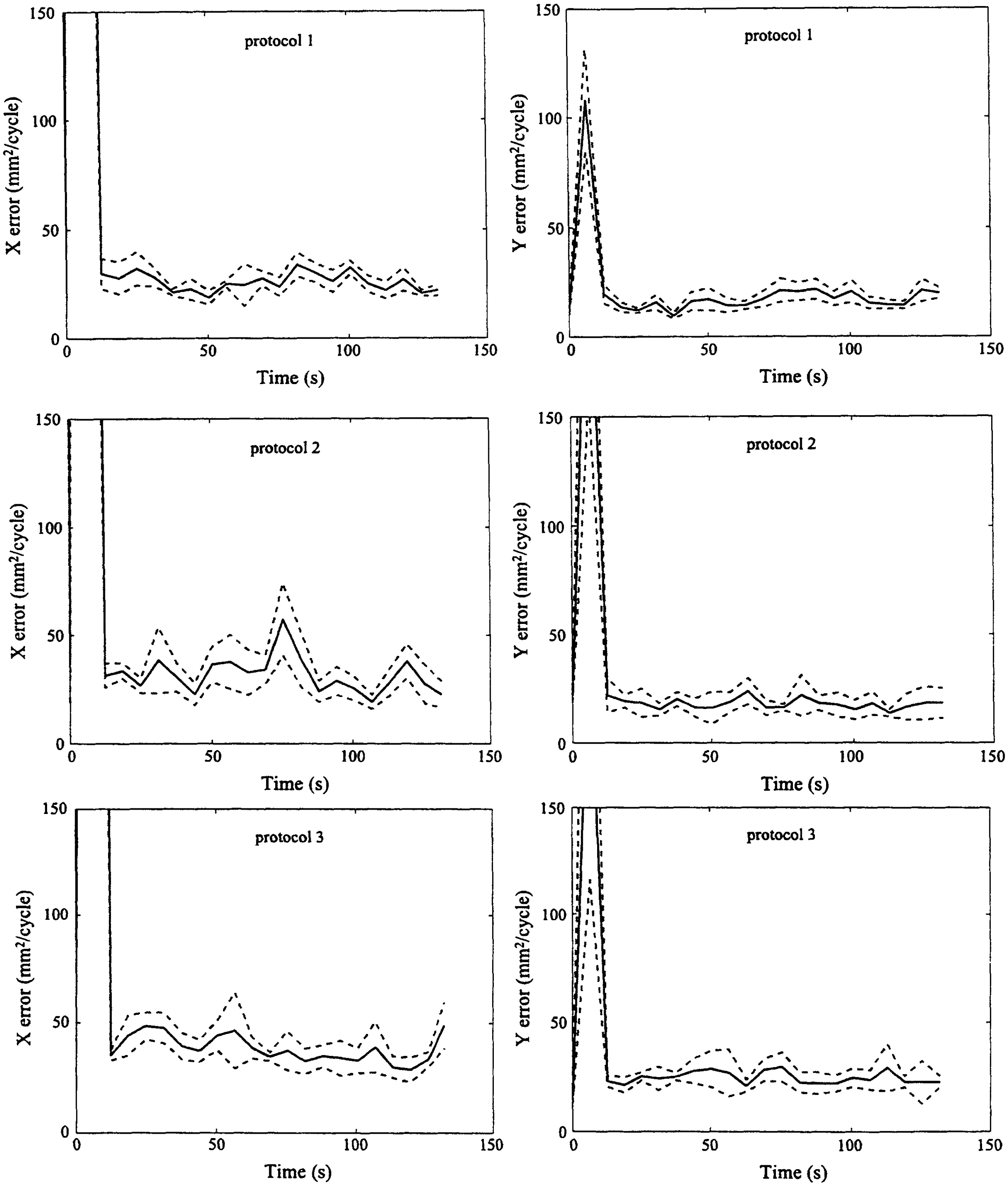
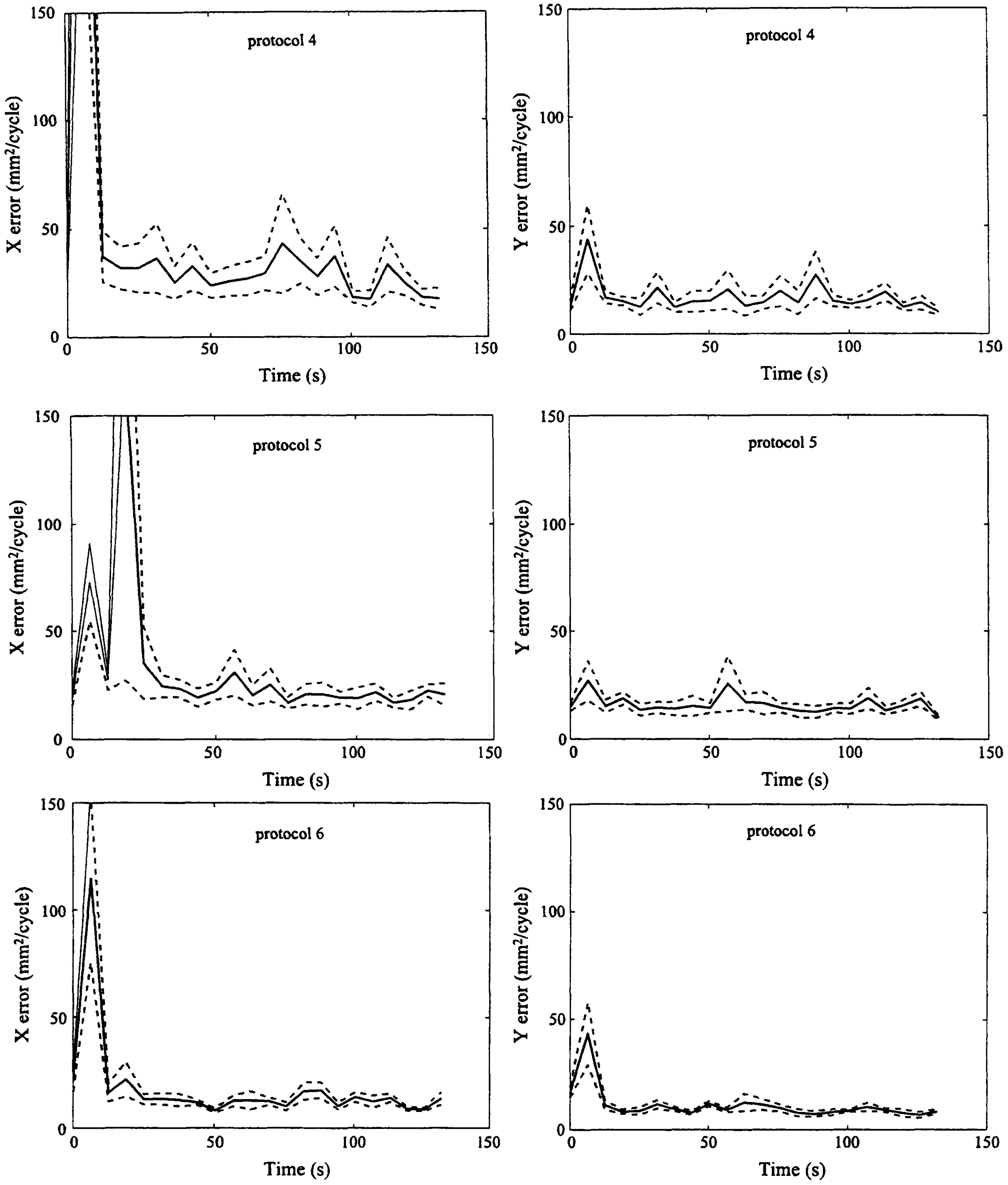
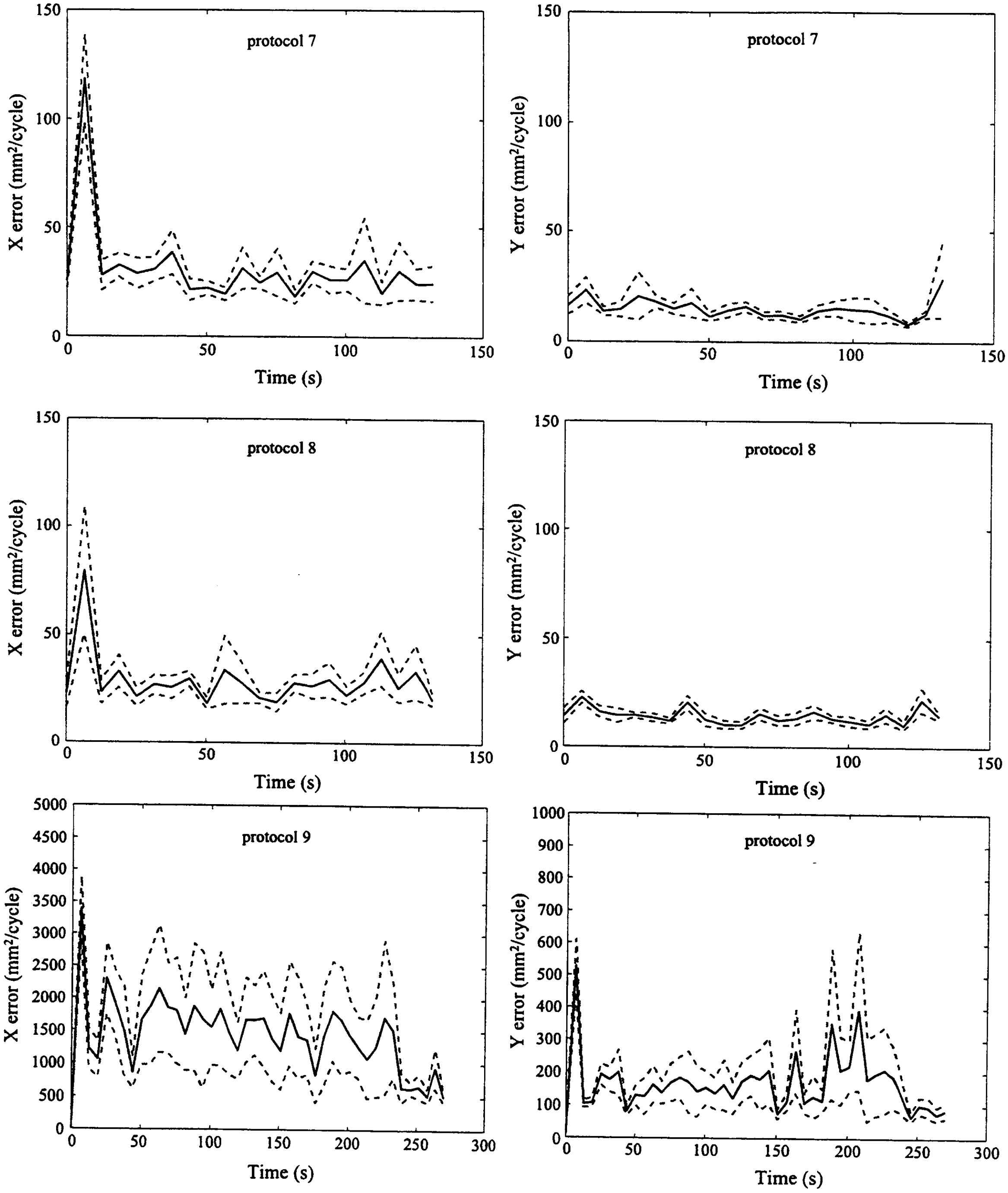


Figure 4.11 The plots include the integrated values of X and Y tracking error per cycle of the desired trajectory versus time (mean ± S.E.). The data is from 6 subjects. Protocols 1 through 3 are shown here.



Protocols 4 through 6 are shown here.



Protocols 7 through 9 are shown here.

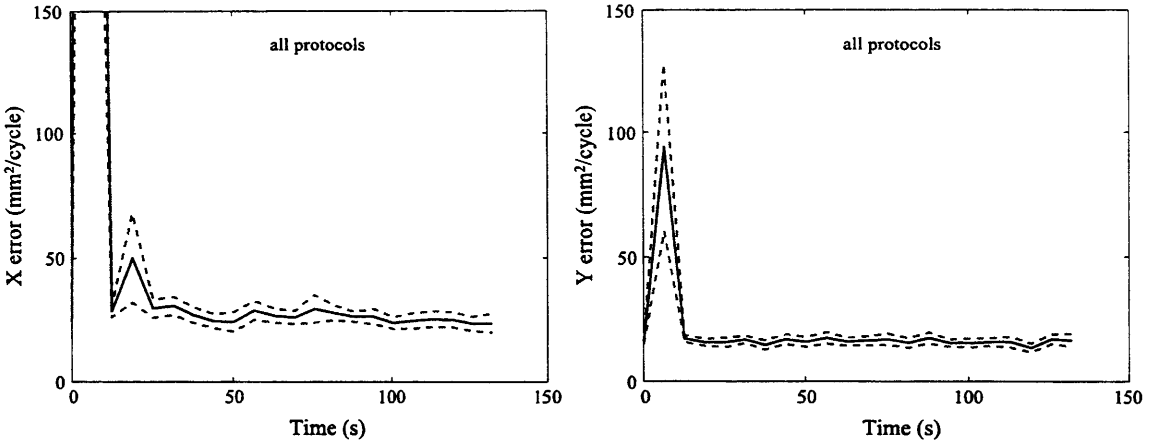


Figure 4.12 The plots include the integrated values of X and Y tracking error per cycle of the desired trajectory versus time (mean ± S.E.). The data is from 6 subjects and protocols 1-8.

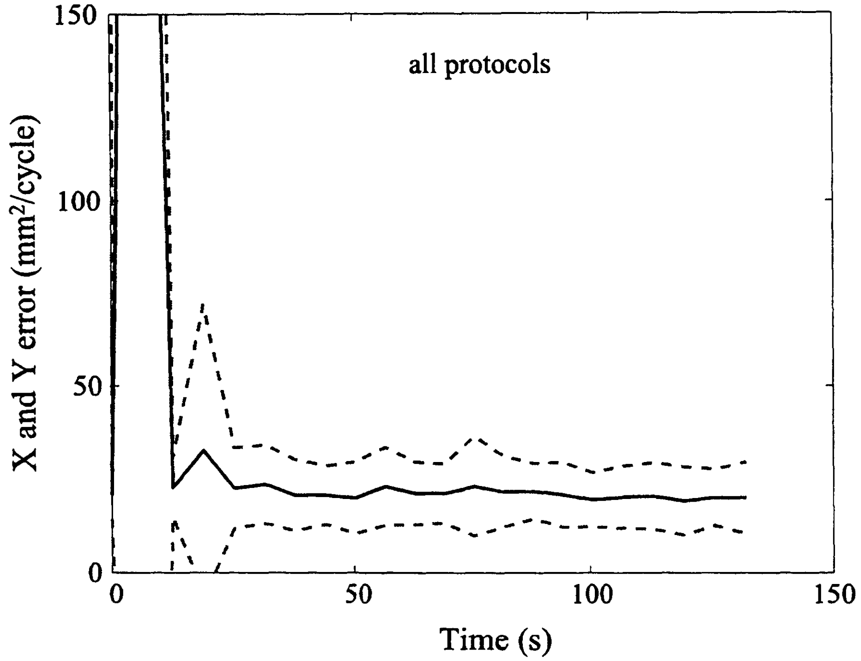


Figure 4.13 The plots include the integrated values of X and Y tracking error per cycle of the desired trajectory versus time (mean ± Std. Div.). The data is from 6 subjects and protocols 1-8.

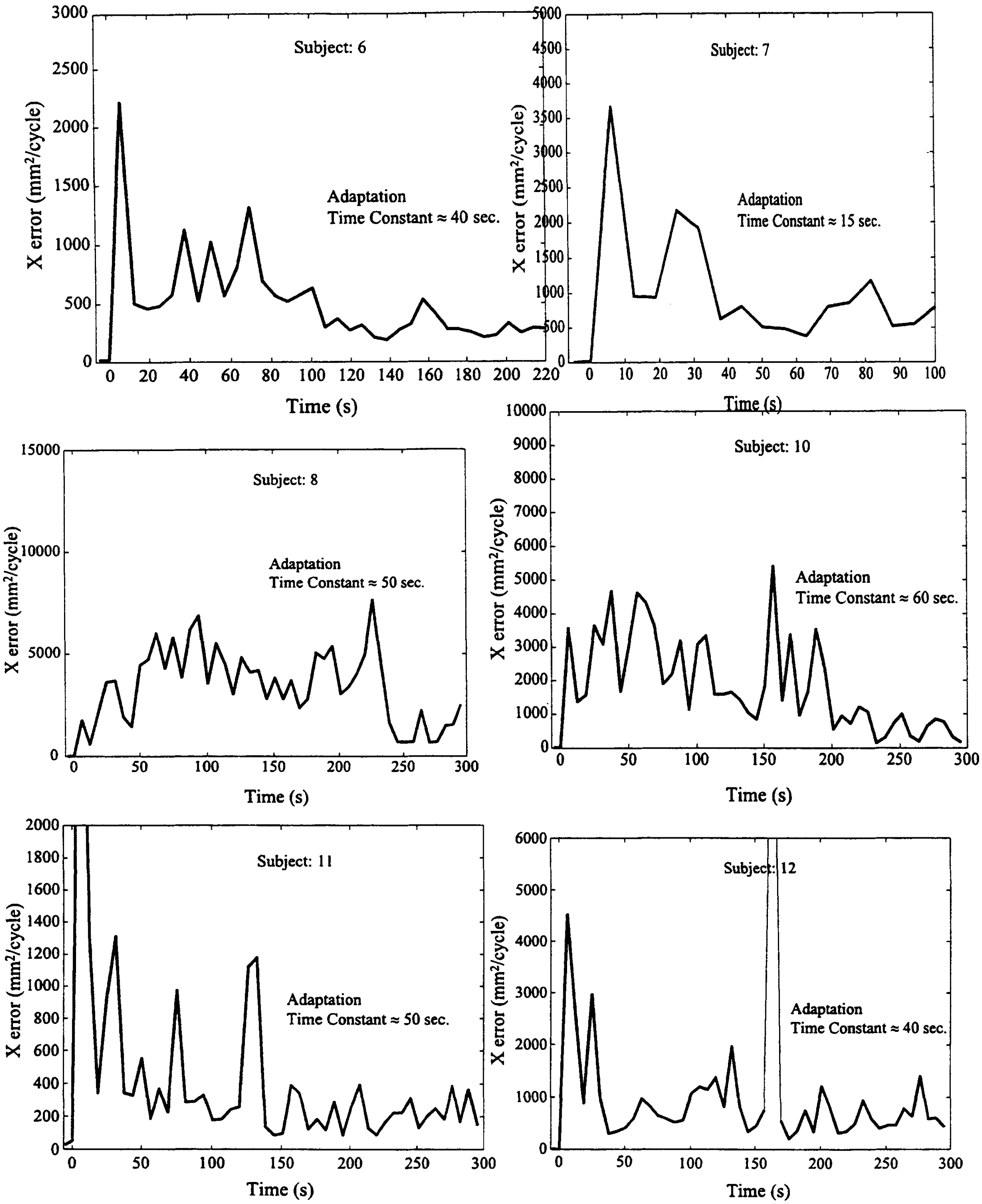


Figure 4.14 X and Y tracking error data versus time from all subjects. Protocol 8 is shown here. These data were the only other data in which the adaptation is visible. For the rest of the protocols the adaptation was almost immediate. We define the adaptation constant to be ¼ of the time to reach steady state.

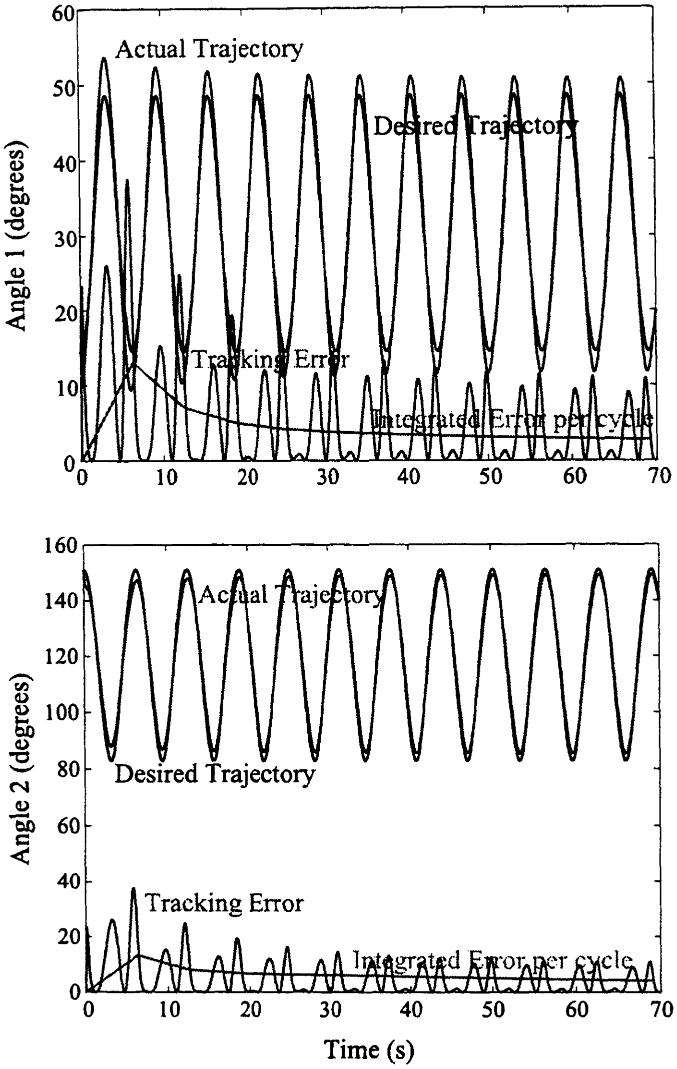


Figure 4.15 The two-link arm model performance is shown here. The desired trajectory, the actual trajectory, the square of the error, and the mean square of the error (per cycle) are plotted versus time. The two plots represent the two joint angles (shoulder-1 and elbow-2). The proposed control method was used to control the negative gain dynamics (similar to protocol 9). Clearly the system is stable. The asymptotic error reaches a non-zero steady-state level.

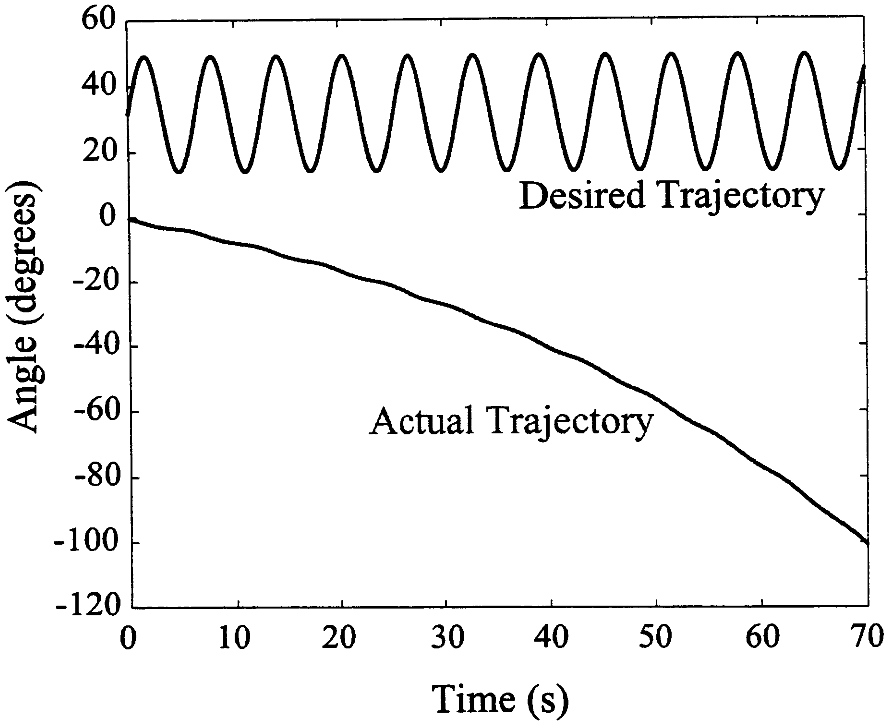


Figure 4.16 The two-link arm model performance is shown here. The desired trajectory, the actual trajectory are plotted versus time. A PID controller was used to control the negative gain dynamics (similar to protocol 9). Clearly the system is unstable.

5. An adaptation study on the performance of subjects during throwing

In this chapter I will describe the throwing experiment study, which was performed onboard the space station Mir. Although this study came before the continuous tracking experiment, its results can be understood better when looked after the results of Chapter 4. There is no direct tie of this study with the proposed control model. In fact, as we will see in the first part of the chapter there seems to be hardly any connection with adaptation altogether. However, its results suggest a different kind of adaptation. Not so much in the performance level, but rather in the throwing "strategy" that subjects employ when exposed to different gravity conditions. The details of this study are presented in the second part of this chapter.

5.1. Throwing Task Experiment in microgravity and 1G

Several cosmonauts and astronauts have participated in a dart throwing task onboard the Russian space station Mir. The astronauts had been exposed to the microgravity environment for at least two weeks before they performed the prescribed throwing task. The goal of this experiment was to determine whether after some exposure to microgravity the astronauts have adapted their throwing strategy to the new gravitational dynamics, even though they have not explicitly performed the prescribed throwing task.

5.1.1. Experimental setup

The throwing task took place in two separate locations. Ground throwing sets were performed in the Man Vehicle Laboratory at MIT for the purpose of setting a comparison level of performance. In addition, cosmonauts and astronauts onboard the space station Mir, performed a series of similar throwing tasks to study the human adaptation in microgravity.

Seven astronauts and cosmonauts took part in this experiment. The subjects threw a ball (approximately 2.5 cm in diameter) to a target at approximately 1.5m away. Accuracy in both the x and the y directions was equally important. First, the subjects *closed* their *eyes*, and they kept them closed. Then, using the *right arm*, they threw the ball towards the target. They opened their eyes to see how they performed. Another member of the crew recorded the x- and y- errors. Each subject repeated the throw 5 times using his/her right arm and with eyes closed, and 5 more times using his/her left arm but now with eyes open. Each subject repeated the process using his/her *left* arm. The left-arm throws were performed immediately after the right-arm throws had been completed.

In addition to the seven astronauts, eight subjects performed a similar protocol on the ground for comparison purposes. The ground subjects' performance was measured to establish a baseline indicating full adaptation to system dynamics. This baseline was meant to be compared to the performance of the Mir crew. The ground protocol was slightly different. The number of throws was the same as well as the number arm/eye combinations. However, due to time constraints onboard the Mir station, the order of execution of the throwing sets was modified from that which was used on the ground. In space, the subjects were limited in the location in which they could have the camera for video-recording the task. Therefore, for the sake of saving time, they chose to finish all the throws with the right arm before they turn 180° to repeat the throws using their left arms. In addition, on the ground the distance from the foot of the subject to the target was exactly 2m, while it was slightly less on the station because of space limitations.

Consider using 'R' and 'L' for right- and left-arm throws correspondingly, and 'C' and 'O' for denoting the eyes-closed and eyes-open conditions. The ground tests were run in the following order: R/C, L/C, R/O, L/O.

5.1.2. Data Error Analysis

The *x*- and *y*- error data were analyzed thoroughly in an attempt to find any trends that would indicate degradation of the targeting ability of the Mir crew during their first sets of throws as compared to the ground subjects. Assuming that both groups of subjects had the same level of targeting ability, one might expect that the subjects exposed to the new dynamics of zero-gravity would perform significantly worse than the ground subjects, especially in the absence of any visual feedback (Watt et al., 1985; Watt et al., 1986). However, the Mir crew had been onboard the space station for a minimum of two weeks before they performed the throwing task. In fact, some cosmonauts had been onboard for over three months prior to completing the task. Figures 4.4 and 4.5 show the *x*- and *y*- error data from several throwing sessions. Notice that the size of the errors does not always decrease as the trial number increases. In fact, when comparing sessions from different days, the performance does not improve by any detectable amount at later days either. Further comparison with the performance of the fully adapted ground subjects indicates that the performance of the space subjects is not in any way worse than the ground subject, indicating further that the adaptation to microgravity was complete by the time the experiment took place.

Looking at isolated sessions in Figure 5.2 it might be tempting to say that some adaptation occurs in the space data. However, such trends are not consistent across all arm/eye conditions. Besides, similar trends appear in the ground data for completely different arm/eye conditions. Further, it is clear that any apparent change in the mean value of the error across eye/hand conditions is very small when compared to the variance of the error during the same throws.

To assure that no evidence of adaptation was missed data was pooled together across subjects. Figures 5.3 and 5.4 have the data from all subjects pooled together (mean ± standard error (SE)) for various eye/arm conditions. Notice that although the mean values seem to indicate an improvement in the performance as the trial number increases, the variances are large enough to make any such trend to be statistically insignificant. The same was true when the data from all subjects and all eye/arm conditions were pooled together (Figure 5.5). The data from several sessions was pooled together and the error of the k+1 throw was plotted versus the error of the k throw. This plot is shown in Figure 5.6. In the event that there was a proportional improvement in the error, the data would have been arranged in form that looks like an ellipse whose major axis has negative slope greater than -1. Instead, the data is arranged in a circular pattern indicating no adaptation (geometric reduction).

Summarizing the data analysis of the throwing task, it is quite clear that the subjects were already adapted to the new gravitational conditions by the time the experiment took place. No further adaptation was detected on either the short (5 trials) or the long (several days apart) time scales. These results are in agreement with the results in the continuous tracking experiment that indicated that subjects adapt to the new system dynamics shortly after they are exposed to it (less than 5 minutes).

5.2. Neuro-adaptive Optimization of Visuomotor Performance

It is well-known that humans are capable of adapting their visuomotor behavior to distortions in the visual field (Dizio and Lackner, 1995; Martin et al., 1996; Thach et al., 1992) as well as in the force field (Dizio and Lackner, 1995; Shadmehr and Mussa-Ivaldi, 1994). Similarly, subjects learn to adjust the production of vowel sounds to compensate for feedback alterations in the auditory sensory signal (Houde and Jordan, 1998). The induction of certain 'internal models' by sensory feedbacks in the brain appears to play a central role in such motor adaptations to changes in the environment (Ito, 1984; Jordan et al., 1994; Kawato et al., 1988; Miall and Wolpert, 1996; Wolpert et al., 1995). However, learning an internal model alone does not guarantee optimal motor performance because there are infinite degrees of freedom for the execution of any motor task (Bizzi et al., 1991). In the classic dart-throwing experiment (Thach et al., 1992), for instance, it is likely that subjects learn a throwing strategy that minimizes not only the mean targeting error, but also the variability of the error under the influence of gravity. To test this hypothesis, we repeated the experiment onboard the Russian space station Mir. It was found that exposing the subjects to microgravity triggered sensorimotor adaptations that not only compensated for the loss of gravitation, but also tended to minimize the variability of the targeting error in the weightless environment. These findings provide strong evidence of an optimization strategy in visuomotor adaptation that conforms to Bayesian probabilistic decision.

5.2.1. Optimization model development and comparison with experimental Data

In a dart-throwing experiment, subjects learn (by trial and error) to compensate for visual distortions introduced by the use of prism glasses (Martin et al., 1996, Thach et al., 1992). The learning is thought to reflect the adaptations of some deterministic internal model of the visuo-motor environment, and is assumed to be complete when the magnitude of the targeting error is reduced to within a small range around zero.

However, throwing is a random process governed by multiple degrees of freedom with infinite possibilities of execution. Figure 5.10A illustrates the infinite possible combinations of release velocities and angles to hit a target in the presence of gravity. In fact, Figure 5.10A shows only half of such combinations. Other release combinations are possible if the angle of release is increased from the critical angle (corresponding to a throw at minimum velocity) to a near vertical direction (at infinite release velocity). However, these steeper projectiles are seldom employed by human subjects—perhaps because it is more difficult to aim and hit the target with a high precision this way. From this observation, it may be surmised that a throwing strategy is probably chosen by the brain (through an internal model) not only to hit the target (an ill-posed problem by itself), but also to do so with a high degree of certainty in the face of random variability in the execution of a throw.

In microgravity, the throwing task is more straightforward since the only trajectory that allows for a hit is a straight line (Figure 5.10B). Despite this, the problem is still ill-posed since the release velocity required for a hit is not unique. In this circumstance, subjects might choose to throw slowly (and hence, more steadily) if they wish to hit the target with a high precision. The predicted differences in throwing strategy with and without the influence of gravity (at 1G and μG, respectively) provide a means of testing the hypothesis that the internal model adaptation acts to minimize the Bayesian mean-square error (i.e., sum of squares of the mean and standard deviation of the targeting error (Rentschler et al., 1994) instead of the mean error alone.

Eight ground subjects took part in this experiment. The subjects threw a small ball (weighing ~20 grams and measuring ~3.75 cm in diameter) to a target at waist level ~1.5 m away. The throws were recorded by a camera positioned perpendicular to the plane of the throw. For purpose of comparison, seven astronauts and cosmonauts performed a similar protocol onboard the Russian space station Mir under weightless conditions. During the experiment the space subjects were anchored by a foot restraint. Errors in the vertical and horizontal directions (relative to the subject) in five consecutive trials were analyzed thoroughly in an attempt to find any trends that would indicate degradation and/or adaptation of the targeting ability of the Mir crew as compared to the ground subjects.

Because the Mir crew had been onboard the space station for several (2-12) weeks prior to the experiment, some motor adaptation to the microgravity environment might be expected in the baseline. Indeed, the mean errors committed by the space subjects were relatively small and not significantly different from those of the ground subjects both in the y-direction (Figure 5.11A) and x-direction (Figure 5.11B), suggesting that gravity had little influence on their targeting ability. Furthermore, the high accuracy attained throughout all experimental trials (Figures 5.2A and 5.2B) indicates that the space subjects were already fully adapted to the microgravity environment by the time the experiment took place. However, the video data in Figure 5.11C show that in order to achieve the high targeting accuracy the subjects in space threw the ball significantly more slowly than the ground subjects did. Thus, precision targeting performance was maintained by the subjects in microgravity by suitable adjustments in release angle as well as release velocity.

We propose a mathematical model (see METHODS) to describe the sensorimotor adaptations underlying the changes in release velocity and release angle that allow for high targeting accuracy in 1G and microgravity. Our hypothesis is that subjects try to optimize their throwing performance on ground and in space by minimizing the corresponding Bayesian mean-square targeting error (Rentschler et al., 1994).

There are two sources of error that could affect targeting accuracy on ground. First, gravity imposes a constant bias in the throw. Second, random errors in the aiming and execution of a throw may cause random variability in the resultant projectile. To correct for gravitational bias subjects must learn an internal model of the gravito-geometric constraint on the projectile (Eq. 5.1). However, the solution of such a deterministic internal model is non-unique (Figure 5.10A). We postulate that subjects may apply this freedom of the internal model to minimize the random variability of the projectile (Eq. 5.3). Indeed, the projectile is least susceptible to aiming and execution errors if subjects throw directly at the target pointblank. However, such a straight-line projectile can be achieved on ground only with a very high (near infinite) release velocity (Figure 5.10A). Throwing at a high speed may incur large variability in the release angle (aiming error) and release velocity (execution error) due to the physical limitations of the visuomotor system (Eq. 5.4). Thus, a high-speed throw increases the aiming and execution errors but decreases the susceptibility of the projectile to these errors, and vice versa. An optimal throw is one that minimizes the variance of the targeting error $\sigma^2(\delta y_t)$ resulting from these two factors (Eqs. 5.5, 5.6).

In particular, analysis of the error sensitivities (Fig. 5.3) shows that $\sigma^2(\delta y_t)$ is always lower in the region $\theta < \theta_{crit}$ than in $\theta > \theta_{crit}$ (see METHODS). This explains why subjects seldom choose to throw at an angle greater than θ_{crit}. Moreover, in the lower projectile region an optimal release velocity always exists that minimizes $\sigma^2(\delta y_t)$ (Eqs. 5.7, 5.8). Fig. 5.4A shows the model fit to the targeting error data in the ground subjects. For a range of feasible values for the exponent β an optimal release velocity and error variance can be found that are compatible with the experimental data (see METHODS)., Interestingly, the predicted value of the exponent α changes only slightly (from 0.97 to 0.36) as β varies from 1 to 5 (Figure 5.13A). Thus, a reasonable estimate for α is in the range (0.3, 1). Furthermore, the same set of parameter values yielded comparable model fits to the space data (Figure 5.13B), with similar predicted decreases in release velocity and target-error variability. Thus, although the model parameters are not uniquely identifiable from the experimental data, the model predictions of an optimal release velocity and target-error variability in 1G and 0G are quite robust and are consistent with the experimental data.

Since the sensitivity of muscle receptors remains unchanged in microgravity (Roll et al., 1993), the decreased throwing velocity in the space subjects is likely a result of central rather than peripheral adaptation. However, naïve subjects may experience increased postural instability in microgravity although postural adaptation in spaceflight is quite rapid (Collins et al., 1995; Grigorova and Kornilova, 1996; Lackner and DiZio, 1996; Paloski, 1998; Reschke et al., 1984). Previous postural control studies in microgravity have demonstrated slower arm movements when astronauts are in orbit (Berger et al., 1997; Clement et al., 1985; Collins et al., 1995). Accordingly, the reduced release velocity in the space subjects (Fig. 5.2C) could represent a neural strategy not only to reduce aiming inaccuracy but also postural instability—both of which increase with V. However, in the present study the space subjects were stabilized by foot restraints. The video data revealed that only a minority (~ 30%) of the space subjects used their arm to stabilize their posture during the throw. Even so, their performance was not appreciably different than other subjects, with no statistically significant differences ($p > 0.1$, Student's t test) in release velocity (0.68±0.21 vs. 0.61±0.12 m/s) as well as targeting error (0.93±4.83 vs. 1.26±6.1 cm). Thus, postural factors probably played a relatively minor role in the visuomotor adaptation in the space subjects.

Previous research on motor adaptation in space has been limited mostly to experiments involving either target pointing or simple trajectory tracking by the hand (Berger et al., 1997; Watt, 1997; Watt et al., 1986; Young et al., 1986). However, because arm weight represents only a partial determinant of arm movement (Bock et al., 1992; Ross et al., 1984), the influence that weightlessness exerts on this type of visuomotor adaptation is relatively small (Bock et al., 1992) compared to those resulting from strong mechanical perturbations of the arm force field (Shadmehr and Mussa-Ivaldi, 1994). In contrast, the present targeting experiment provoked profound visuomotor adaptations in the space subjects by virtue of the projectile dynamics—which is strongly influenced by the presence or absence of gravity (Eqs. 5.1, 5.2).

In conclusion, the present experimental and modeling results demonstrated that visuomotor learning is important not only for error correction through internal model adaptation on ground or in microgravity, but also for the minimization of the total mean-square error in the presence of random variability. Thus human intelligent decision, as with visual perception (Knill and Richards, 1996,Porrill et al., 1999), displays certain attributes that seem to conform with Bayesian statistical games.

5.2.2. Methods for model development

5.2.2.1. Projectile Dynamics

From Fig. 5.1A , the projectile equation in 1G is

$$y_t = \left(-\frac{GL}{V^2 \sin(2\theta)} + 1 \right) L \tan(\theta) \qquad \textbf{eq. 5.1}$$

where G is the acceleration due to gravity (9.81m/s^2) and other terms are defined in Fig. 5.1. Eq. 5.1 represents the gravito-geometric constraint on the environment and, correspondingly, the internal model that must be learned by the visuomotor system in order to correct for the gravitational bias in 1 G.

In microgravity, $G = 0$ and the new internal model to be learnt is (Fig. 5.1B):

$$y_t = L \tan(\theta) \qquad \textbf{eq. 5.2}$$

5.2.2.2. Bayesian Error Analysis in 1 G

Assuming that the internal model of Eq. 1 is eventually learnt exactly by the subjects in 1G, the bias due to gravitational pull is nullified and the only error remaining is that due to random variations in the aiming and execution of the throw. Thus, the targeting error (δy_t) may be modeled as a zero-mean random variable and can be expressed as:

$$\delta y_t(V,\theta) = \frac{\partial y_t}{\partial \theta} \delta\theta(V) + \frac{\partial y_t}{\partial V} \delta V(V) \qquad \textbf{eq. 5.3}$$

Eq. 3 indicates that there are two determinants for δy_t. First, imperfections in the aiming and execution of a throw result in random variations in the release angle and release velocity. These *aiming and execution errors* (denoted $\delta\theta$ and δV, respectively) are assumed to be zero-mean random variables and are independent of the projectile. Second, the random errors in the throw are amplified during the projectile as a result of the gravito-geometric constraint. These *projectile amplification factors* are given by the terms $\frac{\partial y_t}{\partial \theta}$, $\frac{\partial y_t}{\partial V}$ which are the error sensitivities with respective to $\delta\theta$ and δV, respectively.

Generally, aiming and execution of a throw become more variable as the speed of throw increases. As a first approximation, we assume that the aiming and execution errors are both given by a power-law relationship as follows:

$$\sigma_V^2(V) = P^2 V^\alpha \text{ and } \sigma_\theta^2(V) = N^2 V^\beta \qquad \textbf{eq. 5.4}$$

In Eq. 4, $\sigma_V^2(\bullet)$, $\sigma_\theta^2(\bullet)$ denote the variance of the error in V and θ, respectively; P, N, α and β are positive scalar factors that depend on the physiological parameters of individual subjects.

During the throw, subjects try to be as consistently accurate as possible. That is, besides minimizing the mean of the targeting error (which is achieved by learning an internal model, Eq. 1) they also try to minimize its variance $\sigma^2(\delta y_t)$ — or, equivalently, a total risk in the form of the Bayesian mean-square error. Assuming δV and $\delta\theta$ are statistically independent of each other, the optimal release velocity is given by:

$$\frac{d}{dV}\sigma^2(\delta y_t) = \frac{d}{dV}\left[\left(\frac{\partial y_t}{\partial \theta}\right)^2 \eta V^\alpha + \left(\frac{\partial y_t}{\partial V}\right)^2 V^\beta\right] = 0 \qquad \textbf{eq. 5.5}$$

and

$$\frac{d^2}{dV^2}\sigma^2(\delta y_t) > 0 \qquad \textbf{eq. 5.6}$$

$E[\bullet]$ denotes the statistical expectation function, $\eta \equiv (P/N)^2$, and the variables θ and V are related by Eq. 5.1 such that the mean targeting error is zero.

From Figure 5.12, it can be seen that the error sensitivities are monotonically increasing with θ in the region $\theta > \theta_{crit}$. In this projectile region, an increasing θ calls for an increasing release velocity to hit the target by virtue of the gravito-geometric constraint (Eq. 1 and Figure 1A). Therefore, $\sigma^2(\delta y_t)$ is monotonically increasing with increasing V in the region $\theta > \theta_{crit}$ and the minimum is at the boundary $\theta = \theta_{crit}$ adjoining the lower projectile region. It follows that the error variance is always lower in the region $\theta < \theta_{crit}$ than in the upper region.

We now show that an optimal release velocity satisfying Eq. 5.5 always exists in the region $\theta < \theta_{crit}$. Firstly, from the geometry of the projectile (Fig. 1A) the targeting error resulting from an execution error (δV) is always finite if $\theta < \theta_{crit.}$. Consequently, the term $\left(\frac{\partial y_t}{\partial V}\right)^2 V^\beta$

remains finite as $V \rightarrow \infty$. From Eq. 5.1, $\left(\frac{\partial y_t}{\partial V}\right)^2 \sim V^{-6}$ and thus the variance model (Eq. 5.4) is feasible provided $0 \le \beta \le 6$. Indeed, for $\beta < 6$ the term $\frac{d}{dV}\left[\left(\frac{\partial y_t}{\partial V}\right)^2 V^\beta\right] \rightarrow 0^-$ as $V \rightarrow \infty$. Also, from Figure 5.12A $\left(\frac{\partial y_y}{\partial \theta}\right)^2$ is always positive and monotonically decreasing with respect to θ (and hence, monotonically increasing with V since θ and V are inversely related in the region $\theta < \theta_{crit}$). It follows that

$$\forall \beta \in (0,6), \quad \lim_{V \to \infty}\left\{\frac{d}{dV}\left[\left(\frac{\partial y_t}{\partial \theta}\right)^2 \eta V^\alpha + \cancelto{0}{\left(\frac{\partial y_t}{\partial V}\right)^2} V^\beta\right]\right\} > 0 \qquad \textbf{eq. 5.7}$$

Likewise, from Fig. 5.3B $\frac{\partial y_t}{\partial \theta} \rightarrow 0$ as V approaches its minimum value ($V_{\min}$) for hitting the target (or equivalent, as $\theta \rightarrow \theta_{crit.}$). Moreover, for $\beta < 6$ the term $\frac{d}{dV}\left[\left(\frac{\partial y_t}{\partial V}\right)^2 V^\beta\right] < 0$ in the region $\theta < \theta_{crit}$. Hence,

$$\forall \beta \in (0,\ 6), \quad \lim_{V \to V_{\min}}\left\{\frac{d}{dV}\left[\cancelto{0}{\left(\frac{\partial y_t}{\partial \theta}\right)^2} \eta V^\alpha + \left(\frac{\partial y_t}{\partial V}\right)^2 V^\beta\right]\right\} < 0 \qquad \textbf{eq. 5.8}$$

In a similar fashion, it can be shown that the condition $\frac{d^2}{dV^2}\sigma^2(\delta y_t) > 0$ always holds for $\beta \in (0,6)$ when $\frac{d}{dV}\sigma^2(\delta y_t) = 0$. Consequently, Eqs. 5.7 and 5.8 imply that for a power law with feasible values of $\beta \in (0,6)$, the variance of the targeting error always has a minimum at some values of V. In fact, this condition holds even when the power law includes multiple terms, i.e. $\sigma(\delta V) = P_n V^{\alpha_1} + P_{n-1}V^{\alpha_2} + \cdots + P_0$ and $\sigma(\delta\theta) = N_k V^{\beta_1} + N_{k-1}V^{\beta_2} + \cdots + N_0$, where n and k are positive integers and $\beta_i \in (0,6)$, $i = 1,2,\ldots,k$. This model prediction is verified by numerical simulation of the model (Figure 5.13A) which shows that for the range of parameter values $\beta \in (0, 6)$ a unique minimum exists for $\sigma(\delta y_t)$ at some optimal release velocity.

5.2.2.3. Bayesian Error Analysis in 0 G

In microgravity the error analysis is greatly simplified as the projectile equation (Eq. 5.2) is dependent on θ only. Hence, following a similar approach as in the above it can be shown readily that $\sigma^2(\delta y_t)$ tends to zero (minimum) as $V \to 0$. (In practice, $\sigma^2(\delta y_t)$ may not attain the absolute minimum because other limiting factors may set in as $V \to 0$). This model prediction is demonstrated in the computer simulation shown in Figure 5.13B.

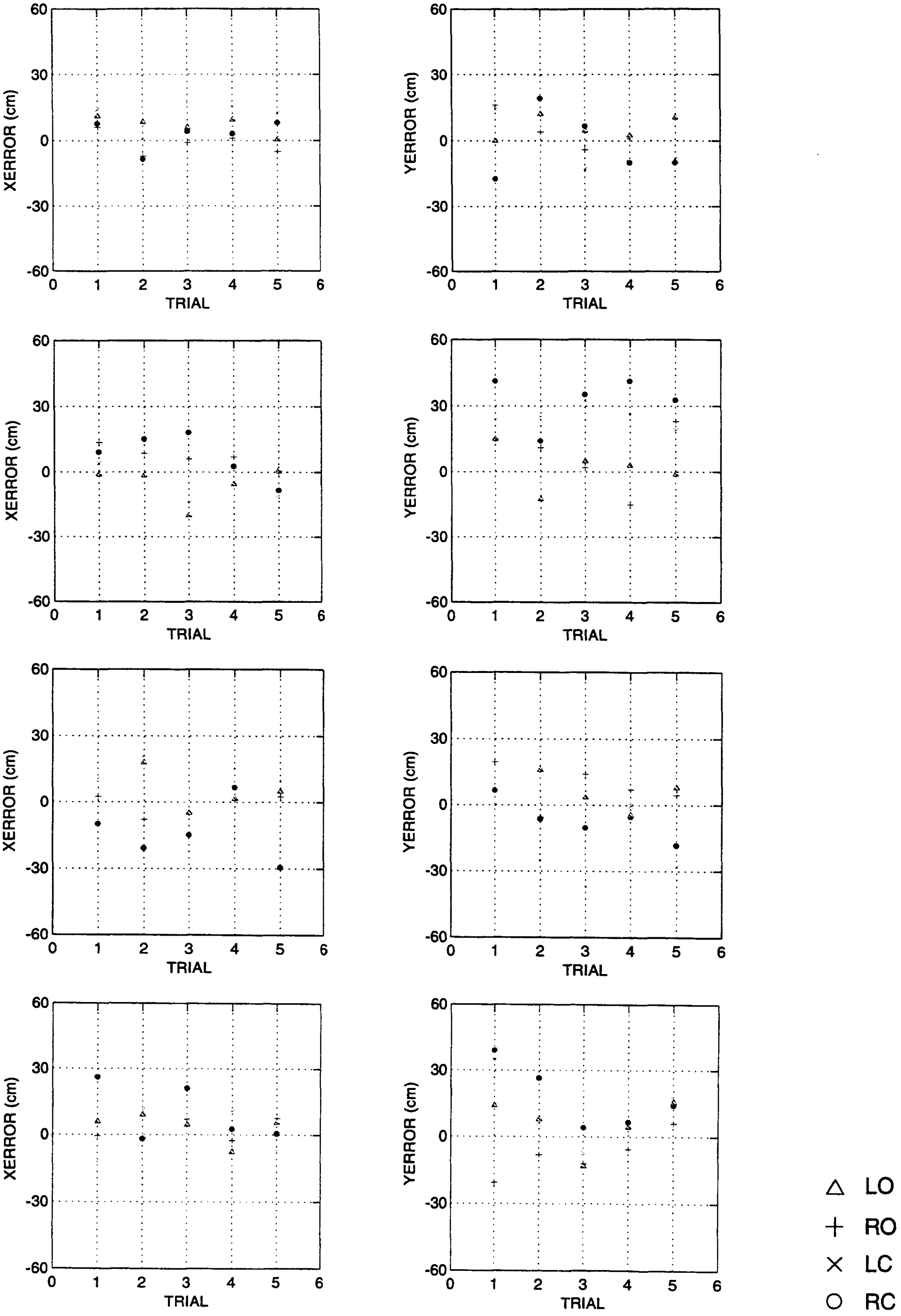


Figure 5.1 X and Y ground error data. Errors for subjects 1 through 4 are shown here. The subject numbers increase starting from the top row. (The rest of the data is shown in Appendix E)

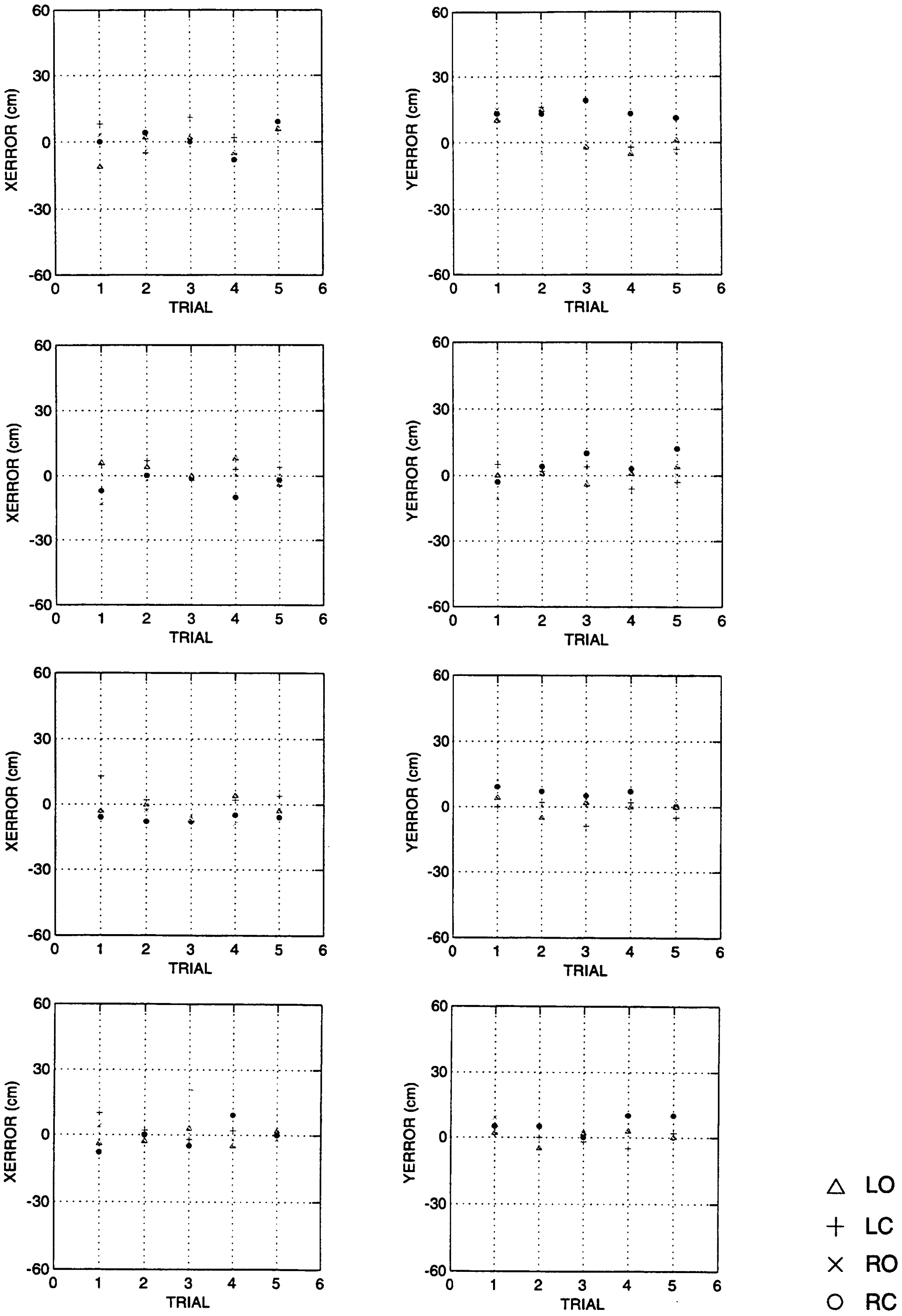


Figure 5.2 X and Y microgravity error data. A) for subject 1 during four different days. Earlier dates are in the higher rows. (The rest of the data is shown in Appendix E)

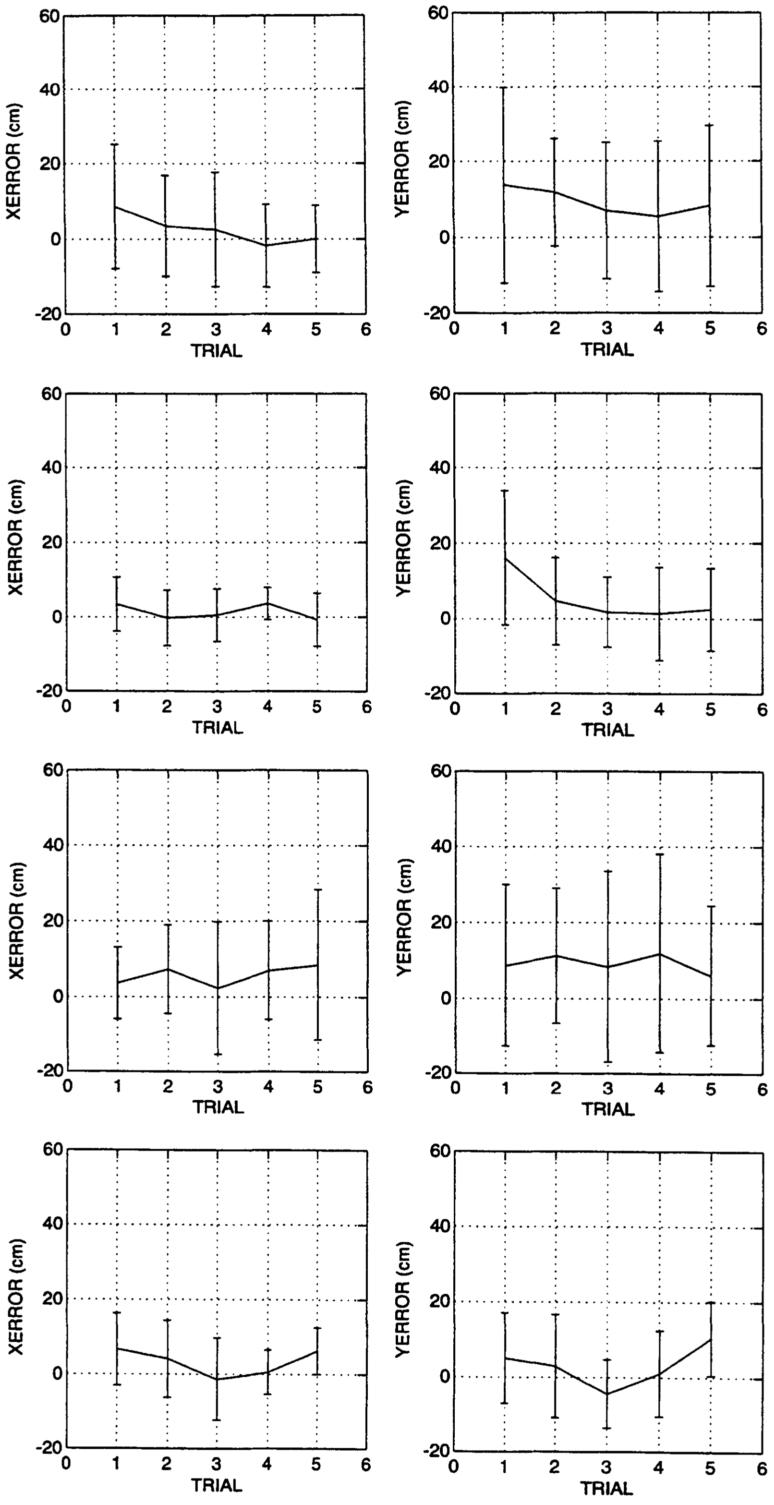


Figure 5.3 Pooled ground error data (mean±st.d.). The data were pooled from all subjects. From top to bottom the conditions were R/C, R/O, L/C and L/O.

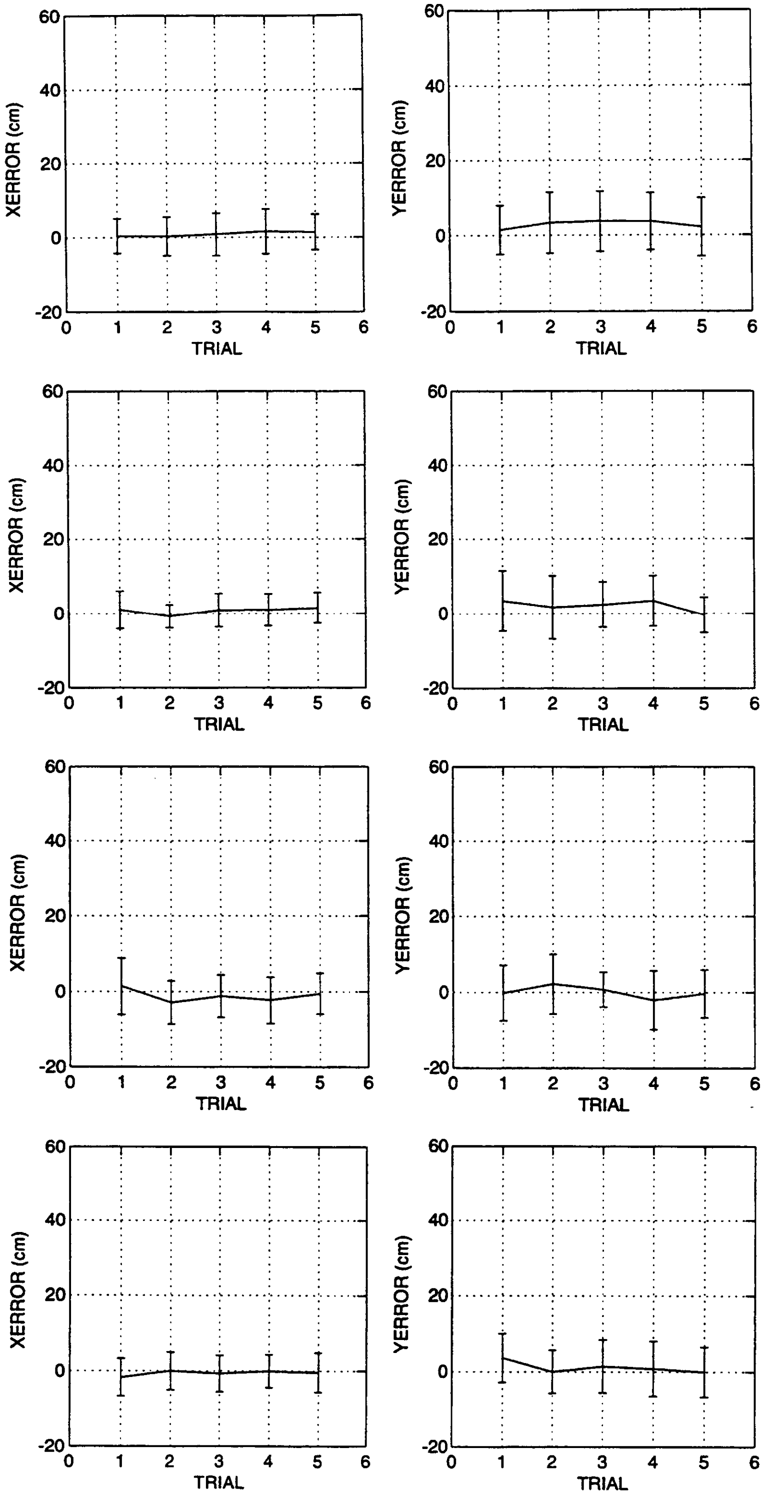


Figure 5.4 Pooled microgravity error data (mean±st.d.). The data were pooled from all subjects during all days. From top to bottom the arm/eye conditions were R/C, R/O, L/C and L/O.

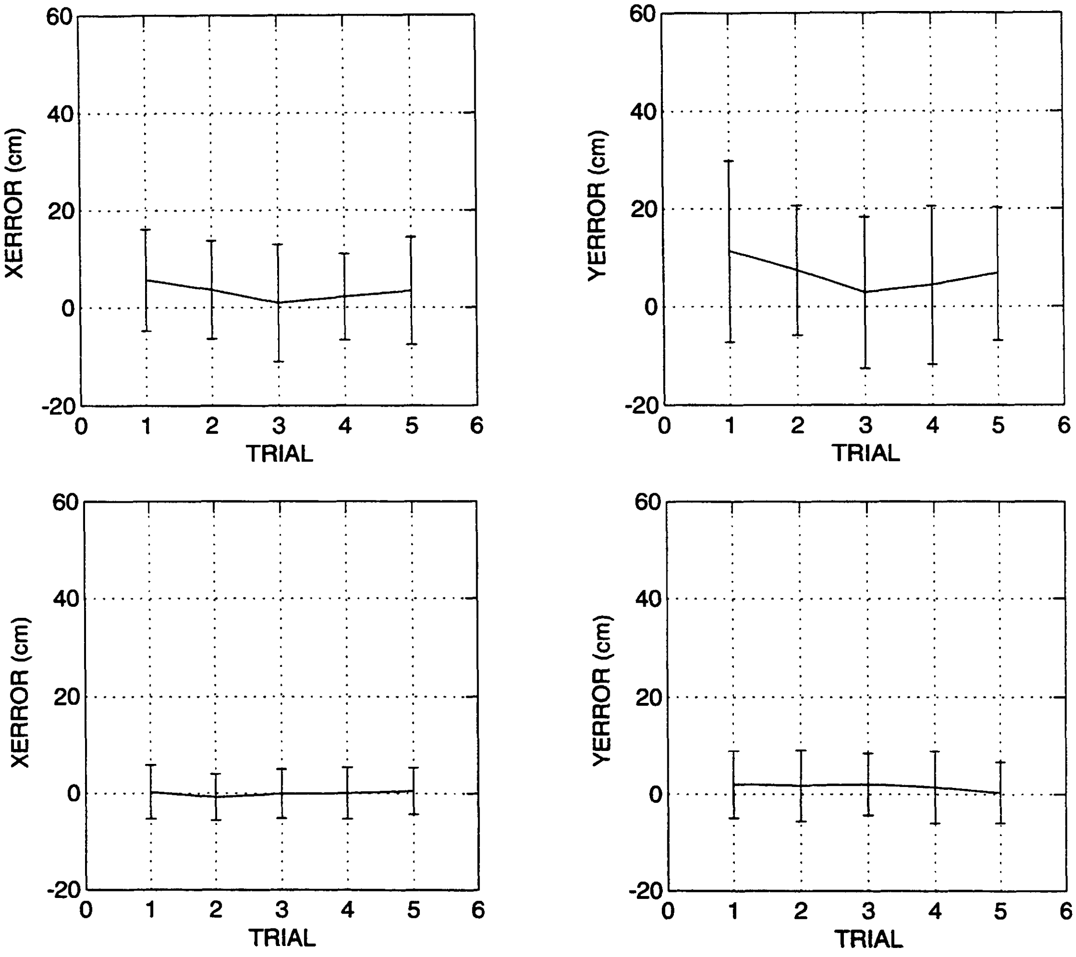


Figure 5.5 Pooled ground error data (mean±st.d.) (Top row). Pooled microgravity error data (mean±st.d.) (Bottom row). The data were pooled from all subjects and all throwing arm/eye conditions.

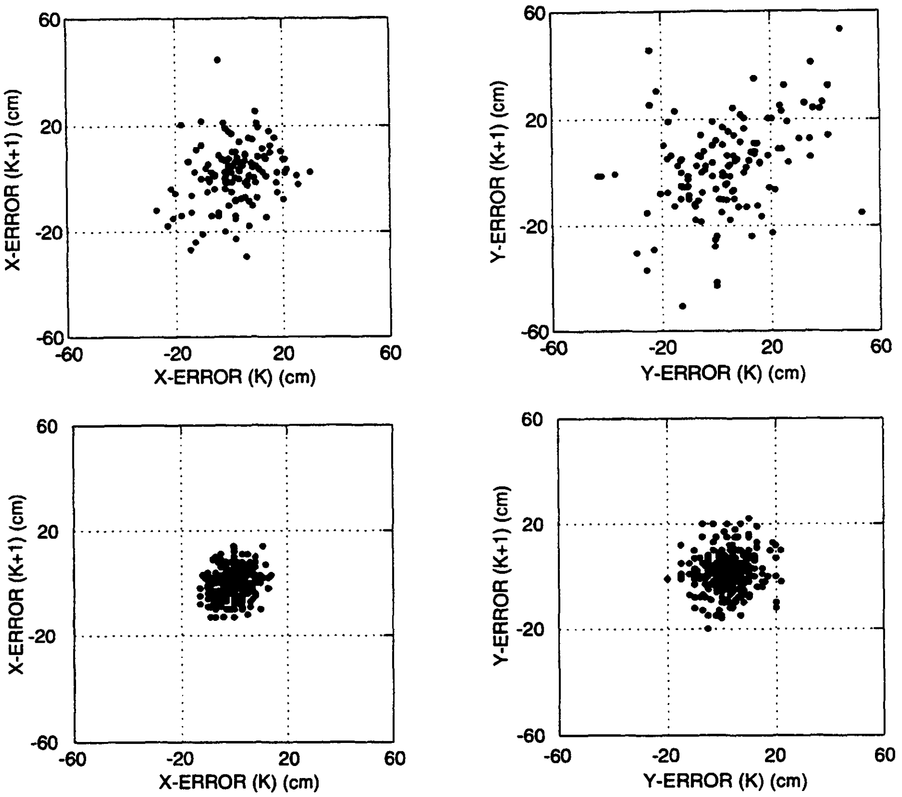


Figure 5.6 Pooled error data plotted against error data of the previous step (e_{k+1} vs. e_k). The data were pooled from all subjects during all days. The top row has the ground data, while the bottom row the data from the space station.

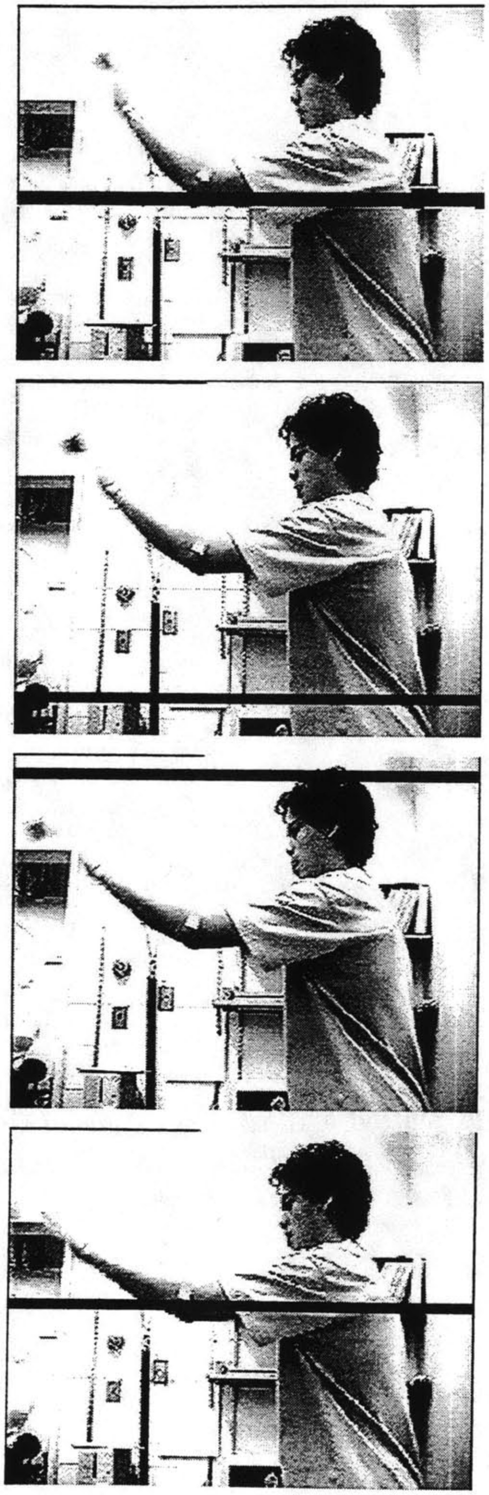


Figure 5.7 Four consecutive frames of the digitized video footage for a throw from the ground set. Sets of frames like this one were used to estimate the velocities and angles of release of the projectile.

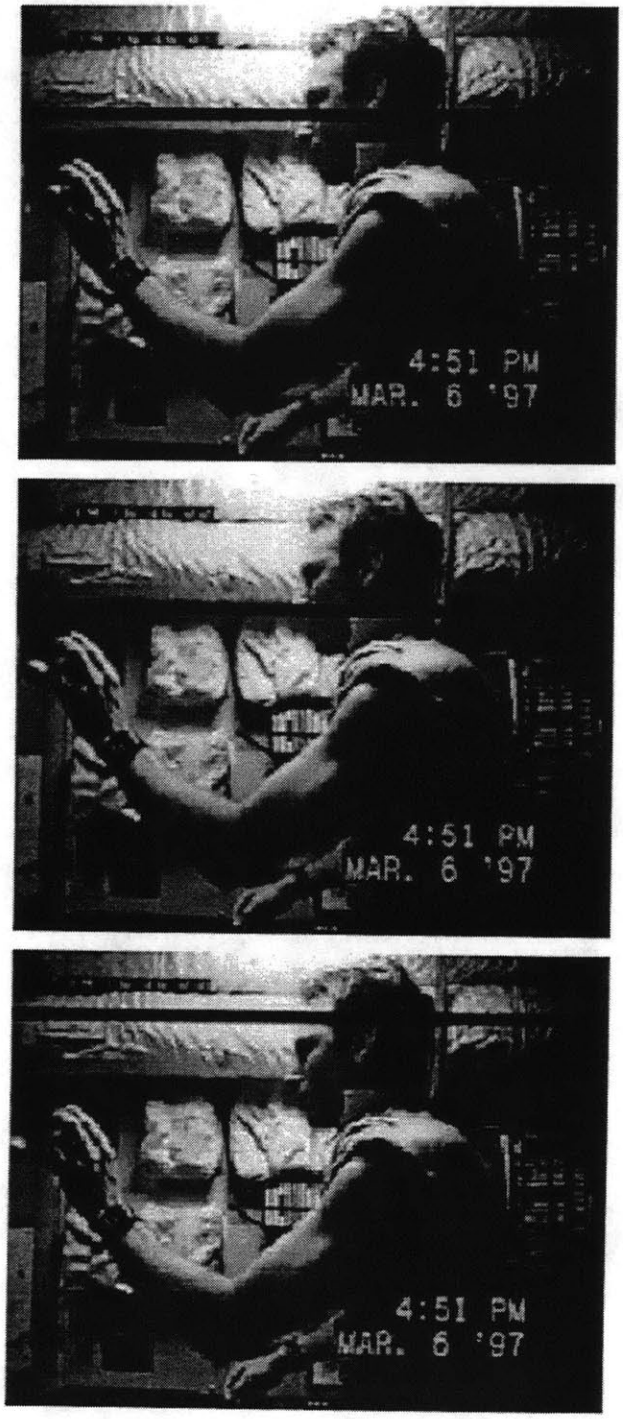


Figure 5.8 Three consecutive frames of the digitized video footage for a throw from the space set.

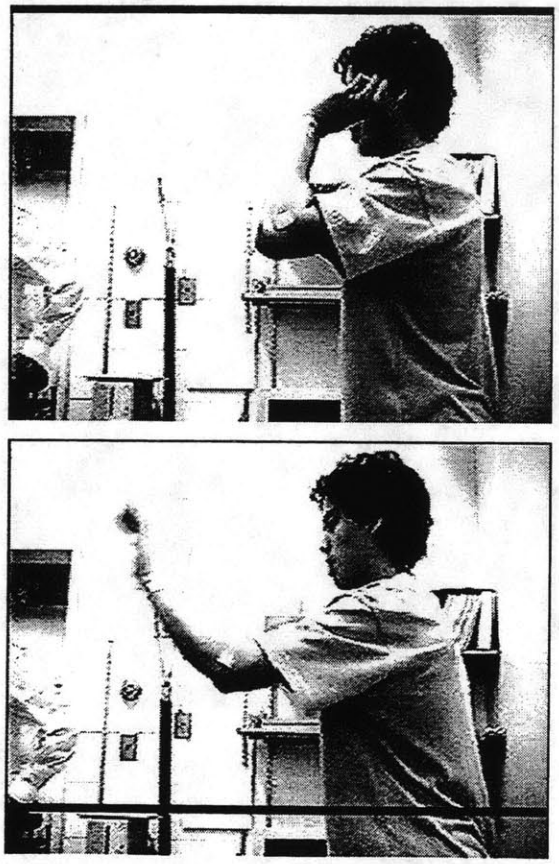


Figure 5.9 The initial and final digitized frames during a ground throw (before the projectile release).

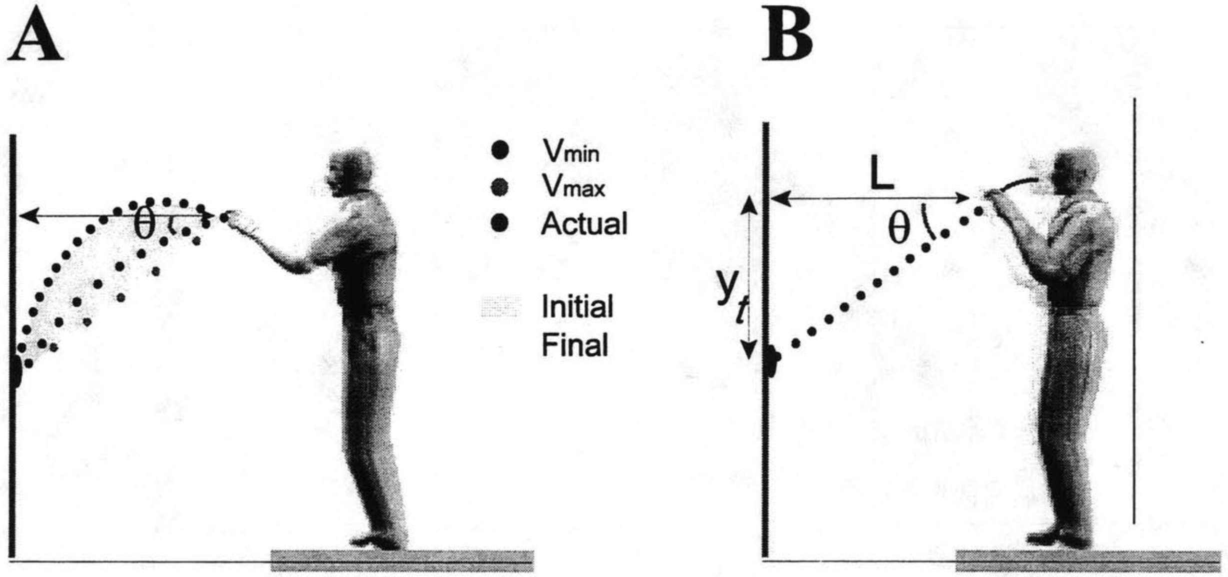


Figure 5.10 Animated illustrations of the throws. A) On ground. B) In microgravity. The figure is showing initial and final body positions, geometry of the projectiles and possible ball trajectories that hit the target at a distance *L* away and height y_t below the point of ball release. A) On ground, trajectories depend on both release angle (θ) and release velocity (*V*). Shaded area shows the range of possible optimal trajectories from a minimum ($V_{\min}$) to infinite release velocity ($V_{\max}$) that result in a minimum Bayesian risk in terms of the mean-square targeting error. The critical release angle (θ_{crit}) is defined as that corresponding to $V_{\min}$. Above θ_{crit} the risk is increasingly higher than in the optimal range. B) In microgravity only a straight-line trajectory at a specific angle will hit the target irrespective of *V*. Bayesian risk is smallest as *V* tends to zero.

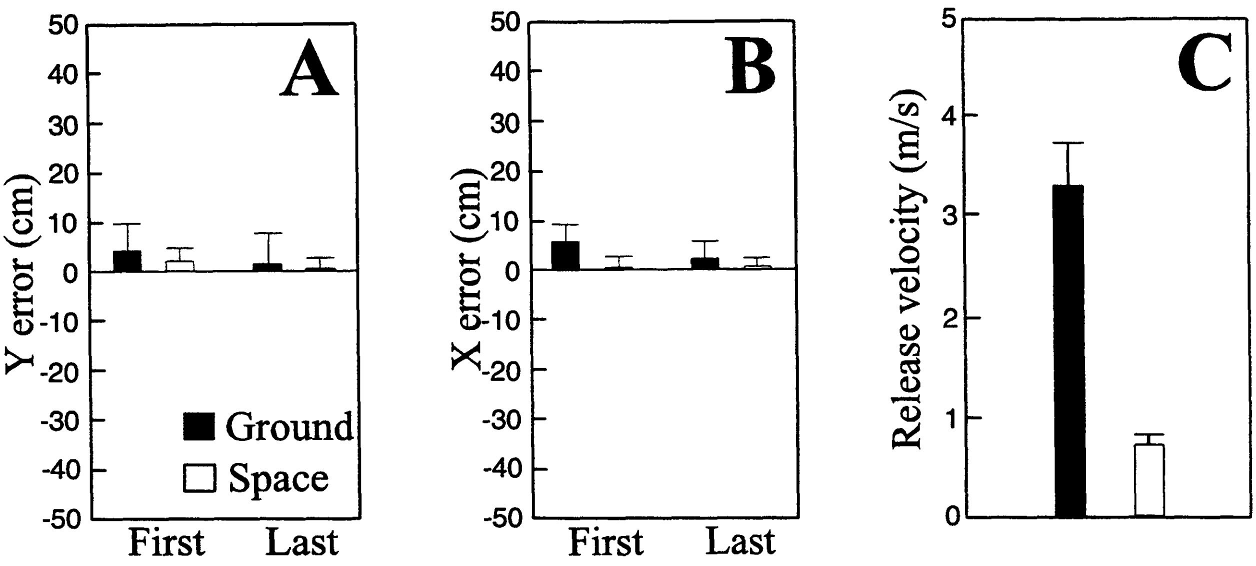


Figure 5.11 Targeting performance in ground subjects and space subjects. Subjects threw with either right or left arm and with eyes either opened or closed in different experimental sessions, each with 5 consecutive trials. Data from experimental sessions under differing throwing conditions were not appreciable different and, hence, were pooled. Mean (±SE) targeting errors in vertical (A) and horizontal direction (B) in both subject groups are not significantly different from zero ($p < 0.05$, Student *t* test) in all experimental trials from the first through the fifth (data from intermediate trials are not shown). Errors are negligible compared with the vertical error that would have occurred (48 cm), had the space subjects thrown the ball at the same mean release angle and velocity as on ground. Throws were recorded by a video camera and the images were digitized frame-by-frame to retrieve hand movement direction and velocity data at point of ball release. Release velocity (C) is markedly decreased in space subjects compared with ground subjects.

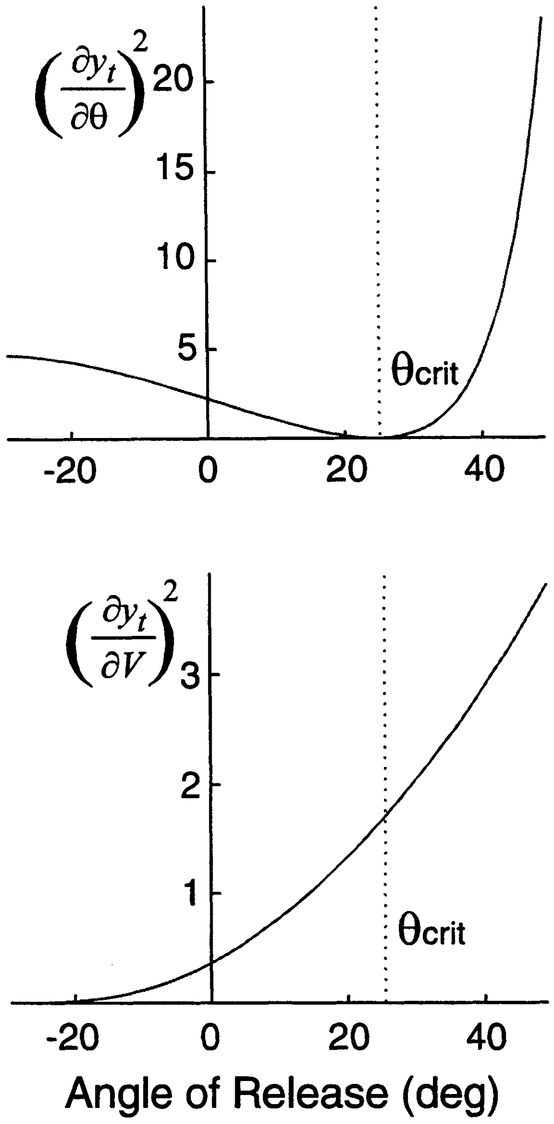


Figure 5.12 Effects of projectile dynamics on targeting performance under the influence of gravity. Plotted are the squared sensitivities of targeting error to imprecisions in aiming (A) and execution of a throw (B) as a function of release angle. Vertical dotted lines indicate the critical angle (6_{crit}) corresponding to the minimum release velocity required for hitting the target. The target-error sensitivity to aiming inaccuracy is minimum (zero) at 6_{crit}. Above 6_{crit} both error sensitivities are higher and the steeper is the throw. For optimal targeting performance, the release angle should be below 6_{crit}.

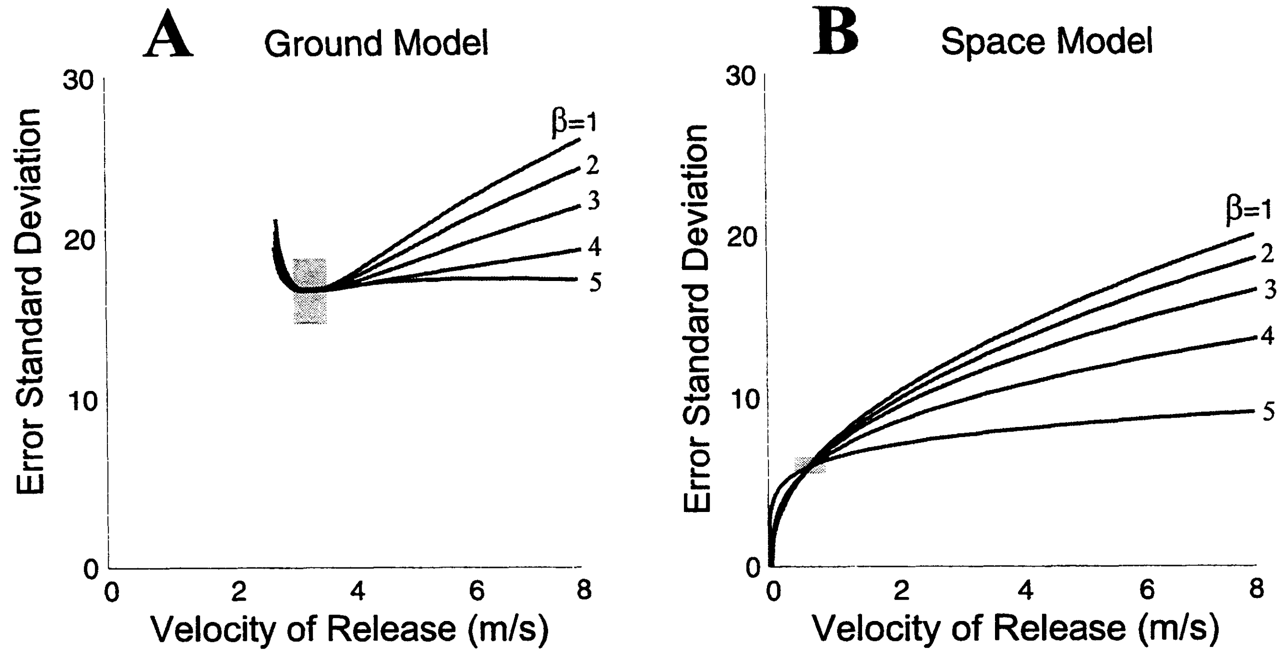


Figure 5.13 Model predictions of variability of targeting error in 1G (A) and in microgravity (B) as compared to experimental data. In 1G, there is a unique minimum for the predicted standard deviation of targeting error corresponding to varying values of the exponent for execution error (β) within the feasible range (0, 6), whereas in microgravity the minimum is zero as release velocity approaches zero. Other model parameters are estimated by fitting the model to the experimental data (shaded areas showing means ± SE).

Bibliography

Arnold, D. B., and Robinson, D. A. (1997). "The oculomotor integrator: testing of a neural network model." *Exp Brain Res*, 113(1), 57-74.

Asada, H., and Slotine, J.-J. E. (1986). *Robot analysis and control*, J. Wiley, New York, N.Y. :.

Barto, A. G., Bradtke, S. J., and Singh, S. P. (1995). "Learning to act using real-time dynamic programming." *Artificial Intelligence*, 72(1-2), 81-138.

Berger, M., Mescheriakov, S., Molokanova, E., Lechner-Steinleitner, S., Seguer, N., and Kozlovskaya, I. (1997). "Pointing arm movements in short- and long-term spaceflights." *Aviat Space Environ Med*, 68, (9 Pt 1): 781-7.

Bizzi, E., Hogan, N., Mussa-Ivaldi, F. A., and Giszter, S. (1992). "Does the nervous system use equilibrium-point control to guide single and multiple joint movements?" *The Behavioral and brain sciences.*, Volume 15(Number 4), 603-613.

Bizzi, E., Mussa-Ivaldi, F. A., and Giszter, S. (1991). "Computations underlying the execution of movement: a biological perspective." *Science*, 253(5017), 287-91.

Bock, O., Howard, I. P., Money, K. E., and Arnold, K. E. (1992). "Accuracy of aimed arm movements in changed gravity." *Aviation space and environmental medicine.*, 63(11), 994.

Chi-Sang Poon, Marina S. Siniaia, Daniel L. Young, and Eldridge, F. L. (1999). "Short-Term Potentiation of Carotid Chemoreflex: An NMDAR-Dependent Neural Integrator." *submitted.*

Ciliz, M. K., and Narendra, K. S. (1996). "Adaptive control of robotic manipulators using multiple models and switching." *International Journal of Robotics Research*, 15(6), 592-610.

Clement, G., Gurfinkel, V. S., Lestienne, F., Lipshits, M. I., and Popov, K. E. (1985). "Changes of posture during transient perturbations in microgravity." *Aviat Space Environ Med*, 56(7), 666-71.

Collins, J. J., De Luca, C. J., Pavlik, A. E., Roy, S. H., and Emley, M. S. (1995). "The effects of spaceflight on open-loop and closed-loop postural control mechanisms: human neurovestibular studies on SLS-2." *Exp Brain Res*, 107(1), 145-50.

Corke, P. I. (1996). "A robotics toolbox for MATLAB." .

Darlot, C., Zupan, L., Etard, O., Denise, P., and Maruani, A. (1996). "Computation of inverse dynamics for the control of movements." *Biol Cybern*, 75(2), 173-86.

Dizio, P., and Lackner, J. R. (1995). "Motor adaptation to Coriolis force perturbations of reaching movements: endpoint but not trajectory adaptation transfers to the nonexposed arm." *J Neurophysiol*, 74(4), 1787-92.

Egeland, O., and Godhavn, J. M. (1994). "Passivity-Based Adaptive Attitude Control of a Rigid Spacecraft." *IEEE transactions on automatic control.*, ume 39(4), 842.

Evarts, E. V. (1981). "Role of motor cortex in voluntary movements in primates." Handbook of Physiology, V. B. Brooks, ed., American Physiological Society, Bethesda, 1083-1120.

Feldman, A. G., and Levin, M. F. (1995). "The origin and use of positional frames of reference in motor control." *The Behavioral and brain sciences.*, 18(4), 723.

Fujita, M. (1982a). "Adaptive filter model of the cerebellum." *Biological Cybernetics*, 45(3), 195-206.

Fujita, M. (1982b). "Simulation of adaptive modification of the vestibulo-ocular reflex with an adaptive filter model of the cerebellum." *Biological Cybernetics*, 45(3), 207-14.

Gaudiano, P., Guenther, F. H., and Zalama, E. (1997). "The neural dynamics approach to sensory-motor control: overview and recent applications in mobile robot control and speech production." *Neural systems for robotics, p. xvii+346*, 153-94.

Gerdes, V. G. J., and Happee, R. (1994). "Use of an internal representation in fast goal-directed movements: A modelling approach." *Biol Cybern*, 70(6), 513-524.

Gomi, H., and Kawato. (1996). "Equilibrium-point control hypothesis examined by measured arm stiffness during multijoint movement." *Science*, 272(5258), 117-20.

Gordon, A. M., Huxley, A. F., and Julian, F. J. (1966). "The variation in isometric tension with sarcomere length in vertebrate muscle fibres." *J Physiol (Lond)*, 184(1), 170-92.

Gottlieb, G. L. (1996). "Equilibrium Point Hypothesis." .

Graziano, M. S. A., Yap, G. S., and Gross, C. G. (1994). "Coding of Visual Space by Premotor Neurons." *Science*, 266(5187), 1054.

Grigorova, V. K., and Kornilova, L. N. (1996). "Microgravity effect on the vestibulo-ocular reflex is dependent on otolith and vision contributions." *Aviat Space Environ Med*, 67(10), 947-54.

Hanneton, S., Berthoz, A., Droulez, J., and Slotine, J. J. (1997). "Does the brain use sliding variables for the control of movements?" *Biol Cybern*, 77(6), 381-93.

Happee, R. (1992). "Time optimality in the control of human movements." *Biol Cybern*, 66(4), 357-66.

Houde, J. F., and Jordan, M. I. (1998). "Sensorimotor adaptation in speech production." *Science*, 279(5354), 1213-6.

Ito, M. (1984). *The cerebellum and neural control*, Raven Press, New York.

Jeannerod, M. (1990). *The neural and behavioural organization of goal-directed movements*, Clarendon Press ; Oxford University Press, Oxford [England] : New York :.

Jordan, M. I., Flash, T., and Arnon, Y. (1994). "A Model of the Learning of Arm Trajectories from Spatial Deviations." *Journal of cognitive neuroscience*, 6(4), 359.

Kalaska, J. F., and Crammond, D. J. (1992). "Cerebral cortical mechanisms of reaching movements." *Science*, 255(5051), 1517-23.

Kawato, M., Furukawa, K., and Suzuki, R. (1987). "A hierarchical neural-network model for control and learning of voluntary movement." *Biol Cybern*, 57(3), 169-85.

Kawato, M., and Gomi, H. (1992). "The cerebellum and VOR/OKR learning models." *Trends Neurosci (WEL)*, 15(11), 445-53.

Kawato, M., Uno, Y., Isobe, M., and Suzuki, R. (1988). "Hierarchical neural network model for voluntary movement with application to robotics." *IEEE Control Systems Magazine*, 8(2), 8-15.

Knill, D. C., and Richards, R. (1996). *Perception as Bayesian inference*, Cambridge University Press, Cambridge, U.K. ; New York :.

Korf, R. E. U. (1990). "Real-time heuristic search." *Artificial Intelligence*, 42, 2-3.

Lackner, J. R., and Dizio, P. (1994). "Rapid adaptation to Coriolis force perturbations of arm trajectory." *J Neurophysiol*, 72(1), 299-313.

Lackner, J. R., and DiZio, P. (1996). "Motor function in microgravity: movement in weightlessness." *Curr Opin Neurobiol*, 6, 744-50.

Martin, D. A., Zomlefer, M. R., and Unal, A. (1989). "Human upper limb dynamics." *Robotics and Autonomous Systems*, 5(2), 151-63.

Martin, T. A., Keating, J. G., Goodkin, H. P., Bastian, A. J., and Thach, W. T. (1996). "Throwing while looking through prisms. I. Focal olivocerebellar lesions impair adaptation." *Brain*, 119, (Pt 4) 1183-98.

McCormick, D. A. (1995). "Motor control. The cerebellar symphony." *Nature (NSC)*, 374, (6521): 412-3.

Miall, R. C., and Wolpert, D. M. (1996). "Forward models for physiological motor control." *Neural Networks*, 9(8), 1265-79.

Miller, W. T., III, Glanz, F. H., and Kraft, L. G. I. I. I. D. N. H. U. S. A. (1987). "Application of a general learning algorithm to the control of robotic manipulators." *International Journal of Robotics Research*, 6(2), 84-98.

Narendra, K. S., and Balakrishnan, J. (1997). "Adaptive control using multiple models." *IEEE Transactions on Automatic Control*, 42(2), 171-87.

Narendra, K. S., and Mukhopadhyay, S. (1997). "Adaptive control using neural networks and approximate models." *IEEE Transactions on Neural Networks*, 8(3), 475-85.

Ojakangas, C. L., and Ebner, T. J. (1994). "Purkinje cell complex spike activity during voluntary motor learning: relationship to kinematics." *J Neurophysiol*, 72(6), 2617-30.

Paloski, W. H. (1998). "Vestibulospinal adaptation to microgravity." *Otolaryngol Head Neck Surg*, 118(3 Pt 2), S39-44.

Pascual-Leone, A., Grafman, J., and Hallett, M. (1994). "Modulation of cortical motor output maps during development of implicit and explicit knowledge." *Science*, 263(5151), 1287-9.

Porrill, J., Frisby, J. P., Adams, W. J., and Buckley, D. (1999). "Robust and optimal use of information in stereo vision." *Nature*, 397, 63-66.

Raikova, R. (1992). "A general approach for modelling and mathematical investigation of the human upper limb." *J Biomech*, 25(8), 857-67.

Rentschler, I., Juttner, M., and Caelli, T. (1994). "Probabilistic analysis of human supervised learning and classification." *Vision Res*, 34(5), 669-87.

Reschke, M. F., Anderson, D. J., and Homick, J. L. (1984). "Vestibulospinal reflexes as a function of microgravity." *Science*, 225(4658), 212-4.

Robinson, D. A. (1989). "Integrating with neurons." *Annu Rev Neurosci*, 12, 33-45.

Roll, J. P., Popov, K., Gurfinkel, V., Lipshits, M., Andre-Deshays, C., Gilhodes, J. C., and Quoniam, C. (1993). "Sensorimotor and perceptual function of muscle proprioception in microgravity." *J Vestib Res*, 3(3), 259-73.

Ross, H., Brodie, E., and Benson, A. (1984). "Mass discrimination during prolonged weightlessness." *Science*, 225(4658), 219-21.

Rothwell, J. C., Traub, M. M., Day, B. L., Obeso, J. A., Thomas, P. K., and Marsden, C. D. (1982). "Manual motor performance in a deafferented man." *Brain*, 105, (Pt 3) 515-42.

Santiago, R. A., and Werbos, P. J. "New progress towards truly brain-like intelligent control." *Proceedings of World Congress on Neural Networks*, San Diego, CA, USA; 5-9 June 1994, 27-33.

Shadmehr, R., and Holcomb, H. H. (1997). "Neural correlates of motor memory consolidation." *Science*, 277(5327), 821-5.

Shadmehr, R., and Mussa-Ivaldi, F. A. (1994). "Adaptive representation of dynamics during learning of a motor task." *J Neurosci*, 14(5 Pt 2), 3208-24.

Shen, L. (1989). "Neural integration by short term potentiation." *Biol Cybern*, 61(4), 319-25.

Slotine, J.-J. E., and Li, W. (1991). *Applied Nonlinear Control*, Prentice-Hall, Englewood Cliffs, NJ 07632.

Soechting, J. F., and Flanders, M. M. (1992). "Moving in three-dimensional space: frames of reference, vectors, and coordinate systems." *Annu Rev Neurosci*, 15, 167-91.

Sutton, R. S. "Integrated architectures for learning, planning, and reacting based on approximating dynamic programming Conference, p. v+427, 216-24." , Austin, TX, USA; 21-23 June 1990.

Thach, W. T., Goodkin, H. P., and Keating, J. G. (1992). "The cerebellum and the adaptive coordination of movement." *Annu Rev Neurosci*, 15, 403-42.

Watt, D. G. (1997). "Pointing at memorized targets during prolonged microgravity." *Aviat Space Environ Med*, 68(2), 99-103.

Watt, D. G., E., M. K., L., B. R., B., T. R., M., G., and P., S.-P. (1985). "Canadian medical experiments on shuttle flight 41-G." *Canadian Aeronautics and Space Journal*, 31(3), 215-226.

Watt, D. G., Money, K. E., and Tomi, L. M. (1986). "M.I.T./Canadian vestibular experiments on the Spacelab-1 mission: 3. Effects of prolonged weightlessness on a human otolith-spinal reflex." *Exp Brain Res*, 64(2), 308-15.

Welsh, J. P., Lang, E. J., Suglhara, I., and Llinas, R. (1995). "Dynamic organization of motor control within the olivocerebellar system." *Nature*, 374(6521), 453-7.

Werbos, P. J., and Xiaozhong, P. "Generalized maze navigation: SRN critics solve what feed-forward or Hebbian nets cannot." *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Beijing, China; 14-17 Oct. 1996, 1764-9.

Widrow, B., McCool, J., Larimore, M. G., and Johnson, C. R. (1976). "Stationary and nonstationary learning characteristics of the LMS adaptive filter." *Proceedings of the IEEE*, 64(8), 1151-62.

Winters, J. M., and Stark, L. (1985). "Analysis of fundamental human movement patterns through the use of in-depth antagonistic muscle models." *IEEE Trans Biomed Eng*, 32(10), 826-39.

Winters, J. M., and Stark, L. (1988). "Estimated mechanical properties of synergistic muscles involved in movements of a variety of human joints." *J Biomech*, 21(12), 1027-41.

Wolpert, D. M., Ghahramani, Z., and Jordan, M. (1995). "An internal model for sensorimotor integration." *Science*, 269(5232), 1880-2.

Wyatt, J. L., Jr., Chua, L. O., Gannett, J., Goknar, I. C., and Green, D. N. "On the Concepts of Losslessness and Passivity in Nonlinear Network." *Proc IEEE Int Symp Circuits Syst*, Houston, 854-855.

Wyatt, J. L., Jr., Chua, L. O., Gannett, J. W., Goknar, I. C., and Green, D. N. (1981). "ENERGY CONCEPTS IN THE STATE-SPACE THEORY OF NONLINEAR n-PORTS EM DASH 1." *IEEE Trans Circuits Syst*, CAS(1), 48-61.

Young, L. R., Oman, C. M., Watt, D. G., Money, K. E., Lichtenberg, B. K., Kenyon, R. V., and Arrott, A. P. (1986). "M.I.T./Canadian vestibular experiments on the Spacelab-1 mission: 1. Sensory adaptation to weightlessness and readaptation to one-g: an overview." *Exp Brain Res*, 64(2), 291-8.

Zames, G. (1966). "On the Input-Output Stability of Time-Varying Nonlinear Feedback Systems. Part I: Conditions Derived Using Concepts of Loop Gain, Conicity, and Passivity." *IEEE Transactions on Automatic Control*, AC-11(2), 228-238.

APPENDIX A

Muscles are major part of the musculoskeletal structure of the human arm. Successful controlling of the arm movements by the CNS requires an accurate representation of the muscles. The muscle model shown is Figure 1 was developed on SIMULINK (Mathworks, Natick, MA) based on a muscle model description used by Hapee (1992). The muscle model of Figure 1 is a third order, nonlinear, lumped parameter model. The CNS sends neural input (u) to the agonist-antagonist muscle pair. The neural excitation (E) is the output of the neural input through a first order system with time constant (T_{ne}). The active state (A) is the output of the neural excitation through another first-order system with time constant (T_a). The active state rescaled is:

$$A_r = A_{\min} + (1 - A_{\min})A$$

The minimum non-zero value of A is denoted by $A_{\min}$. The active force relative to the isometric force that would result from the current active state is:

$$F_r = \frac{F}{F_{\max} A_r}$$

F is the active muscle force, and $F_{\max}$ is the maximum isometric force. The force-velocity relationship of the contractile element is:

$$l'_{ce} = \frac{MV_{vm}}{\tan\left(\frac{1}{0.54} - 1.38\right) + \tan(1.38)} \left[\tan\left(\frac{F_r}{0.54} - 1.38\right) - \tan\left(\frac{1}{0.54} - 1.38\right) \right]$$

MV_{vm} is the maximum shortening velocity.

For the series elastic element:

$$l_{se} = l_m - l_{ce}$$

$$F = \frac{F_{\max}}{e^{SE_{sh}} - 1} \left[e^{l_{se} \frac{SE_{sh}}{SE_{xm}}} - 1 \right]$$

SE_{sh} is the series elastic shape parameter, SE_{xm} is the series elastic stretch with maximal isometric force, l_m is the total muscle length and l_{se} is the length of the series elastic element.

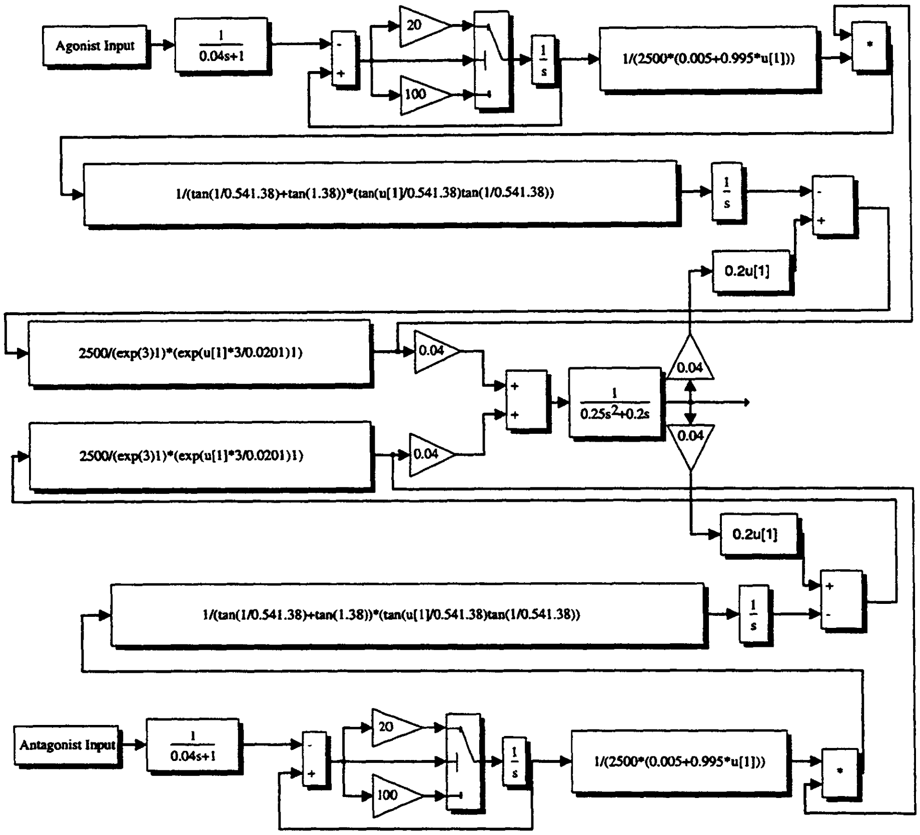

Agonist Input
1
0.04s+1
-
+
20
100
1
s
1/(2500*(0.005+0.995*u[1]))
*
1/(tan(1/0.541.38)+tan(1.38))*(tan(u[1]/0.541.38)tan(1/0.541.38))
1
s
-
+
0.2u[1]
2500/(exp(3)1)*(exp(u[1]*3/0.0201)1)
0.04
+
+
1
0.25s2+0.2s
0.04
0.04
2500/(exp(3)1)*(exp(u[1]*3/0.0201)1)
0.04
0.2u[1]
+
-
1/(tan(1/0.541.38)+tan(1.38))*(tan(u[1]/0.541.38)tan(1/0.541.38))
1
s
Antagonist Input
1
0.04s+1
-
+
20
100
1
s
1/(2500*(0.005+0.995*u[1]))
*

Figure A.1 A simulation block representing a muscle model. This model was built using the description by Gerdes and Hapee. The inputs are EMG patterns at the agonist and antagonist muscles and the output is the rotational displacement of the joint.

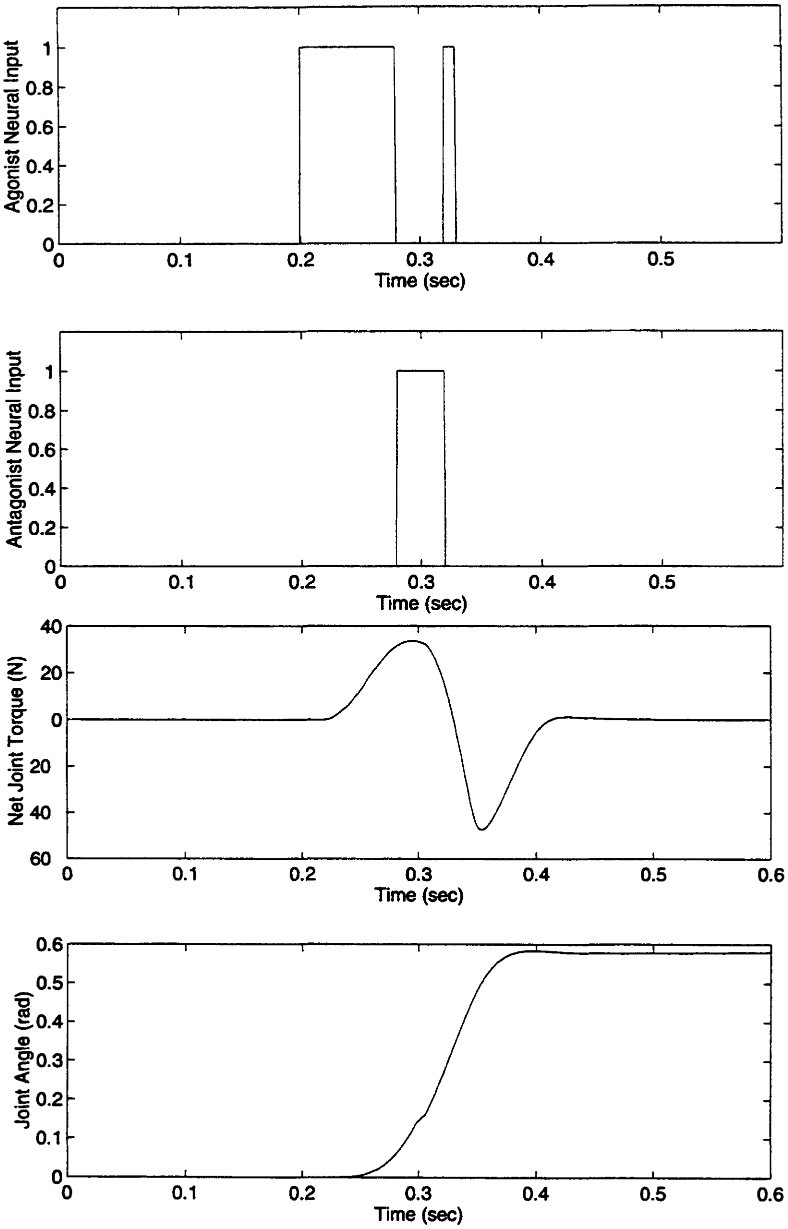


Figure A.2 Agonist and Antagonist inputs to the muscle model of Figure A.1 are plotted versus time (top). The response (joint torque and joint angle) is shown also (bottom).

APPENDIX B

```
> with(linalg):
> alpha0:=0:
> a0:=0:
> d0:=0:
> alpha1:=Pi/2:
> a1:=L1:
> d1:=0:
> alpha2:=Pi/2:
> a2:=0:
> d2:=0:
> alpha3:=-Pi/2:
> a3:=0:
> d3:=0:
> alpha4:=Pi/2:
> a4:=0:
> d4:=L4:
> alpha5:=-Pi/2:
> a5:=0:
> d5:=0:
> alpha6:=Pi/2:
> a6:=0:
> d6:=L6:
> alpha7:=Pi/2:
> a7:=0:
> d7:=0:
> alpha8:=0:
> a8:=L8:
> d8:=0:
> A00:=matrix([[cos(0),-sin(0)*cos(alpha0), sin(0)*sin(alpha0),
  a0*cos(0)],[sin(0), cos(0)*cos(alpha0), -cos(0)*sin(alpha0), a0*sin(0)],[0,
  sin(alpha0), cos(alpha0), d0],[0, 0, 0, 1]]):
> A01:=matrix([[cos(theta1), -sin(theta1)*cos(alpha1),
  sin(theta1)*sin(alpha1), a1*cos(theta1)],[sin(theta1),
  cos(theta1)*cos(alpha1), -cos(theta1)*sin(alpha1), a1*sin(theta1)],[0,
  sin(alpha1), cos(alpha1), d1],[0, 0, 0, 1]]):
> A12:=matrix([[cos(theta2), -sin(theta2)*cos(alpha2),
  sin(theta2)*sin(alpha2), a2*cos(theta2)],[sin(theta2),
  cos(theta1)*cos(alpha2), -cos(theta2)*sin(alpha2), a2*sin(theta2)],[0,
  sin(alpha2), cos(alpha2), d2],[0, 0, 0, 1]]):
> A23:=matrix([[cos(theta3), -sin(theta3)*cos(alpha3),
  sin(theta3)*sin(alpha3), a3*cos(theta3)],[sin(theta3),
  cos(theta3)*cos(alpha3), -cos(theta3)*sin(alpha3), a3*sin(theta3)],[0,
  sin(alpha3), cos(alpha3), d1],[0, 0, 0, 1]]):
> A34:=matrix([[cos(theta4), -sin(theta4)*cos(alpha4),
  sin(theta4)*sin(alpha4), a4*cos(theta4)],[sin(theta4),
  cos(theta4)*cos(alpha4), -cos(theta4)*sin(alpha4), a4*sin(theta4)],[0,
  sin(alpha4), cos(alpha4), d4],[0, 0, 0, 1]]):
> A45:=matrix([[cos(theta5), -sin(theta5)*cos(alpha5), sin(theta5)*sin(alpha5),
  a5*cos(theta5)],[sin(theta5), cos(theta5)*cos(alpha5), -
  cos(theta5)*sin(alpha5), a5*sin(theta5)],[0, sin(alpha5), cos(alpha5),
  d5],[0, 0, 0, 1]]):
```

```
> A56:=matrix([[cos(theta6), -sin(theta6)*cos(alpha6), sin(theta6)*sin(alpha6),
  a6*cos(theta6)],[sin(theta5),
  cos(theta6)*cos(alpha6), -cos(theta6)*sin(alpha6), a6*sin(theta6)],[0,
  sin(alpha6), cos(alpha6), d6],[0, 0, 0, 1]]):
> A67:=matrix([[cos(theta7), -sin(theta7)*cos(alpha7), sin(theta7)*sin(alpha7),
  a7*cos(theta7)],[sin(theta7), cos(theta7)*cos(alpha7), -
  cos(theta7)*sin(alpha7), a7*sin(theta7)],[0, sin(alpha7), cos(alpha7),
  d7],[0, 0, 0, 1]]):
> A78:=matrix([[cos(theta8), -sin(theta8)*cos(alpha8), sin(theta8)*sin(alpha8),
  a8*cos(theta8)],[sin(theta8), cos(theta8)*cos(alpha8), -
  cos(theta8)*sin(alpha8), a8*sin(theta8)],[0, sin(alpha8), cos(alpha8),
  d8],[0, 0, 0, 1]]):
> A02:=multiply(A01,A12):
> A03:=multiply(multiply(A01,A12),A23):
> A04:=multiply(multiply(multiply(A01,A12),A23),A34):
> A05:=multiply(multiply(multiply(multiply(A01,A12),A23), A34),A45):
> A06:=multiply(multiply(multiply(multiply(multiply(A01,
  A12),A23),A34),A45),A56):
> A07:=multiply(multiply(multiply(multiply(multiply(multiply
  (A01,A12),A23),A34),A45),A56),A67):
> A08:=multiply(multiply(multiply(multiply(multiply(multiply (multi-
  ply(A01,A12),A23),A34),A45),A56),A67),A78):
> rc1:=multiply(A01,matrix([[-Lc1],[0],[0],[1]])):
> rc2:=multiply(A02,matrix([[0],[-Lc2],[0],[1]])):
> rc3:=multiply(A03,matrix([[0],[0],[Lc3],[1]])):
> rc4:=multiply(A04,matrix([[0],[-Lc4],[0],[1]])):
> rc5:=multiply(A05,matrix([[0],[0],[Lc5],[1]])):
> rc6:=multiply(A06,matrix([[0],[-Lc6],[0],[1]])):
> rc7:=multiply(A07,matrix([[-Lc7],[0],[0],[1]])):
> rc8:=multiply(A08,matrix([[-Lc8],[0],[0],[1]])):
> rc1:=convert(rc1,vector):
> rc2:=convert(rc2,vector):
> rc3:=convert(rc3,vector):
> rc4:=convert(rc4,vector):
> rc5:=convert(rc5,vector):
> rc6:=convert(rc6,vector):
> rc7:=convert(rc7,vector):
> rc8:=convert(rc8,vector):
> J1:=jacobian(rc1,[theta1,theta2,theta3,theta4,theta5, theta6,theta7,theta8]):
> J2:=jacobian(rc2,[theta1,theta2,theta3,theta4,theta5, theta6,theta7,theta8]):
> J3:=jacobian(rc3,[theta1,theta2,theta3,theta4,theta5, theta6,theta7,theta8]):
> J4:=jacobian(rc4,[theta1,theta2,theta3,theta4,theta5, theta6,theta7,theta8]):
> J5:=jacobian(rc5,[theta1,theta2,theta3,theta4,theta5, theta6,theta7,theta8]):
> J6:=jacobian(rc6,[theta1,theta2,theta3,theta4,theta5, theta6,theta7,theta8]):
> J7:=jacobian(rc7,[theta1,theta2,theta3,theta4,theta5, theta6,theta7,theta8]):
> J8:=jacobian(rc8,[theta1,theta2,theta3,theta4,theta5, theta6,theta7,theta8]):
> Ja0:=matrix([[0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0],[0,0,0,0, 0,0,0,0]]):
> Ja1:=add(multiply(multiply(matrix([[1,0,0,0],[0,1,0,0],[0,
  0,1,0]]),multiply(A00,matrix([[0],[0],[1],[0]]))),matrix([[1,0,0,0,0,0,0,0]])
  ),Ja0):
> Ja2:=add(multiply(multiply(matrix([[1,0,0,0],[0,1,0,0],[0,
  0,1,0]]),multiply(A01,matrix([[0],[0],[1],[0]]))),matrix([[0,1,0,0,0,0,0,0]])
  ),Ja1):
```

```
> Ja3:=add(multiply(multiply(matrix([[1,0,0,0],[0,1,0,0],[0,
  0,1,0]]),multiply(A02,matrix([[0],[0],[1],[0]]))),matrix([[0,0,1,0,0,0,0,0]])
  ),Ja2):
> Ja4:=add(multiply(multiply(matrix([[1,0,0,0],[0,1,0,0],[0,
  0,1,0]]),multiply(A03,matrix([[0],[0],[1],[0]]))),matrix([[0,0,0,1,0,0,0,0]])
  ),Ja3):
> Ja5:=add(multiply(multiply(matrix([[1,0,0,0],[0,1,0,0],[0,
  0,1,0]]),multiply(A04,matrix([[0],[0],[1],[0]]))),matrix([[0,0,0,0,1,0,0,0]])
  ),Ja4):
> Ja6:=add(multiply(multiply(matrix([[1,0,0,0],[0,1,0,0],[0,
  0,1,0]]),multiply(A05,matrix([[0],[0],[1],[0]]))),matrix([[0,0,0,0,0,1,0,0]])
  ),Ja5):
> Ja7:=add(multiply(multiply(matrix([[1,0,0,0],[0,1,0,0],[0,
  0,1,0]]),multiply(A06,matrix([[0],[0],[1],[0]]))),matrix([[0,0,0,0,0,0,1,0]])
  ),Ja6):
> Ja8:=add(multiply(multiply(matrix([[1,0,0,0],[0,1,0,0],[0,
  0,1,0]]),multiply(A07,matrix([[0],[0],[1],[0]]))),matrix([[0,0,0,0,0,0,0,1]])
  ),Ja7):
> I1:=matrix([[I1x,0,0],[0,I1y,0],[0,0,I1z]]):
> H1:=add(m1*multiply(transpose(J1),J1),multiply(multiply( trans-
  pose(Ja1),I1),Ja1)):
> I2:=matrix([[I2x,0,0],[0,I2y,0],[0,0,I2z]]):
> H2:=add(m2*multiply(transpose(J2),J2),multiply(multiply( trans-
  pose(Ja2),I2),Ja2)):
> I3:=matrix([[I3x,0,0],[0,I3y,0],[0,0,I3z]]):
> H3:=add(m3*multiply(transpose(J3),J3),multiply(multiply( trans-
  pose(Ja3),I3),Ja3)):
> I4:=matrix([[I4x,0,0],[0,I4y,0],[0,0,I4z]]):
> H4:=add(m4*multiply(transpose(J4),J4),multiply(multiply( trans-
  pose(Ja4),I4),Ja4)):
> I5:=matrix([[I5x,0,0],[0,I5y,0],[0,0,I5z]]):
> H5:=add(m5*multiply(transpose(J5),J5),multiply(multiply( trans-
  pose(Ja5),I5),Ja5)):
> I6:=matrix([[I6x,0,0],[0,I6y,0],[0,0,I6z]]):
> H6:=add(m6*multiply(transpose(J6),J6),multiply(multiply( trans-
  pose(Ja6),I6),Ja6)):
> I7:=matrix([[I7x,0,0],[0,I7y,0],[0,0,I7z]]):
> H7:=add(m7*multiply(transpose(J7),J7),multiply(multiply( trans-
  pose(Ja7),I7),Ja7)):
> I8:=matrix([[I8x,0,0],[0,I8y,0],[0,0,I8z]]):
> H8:=add(m8*multiply(transpose(J8),J8),multiply(multiply( trans-
  pose(Ja8),I8),Ja8)):
> H:=add(add(add(add(add(add(add(H1,H2),H3),H4),H5),H6),H7), H8):
```

APPENDIX C

This appendix gives the Maple scripts for the derivation of equations and plots related to:

i. Figure 3.5

Sliding Mode Simulation for thesis.m

```
clear
global GAMMA LAMBDA eta1 eta2 Kd
m1=1;
l1=1;
me=2;
de=pi/6;
I1=0.12;
lc1=0.5;
Ie=0.25;
lce=0.6;
a1=I1+m1*lc1^2+Ie+me*lce^2+me*l1^2;
a2=Ie+me*lce^2;
a3=me*l1*lce;
a5=lc1*m1*9.81;
a6=l1*me*9.81;
a7=lce*me*9.81;
a=[a1 a2 a3 a5 a6 a7];
G=1;
GAMMA=G*diag([0.6 0.04 0.03 300 40 225]);
LAMBDA=20*eye(2);
eta1=0.1;
eta2=0.1;
Kd=70*eye(2);
t0=0;
tf=30;
ahatinit=[0 0 0 0 0 0]';
q0=[0 0 0 0 ahatinit']';
tic
[t,q]=ode23('phd2_sys',t0, tf, q0);
toc
q_d=[(1-cos(2*pi*t)) (2*(1-cos(2*pi*t)))]';
qtilde=[q(:,1) q(:,3)]'-q_d;
atilde=q(:,5:10)-ones(length(t),6)*diag(a');
junk=['Gamma=' num2str(G)];
figure(1)
subplot(221)
plot(t,qtilde(1,:)*180/pi,'-')
ylabel('Position Error angle1 (deg.)')
xlabel('Time (s)')
title(junk)
subplot(222)
plot(t,qtilde(2,:)*180/pi,'-')
ylabel('Position Error angle2 (deg.)')
xlabel('Time (s)')
title(junk)
subplot(223)
plot(t,atilde(:,1),'-')
ylabel('Parameter Error a1')
xlabel('Time (s)')
subplot(224)
plot(t,atilde(:,2),'-')
ylabel('Parameter Error a2')
xlabel('Time (s)')
```

```
figure(2)
subplot(221)
plot(t,atilde(:,3),'-')
ylabel('Parameter Error a3')
xlabel('Time (s)')
title(junk)
subplot(222)
plot(t,atilde(:,4),'-')
ylabel('Parameter Error a4')
xlabel('Time (s)')
title(junk)
subplot(223)
plot(t,atilde(:,5),'-')
ylabel('Parameter Error a4')
xlabel('Time (s)')
subplot(224)
plot(t,atilde(:,6),'-')
ylabel('Parameter Error a4')
xlabel('Time (s)')
```

phd2 sys.m

```
function qdot=f(t,q)
global GAMMA LAMBDA eta1 eta2 Kd
q_d=[(1-cos(2*pi*t)) (2*(1-cos(2*pi*t)))]';
qd_d=[2*pi*sin(2*pi*t) 4*pi*sin(2*pi*t)]';
qdd_d=[4*pi*pi*cos(2*pi*t) 8*pi*pi*cos(2*pi*t)]';
qtilde=[(q(1)-q_d(1)) (q(3)-q_d(2))]';
qtildedot=[(q(2)-qd_d(1)) (q(4)-qd_d(2))]';
qd_r=qd_d-LAMBDA*qtilde;
qdd_r=qdd_d-LAMBDA*qtildedot;
s=qtildedot+LAMBDA*qtilde;
Y11=qdd_r(1);
Y12=qdd_r(2);
Y13=cos(q(3))*(2*qdd_r(1)+qdd_r(2))-
sin(q(3))*(q(4)*qd_r(1)+q(2)*qd_r(2)+q(4)*qd_r(2));
Y15=cos(q(1));
Y16=cos(q(1));
Y17=cos(q(1)+q(3));
Y21=0;
Y22=qdd_r(1)+qdd_r(2);
Y23=cos(q(3))*qdd_r(1)+sin(q(3))*q(2)*qd_r(1);
Y25=0;
Y26=cos(q(1));
Y27=cos(q(1)+q(3));
Y=[Y11 Y12 Y13 Y15 Y16 Y17;Y21 Y22 Y23 Y25 Y26 Y27];
ahat=q(5:10);
tau=Y*ahat-Kd*qtildedot;
m1=1;
l1=1;
me=2;
de=pi/6;
I1=0.12;
lc1=0.5;
Ie=0.25;
lce=0.6;
a1=I1+m1*lc1^2+Ie+me*lce^2+me*l1^2;
a2=Ie+me*lce^2;
a3=me*l1*lce;
```

```
a4=0;
a5=lc1*m1*9.81;
a6=l1*me*9.81;
a7=lce*me*9.81;
H11=a1+2*a3*cos(q(3))+2*a4*sin(q(3));
H12=a2+a3*cos(q(3))+a4*sin(q(3));
H21=H12;
H22=a2;
h=a3*sin(q(3))-a4*cos(q(3));
H=[H11 H12;H21 H22];
C=[-h*q(4) -h*(q(2)+q(4));h*q(2) 0];
gravity=[-9.8*lc1*cos(q(1))*m1-9.8*(l1*cos(q(1))+lce*cos(q(1)+q(3)))*me -
9.8*(l1*cos(q(1))+lce*cos(q(1)+q(3)))*me]';
qdot(1)=q(2);
qdot(3)=q(4);
dynamics=[0 0]';
dynamics=inv(H)*(tau-C*[q(2) q(4)]' + gravity);
qdot(2)=dynamics(1);
qdot(4)=dynamics(2);
qdot(5:10)=-GAMMA*Y'*s;
```

APPENDIX D

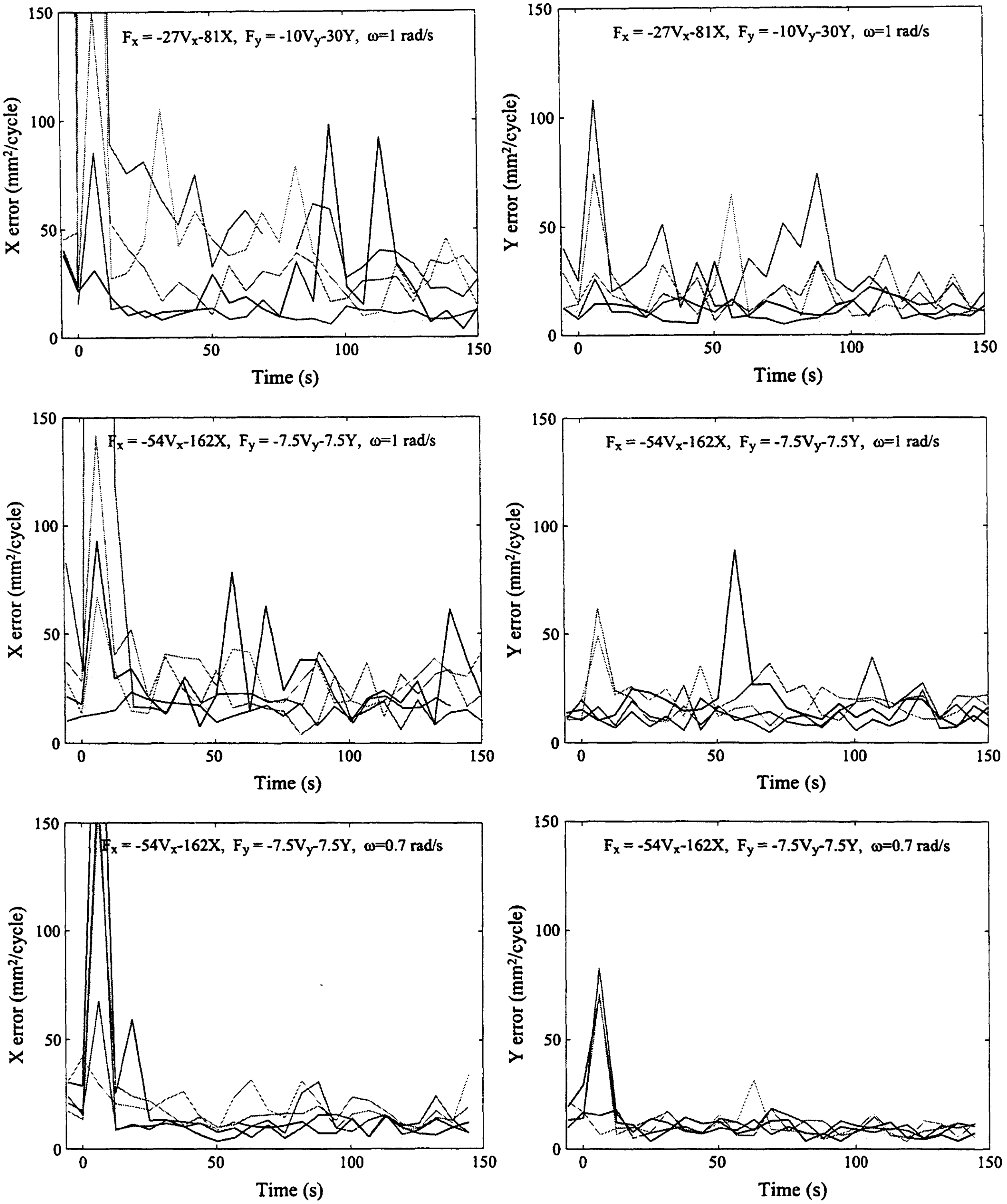


Figure 4.10 (Continued) Protocols 4 through 6 are shown here.

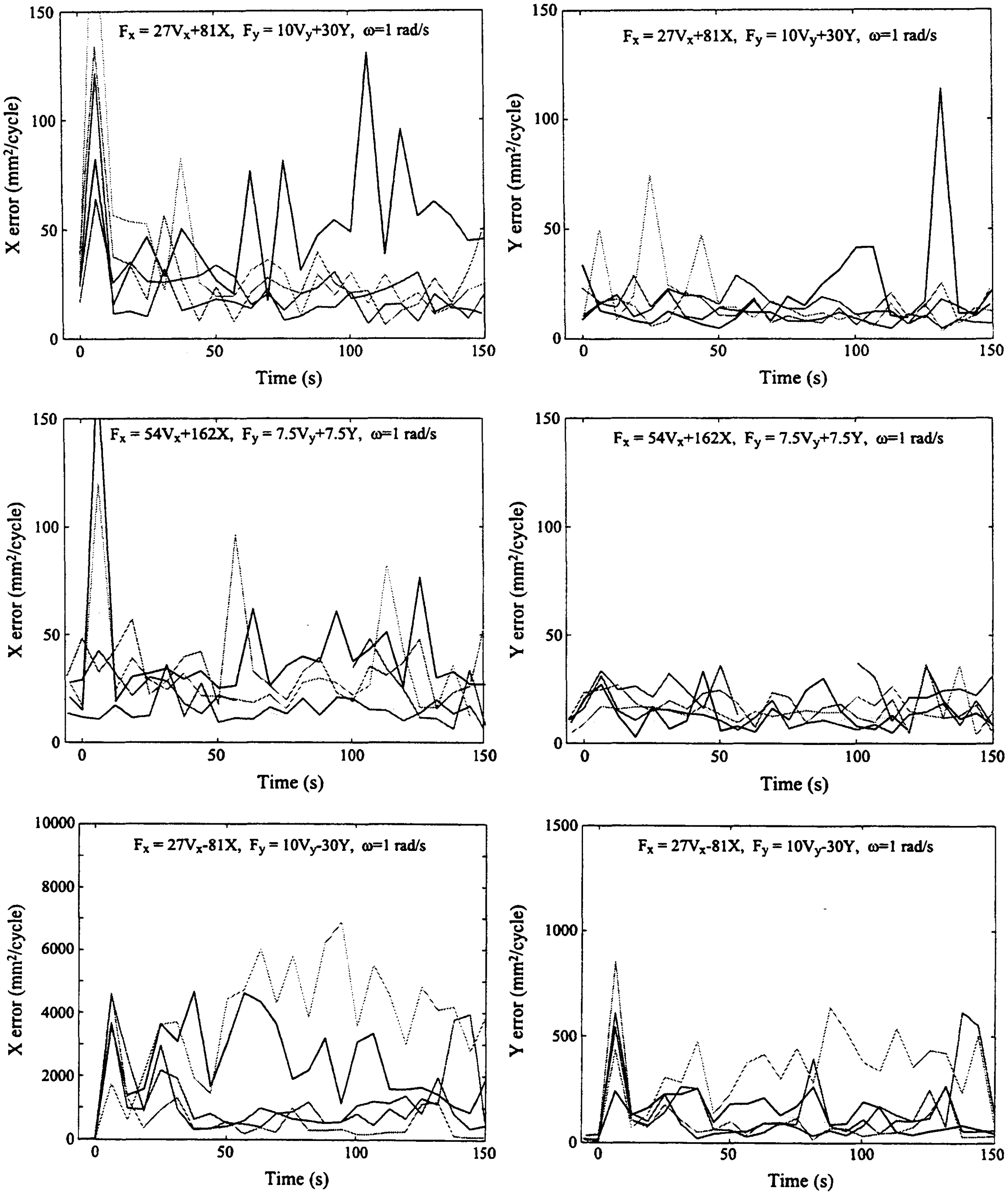


Figure 4.10 (Continued) Protocols 7 through 9 are shown here.

These Maple scripts for the derivation of equations and plots related to:

i. Figure 4.2 (The Matlab© simulation script that compares with the experimental data of the reaching experiment in the rotating room)

dizio matlab simulation.m

```
clear
global GAMMA Kd
G=2;
GAMMA=G*diag([0.03 0.05 0.1 0.3 0.1]);
Kd=60*[1 0;0 1];
t0=0;
tf=300;
ahatinit=[0 0 0 0 0]';
q0=[0.2527 0 2.6362 0 ahatinit']';
tic
[t,q]=ode23('dizio matlab simulation_sys',t0,tf,q0);
toc
time=t;
L=length(time);
vy=5*sin(time);
ay=5*cos(time);
y=10-5*cos(time);
q_d(:,2)=acos(y.^2/(2*10^2)-1);
qd_d(:,2)=-y.*vy./(10^2*sin(q_d(:,2)));
qdd_d(:,2)=-(10^2*(vy.^2+ay.*y).*sin(q_d(:,2))-
y.*vy.*cos(q_d(:,2)).*qd_d(:,2))./ ...
            (10^4*sin(q_d(:,2)).^2);
q_d(:,1)=atan((1+cos(q_d(:,2)))./sin(q_d(:,2)));
qd_d(:,1)=-qd_d(:,2).*cos(q_d(:,2)).^2./sin(q_d(:,2)).^2;
qdd_d(:,1)=-(qdd_d(:,2).*cos(q_d(:,1)).^2.*sin(q_d(:,2)).^2 ...
           -qd_d(:,2).*qd_d(:,1).*sin(2*qd_d(:,1)).*sin(q_d(:,2)).^2 ...
           -
qd_d(:,2).*qd_d(:,1).*sin(2*qd_d(:,2)).*cos(q_d(:,2)).^2)./sin(q_d(:,2)).^4;
qtilde=[q(:,1) q(:,3)]-q_d;
figure(1)
clf
subplot(211)
plot(time,q(:,1)*180/pi,'y', time, q_d(:,1)*180/pi,'ro')
ylabel('Position Angle 1')
xlabel('Time (s)')
hold on
subplot(212)
plot(time,q(:,3)*180/pi,'y', time, q_d(:,2)*180/pi,'ro')
ylabel('Position Angle 2')
xlabel('Time (s)')
hold on
figure(2)
clf
plot(10*cos(q(:,1))+10*cos(q(:,1)+q(:,3)),10*sin(q(:,1))+...
        10*sin(q(:,1)+q(:,3)),'y')
axis([-8 8 0 16])
grid
hold on
plot(10*cos(q_d(:,1))+10*cos(q_d(:,1)+q_d(:,2)),10*sin(q_d(:,1))+...
        10*sin(q_d(:,1)+q_d(:,2)),'ro')
```

dizio matlab simulation sys.m

```
function qdot=f(t,q)
global GAMMA Kd
LAMBDA=20*eye(2);
eta1=0.1;
eta2=0.1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
time=t;
L=length(time);
vy=5*sin(time);
ay=5*cos(time);
y=10-5*cos(time);

q_d(:,2)=acos(y.^2/(2*10^2)-1);
qd_d(:,2)=-y.*vy./(10^2*sin(q_d(:,2)));
qdd_d(:,2)=-(10^2*(vy.^2+ay.*y).*sin(q_d(:,2))-
y.*vy.*cos(q_d(:,2)).*qd_d(:,2))./ ...
            (10^4*sin(q_d(:,2)).^2);
q_d(:,1)=atan((1+cos(q_d(:,2)))./sin(q_d(:,2)));
qd_d(:,1)=-qd_d(:,2).*cos(q_d(:,2)).^2./sin(q_d(:,2)).^2;
qdd_d(:,1)=-(qdd_d(:,2).*cos(q_d(:,1)).^2.*sin(q_d(:,2)).^2 ...
           -qd_d(:,2).*qd_d(:,1).*sin(2*qd_d(:,1)).*sin(q_d(:,2)).^2 ...
           -qd_d(:,2).*qd_d(:,1).*sin(2*qd_d(:,2)).*...
            cos(q_d(:,2)).^2)./sin(q_d(:,2)).^4;
q_d=q_d';
qd_d=qd_d';
qdd_d=qdd_d';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
qtilde=[(q(1)-q_d(1)) (q(3)-q_d(2))]';
qtildedot=[(q(2)-qd_d(1)) (q(4)-qd_d(2))]';
qd_r=qd_d-LAMBDA*qtilde;
qdd_r=qdd_d-LAMBDA*qtildedot;
s=qtildedot+LAMBDA*qtilde;
Y11=qdd_r(1);
Y12=qdd_r(2);
Y21=qdd_r(2);
Y22=qdd_r(1);
Y13=qd_r(1);
Y14=qd_r(2);
Y23=qd_r(2);
Y24=qd_r(1);
Y15=2.5;
Y25=0.15;
Y=[Y11 Y12 Y13 Y14 Y15;Y21 Y22 Y23 Y24 Y25];
ahat=q(5:9);
tau=Y*ahat-Kd*s+vy*2*pi*5;                       %% Coriolis
m1=1;
l1=1;
me=2;
```

```
de=pi/6;
I1=0.12;
lc1=0.5;
Ie=0.25;
lce=0.6;
a1=I1+m1*lc1^2+Ie+me*lce^2+me*l1^2;
a2=Ie+me*lce^2;
a3=me*l1*lce*cos(de);
a4=me*l1*lce*sin(de);
H11=a1+2*a3*cos(q(2))+2*a4*sin(q(2));
H12=a2+a3*cos(q(2))+a4*sin(q(2));
H21=H12;
H22=a2;
h=a3*sin(q(2))-a4*cos(q(2));
H=[H11 H12;H21 H22];
C=[-h*q(4) -h*(q(2)+q(4));h*q(2) 0];
gravity=[-9.8*(2.5*cos(q(1))+0.15*cos(q(1)+q(2))) -
9.8*0.15*cos(q(1)+q(2))]';
qdot(1)=q(2);
qdot(3)=q(4);
dynamics=[0 0]';
dynamics=inv(H)*(tau-C*[q(2) q(4)]' - gravity);
qdot(2)=dynamics(1);
qdot(4)=dynamics(2);
qdot(5:9)=-GAMMA*Y'*s;
```

ii. This is the Matlab script used to analyze the digitized video footage from the throwing experiments. The velocities and angles of release were recorded.

Vidanal.m

```
clc
clf
clear
clear functions
%xpixpercm=6.4; (This was used for space subjects.  It was calculated)
%ypixpercm=6.4; (using length comparison from the Mir footage)
directory='This Directory Does NOT exist';
while exist(directory)~=7
  directory=input('Enter the working directory: ','s');
  clc
end
eval(['cd ''' directory '''']);
clc
current_file='This File Does NOT exist';
while exist(current_file)~=2
   eval(['dir ''' directory '''']);
   current_file=input('Now, enter the first file to analyze: ','s');
   if length(current_file)==0
      current_file='This File Does NOT exist';
   end
   clc
end
current_file=[current_file blanks(80-length(current_file))];
lcf=80;
for counter=80:-1:1
   junk=blanks(80-counter+1);
   if current_file(counter:80)==junk
      lcf=counter-1;
   end
end
dx=[];
dy=[];
FILES=[blanks(80)];
while eval(['exist(''' current_file(1:lcf) ''')'])==2 | current_file~=['0'
blanks(79)]
   clc
   clf
   eval(['[PIC,MAP]=imread(''' current_file(1:lcf) ''');']);
   image(PIC)
   colormap(MAP)
   if exist('xpixpercm')==0
      disp('Now Pick The Origin of the Calibration Figure')
      title('CALIBRATION - PICK THE ORIGIN')
      [xc0,yc0]=ginput(1);
      xc1=xc0;
      yc1=yc0;
      while xc1<=xc0
         disp('Now Pick The X=+11 inches point of the Calibration Figure')
         title('CALIBRATION - X=+11')
         [xc1,yc1]=ginput(1);
      end
      xc2=xc0;
      yc2=yc0;
      while yc2>=yc0
         disp('Now Pick The Y=+8.5 inches point of the Calibration Figure')
         title('CALIBRATION - Y=+8.5')
         [xc2,yc2]=ginput(1);
      end
      xpixpercm=(xc1-xc0)/(11*2.54*39/110)
      ypixpercm=-(yc2-yc0)/(8.5*2.54*39/110)
   end
   title(current_file(1:lcf))
   clear XS YS
   disp('Using the mouse chose the point on the subject to record!')
   [XS(:,1),YS(:,1)]=ginput(1);
   dx=[dx;XS/xpixpercm];
   dy=[dy;YS/ypixpercm];
   clc
   dir
   temp_file=input('Now, enter the NEXT file to analyze: ','s');
   if exist(temp_file)==2
      current_file=[temp_file blanks(80-lcf)];
   else
      disp(['V='
num2str(mean(sqrt((diff(dx)*27/100).^2+(diff(dy)*27/100).^2)))])
      disp(['q='
num2str(mean(atan((diff(dy)*27/100)./(diff(dx)*27/100)))*180/pi)])
         break
   end
end
```

iii. This is the C++ code that I wrote to control the manipuladum during the continuous trajectory tracking experiment. This only includes the code that I edited (rewrote). More code is necessary to compile this piece, but since those scripts where written by other authors, they are not included here.

```
//miket010.cpp


#include <conio.h>
#include <time.h>
#include <stdio.h>
#include "apoint.h"
#include "textdisp.h"
#include "manip.h"
#include "dataman4.h"
#include <iostream.h>
#include <fstream.h>
#include <string.h>
#include <bios.h>
#include <math.h>
#include <dos.h>
#include <graphics.h>
#include <stdlib.h>
#ifndef FALSE
   #define FALSE 0
   #define TRUE 1
#endif /* FALSE */
int read_targets(int targetx[], int targety[], int viax[], int viay[],
   int targetrec[], int targetvfb[], int ffield[], int gain[],
   float reachtime[], float waittime[], char colortarg[], int);
REAL t1=0.0, t2=0.0;
int read_matrix(float* matrix);
#define SAMP_FREQ 100   /* Hz */
#define INT( X ) int( floor( X + 0.5 ) )
#define scalex( O ) INT(-(O)/0.7+maxx/2.0)
#define scaley( O ) INT((O-450.0)/0.7+maxy/2.0)
#define scaledx( O ) INT(O/0.7)
#define max(x,y) ((x)>(y)?(x):(y))
#define mat_size 120
//size of input matrix x size
#define xinput_size 15
#define mm_per_grain 20
// origin of the field(absolute distance from center)
#define xmin_from_center 160
#define xmax_from_center 120
// opposite corner of the field
#define ymin_from_center 160
#define ymax_from_center 60

//#define sqr_size 1
extern Manipulandum manip;
extern TextDisplay tDisp;

//polhemus x y z data   <polhemus.c>
extern  double pol_x(void);
extern  double pol_y(void);
extern  double pol_z(void);
extern  void initrs232( void );
extern  void input(void);
int maxx, maxy;

void drawtarget(REAL xpos, REAL ypos)    // in robot coordinates
{
   int i, size = 4; /* mm */
   moveto(scalex(xpos+size/2),scaley(ypos-size/2));
   linerel(scaledx(size),0);
   linerel(0,scaledx(size));
   linerel(-1*scaledx(size),0);
   linerel(0,-1*scaledx(size));
}

void env(REAL xx, REAL xy, REAL yx, REAL yy)
{
   // input variables should be in N/m/sec, N/m, or Newtons
   // for cartesian viscosity, stiffness, and parallel field,
   // N.m/rad/sec, N.m/rad, or N.m for joint based items

   #define tottargets 1 //max targets (we set this for purpose of bounds on
vectors)
   #define Pi 3.141592654
   #define ERRMARGIN -20 //how close the cursor must be to the target to
count as close enough (used to be 10)
   #define bias_t1 0.0
   #define bias_t2 0.0

   ManagedData4 theData; //4 column data array
   REAL sh_angvbias, el_angvbias; //velocity biases from tachometer
   REAL xv, yv, fx, fy, fdir, staystillperiod,
          xpos, ypos, shv, elv, oldfx, oldfy, axx[903],
          ayy[903], c1, s1,
          tottime, delaytime, MVMTTIME /* in sec */;
          //extracted t1 and t2;
   float reachtime[tottargets], waittime[tottargets];
   REAL movestart_time, moveend_time, summove_time, timeattarget, oldxp,
oldyp, omega;
   int targetindex = -1, testindex = 0, testfileindex = 1; /* first file
number*/
   int targetx[tottargets+1], targety[tottargets+1],  viax[tottargets],
         viay[tottargets], gain[tottargets], //general targets
         targetrec[tottargets], targetvfb[tottargets], ffield[tottargets];
   int last_tarindex, odd_detector, index, axy[903], ayx[903];
   float matrix[mat_size];
   char file_info[20], colortarg[tottargets];

   int numtargetsexploded = 0;
   int done = FALSE, trace = FALSE, applyb = FALSE,
         eraseobject = FALSE, timetodrawobj = FALSE, objonscreen = FALSE,
         movedyet = FALSE, soundcount = 0, drawobjdelaycounter = 11, wait-
counter = 1000,
         numof_moves = 0, movepfileopen = FALSE, cursorON = FALSE;
   int keepcount, key, i, j, k, waitedbeepcount;
   char string[39],fname[20],subjid[5];
   int gdriver = DETECT, gmode;

   ostream *pOuts; /* Temporary local output stream */
   ostream *pOuts2;
   int polhemus=0; //toggle var for polhemus input

   //initialize polhemus serial port
   initrs232();

   // Initialize the data buffer, 40 sec max
   theData.init(300);

   cout << "Enter one-character subject identifier: ";
   cin >> subjid;
   subjid[1] = '\0'; // end of string

   if(!read_targets(targetx, targety, viax, viay, targetrec,
                      targetvfb, ffield, gain, reachtime, waittime, color-
targ,
                      tottargets)) {
         cout << "Error in reading targets from file. \n";
         return;
   }
   targetx[2] = targetx[1];
   targety[2] = targety[1];

   if(!read_matrix(matrix)) {
         cout << "Error in reading field information from file. \n";
         return;
   }

   // initialize graphics
   initgraph(&gdriver, &gmode, "");
   setfillstyle( SOLID_FILL, RED );
   maxx = getmaxx() + 1;
   maxy = getmaxy() + 1;
   cleardevice();
   settextjustify(LEFT_TEXT, BOTTOM_TEXT);
   settextstyle(DEFAULT_FONT, HORIZ_DIR, 3);
   setwritemode(XOR_PUT);

   //Clear the text screen's upper section
    for(i=0; i<10; i++) {
         tDisp .put(i,0, "
");
   }

   // describe the available commands
   tDisp.put(0,0,"  b: x-viscous N/m/s    w: j-viscous Nms/rad          p:
x-force N");
   tDisp.put(2,0,"  r: release fields     x: subject's kinematics");
   tDisp.put(3,0,"  t: start targets");
   tDisp.put(4,0,"  c: end targets");
   tDisp .put(5,0,"  q: QUIT                Max. Force recorded at Lord:
");
   tDisp.put(8,0,"Test Target Num. : 0
Current Field: NONE");
   tDisp.put(9,0,"Test Files Stored: 0");
   tDisp.put(10,0,"Movement time:   ");
   theData.activate();
   theData.init();         /* Initialize the data */
   manip.torque(0.0, 0.0);
   manip.update();             /* Update state */
   fx=0;
   fy=0;
```

```
xpos = manip.x();
ypos = manip.y();

// initialize counts
keepcount = 8;      // will keep every tenth point
index = 0;      // will keep every tenth point
tottime = 0.0;
timeattarget = 0.0;
staystillperiod = 0.0;
movestart_time = 900.0;        // some non-zero start time
summove_time = 0.0;
delaytime = 0.0;
omega = 1.0;

manip.clock.set(ONE_ms*(1000/SAMP_FREQ));         /* sampling period */
manip.clock.go();

// get shoulder and elbow tachometer biases
sh_angvbias = el_angvbias = 0.0;
i = 0;
while(manip.clock.wait() && (i<101)) {
      manip.update();
      sh_angvbias += manip.tj1();
      el_angvbias += manip.tj2();
      i += 1;
}
sh_angvbias = sh_angvbias/100.0;
el_angvbias = el_angvbias/100.0;

oldxp= xpos;
oldyp= ypos;
xpos = manip.x();
ypos = manip.y();


// The Main Loop
while(manip.clock.wait() && !done) {
      manip.update(); /* Update state */
      oldfx=fx;
      oldfy=fy;
      oldxp=xpos;
      oldyp=ypos;
      xpos = manip.x(); // mm
      ypos = manip.y();
      shv = manip.tj1()-sh_angvbias; // rad per sec
      elv = manip.tj2()-el_angvbias;
      xv = (shv*manip.jac[0][0] + elv*manip.jac[0][1])*0.001; // m/sec
      yv = (shv*manip.jac[1][0] + elv*manip.jac[1][1])*0.001;
      if(polhemus){
        input();
        xpos=pol_x();
        ypos=pol_y();
        xv=pol_z();
      }

      tottime += 0.01; // update total time (in seconds);
      keepcount++;
      if(keepcount == 9){
            axx[index] = xpos;      //x-position in mm
            ayy[index] = ypos;      //y-position in mm
            axy[index] = xv*1000;   //x-velocity in mm/s
```

```
               ayx[index] = yv;        //y-velocity in mm/s
               keepcount = -1;
               index++;
         }

         // erase the cursor
         if(cursorON) {
               setcolor(RED);
               rectangle(scalex(oldxp)-scaledx(2),scaley(oldyp)-scaledx(2),
               scalex(oldxp)+scaledx(2),scaley(oldyp)+scaledx(2));
               cursorON = FALSE;

         }

         // draw the target
         setcolor(WHITE);
         targetindex = 1;
         drawtarget(targetx[2],targety[2]);
         setcolor(WHITE);
         targetx[1] = -80*cos(omega*tottime);
         targety[1] = 440-80*sin(omega*tottime);
         drawtarget(targetx[targetindex],targety[targetindex]);
         targetx[2] = targetx[1];
         targety[2] = targety[1];
         // draw the cursor
         if((!trace && !cursorON) ||
               (trace && !cursorON && targetvfb[testindex])) {
               setcolor(RED);
               rectangle(scalex(xpos)-scaledx(2),scaley(ypos)-scaledx(2),
               scalex(xpos)+scaledx(2),scaley(ypos)+scaledx(2));
               cursorON = TRUE;
         }
         // save data if needed
         if(index==300)  {
               strcpy(fname, "set1"); /* Copy file prefix */
               strncat(fname, subjid, 79); /* Append subject ID */
               itoa(testfileindex, string,10);
               strncat(fname, string, 79);
               strncat(fname, ".dat", 79); /* Append file suffix */
               pOuts = new ofstream(fname);
               while (index>-1){
                       theData.addRec(axx[index], ayy[index], testfilein-
dex, axy[index]);
//                     theData.addRec(axx[index], ayy[index], axy[index],
ayx[index]);
                       index--;
               }
               *pOuts << theData; // write the data to file
               delete pOuts; // close the data file
               sprintf(string, "%d", testfileindex);
               tDisp.put(9,19,"    ");
               tDisp.put(9,19, string);
               theData.init();
               index=0;
               testfileindex += 1;
         }

         if (tottime<150) {
               fx = 0; //Newtons
               fy = 0;
         }
```

```
    if (tottime>=150 && tottime<300) {
        fx = 10; //Newtons
        fy = 0;
    }
    if (tottime>=300 && tottime<450) {
        fx = 0; //Newtons
        fy = 10;
    }
    if (tottime>=450 && tottime<600) {
        fx = 20; //Newtons
        fy = 0;
    }
    if (tottime>=600 && tottime<750) {
        fx = -27*xv - 81*xpos/1000.0 ; //Newtons
        fy = -10*yv - 30*(ypos-440)/1000.0 ;
    }
    if (tottime>=750 && tottime<900) {
        fx = -54*xv - 162*xpos/1000.0 ; //Newtons
        fy = -7.5*yv - 7.5*(ypos-440)/1000.0 ;
    }
    if (tottime>=900 && tottime<1050) {
        omega = 0.7;
        fx = -54*xv - 162*xpos/1000.0 ; //Newtons
        fy = -7.5*yv - 7.5*(ypos-440)/1000.0 ;
    }
    if (tottime>=1050) {
        done = TRUE;
    }

    t1 = (manip.jac[0][0] * fx + manip.jac[1][0] * fy)*0.001+bias_t1;
    t2 = (manip.jac[0][1] * fx + manip.jac[1][1] * fy)*0.001+bias_t2;

        // safety region: will not produce torques outside this win-
dow
    if(xpos>300||xpos<-300||ypos>650||ypos<250){
        t1=t2=0.0;
        tDisp.put(8,73,"None");
        applyb = FALSE;
    }
manip.torque(t1, t2); //give the computed torques
    if((keepcount == 3) && bioskey(1)) { // check every 50 msec
        key = bioskey(0)&0xff; // take the key
        switch(key) {
            case 'b':
                tDisp.put(8,76,"    ");
                tDisp.put(8,76,"B");
                applyb = TRUE;
            break;
            case 'c':
            break;
            case 'p':       //polhemus input on
                polhemus=(polhemus==1)?0:1;
                if(polhemus) tDisp.put(7,66,"Pol on");
                else  tDisp.put(7,66,"Pol off");
            break;
            case 'q':
                done = TRUE;
            break;
            case 'r':
                tDisp.put(8,76,"None");
```

```
                           applyb = FALSE;
                     break;
                     case 't':
                           if(trace) break;
                           if(cursorON) {     // turn off cursor
                                 setcolor(RED);
                                 rectangle(scalex(xpos)-scaledx(2),
                                       scaley(ypos)-
scaledx(2),scalex(xpos)+scaledx(2),
                                       scaley(ypos)+scaledx(2));
                                 cursorON = FALSE;
                           }
                     break;
               }
         } // end of if
   }   // end of while
   manip.torque(0.0,0.0);
   if(movepfileopen) delete pOuts2;     // close the file
   theData.deactivate();   /* Turn storage off everytime*/
   nosound();
   return;
   }
```

The bold letters in the code above indicate the instructions for the protocols. The ones shown here are for protocols one through six (i.e., session 1). For session 2 instructions simply replace that part of the code by the following piece.

```
if (tottime<150) {
      fx = 0; //Newtons
      fy = 0;
}
if (tottime>=150 && tottime<450) {
      fx = -27*xv + 81*xpos/1000.0 ; //Newtons
      fy = -10*yv + 30*(ypos-440)/1000.0 ;
}
if (tottime>=450 && tottime<750) {
      fx = -54*xv + 162*xpos/1000.0 ; //Newtons
      fy = -7.5*yv + 7.5*(ypos-440)/1000.0 ;
}
if (tottime>=750 && tottime<1050) {
      fx = 27*xv - 81*xpos/1000.0 ; //Newtons
      fy = 10*yv - 30*(ypos-440)/1000.0 ;
}
if (tottime>=1050) {
      done = TRUE;
}
```

APPENDIX E

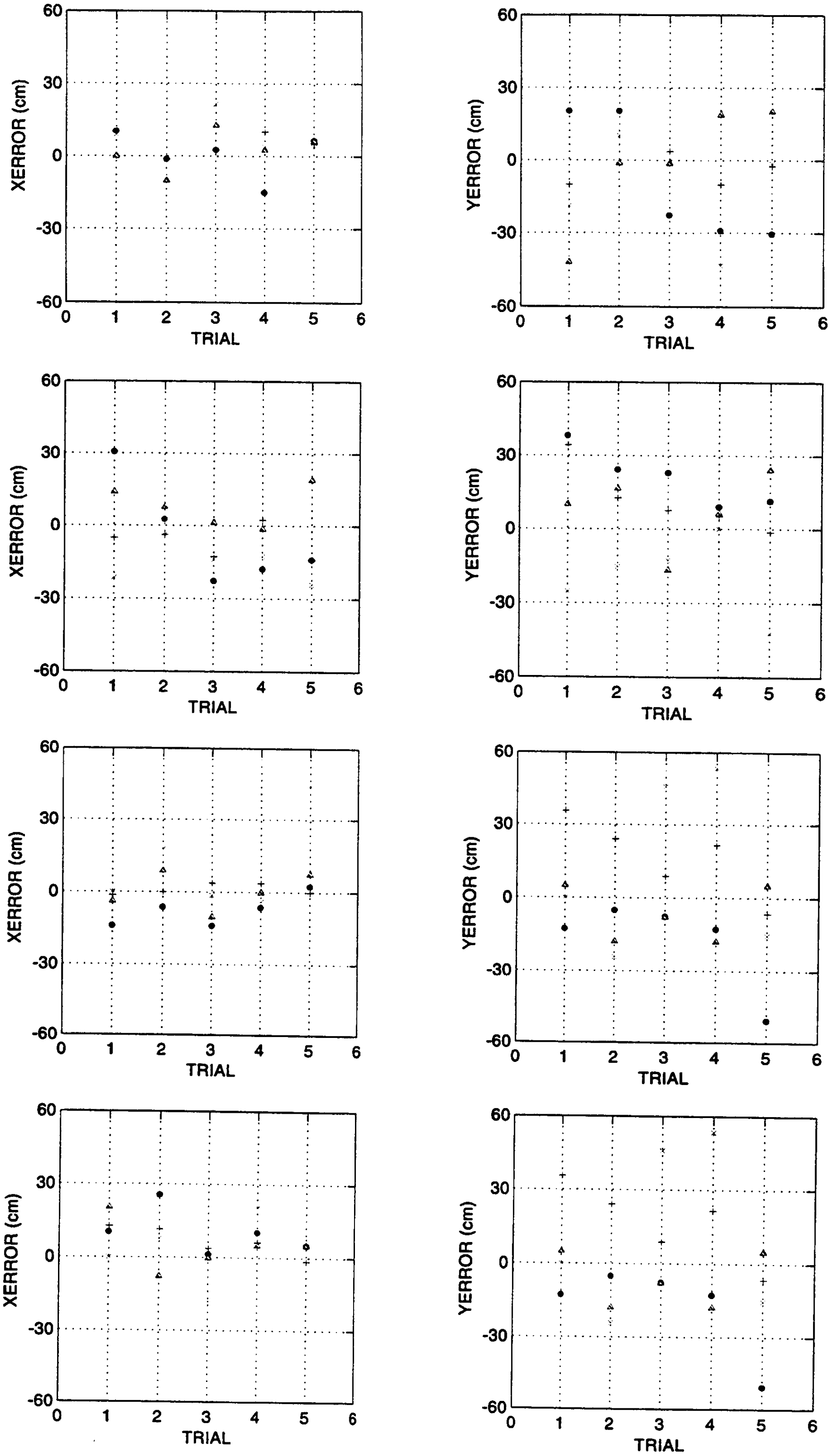


Figure 5.1 X and Y ground error data (Continued). Errors for subjects 5 through 8 are shown here. The subject numbers increase starting from the top row down.

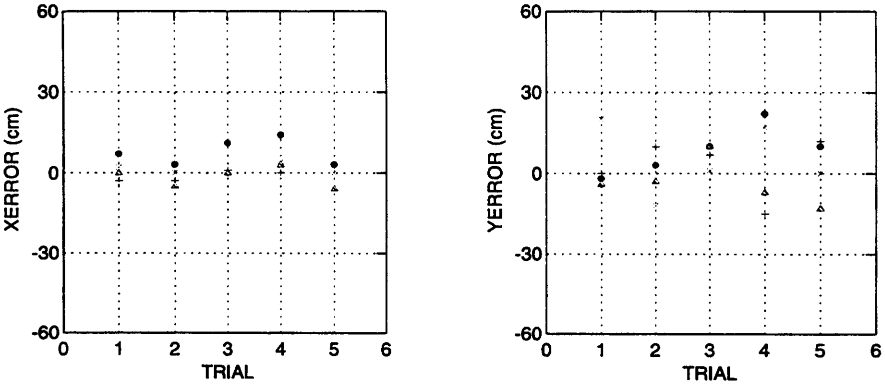
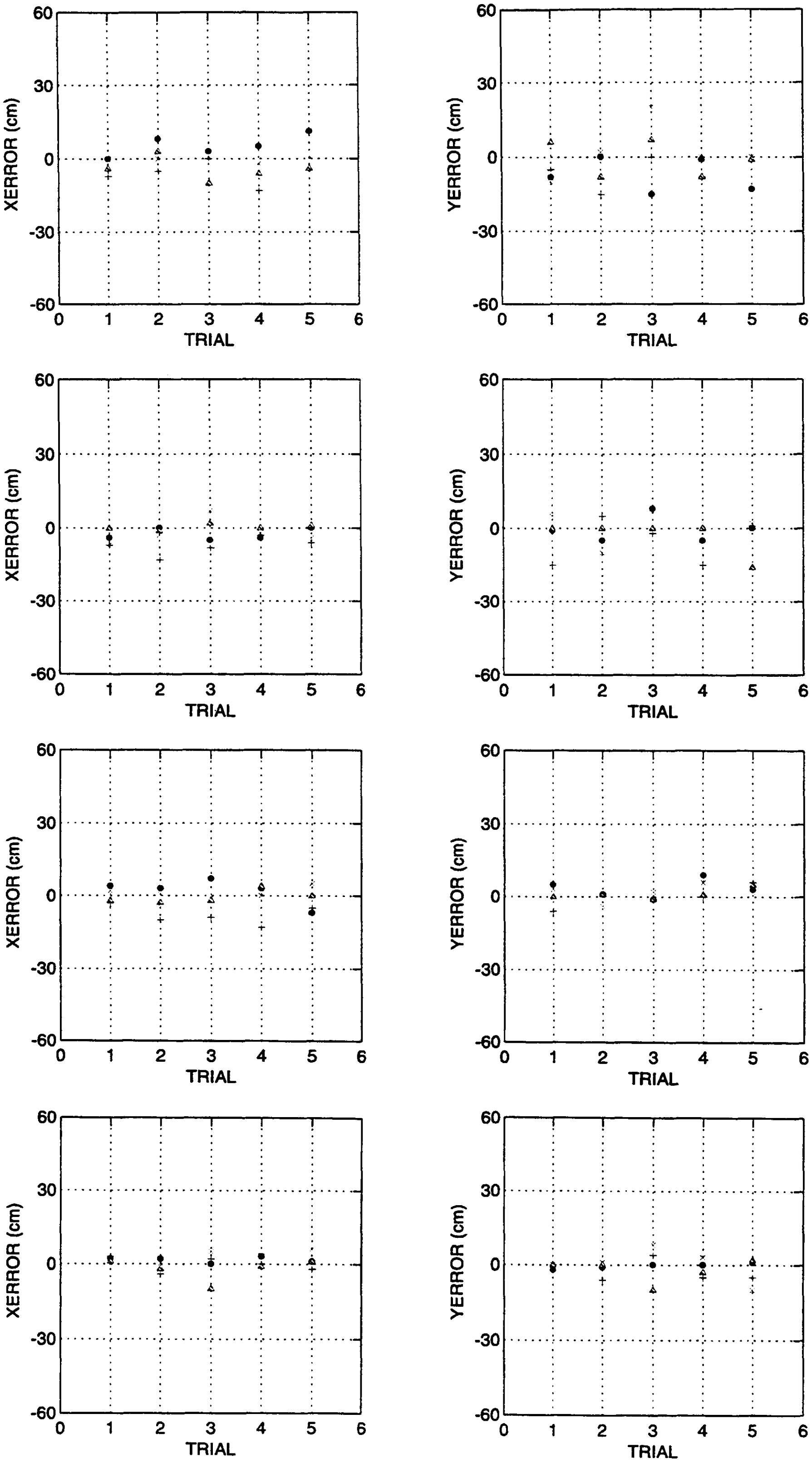
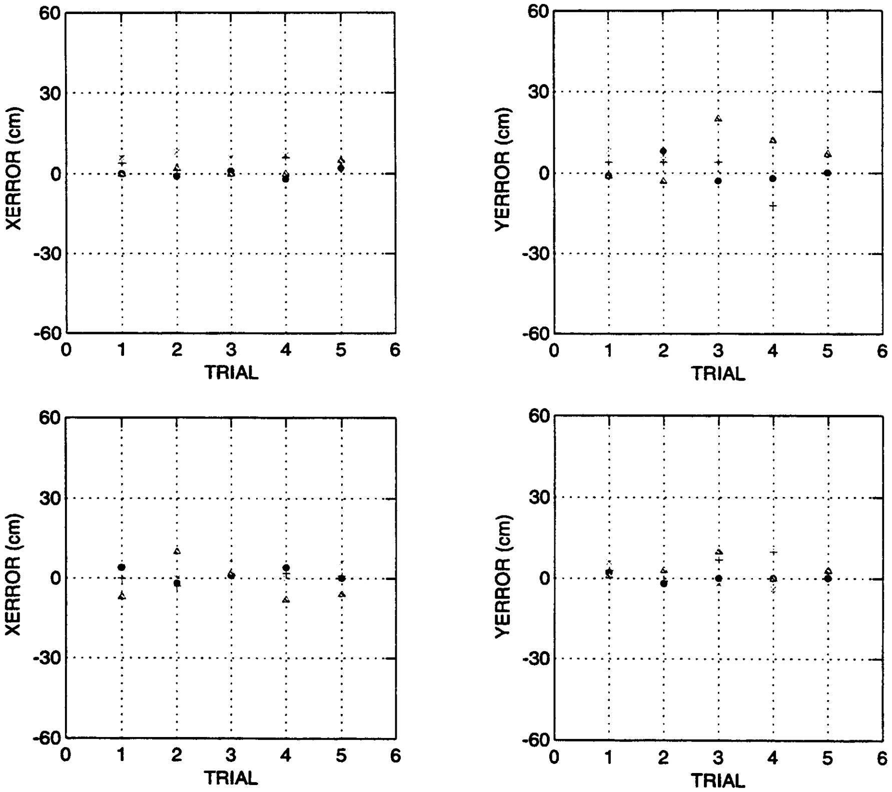


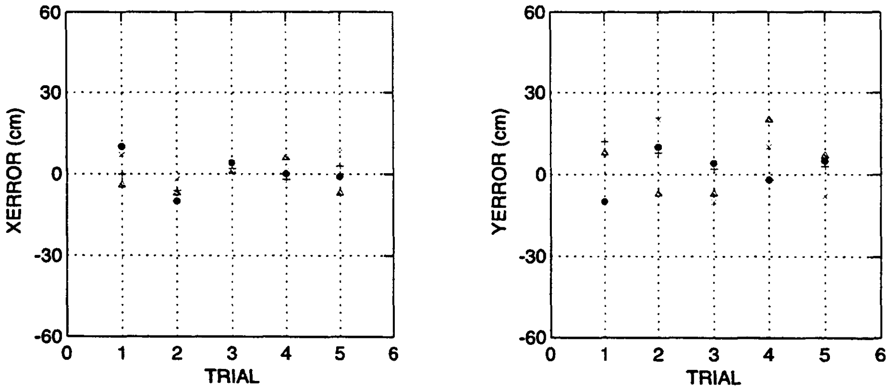
Figure 5.2 X and Y microgravity error data (Continued). B) for subject 2 during just one day.



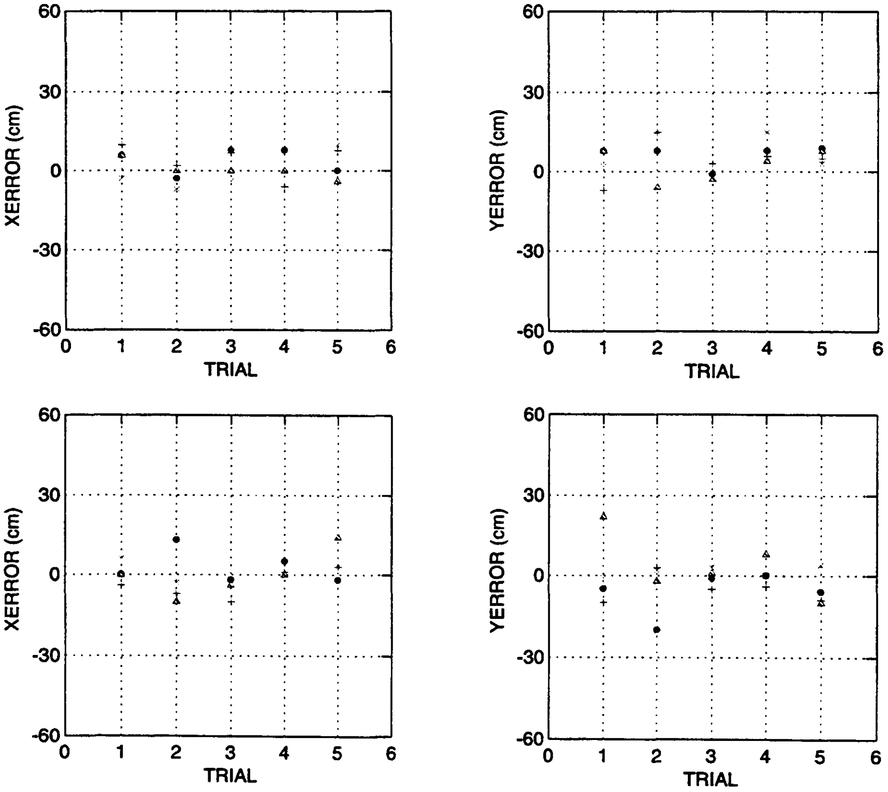
C) for subject 3 during four different days. Earlier dates are in the higher rows.



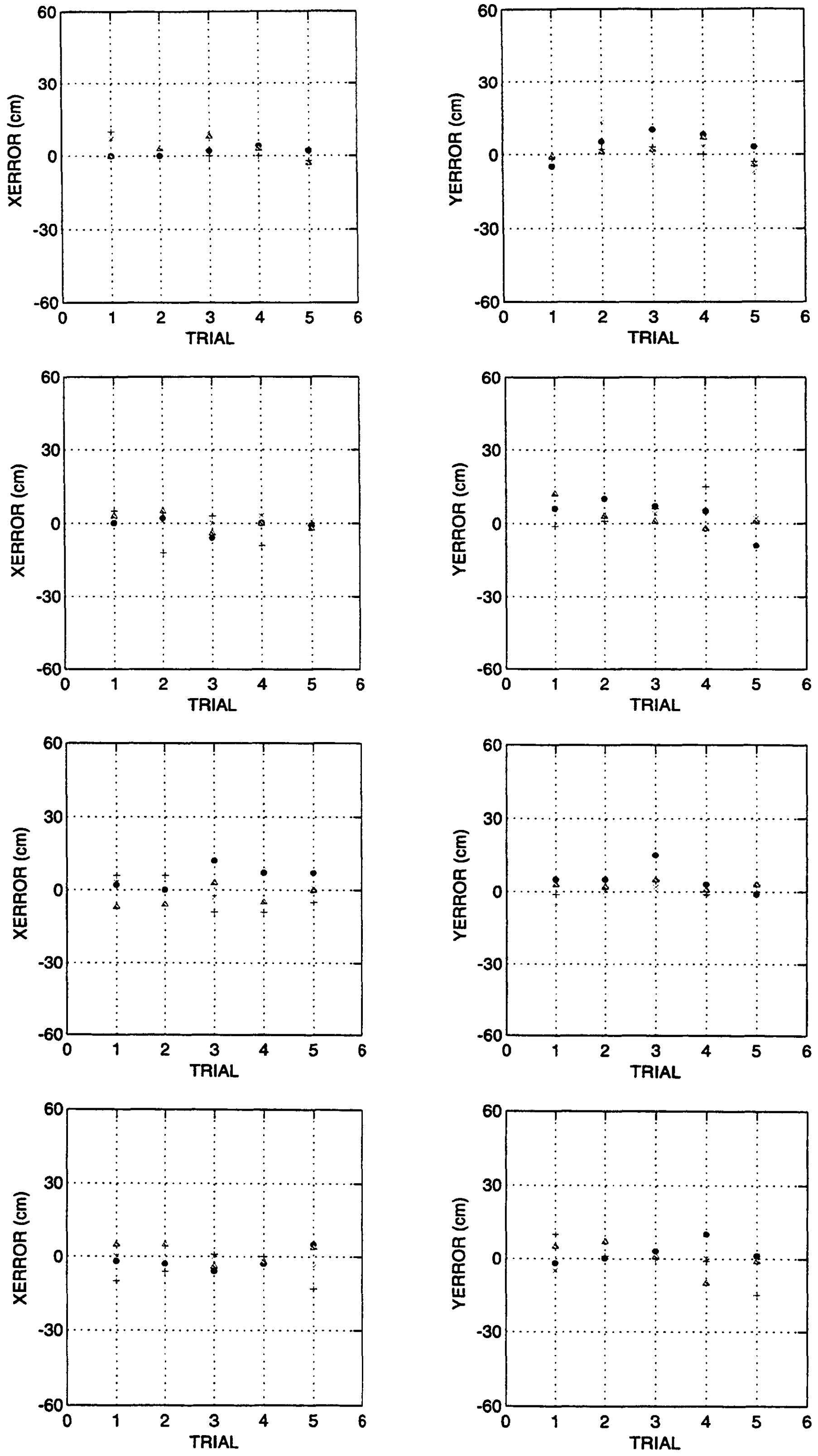
D) for subject 4 during two different days. Earlier dates are in the higher rows.



E) for subject 5 during just one day.



F) for subject 6 during two different days. Earlier dates are in the higher rows.



G) for subject 7 during four different days. Earlier dates are in the higher rows.

These are the Maple scripts for the derivation of equations and plots related to:

i. Figure 5.12

```
>y_g:=(-9.81*1.5^2/V_g^2+1.5*sin(2*q_g))/(1+cos(2*q_g)):#(Projectile equation for the ground)
>y_s:=(1.5*sin(2*q_s)/(1+cos(2*q_s))):#(Projectile equation for the space)
>s_q_g:=simplify(diff(y_g,q_g)^2):#(sensitivity of the throwing error with respect to the angle of release on the ground)
>s_q_s:=simplify(diff(y_s,q_s)^2):#(sensitivity of the throwing error with respect to the angle of release in space)
>s_v_g:=simplify(diff(y_g,V_g)^2):#(sensitivity of the throwing error with respect to the velocity of release on the ground)
>s_v_s:=simplify(diff(y_s,V_s)^2):#(sensitivity of the throwing error with respect to the velocity of release in space)
>q:=solve({y_g=-1},V_g):#(Calculate the angle of release on the ground necessary for to hit the target at a given velocity of release)
>V_g:=90/(2600*sin(2*q_g)^2+2400*sin(2*q_g))*(-218*sin(2*q_g)*(13*sin(2*q_g)+12)*(-3*sin(2*q_g)-2+2*cos(2*q_g)))^(1/2):
>q_s:=-17*Pi/180:#(For the space angle of release we used the measured angle of release)
>plot({s_v_g,s_q_g},q_g=-arctan(1/1.5)..7*Pi/16,0..10):
```

ii. Figure 5.13

```
>y_g:=(-22.0725/V_g^2+1.5*sin(2*q_g))/(1+cos(2*q_g)):#(Projectile equation
for the ground)
>y_s:=(1.5*sin(2*q_s)/(1+cos(2*q_s))):#(Projectile equation for the space)
>s_q_g:=simplify(diff(y_g,q_g)^2):#(sensitivity of the throwing error with
respect to the angle of release on the ground)
>s_q_s:=simplify(diff(y_s,q_s)^2):#(sensitivity of the throwing error with
respect to the angle of release in space)
>s_v_g:=simplify(diff(y_g,V_g)^2):#(sensitivity of the throwing error with
respect to the velocity of release on the ground)
>s_v_s:=simplify(diff(y_s,V_s)^2):#(sensitivity of the throwing error with
respect to the velocity of release in space)
>q:=solve(y_g=-1,q_g):q_g:=q[1]:#(Calculate the angle of release on the
ground necessary for to hit the target at a given velocity of release)
>q_s:=-17*Pi/180:#(For the space angle of release we used the measured an-
gle of release)
>c_g_1:=evalf(diff((eta1*V_g^q1*s_q_g+s_v_g*(V_g^q2)),V_g)):#(Constraint 1
for ground throws - The derivative of the cost becomes 0 at the obseved
conditions)
>c_g_2:=evalf(sqrt(eta1*V_g^q1*s_q_g+s_v_g*(V_g^q2))):#(Constraint 2 for
ground throws - The cost equals the observed values at the obseved release
conditions)
>c_s_1:=evalf(sqrt(eta1*V_s^q1*s_q_s+s_v_s*(V_s^q2))):#(Constraint 1 for
space throws - The cost equals the observed values at the obseved release
conditions)
>Z:=11:V_g:=3.4:V_s:=0.66:sols:=solve({c_g_1=0,c_g_2=17/Z,c_s_1=6/Z},{q1,q2
,eta1}):alpha:=subs(sols,q1):beta:=subs(sols,q2):eta:=subs(sols,eta1):
>Y_G:=(-9.81*1.5^2/V_G^2+1.5*sin(2*Q_G))/(1+cos(2*Q_G)):S_Q_G:= sim-
plify(diff(Y_G,Q_G)^2):S_V_G:=simplify(diff(Y_G,V_G)^2):Q:=solve(Y_G=-
1,Q_G):Q_G:=Q[1]:C_G_2:=Z*sqrt(eta2*V_G^alpha*S_Q_G+S_V_G*
V_G^beta):Y_S:=(1.5*sin(2*Q_S))/(1+cos(2*Q_S)):S_Q_S:=simplify(diff(Y_S,Q_S
)^2):S_V_S:=simplify(diff(Y_S,V_S)^2):Q_S:=17*Pi/180:C_S_1:=Z*sqrt(S_Q_S*et
a2*V_S^alpha):
>eta2:=eta*1:C11:=C_G_2:C21:=C_S_1:eta2:=eta:C12:=C_G_2:C22:=C_S_1:eta2:=et
a/1:C13:=C_G_2:C23:=C_S_1:plot({(C11,C12,C13)},V_G=0..8,0..30):plot({(C21,C
22,C23)},V_S=0..8,0..30):
```