CONTROL FORCE CHANGE DUE TO ADAPTATON OF FORWARD MODEL IN HUMAN MOTOR CONTROL

by

Tie Wang

A thesis submitted to The Johns Hopkins University in conformity with the requirements for the degree of Master of Science in Engineering

Baltimore, Maryland

July 15, 2000

Abstract

Learning to make point-to-point reaching movements in a curl force field was used as a paradigm to explore the system architecture of the human motor adaptive controller. The concept of internal model, a system for predicting behavior of the controlled movements, is divided into a forward and an inverse model. The existence and learning ability of the inverse model in the brain is more straightforward than forward model. I try to ask whether forward mode does play a role in the human control system and what happened to the forward model during the adaptation of the controller? A forward model transforms an efferent copy of descending commands into a prediction of the current state, position and velocity of the arm. Our work has focused on a forward model of dynamics as a role of feedback control. We compute control force change caused by perturbation in the null field and in the force field. From the control architecture we provided in this thesis, the control force change is only due to computation result of forward model and spinal reflex feedback. In order to compare the control force change in the null field and in the force field in the same state, a parameterized model was built for the null field to compute what the control force change would be in the state of force field. We see the adaptation of control force change to exactly compensate the expected the force field. This suggests that forward model plays a dominant role in the motor control and learning.

Contents

Abstract 3

Contents 3

1 Understanding of the Computational Motor Control for Human Arm Movements 1

1.1 Motor Planning 1

1.1.1 Optimal Control Approach and Kinematic Cost Function 2

1.1.2 Minimum Jerk Trajectory 3

1.2 Internal Models 4

1.2.1 Forward Model 4

1.2.2 Inverse Model 6

1.2.3 State Estimation 7

1.3 An Example Model of Human Arm Control Using Forward-Inverse Model Feedback Control 7

1.4 Adaptive Motor Control using Internal Models 8

2 Modeling of the Human Motor Control 10

2.5 Kinematics of the Human Arm 10

2.6 Dynamics of the Human Arm 13

2.7 Mathematical Modeling of the Control of Human Arm 15

2.8 Impedance of the Human Arm Controller 17

2.9 Curl Force Field 19

2.10 Savitsky-golay Smoothing and Derivative 21

2.11 Principal Component Analysis Methods 23

3 Experimental Apparatus and Protocol 28

3.12 Introduction 28

3.13 Material and Methods 31

3.13.1 Experimental Setup 31

3.13.2 Experiment Procedures 33

3.14 Data Acquisition and Preprocessing 36

3.15 Data analysis: Formation of the Internal Model 42

3.16 Estimating Inertial Model of the Human Arm 44

3.17 Predicting the Un-perturbed Trajectory 51

4 Data Analysis and Results 61

4.18 Introduction 61

4.19 Mathematical Modeling of Human Motor Control 62

4.20 Calculation of Control Force Produced by the Human Controller 70

4.21 Control Force Change Caused by Perturbations 71

4.22 Modeling the Control Force Change in the Null Field 73

4.23 Adaptation of the Control Force Change after Learning the Novel Dynamics Environment 89

5 Discussion and Conclusion 104

Chapter 1

Current Understanding of the Computational Motor Control for Human Arm Movements

A current controversy in motor control is whether the CNS makes use of an internal model of the motor apparatus in planning and executing goal-directed movements. A number of investigators have suggested that an internal models is used either to predict the movement consequences of motor commands (forward model) (Jordan and Rumelhart, 1992; Miall et al., 1993; Jordan et al., 1994; Wolpert et al., 1995) or determine the commands needed to achieve a desired movement trajectory (inverse model) (Saltzman, 1979; Atkeson, 1989; Uno et al., 1989; Hollerbach, 1990). However, other workers have proposed control theories that explicitly reject the notion of an internal model (Bizz et al., 1984; Flash, 1987; Bullock and Grossberg, 1988; Feldman et al., 1990; Flanagan et al., 1993). We introduce some of the computational approaches that have been developed in the area of motor control. We focus on areas of motor control, which have been enriched by control system models: motor planning, internal model, motor prediction and motor learning.

1.1 Motor Planning

The computational problem of motor planning arises from a fundamental property of the motor system; the reduction in the degrees of freedom from neural commands through muscle activation to movement kinematics. Even for the simplest of tasks, such as moving the hand to a target location, there are an infinite number of possible paths that the hand could move along and for each of these paths there are an infinite number of velocity profiles (trajectories) the hand could follow. Having specified the hand path and velocity, each location of the hand along the path can be achieved by multiple combinations of joint angles and, due to the overlapping actions of muscles and the ability to co-contract, each arm configuration can be achieved by many different muscle activations. Motor planning can be considered as the computational process of selecting a single solution or pattern of behavior at the levels in the motor hierarchy, from the many alternatives, which are consistent with the task.

Optimal Control Approach and Kinematic Cost Function

One computational framework, which is natural for such a selection process, is optimal control in which a cost function is chosen in order to evaluate quantitatively the performance of the system under control. The cost function is usually defined as the integral of an instantaneous cost, over a certain time interval, and the aim is to minimize the value of this cost function. Every possible solution, that is each possible movement, has an associated cost and the solution with the lowest cost is selected as the plan. In this framework the cost function is a mathematical means for specifying the plan. The variables that appear in the cost function, and that are therefore planned, determine the patterns of behavior observed.

While many possible cost-function have been examined there are two main classes of model proposed for point-to-point movements: kinematics and dynamics based models. Here only the kinematics based models are introduced and used for the computation of a desired movement or trajectory for the arm. Kinematic cost function contain only geometrical and time-based properties of motion and variables of interest are the

positions (e.g. joint angles or hand Cartesian coordinates) and their corresponding velocities, acceleration and higher derivatives.

1.1.2 Minimum Jerk Trajectory

Based on the observation that point-to-point movements of the hand are smooth when viewed in a Cartesian framework, it was proposed that the squared first derivative of Cartesian hand acceleration or 'jerk' is minimized over the movement. The minimum jerk hypothesis produces a unique solution given the movement duration and suitable boundary conditions of initial and final position and velocity. The model predicts straight-line Cartesian hand paths with bell-shaped velocity profiles, which are consistent with empirical data for rapid movements made without accuracy constraints.

Assume that we know where our arm is currently, x_0 or q_0, we know where we want our arm to be at the end of movement, x_f or $\mathrm{q_f}$, how should the arm/hand move from start to final position? Infinite numbers of trajectories are possible. There is "regularity" in the way people move.

Hypothesis: movements are planned so that hand path is minimum in jerk, i.e., maximally smooth. Assume $x(t)$ is position,

$$J = \int_0^T \left[\frac{d^3 x}{dt^3} \right]^2 dt$$

The trajectory $x(t), t = 0...T$ that minimizes above function is a trajectory that is a minimum jerk. If $x_0 = x(0), x_f = x(f)$ and given $\dot{x}, \ddot{x} = 0$ at start and end of movement, we have

$$x(t) = x_0 + (x_f - x_0)\left[10(\frac{t}{T})^3 - 15(\frac{t}{T})^4 + 6(\frac{t}{T})^5\right]$$

1.2 Internal Models

The concept of an internal model, a system for predicting behavior of a controlled process, is central to the current theories of motor control and learning. Theoretical studies have proposed that internal models may be divided into two varieties: forward models and inverse models. The forward model accomplishes the transformation from motor variables to sensory variables by the environment and the musculoskeletal system (these physical systems transform efferent motor actions into reafferent sensory feedback). It is also possible, however, to consider internal transformation, implemented by neutral circuitry, that mimic the external motor-to-sensory transformation. Such internal transformations are known as internal forward models. Forward dynamic models predict the next state (e.g. position and velocity) given the current state and the motor command whereas forward output models predict the sensory feedback. This in contrast to inverse model which invert the system by providing the motor command required to achieve some desired result they have a natural use as a controller. Based on computational principles, this classification is relevant for adaptive control of a nonlinear system. The CNS appears to learn control of the arm through the formation of both forward and inverse internal model of the environment's mechanical dynamics. Memory consolidation maybe related to transform of one kind of internal model into another (Bhushan & Shadmehr, 1999). In this chapter, control methods based on inverse and forward modalities that the brain can use for controlling arm movements are introduced.

1.2.1 Forward Model

The feedback forward model control is a method where the brain uses sensory feedback information about the state of the arm from vision and proprioception to generate or modify the motor commands sent to the arm based on error in the measured state and the desired trajectory. This method places prime emphasis on the role of joint stiffness and viscosity for the generation of movements by the central nervous system (CNS). The forward dynamics model refers to a hypothetical computational network in the brain and has been defined as an internal model that mimics the casual flow of a process by predicting its next state given the current state and the motor commands (Miall and Wolpert, 1996). The forward model is a model of the input-output mapping of the human arm from muscle activation to arm movement. It can be estimated as an estimate of the forward dynamics of the human arm, which predicts hand acceleration from neural signal and hand state (position and velocity).
The feedback forward model control of a system has the following advantage:

- It is robust to noise in the plant and changes in the plant
- With high feedback gains it is possible to emulate the inverse dynamics of the system with a simple linear controller and achieve close to exact tracking of the desired trajectory.

Its disadvantage:

- Extremely sensitive to noise in sensory measurements and feedback
- Affected greatly by time delays in the feedback loop
- The actual trajectory can never track the desired trajectory exactly

In the human motor system, feedback control is very attractive, humans have to interact with different environments that continually alter the dynamics of the system they are trying to control. And, visual control of movement is a form of feedback control that is very useful. However, time delays in the feedback loop severely limit the scope of feedback control and do not allow simple feedback control of the system. The reason is that information about the outcome of a control action is not available instantly and has to

go through a delay before reaching the controller. By the time the control action is taken it may no longer be appropriate for dealing with the current errors in the output of the system. To overcome delays in the feedback, what is required is a method to obtain the current state or output of the system without having to wait for it to feed back. For stable feedback control the brain has to compute the state of the arm at current time t from a delayed measurement of the state at time $t - t_0$, where t_0 is the feedback delay, and a history of motor commands sent out by the brain until the current time t. In control literature, a computational unit that estimates or predicts current state is called an observer. Hence an observer has to be designed to solve the time-delay problem. In the context of movements, the concept of a Forward Model has been proposed to construct an observer and achieve feedback control in the presence of time delays in the system (Miall and Wolpert, 1996).

1.2.2 Inverse Model

This refers to a control method by the brain that uses only the predetermined desired trajectory to generate control signals for movement of the arm. It is a feedforward control system and does not rely on feedback during the movement. Stability of the system is achieved by the spinal reflex loop and equilibrium properties of the muscle. An important distinction of this method to feedback methods discussed later is that desired trajectory directly drives the system without the use of intermediate variables like muscle activation or torque. The brain does not have independent control over torque or muscle activation generated in the system and controls only the trajectory. Muscle can be used as a trajectory controller or as a torque generator. In the feedforward methods discussed here, the controller relies on the trajectory control properties of the muscle.

1.2.3 State Estimation

Delays in forward model feedback cause instability and uncertainty. Therefore, human motor control produces its estimate of the current state by monitoring the stream of input (motor neural signal) and sensory feedback at some earlier time. Based on the current state estimate and the estimated error in trajectory, the desired trajectory is corrected using a linear feedback controller.

1.3 An Example Model of Human Arm Control Using Forward-Inverse Model Feedback Control

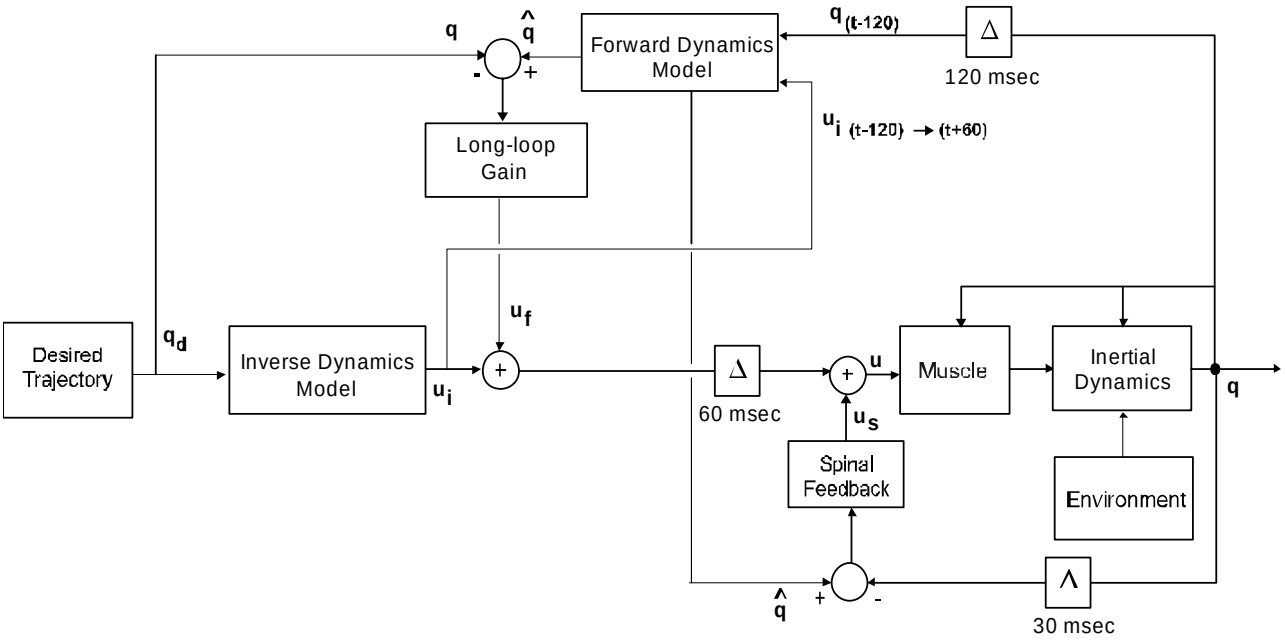


Figure 1.1: Block diagram illustrating the control method using both feedforward and feedback control

The collective system of “muscle” and manipulated “Inertial Dynamics” is often referred as the “plant”, which is the controlled object. The plant transforms the neural motor command into the trajectory (position and velocity) of the human arm. One of the problems associated with feedback control is that of tracking the desired trajectory

exactly. With feedback gains that are not sufficiently high, the actual trajectory only approximates the desired trajectory as was seen with only feedback control of movement for neural activation and torque control methods. The other disadvantage with above methods was that of adapting to altered dynamics by changing the forward model, which improved performance but still caused errors in tracking the desired trajectory. One way to overcome this drawback is to combine inverse model feedforward control with feedback control. In the new configuration, the feedforward signal controls the system and the feedback signal corrects for unmodeled disturbance to the system, hence this method integrates the advantage of both the feedforward and feedback techniques.

The properties of this control method are:

1. Use of an inverse model to generate the feedforward signals, and a forward model to generate the estimates for feedback control. Hence the control on both the forward and inverse plant models.
2. Exact tracking the desired trajectory can be achieved.
3. Stable control of the arm in the presence of external force fields.
4. Adaptation to the altered dynamic environments through changes in the forward and inverse models.

Greater sensitivity of performance of the system to changes or errors in the forward model that to the inverse model.

1.4 Adaptive Motor Control using Internal Models

Adaptive control of a nonlinear system which has large sensory feedback delays, such as the human arm, can be accomplished by using two different internal model architectures. One method uses only an adaptive inverse dynamics model to control the system (Shadmehr and Mussa-Ivaldi, 1994). The adaptive controller is feedforward in nature and ignores delayed feedback during the movement. The control system is stable because it relies on the equilibrium properties of the muscle and the spinal reflex to correct for any deviations from the desired trajectory. The other uses a rapidly adapting forward

dynamics model and delayed sensory feedback in addition to an inverse dynamics model to control arm movements (Miall and Wolpert, 1996). In this case, the corrections to deviations from the desired trajectory are a result of a combination of supraspinal feecback as well as spinal/muscular feedback. For reaching movements of the hand in novel force fields, only the learning of the forward model results in key characteristics of performance that match the kinematics of human subjects. In contrast, the adaptive control system that relies only on the inverse model fails to produce the kinematic patterns observed in the subjects, despite the fact that it is more stable (Bhushan N, Shadmehr R, 1999).

Chapter 2

Modeling of the Human Motor Control

2.1 Kinematics of the Human Arm

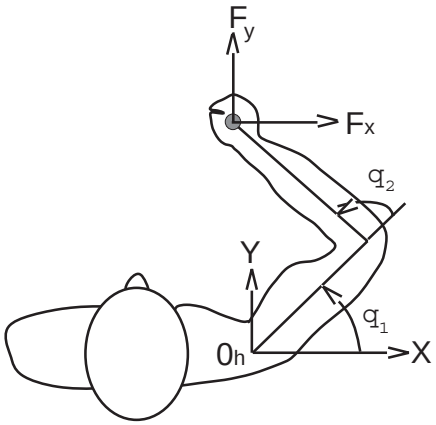


Figure 2.1: Schematic picture of the arm model. The upper arm (angle q_1) and the forearm (angle q_2). The hand with Cartesian coordinates (x, y). An external force (F_x, F_y) acting on the hand from the robot handle.

The kinematics of the human arm refer to the configuration relationships between joint positions and hand positions and the transformation between these two coordinates system. Human arm movements in the horizontal plane are described by 2-dof arm model which is depicted in Fig. 2.1. The upper arm and forearm are presented by rigid links that rotate in 1-dof joints, modeling the shoulder and elbow joint. The following are the equations that govern the forward kinematics of the arm and represent Cartesian or hand state in terms of joint state. The transformation from joint angles to handle position (in subject Cartesian coordinates) is given by

$$x = l_1 \cos q_1 + l_2 \cos(q_1 + q_2)$$
$$y = l_1 \sin q_1 + l_2 \sin(q_1 + q_2)$$

$$\dot{x} = -l_1 \dot{q}_1 \sin q_1 - l_2 (\dot{q}_1 + \dot{q}_2) \sin(q_1 + q_2)$$
$$\dot{y} = l_1 \dot{q}_1 \cos q_1 + l_2 (\dot{q}_1 + \dot{q}_2) \cos(q_1 + q_2)$$

The last two equations can be represented in vector notation by:

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

$$\mathrm{x} = \begin{bmatrix} x \\ y \end{bmatrix}; \quad q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \quad j(q) = \frac{dx}{dq} = \begin{bmatrix} -l_1 \sin q_1 - l_2 \sin(q_1 + q_2) & -l_2 \sin(q_1 + q_2) \\ l_1 \cos q_1 + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \end{bmatrix}$$

where,

x, y are the Cartesian x-y hand (handle) position

$\dot{x}, \dot{y}$ are the Cartesian hand (handle) velocity

q_1, q_2 are the relative shoulder and elbow joint angles

$\dot{q}_1, \dot{q}_2$ are the relative shoulder and elbow joint velocities

J is the Jacobian of joint to Cartesian coordinate transform

l_1, l_2 are the upper and lower arm lengths respectively

It is possible to express the inverse kinematics relationship that represents the joint angles as a function of hand position for the human arm. A unique joint position q exists for a given hand position in the workspace, when planar movements are considered. The uniqueness of the solution is ensured because of the constraint on the joint angle, $0 \le q_2 \le \pi$. The solution is given by the following equations,

$$q_2 = \cos^{-1}\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right)$$

$$q_1 = q_h - \sin^{-1}\left(\frac{l_2 \sin q_2}{\sqrt{x^2 + y^2}}\right)$$

$$\dot{q} = J^{-1}(q)\dot{x}$$

where,

q_h is the angle made by the hand with respect to the x-axis and is equal to $\arctan(x, y)$

J^{-1} is the inverse of 2×2 matrix J

J^{-1} exists except at the boundary of the workspace where $q_1 = q_2$. This singularity is ignored here because the movements of the arm considered in the current study are well within the boundaries of the workspace, and it is assumed that J^{-1} exists at all points during a movement of the arm. The equation relating hand acceleration $\ddot{x}$ joint acceleration $\ddot{q}$, is obtained by differentiating Eq. 1.1 that gives,

$$\ddot{x} = J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q}$$

where,

$$\dot{J} = \frac{dJ}{dt} = \frac{dJ}{dq}\frac{dq}{dt}$$

2.2 Dynamics of the Human Arm

The dynamics of the arm refers to the interaction between forces in the system and change of state of the system. A torque acting on the joints causes a change in the joint position and velocity. For the two-link two-joint system in Fig 2.1, the dynamics can be represented in terms of the link lengths and mass and inertia of the links with these equations. To estimate inertia parameters by shaking human arm, the following equations can be utilized:

$$T(q,\dot{q},\ddot{q}) \equiv H(q)\ddot{q} + C(q,\dot{q})\dot{q}$$

Here, $T = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$ (τ_1 and τ_2 are the relative shoulder and elbow torque respectively). The subscript '1' and '2' represent shoulder and elbow respectively. τ_1 and τ_2 represent the net torque produced by all the muscles about each joint. $H(q)$ and $C(q,\dot{q})$ denote the inertial matrix (2×2) and coriolis-centrifugal force vector respectively.

$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}$$

where,

$$\begin{aligned} H_{11} &= i_{1c} + i_{2c} + m_1 l_{c1}^2 + m_2(l_1 + l_{c2}^2) + 2m_2 l_1 l_{c2}\cos q_2 \\ &= a_1 + a_2 + 2a_3 \cos q_2 \\ H_{12} &= H_{21} = i_{2c} + m_2 l_{c2}^2 + m_2 l_1 l_{c2}\cos(q_2) \\ &= a_2 + a_3 \cos q_2 \\ H_{22} &= i_{2c} + m_2 l_{c2}^2 = a_2 \end{aligned}$$

$$C = \begin{bmatrix} -m_2 l_1 l_{c2} \dot{q}_2 \sin q_2 & -m_2 l_1 l_{c2} (\dot{q}_1 + \dot{q}_2) \sin q_2 \\ -m_2 l_1 l_{c2} \dot{q}_1 \sin q_2 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} -a_3 \dot{q}_2 \sin q_2 & -a_3 (\dot{q}_1 + \dot{q}_2) \sin q_2 \\ -a_3 \dot{q}_1 \sin q_2 & 0 \end{bmatrix}$$

Here, m_1 and m_2 denote the masses of upper arm and lower arm (hand plus forearm) links, l_{1c} and l_{2c} denote the length from each joint to the center of gravity for each link, i_{1c} and i_{2c} denote the inertia of each link, and l_1 and l_2 denote the length of the upper arm and forearm. These parameters can be merged into three parameters a_1, a_2 and a_3. It is obvious that

$$a_1 = i_{1c} + m_1 l_{c1}^2 + m_2 l_1^2$$
$$a_2 = i_{2c} + m_2 l_{c2}^2$$
$$a_3 = m_2 l_1 l_{c2}.$$

We call these parameters ‘structural parameters’ since they are independent of arm position, velocity, acceleration and torque. They are constant values under all conditions. So we can pre-estimate $a_{i,} i = 1,2,3$, the inertial models of human arms, and then use it to compute the dynamics for reaching movement of the human arm.

It is also possible to represent the torque in terms of hand velocities and accelerations as,

$$T = HJ^{-1}\left[\ddot{x} - \dot{J}J^{-1}\dot{x}\right] + CJ^{-1}\dot{x} \qquad (1.1)$$

The forward dynamics of the arm is the functional relationship that gives change in state of the hand in terms of the input joint torques. This is given by,

$$\ddot{x} = JH^{-1}\left[T - CJ^{-1}\dot{x}\right] + \dot{J}J^{-1}\dot{x} \tag{1.2}$$

and can be expressed in a simpler form by a nonlinear function f_D,

$$\ddot{x} = f_D(T, x, \dot{x}) \tag{1.3}$$

The relationship that gives the torque required for moving the arm from one point to another along a certain trajectory is called the inverse dynamics of the arm and is represented by Equation (1.1) or simply as f_D^{-1}. If the hand position, velocity and acceleration are given for any instant of time then torque can be computed using Equation 1.1.

Note: The representation of kinematics and dynamics of the human arm is in relative joint coordinate system throughout this thesis. This implies that the joint angles and joint torques for the human arm are expressed in relative joint coordinates.

2.3 Mathematical Modeling of Human Arm Motor Control

It is clear from our simulation studies and those of others (Hollerbach and Flash 1982) that the generation of coordinated multijoint arm movements requires the CNS to account for joint interactional effects. Several different strategies for controlling multijoint movements have been advanced in the literature. Proponents of the equilibrium point hypothesis (Bizzi et al. 1984; Feldman 1966) believe that the CNS controls movement by defining a sequence of equilibrium positions for the limb (the virtual trajectory). According to this hypothesis, muscle torques arise from the interaction of limb stiffness and differences between the actual and virtual trajectories, without explicit solution of the inverse dynamics problem. Rule-based control schemes (Gottlieb et al. 1997; Karst and Hasan 1991) also allow the CNS to circumvent the computation of inverse dynamics.

Alternatively, and more plausibly in our view, the CNS may utilize an internal model of limb dynamics to transform the desired movement kinematics into appropriate torque or muscle activation patterns (Atkeson 1989). Such a model, perhaps acquired through "motor learning," would provide explicit predictive control of segmental interactions. Several recent studies have provided evidence that the CNS utilizes internal models in the control of motor behavior (Flanagan and Wing 1997; Lackner and Dizio 1994; Lacquaniti et al. 1992; Sainburg et al. 1999; Shadmehr and Mussa-Ivaldi 1994; Wolpert et al. 1995). In particular, the study of Sainburg et al. suggests that the anticipatory control of intersegmental dynamics is achieved using an internal model of the intrinsic dynamics of the limb. The neural correlates of this model remain unclear. Schweighofer et al. (1998) have proposed a distributed representation of limb dynamics in which the motor cortex provides compensation for the inertial anisotropy of the limb and the cerebellum accounts for segmental interactions. This representation is consistent with recent studies of patients with cerebellar lesions that have linked trajectory disturbances to an impaired control of interaction torques (Bastian et al. 1996; Topka et al. 1998).

The purpose of the mathematical modeling was to help describe the concept of an "internal model". Let us start by considering the arm's dynamics in generalized coordinates: We indicate by q a point in configuration space (e.g., an array of joint angles) and by $\dot{q}$ and $\ddot{q}$ its first and second time derivatives. The dynamics of the motor control system coupled (in parallel) with its environment can be described by the sum of the following terms: a time-invariant component, $\Psi(q,\dot{q},\ddot{q})$. $\tau_{\text{ext}}(t)$, represents forces which depend on dynamics of the environment. $M(q,\dot{q},\mu)$ represents the forces which depend on the operation of the controller.

$$\Psi(q,\dot{q},\ddot{q}) = M(q,\dot{q},\mu) + \tau_{\text{ext}}(t) \tag{1.4}$$

If we assume the arm to be rigid body serial link system, the forces represented by Ψ is itself a sum of inertial, Coriolis, centripetal, and friction forces:

$$\Psi(q,\dot{q},\ddot{q}) \cong H(q)\ddot{q} + C(q,\dot{q})\dot{q} \tag{1.5}$$

Here, $\Psi(.)$ denotes a two-link arm dynamics, and $q, \dot{q}$ and $\ddot{q}$ are angular positions ($q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$, where q_1 is shoulder angle and q_2 is elbow angle), velocity and acceleration vector, respectively.

With respect to the function $M(q,\dot{q},\mu)$, simulations have previously suggested a reasonable lumped model of the subject's biomechanical motor controller in the case of point-to-point movements is as follows (Shadmehr and Brashers-Krug, 1997):

$$M = \hat{H}(q_d)\ddot{q}_d + \hat{C}(q_d,\dot{q}_d)\dot{q}_d - E(q_d) - K(q - q_d) - B(\dot{q} - \dot{q}_d) \tag{1.6}$$

where $\hat{H}$ and $\hat{C}$, which is the inverse model built by human motor controller for the dynamic of human arm, are the approximation of H and C.

2.4 Impedance of the Human Arm Controller

The mathematical modeling of the two-link human arm dynamics on the horizontal plane were modeled by the following second-order nonlinear differential equations:

$$\Psi(q,\dot{q},\ddot{q}) = M(q,\dot{q},\mu) + \tau_{ext}(t)$$

$\tau_{ext}(t)$ denotes the external force acting on the hand from the robot handle. Considering the length-tension and velocity-tension relationships of muscle force, the generated torque, $M(q,\dot{q},u)$, can be represented as a function of angular position, velocity, and motor command, u, descending from the supraspinal central nervous system (CNS). Let us consider the human arm controller is capable of guiding a limb along a desired

trajectory $q_0(t)$. When arm follows a particular trajectory $q_0(t)$, we assume that $\tau = \tau_0(t)$ and $\mu = \mu_0(t)$,

$$\Psi(q_0, \dot{q}_0, \ddot{q}_0) = M(q_0, \dot{q}_0, \mu_0) + \tau_0(t) \tag{1.7}$$

If human arm is perturbed $\Delta q(t)$ from the desired trajectory $q_0(t)$ by applying small and smooth perturbations for a short duration, the following variational equations can be utilized. Impose a perturb force $\Delta\tau$ resulting in $q_0 + \Delta q$ and $u_0 + \Delta u$,

$$\Psi(q_0 + \Delta q, \dot{q}_0 + \Delta\dot{q}, \ddot{q}_0 + \Delta\ddot{q}) = M(q_0 + \Delta q, \dot{q}_0 + \Delta\dot{q}, \mu_0 + \Delta\mu) + \tau_0(t) + \Delta\tau(t) \tag{1.8}$$

$$(1.8) - (1.7): \ \Psi(q_0 + \Delta q) - \Psi(q_0) = M(q_0 + \Delta q, \dot{q}_0 + \Delta\dot{q}, u_0 + \Delta u) - M(q_0, \dot{q}_0, u_0) + \Delta\tau$$

Using Fourier Expansion, we have

$$M(q_0 + \Delta q, \dot{q}_0 + \Delta\dot{q}, u_0 + \Delta u) \approx M(q_0, \dot{q}_0, u_0) + \left.\frac{dM}{dq}\right|_{q_0} \Delta q + \left.\frac{dM}{d\dot{q}}\right|_{\dot{q}_0} \Delta\dot{q} + \left.\frac{dM}{du}\right|_{q_0} \Delta u + \ldots$$

$$\Psi(q_0 + \Delta q) - \Psi(q_0) \approx \left.\frac{dM}{dq}\right|_{q_0} \Delta q + \left.\frac{dM}{d\dot{q}}\right|_{\dot{q}_0} \Delta\dot{q} + \left.\frac{dM}{du}\right|_{u_0} \Delta u + \Delta\tau$$

K and B represent muscle stiffness and viscosity matrix whose size is 2×2, so we have:

$$\text{Stiffness: } \left.\frac{dM}{dq}\right|_{q_0} \equiv K = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix},$$

$$\text{Viscosity: } -\left.\frac{dM}{d\dot{q}}\right|_{\dot{q}_0} \equiv B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}.$$

$$\Psi(q_0 + \Delta q) - \Psi(q_0) \approx -K\Delta q - B\Delta\dot{q} + \left.\frac{dM}{du}\right|_{u_0} \Delta u + \Delta\tau$$

We can measure external torque change acting on the human arm from the environment. By measuring inertial dynamics of the arm: $\Psi(q, \dot{q}, \ddot{q}) = H(q)\ddot{q} + C(q, \dot{q})\dot{q}$ and predicting the path $q_0(t)$ that the human arm would follow if there were no perturbations, we can estimate the torque or force $-K\Delta q - B\Delta\dot{q} + \left.\frac{dM}{du}\right|_{u_0} \Delta u$ that was generated by the human motor to activate and control reaching movements of human arm

2.5 Curl Force Field

In some cases of the experiments, the manipulandum was programmed to produce forces on the hand of subject as the subject performed reaching movements. These forces, indicated by the vector F, was computed as a function of the velocity of the hand:

$$F = B\dot{x}$$

where $\dot{x}$ was the hand instant velocity and B was a constant matrix representing viscosity of the imposed environment in subject Cartesian coordinates. The viscosity matrix B we used in the experiment has the following format:

$$B = \begin{bmatrix} 0 & k \\ -k & 0 \end{bmatrix}$$

It is viscous (proportional in strength to the instantaneous hand velocity) and directed orthogonal to the instantaneous hand velocity. This force field is considered here because

human motor behavior in this filed is used as the main source of data for understanding human motor control and adaptation in this thesis.

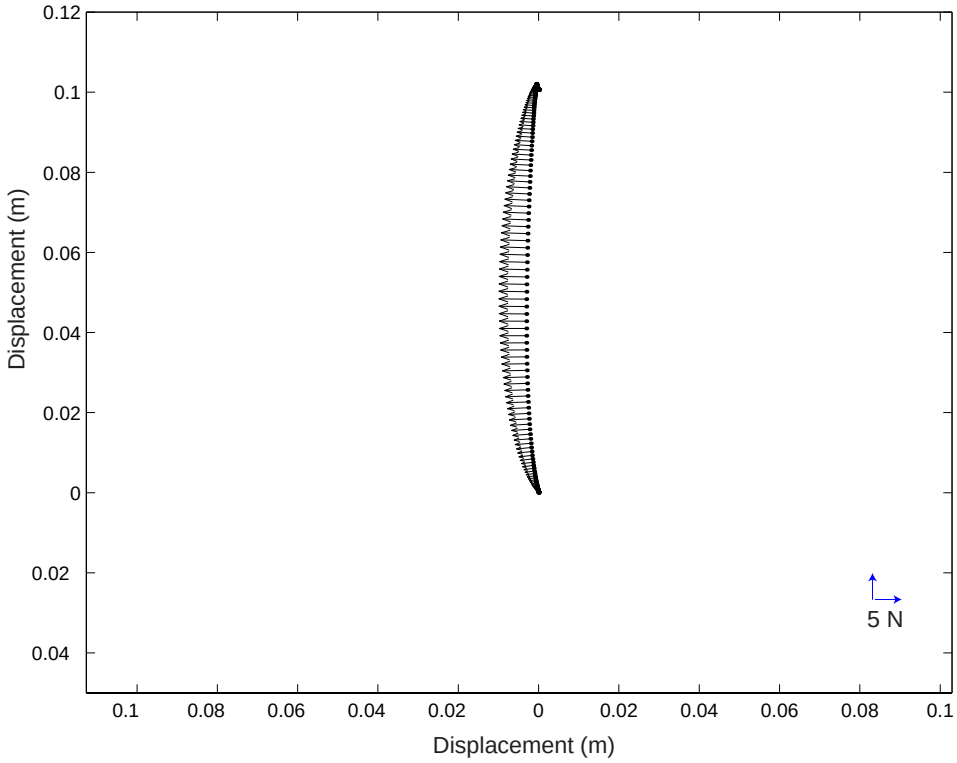


Figure 2.2: An environment as described by the curl force field with the viscosity matrix equals to $\begin{bmatrix} 0 & -13 \\ 13 & 0 \end{bmatrix} N.\sec/m$. Forces acting on the hand while making reaching center-up movements. The movement here is typical subject averaged data with a period of 0.5 sec and amplitude of 10 cm.

The forces field is chosen because it provides a new dynamic environment that has not been previously encountered by the subjects, and therefore their behavior and performance in this filed is unaffected by previous learning of other every day tasks. A wealth of data on human learning in this field has been collected in our laboratory and will be used in subsequent chapters. Hence it is important the structure and nature of this novel dynamic perturbation. The curl force field causes a force on the hand that is

perpendicular and proportional to the hand velocity at any instant. The work done by the field is always zero; therefore it does not affect the energy of the system. The interaction force acting on the arm due to the curl force field can be mathematically represented as, In particular, we chose B to be:

$$B = \begin{bmatrix} 0 & -13 \\ 13 & 0 \end{bmatrix} \; N.\text{sec}/m$$

The torque change required with the altered dynamics for reaching movement is significant compared to the torque required for the unload arm. As a subject made reaching movements center-up in this force field, the robot handle produced forces shown in Figure 2.2 (the movement we used here are the average of reaching movements of a typical subject).

2.6 Savitsky-Golay Smoothing and Derivative

The Savitsky-Golay algorithm is based on performing a least squares linear regression fit of a polynomial of degree k over at least $k+1$ data points around center point in the spectrum to smooth the data. The derivative is then the derivative of the fitted polynomial at the center point. Since the coefficients of the fitted polynomial are linear in the data values, the filters can be precomputed. "Order of the filter k" is the highest polynomial power used in the fit. "Half-width of the filter" are the number of points to the right and left of the center point (note: if the number of points to the right is not equal to the number of points to the left, a non-symmetric filter will be generated; otherwise, symmetry is assumed and the total width of the filter will be $2 \times halfwidth + 1$). Given order and half-width of the filter, Outputs, we can precompute the coefficients of this filter. We construct a coefficients matrix C whose each row represents the coefficients used to calculate the corresponding derivative at the center point. For example, this matrix has dimensions $(order + 1) \times N$, where N is the total width of the filter

$$N = 2 \times halfwidth + 1 .$$

To compute the smoothed estimate of the value at the center point, take the dot product of the first row of C with the data around the center point. Similarly, for the first derivative, use the second row of C, etc.

Note: When using the Savitsky-Golay filter to compute derivatives, dot product need to be divided by the sampling interval raised to the power of the desired derivatives.

The calculation uses the matrix formalism described above to calculate 1st through d th derivatives.

Legend:

d : the order of the derivative

k : the degree pf the polynomial

s : the number of points to be fitted by the polynomial

P : An s -element array with values [-m,...,0,..., m]

θ : A k -element array parameters

y : the array of actual data points

$$X = \begin{bmatrix} 1 & \dots & 0 & \dots & 1 \\ -m & \dots & 0 & \dots & m \\ \dots & \dots & 0 & \dots & \dots \\ (-m)^{k+1} & \dots & 0 & \dots & m^{k+1} \end{bmatrix}$$

Then $\mathrm{y} = \mathrm{X}\theta$ and the least squares fit is given by minimizing

$$S = (y - X\theta)^T (y - X\theta)$$

which is given by the condition: $\frac{dS}{d\theta} = 0$. This yields

$$\theta = (X^T X)^{-1} X^T y = Ty$$

where $T = (X^T X)^{-1} X^T$. The d th derivative is then given by $d!$ times the $(d+1)th$ rows if the T convolved with the trace data:

$$\frac{d^d y}{dx^d} = (d!) \sum_{j=-m}^{+m} y_i T_{d+1,i-j} \ .$$

where $m = (s-1)/2$. Note that this convolution truncates the trace by m points on each side.

2.7 Principal Component Analysis Methods

The basic concept of principle component is introduced in this section. This statistics tool is used in Chapter 3 to predict what the movement would be if there is no perturbation. In real samples, there are usually many different variations that make up a spectrum: the constituents in the sample mixture, inter-constituent interactions, instrument variations such as detector noise, changing environmental conditions that affect the baseline and absorbance, and differences in sample handling. Yet, even with all of these complex changes occurring, there should be some finite number of independent variations occurring in the spectral data. Hopefully, the largest variations in the calibration set would be the changes in the spectrum due to the different concentrations of the constituents of the mixtures. If it were possible to calculate a set of "variation spectra" that represented the changes in the sample data at all the wavelengths in the spectra, then this data could be used instead of the raw spectral data for building the calibration model. There should be fewer common variations than the number of calibration spectra (in most cases), and thus, the number of calculations for the calibration equations will be reduced as well.

Presumably, the "variation spectra" could be used to reconstruct the spectrum of a sample by multiplying each one by a different constant scaling factor and adding the results together until the new spectrum closely matches the unknown spectrum. Obviously, each spectrum in the calibration set would have a different set of scaling constants for each variation since the concentrations of the constituents are all different. Therefore, the fraction of each "spectrum" that must be added to reconstruct the unknown data should be related to the concentration of the constituents.

The "variation spectra" are often called eigenvectors (a.k.a., spectral loadings, loading vectors, principal components or factors), for the methods used to calculate them. The scaling constants used to reconstruct the spectra are generally known as scores. This method of breaking down a set spectroscopic data into its most basic variations is called Principal Components Analysis (PCA).

Since the calculated eigenvectors came from the original calibration data, they must somehow relate to the concentrations of the constituents that make up the samples. The same loading vectors can be used to predict "unknown" samples; thus, the only difference between the spectra of samples with different constituent concentrations is the fraction of each loading vector added (scores).

Before PCA is applied to a training set, the data is commonly mean centered. This means that the mean spectrum (average spectrum) is calculated from all of the calibration spectra and then subtracted from every calibration spectrum. Mean centering has the effect of enhancing the subtle differences between the spectra. Remember, eigenvector methods calculate the principal components based on changes in the absorbance data, and not the absolute absorbance. Therefore, anything that improves the ability of the calculation to detect the differences between the calibration spectra, will improve the model. This actually makes a lot of sense when considered in the context of how PCA calculates the eigenvectors. Since the eigenvectors represent the changes in the spectral data that are common to all the calibration spectra, removing the mean simply removes the first most common variation before the data is even processed by the PCA algorithm.

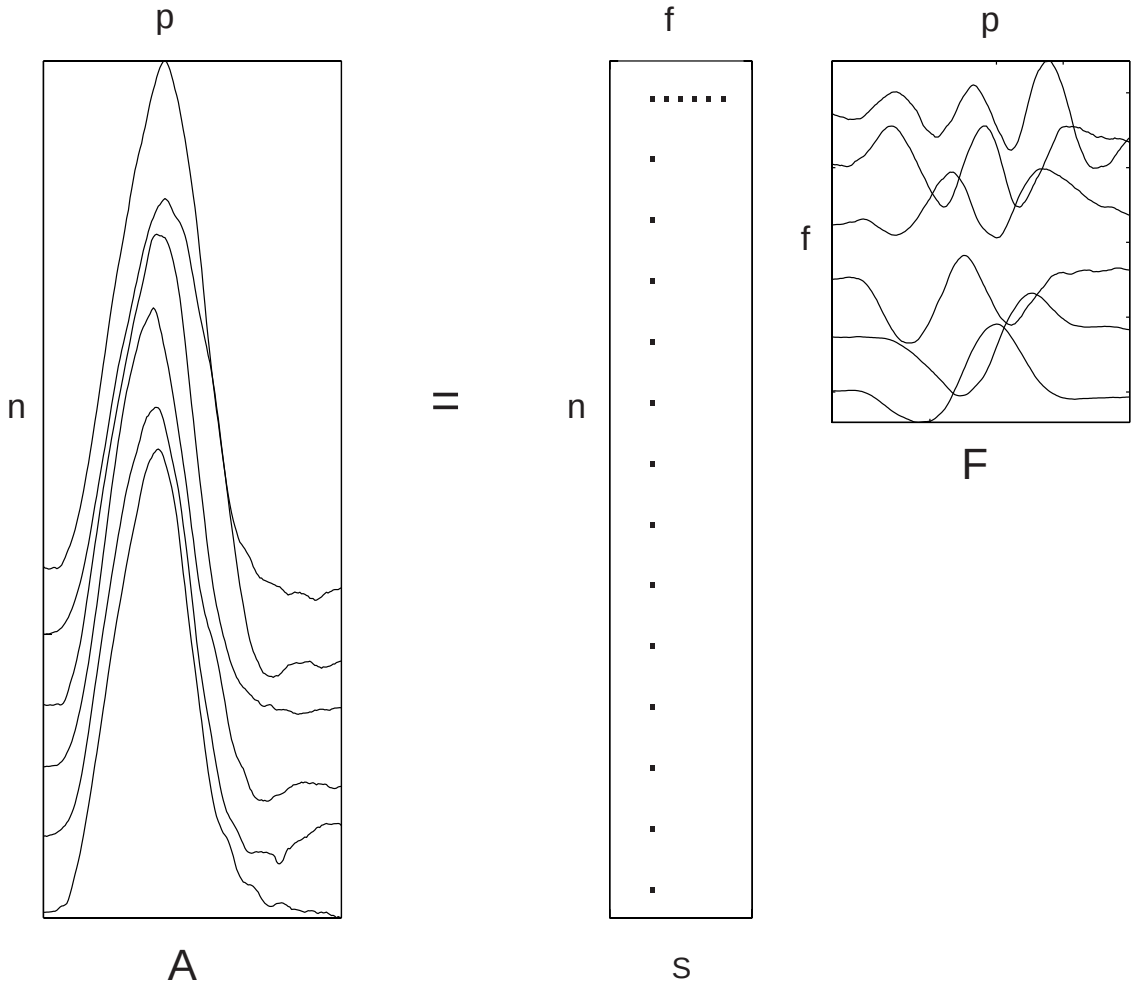


Figure 2.3: PCA breaks apart the spectral data into the most common spectral variations (factors, eigenvectors, loadings) and the corresponding scaling coefficients (scores). The original spectral data matrix A is joint velocity data of point-to-point reaching movements of a typical subject. The dimension of each matrix is indicated in the figure.

PCA is effectively a process of elimination. By iteratively eliminating each independent variation from the calibration spectra in series, it is possible to create a set of eigenvectors (principal components) that represent the changes in the sample data that are common to all. When the training data has been fully processed by the PCA algorithm, it is reduced

to two main matrices: the eigenvectors (spectra) and the scores (the eigenvector weighting values for all the calibration spectra). The matrix expression of the model equation for the spectral data looks something like:

$$A = SF + E_A$$

where A is an n by p matrix of spectral sample data, S is an n by f matrix of score values for all of the spectra, and F is an f by p matrix of eigenvectors. The E_A matrix is the errors in the model's ability to predict the calibration sample data and has the same dimensionality as the A matrix. In the case of eigenvector analysis, the E_A matrix is often called the matrix of residual spectra. The dimensions of the matrices are representative of the data they hold; n is the number of samples (spectra), p is the number of data points (wavelengths) used for calibration, and f is the number PCA eigenvectors. As will be shown later, this is actually a simplification of the true model equation.

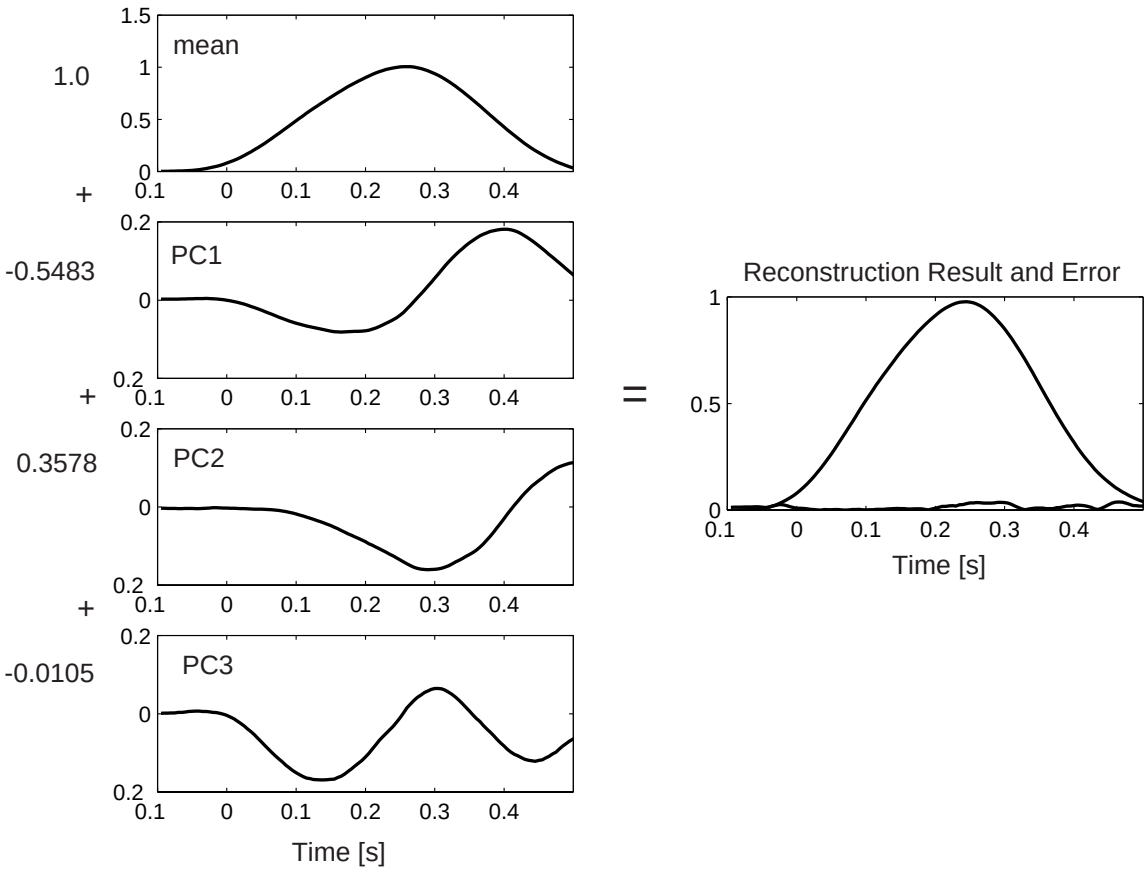


Figure 2.4: By multiplying PC1, PC2 and PC3 (Eigenvectors) by the set of representative scalar fractions (Scores) and summing the results (along with the Mean spectrum if the data was mean centered), the original joint velocity data can be recreated. The "spectral residual" is the difference between this reconstruction and the original. The difference (error) is because we not all principle components are used to recreate the real data.

Chapter 3

Experimental Apparatus and Protocol

3.1 Introduction

People learn to move novel objects along desired trajectory, in any direction, by simply practicing the task a few times. This adaptation is remarkable because of the computational complexity inherent to learning dynamics (Atkeson, 1989). Previous studies have hypothesized that adaptation of a neural internal model (IM), transforming desired trajectory of the hand into appropriate muscle activations, likely underlies this ability (Jordan, 1995; Wolpert et al., 1995). Aftereffects, errors that people make when learned dynamics are unexpectedly changed, suggest that IMs are built gradually with practice (Shadmehr and Mussa-Ivaldi, 1994), that learning one IM can interfere with the learning of a second IM (Brashers-Krug, 1996), and that the interference fades over the courses of hours (Shadmehr and Brashers-Krug, 1997).

When subjects make reaching movements against a curl force instead of the normal null field, the trajectories would be quite different at the very beginning. But after 15 minutes (more or less time would be needed for different subject) training and practice, the trajectory (position, velocity and acceleration of human hand) of the movement remain nearly unchanged compared with unloaded trials (See fig 3.2). The interaction force acting on the hand from the robot handle is still quite different from the ones under the normal null field and remains constant across movements. If we model the two-link

human arm dynamics on the horizontal plane by the following second-order nonlinear differential equations:

$$\Psi(q,\dot{q},\ddot{q}) = M(q,\dot{q},\mu) + J^T(q)F_{ext}(t)$$

So we have,

$$M(q,\dot{q},\mu) = \Psi(q,\dot{q},\ddot{q}) - J^T(q)F_{ext}(t)$$

$F_{ext}(t)$ denotes the external force acting on the hand from the robot handle. We can see that since trajectory $q,\dot{q},\ddot{q}$ remain nearly unchanged, $\Psi(q,\dot{q},\ddot{q})$ remain unchanged. Then the change of $F_{ext}(t)$ reflects the change of $M(q,\dot{q},\mu)$, the adaptation of descending control commands, which use proprioceptive information to produce an error-feedback action (Marsden et al., 1978). In computational studies, the changes in descending commands are attributable to adaptation of an IM (Wada and Kawato, 1993; Miall and Wolpert, 1996; Barto et al., 1998; Bhushan and Shadmehr, 1999). An elegant idea is that adaptation may be driven by error-feedback motor responses generated by reflex circuits (Kawato et al., 1987; Stroeve, 1997). In other words, the delayed, reflex-based error feedback might serve as a “blueprint” for how the CNS needs to change descending commands. Here I wanted to quantify the changes of descending commands $M(q,\dot{q},\mu)$, represented as control force or torque, after the subjects adapted to the novel dynamic field.

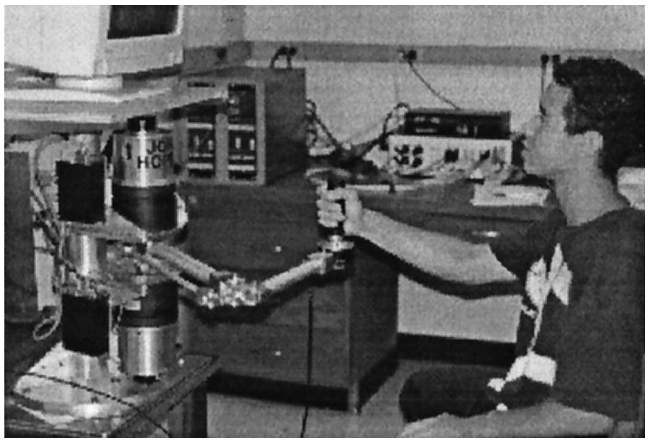


Figure 3.1 The robot manipulandum and the experimental setup. The manipulandum is a very low-friction, planar mechanism powered by two high-performance torque motors. The subject grips the handle of the robot. The handle houses a force transducer. The video monitor facing the subject displays a cursor corresponding to the position of the handle. A target position is displayed, and the subject makes a reaching movement. With practice, the subject learns to compensate for the forces produced by the robot.

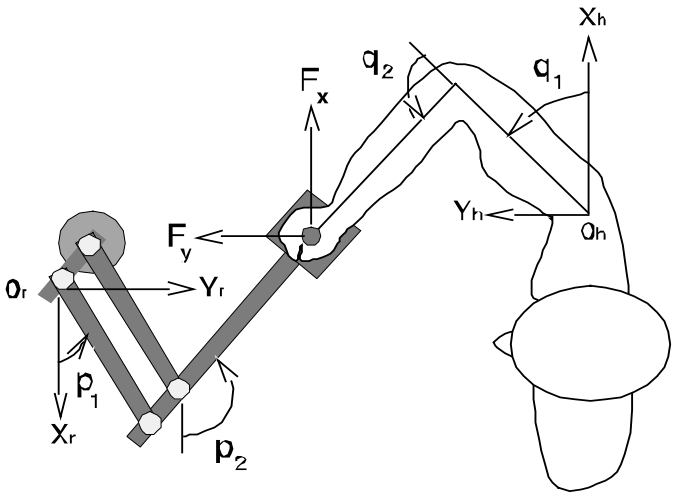


Figure 3.2 Overhead view of a subject seated for experimental. The arm is supported in the horizontal plane. Two Cartesian coordinate frames: the subject frame (right) with the origin O_h and the robot frame (left) with the origin O_r in this experiment are also sketched.

3.2 Material and Methods

The purpose of our experiment was to observe how a subject adapted to the changed dynamics of a reaching task by examining control force $M(q,\dot{q},\mu)$, which is solely

attributed to the feedback control strategy of the human motor controller. A robot manipulandum whose handle was grasped by the subject produced these variable dynamics. A mathematical model was developed to compute the control force. Both the experiments, data acquisition and processing, and some of the modeling procedures are described in this chapter.

3.2.1 Experimental Setup

Two males and one female, a total of 3 right-handed subjects (age range, 22-36 years old), participated in these experiments after giving informed consent. None of the subjects reported sensorimotor or neurological problems, and all had correct-for-normal vision. All of the subjects were naïve with respect to the hypotheses under study. Subjects were seated in front of the manipulandum, with the right elbow supported by a long rope (3m) attached to the ceiling, and the right, dominant hand grasped the end effector (handle) of a 2 degree-of-freedom (df) manipulandum mounted in the horizontal). Subjects learned to make reaching movements while interacting with a force producing manipulandum. A schematic and photo of the measurement apparatus are shown in Figure 3.1 and Figure 3.2.

The manipulandum is a two degree of freedom, lightweight, low friction (0.02 and 0.06 N·m·sec viscous friction for shoulder and elbow joints) robot (Faye 1986) with 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 parallelogram configuration. Position and velocity measurements are made using two optical encoders (Teledyne Gureley) and tachometers (PMI), respectively, mounted on the axes of the mechanical joints. An accelerometer mounted under the base of the handle. A video display monitor mounted directly above the base of the robot (approximately at eye level with the subject).

The subject was instructed to move his or her right hand from the start to the end position, both of which were displayed on the computer monitor. High-resolution sensors mounted on the axes of the mechanical joints were used to accurately measure joint position, velocity of the robot linkage. The position and velocity of the manipulandum in Cartesian coordinate were computed by the kinematical equation for the manipulandum linkage. Manipulandum acceleration and interaction force on it were monitored in Cartesian coordinate by the accelerometer and force transducer mounted on the base of the manipulandum. Two motors, mounted on the base of the manipulandum, could independently produce torque on the proximal and distal joints of the robot arm. A computer monitor mounted above the robot displayed a yellow cursor representing hand position and a green box representing targets. Subjects were instructed to move the cursor to a green plus on the monitor, which represented both the center of the manipulandum working and center of the monitor. The subject was asked not to move until the target box turned to green. We record hand position, hand velocity, and hand acceleration and interaction force at 200 samples/s. Subjects were instructed to make 10 cm movements in the horizontal plane "from point A (the starting target) to point B (the final target)" as accurately as possible in 50 ms. The target is represented by a 8mm square on the computer monitor. Subjects made movements in the direction of north from the center of the workspace and then back to the center. Movement duration (MD) was estimated as the amount of time during which the hand's speed exceed 0.03 m/s. If the subject completed the movement in 500 ± 50 msec, the box 'exploded,' and the computer generated a pleasing sound; if the cursor reached the box too slow (in $MD \geq 550$ msec), the target filled in blue; if the cursor reached the box too quickly (in $MD \leq 450$ msec), the target filled in red. The only instruction provided was to explode as many targets as possible. I provided no instructions regarding straightness of movements or smoothness of trajectories.

Each block of the experiment was composed of 96 forth and back reaching movements (48 forth and 48 back). All subjects exploded most of the targets after two or three blocks training. Movements were performed in two different dynamics environments. One, termed the "null field", is the normal condition in which the torque motors do not create

any forces; so in the null field the subjects encounter only the inertial dynamics of the manipulandum. In the second environments, torque motors produce an additional force as described by the equation:

$$\vec{F}_M = B\dot{x}$$

where $\vec{F}_M$ is the force produced at the end of effector by the robot's motor, $\dot{x}$ is the instantaneous velocity vector of the handle or hand in the robot frame which was obtained online at $200Hz$ sample rate, and B is a viscosity matrix. Generally, in the first environment of the null field, viscosity matrix B equals $[0 \quad 0; 0 \quad 0]\,\mathrm{N\,sec\,m^{-1}}$. In the second environment of force field, B is not zero, equals for example $[0 \quad -13; 13 \quad 0]\,\mathrm{N\,sec\,m^{-1}}$. This force field exerted a force by the robot handle proportional in strength to the instantaneous speed of the hand, in the direction perpendicular to the instantaneous velocity vector. Choosing different parameters for B will design different force fields.

3.2.2 Experiment Procedures

Experiment 1. Movements in the null field. The subject are initially trained to make reaching movements with the robotic manipulandum for 2 or 3 blocks of 96 movements each in the null field since the robot does not produce any active force on the hand. Subjects were required to explode targets as many as possible. This is to train the subjects thoroughly with the experimental paradigm, the visuomotor transformation from the vertical plane visual coordinates to the horizontal plane hand coordinates, and the passive dynamics of the robot manipulandum. A typical movement is shown in Fig. 3.2. It has been shown previously (Flash and Hogan, 1985) that human plan reaching movements in order to follow a minimum jerk trajectory. The results here are consistent with the previous findings. The minimum jerk trajectory seems to present the desired behavior for subjects when making reaching movements. In other words, it is their kinematic plan for moving from one point to another. After the training, subjects will perform constantly

and explode most of the targets. We assume that the internal model was adapted to the dynamic environment. Now a small, smooth and bell-shaped perturbation was given from 100 msec to approximately 200 msec into the movement. The frequency of perturbations was approximately one out of three or four movements. Each block consisted of 23 randomly selected perturbing directions. Three blocks of 96 movements each were used in the experiments for a single perturbing magnitude. After 27 blocks, we had perturbed movements with 9 different perturbing magnitudes and 23 directions. Movements with the same perturbing direction and magnitude were averaged for future computation and estimation. The typical perturbed movement with one kick magnitude is shown in Fig. 3.4.

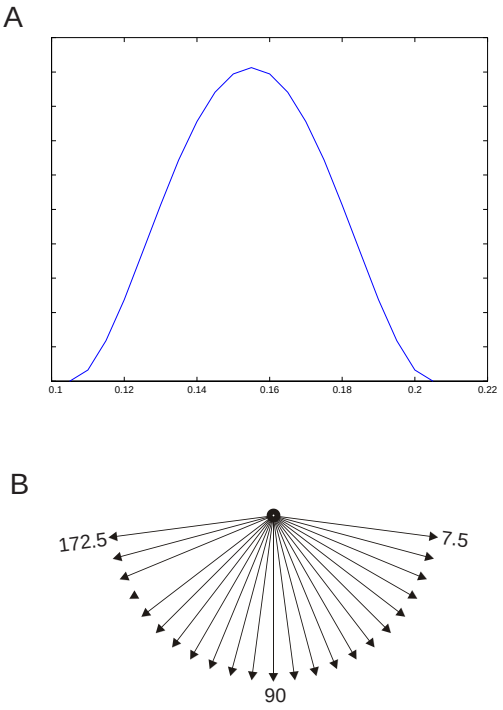


Figure A is the shape of perturbing force with the duration around 100 msec. Human arm is perturbed 100 msec after the beginning of movement. Movements are perturbed in 23 equally spaced directions as shown in Figure B.

Figure 3.3: Shape of the perturbation A and All perturbing directions B. Each perturbation with the same kick magnitude and kick direction is repeated 3 times and data with the same kick was averaged.

Experiment 2. All subjects performed movements in a consistently applied viscous force field. After the initial training and perturbation in the null field, the robot is programmed to generate active forces on the hand of the subject as the subject performed reaching

movements. The forces, produced by the robot and indicated by the vector $\vec{F}_M$, simulate a function of the velocity of the hand $\vec{F}_M = B\dot{x}$, where $\dot{x}$ was the hand velocity vector, and B was a constant matrix respresenting viscosity of the imposed environment in subject Cartesian coordinates, we chose B to be:

$$B = \begin{bmatrix} 0 & -13 \\ 13 & 0 \end{bmatrix} \text{N sec /m}$$

This alters the dynamics of the environment, which is quite unfamiliar with the usual natural environments. Subjects were also required to explode targets as many as possible. Subjects' training was divided into sets of 96 movements each in novel dynamic environments. There is an initial deviation from the straight line desired path as the force fields pushes the hand to the left when the subject move from the center to the target above the center, followed by a quick corrective movement to the target. As subjects trains in the force field the performance shows a gradual improvement until it converge back to the minimum trajectory and remained nearly unchanged compared with unloaded trials (in the null field). Fig 3.3 show one of the typical movement after adaptation and one can see that the hand paths are almost a straight line to the target with smooth bell shaped speed profile. Having completed the adaptation phase of the environment, the subject's arm movement was perturbed. The same perturbation as in the null field was added to the force field and was imposed on the hand from the robot manipulandum. The smaller kick magnitude was chosen here so the perturbed trajectories in the force field were in the same working space and easier to compare with the trajectories in the null field. The perturbation was applied from 100 msec to 200 msec into the movement. One out of three or four movements was perturbed. 23 randomly selected perturbing directions in each block). We did two different methods here in the force field. For the first 3 blocks, after the perturbing force the curl force field was turned off and the subject came back to the target unloaded; For the second 3 blocks, the force fields were still there after the perturbing force and the subject came back to the target loaded. Recorded data was also averaged for the same perturbing direction and magnitude.

3.3 Data Acquisition and Preprocessing

Subjects move the robot handle from a starting point to a target at 10 cm in upward direction with a movement time of 0.5 s. We consider two conditions. First, movements in a null field, i.e., the subject's arm is unloaded. Second, movements in a curl force field $\vec{F}_M = B\dot{x},$ with the field described by $B = \begin{bmatrix} 0 & -13 \\ 13 & 0 \end{bmatrix}$ N sec/m . A two-degree-of-freedom model of the kinematic linkage of the human arm was used to analyze the data (See fig 3.1). This model include elbow and shoulder joint rotations. The motion of the human hand could be descried in two Cartesian coordinates: the subject's frame O_h and the robot frame O_r. The origin of the subject's frame O_h was the center of rotation of the subject's shoulder, whereas the origin of the robot frame O_r was the intersection of the rotation axis of the motors with the horizontal plane. The X and Y axes of the two systems where parallel to each other but were pointing in opposite directions. The subject's X axis was lying in the frontal plane passing through the centers of rotation of both shoulders. The transform from robot coordinates to subject coordinates is then given by

$$x_h = x_r - X_0$$
$$y_h = y_r - Y_0$$

where X_0 and Y_0 are the coordinates of the robot origin in the subject's frame; x_h and y_h are the coordinates of a point (the manipulandum) in the subject's frame; and x_r and y_r are the coordinates of the same point in the robot frame. The inverse transform from subject coordinates to human coordinates is straightforward. The transformation of velocity, acceleration and force from one frame to the other are just inversing direction of these vectors and keep the magnitudes same.

The rotation of the human arm was described in the subject joint coordinates with the origin O_h. $q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$, where q_1 is shoulder angle and q_2 is elbow angle. Here the relative joint coordinates were used considering the configuration feature of the human arm. The kinematics of the human arm refer to the configuration relationships between joint positions and hand positions and the transformation between these two systems were introduced in Chapter 2 in detail. The transformation from interaction force acting on the human hand in the subject Cartesian coordinates to the rotational torque acting on the joints in the subject joint coordinates is given by

$$T = \left[J^T(q)\right]F$$

The inverse transformation from torques to forces is given by

$$F = \left[J^T(q)\right]^{-1}T$$

$$F_x = \left[\frac{\cos(q_1+q_2)}{l_1}\tau_1 - \frac{\cos(q_1)}{l_2}\tau_2\right]\left[\sin(q_2)\right]^{-1}$$

$$F_y = \left[\frac{\sin(q_1+q_2)}{l_1}\tau_1 - \frac{\sin(q_1)}{l_2}\tau_2\right]\left[\sin(q_2)\right]^{-1}$$

where F_x and F_y are the components of the corresponding force at the handle taken in the subject Cartesian coordinates. q_1 is shoulder angle and q_2 is elbow angle in the subject joint coordinates. J is the Jacobian of joint to Cartesian coordinate transform.

We sampled hand position, velocity, acceleration and interaction force on the hand at 5 msec intervals as the subject reached to a target at a distance of 10 cm. The hand kinematic information was recorded online in the robot Cartesian coordinates by $C++$ software at the sample rate $200\,Hz$. Data was collected in blocks of 96 trials each and

aligned using velocity threshold at the onset of movement. In order to reduce the effect of small hand tremors or unintentional movements, the perturbed paths were averaged for the same condition (kick magnitude, direction and dynamic force fields). We represent each trajectory as a time series of vectors for position, velocity, acceleration and interaction force on the hand. Raw data was converted into subject Cartesian coordinates and subject joint coordinates by *Matlab* offline for future processing. Typical hand trajectories from the center of the monitor to the target 10 cm above it both in the null field and force field were plotted in the following figures. (Figure 3.3 and Figure 3.4).

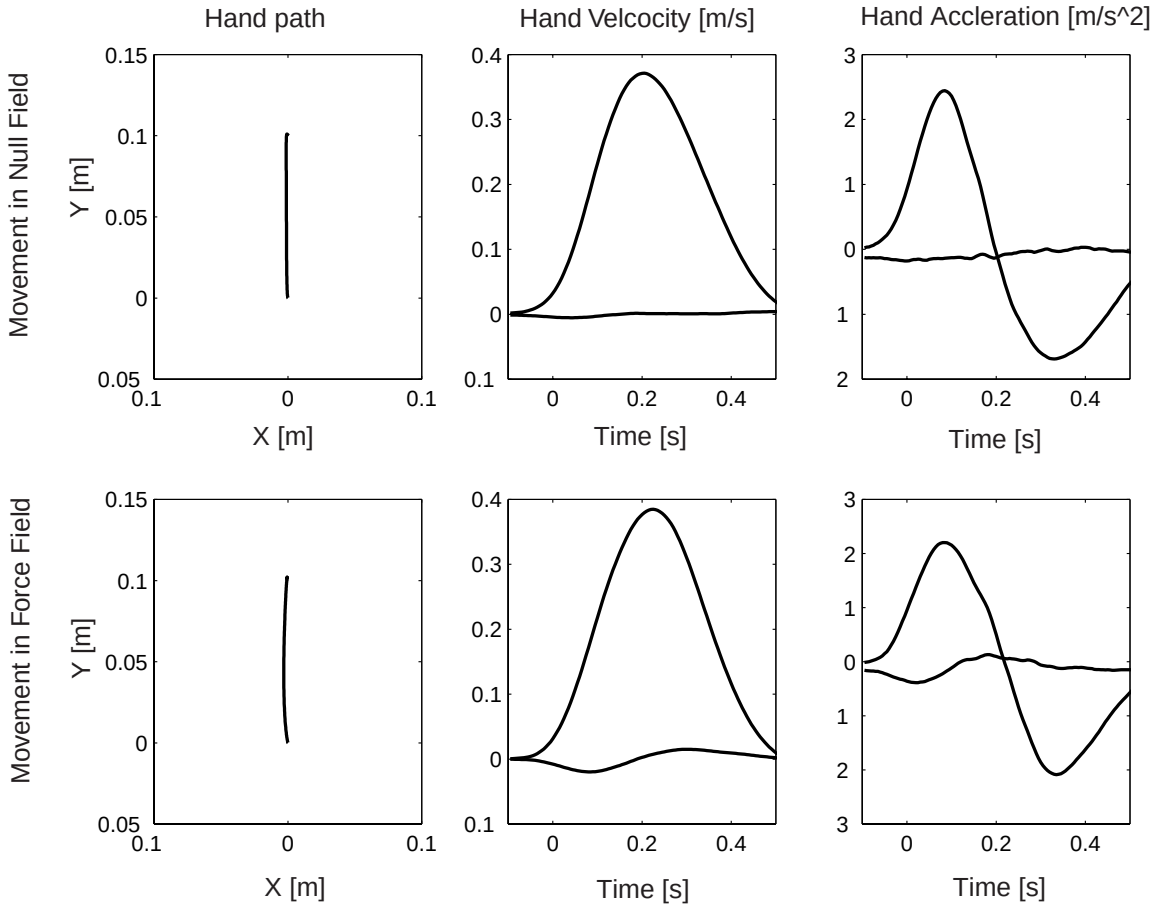


Figure 3.4 Typical hand path, velocity and acceleration in the subject Cartesian coordinates of movements in the late training period. Top row is movements in the null field. Bottom row is movements in the force field with the viscosity matrix B equals to $\begin{bmatrix} 0 & -13 \\ 13 & 0 \end{bmatrix}$ N sec m^{-1}. *Left*, Hand path of a typical subject in his movements toward a target. *Middle and Right* are hand velocity and hand acceleration respectively. Units for x-axis is sec and for y-axis is m hand path, m/s for hand velocity and m/s^2 for hand acceleration.

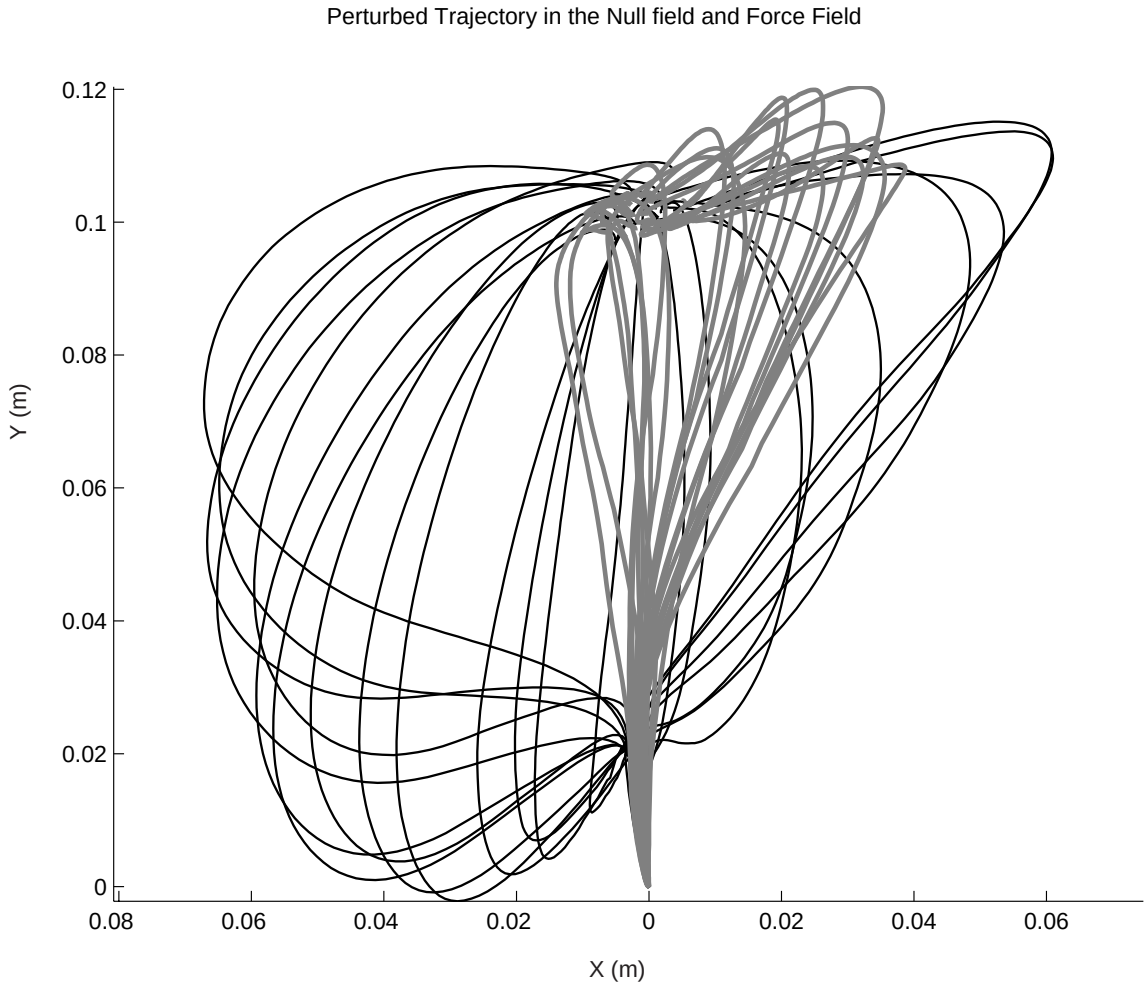


Figure 3.5: Perturbed trajectories in the null field (black) and perturbed trajectories in the force field (gray). Trajectories are averaged over 3 individual trials in 23 different perturbing directions. Force field trajectories are the data with the force field on after the perturbing force.

The recorded position and velocity of the human arm movements were noise free and there is no time delay for the sample rate $200\,Hz$. But the measurements of accelerometer and force traducer were noisier compared to position and velocity. $4\,m\sec$ time delay for the measurement of interaction force was also observed. We used the Savitsky-Golay smoothing technique to delete the noise in the raw data. The order of the filter (the highest polynomial power used in the fit) we used here is 3 and half-width of the filter is $[4 \quad 4]$ (Half-width is given in the form [NL, NR], where NL and NR are the number of points to the right and left of the point will be filtered. Savitsky-Golay filter can also be used to compute the derivatives. We can compute acceleration by filtering velocity and compared it with the measured acceleration. This filter added no time delay to the filtered data. Spikes were also observed in the raw data. Spikes were removed simply by replacing them with the average of the two points to the right and left.

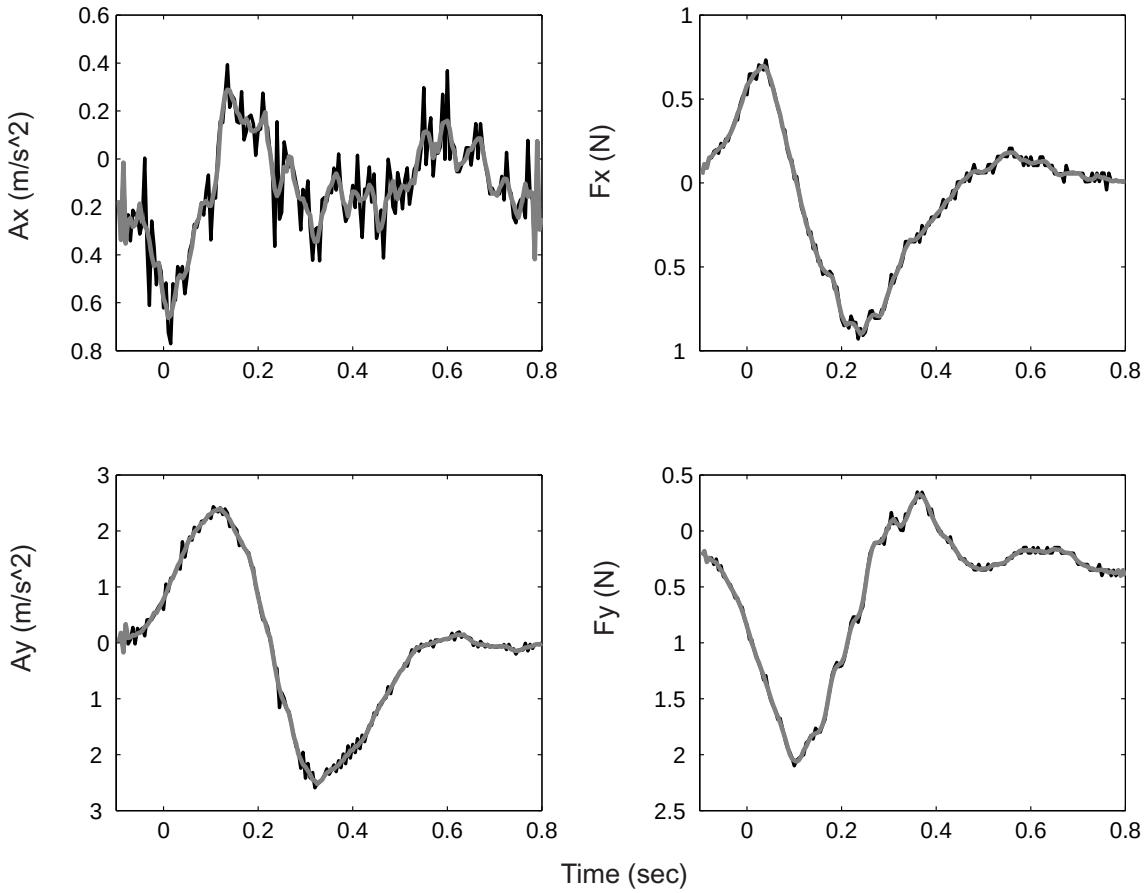


Figure 3.6: Comparing Savitsky-Golay (order is 3, half-width is [4 4]) filtered data (gray) and raw data (black) of hand acceleration and interaction force acting on the human hand in subject's Cartesian coordinate frame.

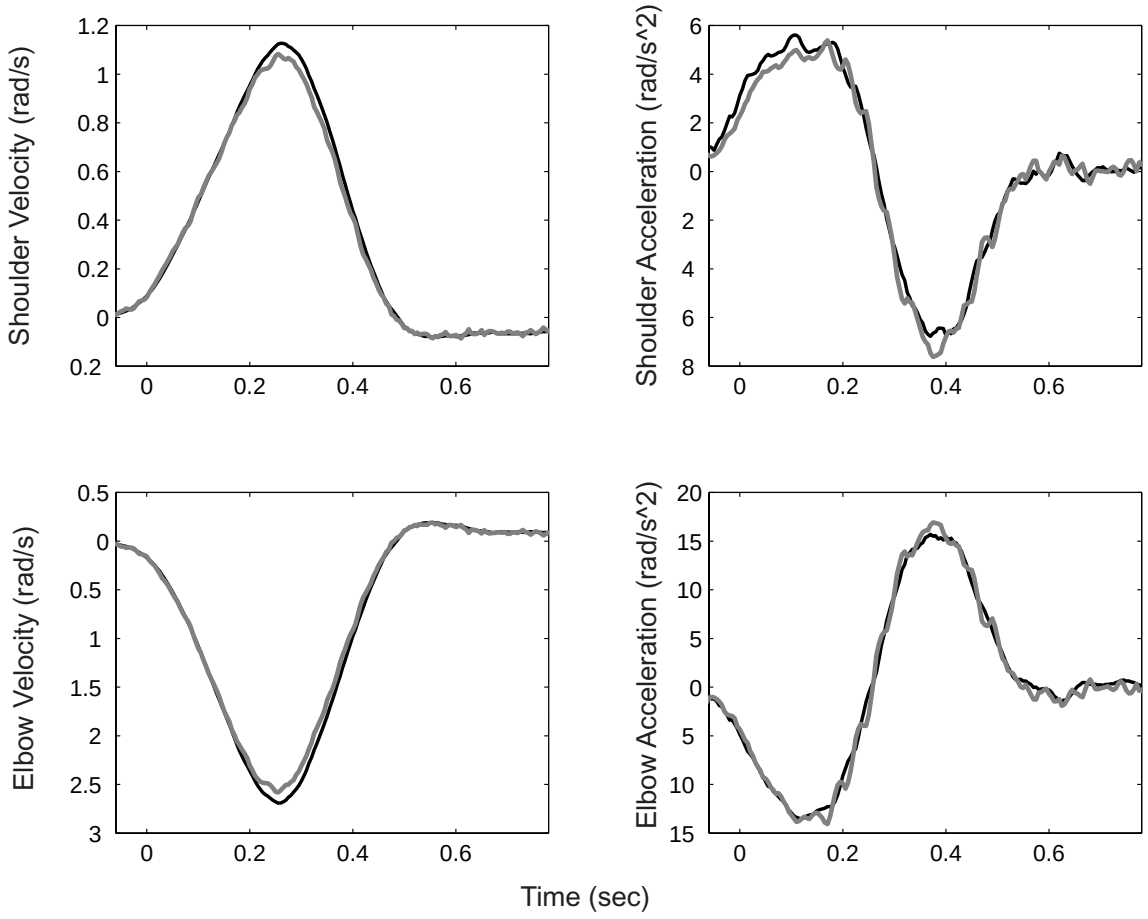


Figure 3.7: Comparing computed data (gray) and measured data (black) of human joint velocity and acceleration. The velocity and acceleration in gray were computed using Savitsky-Golay filter (order is 3, half-width is [4 4]) from raw position and velocity respectively. The velocity and acceleration in black were raw data from measurements of sensor and accelerometer.

3.4 Data analysis: Formation of the Internal Model

The experimental results from fig. 3.2. establish the learning of the force field as the subjects trained over time. It has been proposed that this process of learning occurs by adaptation of internal models that predict the dynamics of the force field (Shadmehr and Mussa-Ivaldi, 1994). After-effect movements provide evidence in support of this theory.

In the absence of external force field, the subjects' hand trajectories displayed approximately straight paths and smooth, bell shaped velocity profiles (Fig. 3.6). However, when the robot manipulandum generated velocity-dependent forces that interfered with the execution of the reaching movements, the hand trajectories were distorted. With training, movement error (integral of perpendicular velocity, that is, displacement from a straight line) gradually decreases, and the subjects' distorted hand trajectories converged to the trajectories observed before the application of the force field. This convergence was gradually but monotonic and consistent with an adaptive process whose goal was to compensate for the forces imposed by the external field and return the hand's trajectory to the path produced in the null field.

Subjects' training was divided into sets of 96 movements. Each two sets were followed by a 3-minute rest period. We gathered quantitative evidence motor learning by recording the path and the velocity of each subject's movements and by computing a correlation coefficient between the velocity of an ideal, straight trajectory and the velocity of the actual trajectories. The correlation coefficients allowed us to evaluate whether the internal model has adapted to the new dynamic environment.

In the experiments described here, a key feature of the task to which the human subjects were exposed involved a change in the mechanical environment with which their hand interacted. Because of this change, the neural representation of the arm would have to develop a new model to deal with the new dynamics of the environment. In this thesis, we present psychophysical evidence for the formation of this new internal model and we described the control force and torque changes observed by applying perturbations as the new internal model was formed (see figure 3.4).

3.5 Estimating Inertial Model of the Human Arm

As described above, the model for the two-link human arm dynamics on the horizontal plane is,

$$\Psi(q,\dot{q},\ddot{q}) = M(q,\dot{q},\mu) + J^T(q)F_{ext}(t)$$

Our goal is trying to quantify the changes of descending commands $M(q,\dot{q},\mu)$, represented as control force or torque, after the subjects adapted to the novel dynamic field. In order to get $M(q,\dot{q},\mu)$, we measure the interaction acting on the hand $F_{ext}(t)$, the upper and lower arm lengths of the subject to compute $J^T(q)$, and kinematic information $q,\dot{q},\ddot{q}$ for all the movements for $\Psi(q,\dot{q},\ddot{q})$. One more thing, we need to estimate the inertial model of the human arm $\Psi(q,\dot{q},\ddot{q})$.

We described the theory derivation of human arm inertial model in Chapter 2. Here we explained how we design the experiment to estimate this intertial model. We know that because the three inertial parameters, a_1, a_2 and a_3 for $\Psi(q,\dot{q},\ddot{q})$, are independent of posture and movement, their values can be fixed in any posture for each subject. This reduces estimation errors caused by partially correlated data under some conditions. To measure the inertial parameters, we use the same experimental apparatus and protocol as explained in 2.2. A small box was displayed on the center of the computer monitor representing the target. The cursor on the same monitor represented hand position of the human arm. First we asked the subject to move his/her hand to the target (the small box on the center of monitor) by moving the cursor on top of the target. Right-handed subjects sat straight in front of the monitor grasping the handle, with the right arm supported by a sling. Then we shake subject's arm by grasping and shaking the robot arm. We asked subjects relax his arm when we shake his arm but keep their wrists stiff. The cursor and target on the monitor help us not to move the hand too far away from the target, the original position of right hand. Since the arm was shaken in a small area, we

assumed that muscle stiffness and viscosity (2×2) matrix K and B remained constant during the shaking experiment.

As explained on 2.2 and 2.3, the arm dynamics on the horizontal plane could be modeled by the following second-order nonlinear differential equation,

$$\Psi(q,\dot{q},\ddot{q}) = M(q,\dot{q},\mu) + J^T F_x$$

Here, $q,\dot{q}$ and $\ddot{q}$ are the relative joint position, velocity and acceleration in the subject joint coordinates. [$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$, where q_1 is shoulder angle and q_2 is elbow angle. $\Psi(.)$ denotes a two-link arm dynamics. M is the joint torque generated by the muscles and it can be represented as a function of angular position, velocity and motor command μ, descending from the supraspinal central nervous system (CNS). J is the Jacobian for hand position and joint angle transform, and F_x is the external dynamic interaction force on the hand. From Chapter 2, we knew that if we assume the arm to be rigid body two-link system, we have

$$\Psi(q,\dot{q},\ddot{q}) \cong H(q)\ddot{q} + C(q,\dot{q})\dot{q}$$

H and C are the inertia and coriolis matrices of the arm. $H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}$

$$H_{11} = a_1 + a_2 + 2a_3 \cos q_2$$

$$H_{12} = H_{21} = a_2 + a_3 \cos q_2$$

$$H_{22} = a_2$$

$$C = \begin{bmatrix} -a_3\dot{q}_2 \sin q_2 & -a_3(\dot{q}_1 + \dot{q}_2)\sin q_2 \\ -a_3\dot{q}_1 \sin q_2 & 0 \end{bmatrix}$$

$$\Psi(q,\dot{q},\ddot{q}) = \begin{bmatrix} a_1 + a_2 + 2a_3\cos q_2 & a_2 + a_3\cos q_2 \\ a_2 + a_3\cos q_2 & a_2 \end{bmatrix}\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} -a_3\dot{q}_2\sin q_2 & -a_3(\dot{q}_1+\dot{q}_2)\sin q_2 \\ -a_3\dot{q}_1\sin q_2 & 0 \end{bmatrix}\begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

where

$$a_1 = i_{1c} + m_1 l_{c1}^2 + m_2 l_1^2$$
$$a_2 = i_{2c} + m_2 l_{c2}^2$$
$$a_3 = m_2 l_1 l_{c2}.$$

We represent muscle stiffness and viscosity matrix (2×2) K and B such as:

$$\text{Stiffness: } -\frac{dM}{dq}_{q_0} \equiv K = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$$

$$\text{Viscosity: } -\frac{dM}{dq}_{q_0} \equiv B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

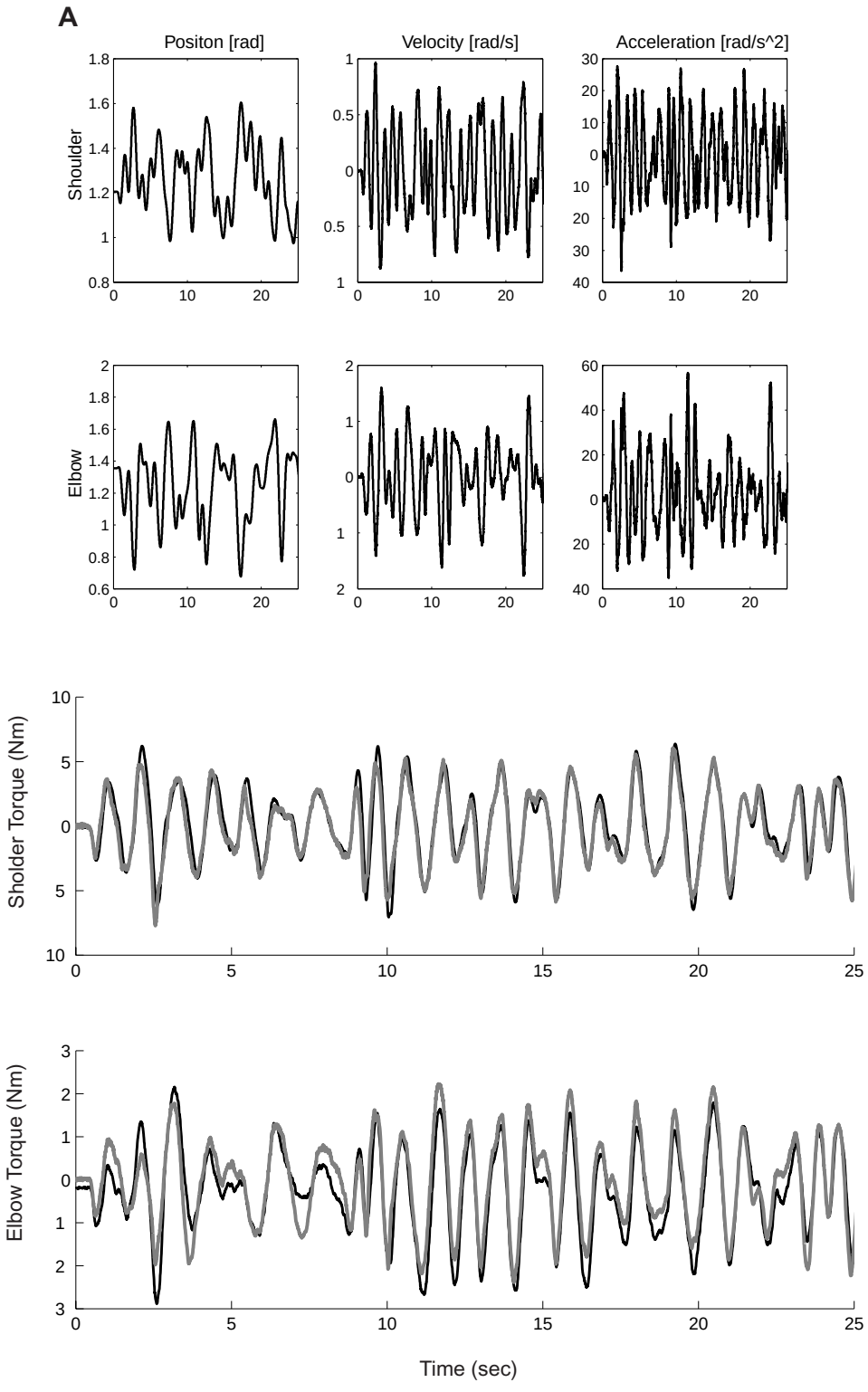
So, we have

$$M(q,\dot{q},\mu) \cong -Kdq_{q_0} - Bd\dot{q}_{q_0}$$

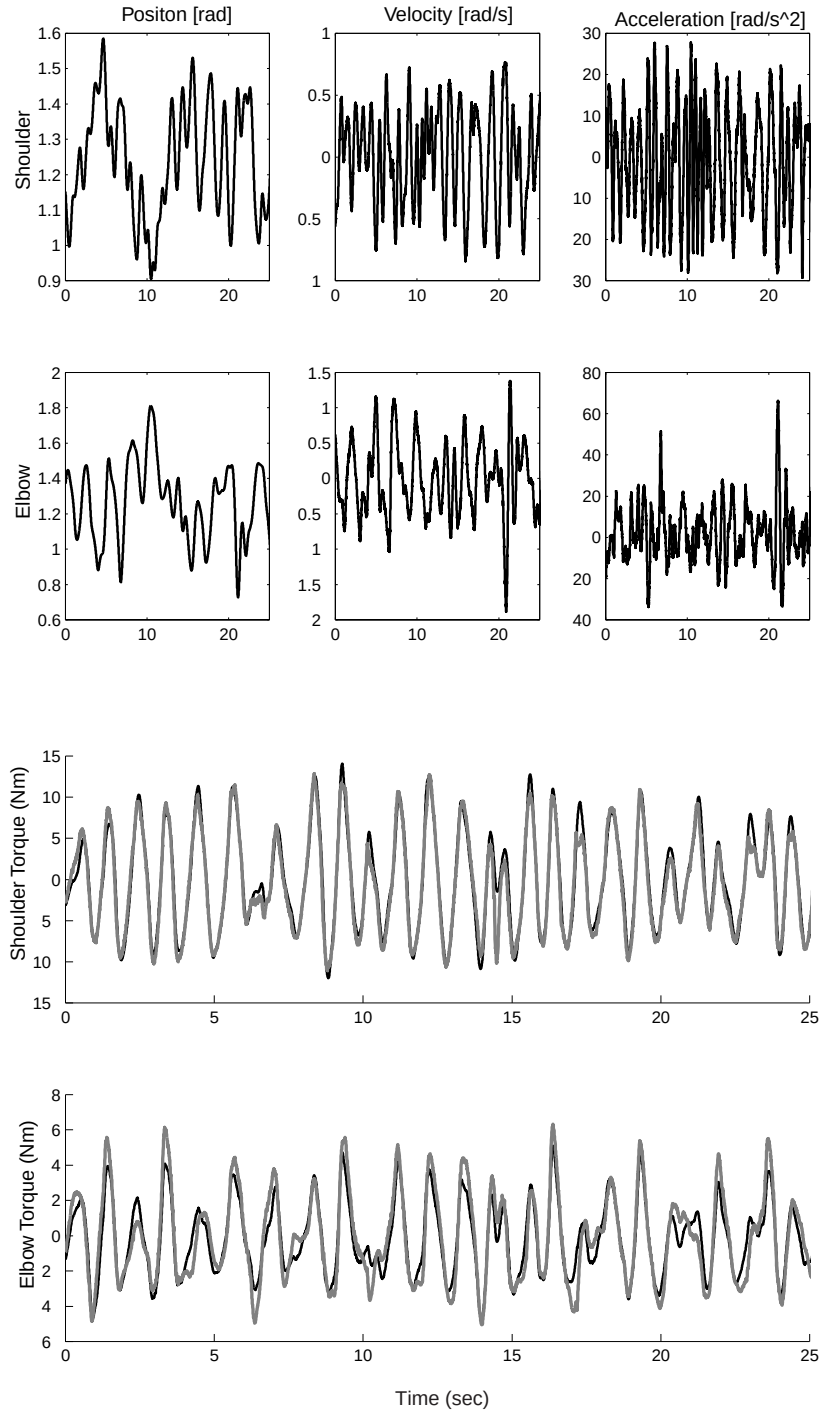
where $dq_{q_0} = q - q_0$ and $d\dot{q}_{q_0} = d\dot{q}$

$$\Psi(q,\dot{q},\ddot{q}) = -K(q-q_0) - Bd\dot{q}_{q_0} + J^T F_x$$

We suppose that K and B are constant for postural maintenance control. $q,\dot{q},\ddot{q}$ and τ_{ext} are measured where $q_0 = q(0), \dot{q}_0 = 0, \ddot{q}_0 = 0$. To fix the inertial parameters, we have quantified all these parameters $a_1, a_2, a_3, k_{11}, k_{12}, k_{21}, k_{22}, b_{11}, b_{12}, b_{21}, b_{22}$ using data from different trials. We chose the inertia parameters that are nearly constant among different trials.


A
Positon [rad]
Velocity [rad/s]
Acceleration [rad/s^2]
Shoulder
Elbow
Sholder Torque (Nm)
Elbow Torque (Nm)
Time (sec)

B


Positon [rad]
Velocity [rad/s]
Acceleration [rad/s^2]
Shoulder
Elbow
Shoulder Torque (Nm)
Elbow Torque (Nm)
Time (sec)

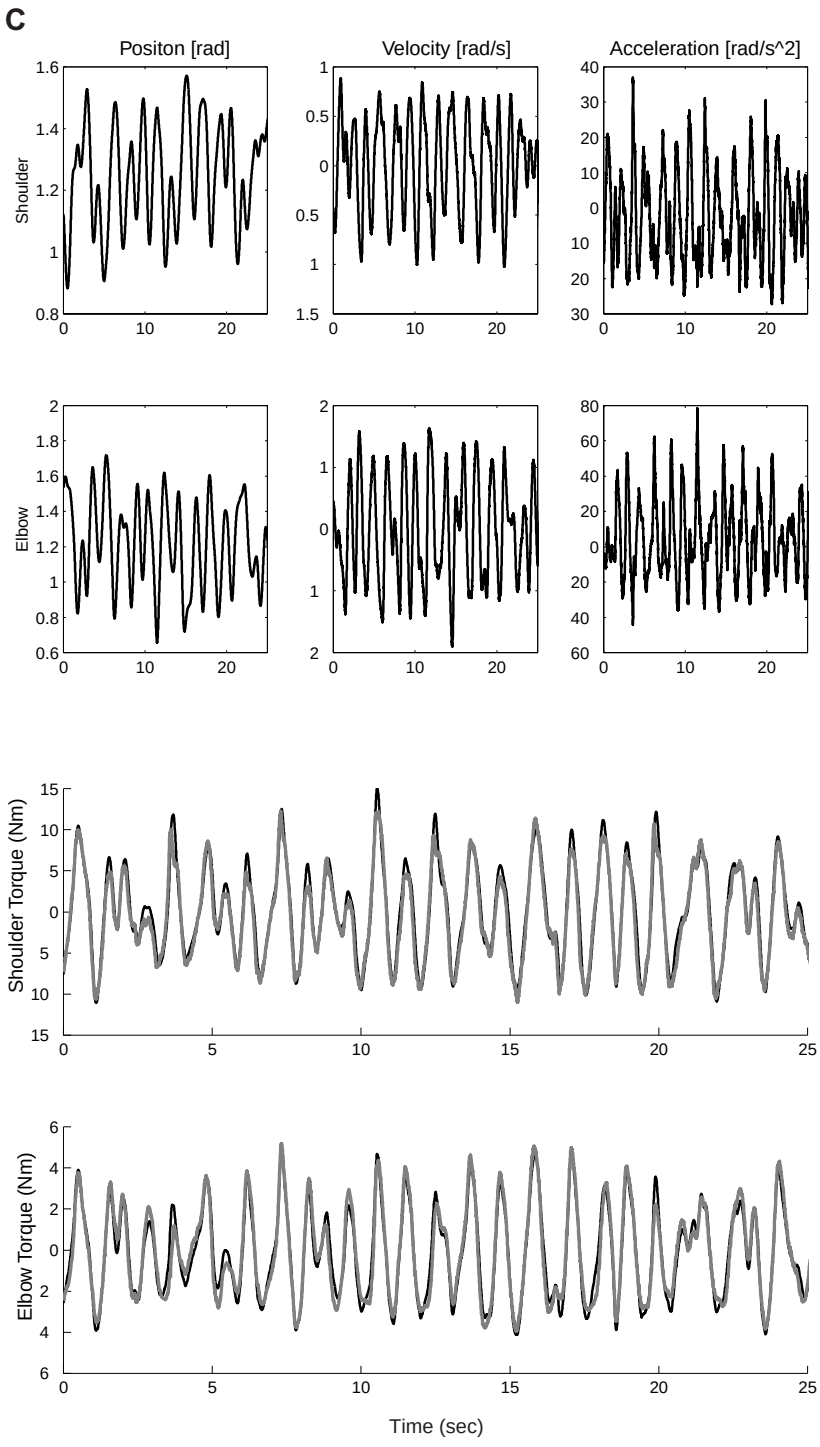


Figure 3.8: The comparison between inertia model output and the measured data for subject **A, B** and **C**.

Table 3.1. The inertial parameters estimated for each subject

Subject	a_1 N m/(rad/s^2)	a_2 N m/(rad/s^2)	a_3 N m/(rad/s^2)
A	0.2347 ± 0.0034	0.0990 ± 0.0014	0.0730 ± 0.0014
B	0.2936 ± 0.0041	0.0788 ± 0.0009	0.0882 ± 0.0012
C	0.4296 ± 0.0081	0.1433 ± 0.0017	0.1323 ± 0.0010

Table 3.2. The average of absolute fitting error and average of absolute measured torque for each subject in Figure 3.8.

Subject	Average Abs Err [Nm]/ Average Abs Torque [Nm] (Shoulder)	Average Abs Err [Nm]/ Average Abs Torque [Nm] (Elbow)
A	0.7000/2.7416	0.2803/1.0249
B	0.6737/4.3433	0.3128/1.4544
C	0.8992/5.0522	0.4028/1.8062

Model validation: The measured values of stiffness and viscosity parameters $k_{11}, k_{12}, k_{21}, k_{22}, b_{11}, b_{12}, b_{21}, b_{22}$ for one subject is not necessary same across different trials. It depends on how subject relaxes his/her arm during each trial. Impedance of human arm does have different value when the arm is posed in different state (position and velocity, which is the property of human arm muscle). So when we use the robot to shake subject's arm in, we keeping the arm in a small working space. In this condition, we assume that the value of stiffness and viscosity does not change during shaking. This is the main reason that the modeled dynamic torque does not perfectly fit the measured torque. The inertia parameters should be constant across different trials. To modify the inertia model, for example, we measure the inertia parameters for subject **A** in 3 trials, the inertia value we have for each trial is

Table 3.3: The inertial parameters of subject **A** in three different trials

Subject **A**	a_1 N m/(rad/s^2)	a_2 N m/(rad/s^2)	a_3 N m/(rad/s^2)
Trial 1	0.2310	0.1003	0.0698
Trial 2	0.2354	0.0992	0.0703
Trial 3	0.2376	0.0976	0.0789

One can see that the inertia values we measured across different trials are very stable, which testify that the inertial model we build for human arm can perfectly predict the inertia torque during the arm movements.

3.6 Predicting the Un-perturbed Trajectory

When subjects learned the new dynamic environment and internal model has adapted to the force field, small force perturbations of brief periods (about $0.11\,s$), with randomly selected perturbing directions, were given in one out of two or three movements. The

subject was also instructed to move his or her hand in as relaxed a way as possible and not to intervene voluntarily (not to correct his or her movements even if the target was missed because of the perturbation). Trajectories are selected based on the recorded data before the perturbation is given. Only the beginning trajectories (part of trajectories before the perturbations) close to the normal sample trajectories were used for data analysis. From Chapter 3, we described that when there was no perturbation, the hand path followed a desired trajectory q_{d} during reaching movements. When movements were perturbed, it is deviated from the desired path q_{d} by Δq. The perturbed resulting trajectory was $q = q_{\mathrm{d}} + \Delta q$. We need to predict the path that the human arm system would follow if there were no perturbations based on the data we measured q and the sample data without perturbations recorded before giving perturbations for the same subject.

For easy understanding the movement prediction method, a row vector was constructed for each movement and was organized as a row for the data matrix. So we have a data matrix with the matrix with the size equal to number of movements by duration of each movement for position, velocity, acceleration and interaction force. We call this matrix S. This matrix S can also be expressed by catenation of two small matrices S^1, S^2.

$$S = [S^1 \quad S^2]$$

S^1 is all the unperturbed movements with the time interval from the beginning to perturbations appear. S^2 is all the unperturbed movements with the interval from perturbations appear to the end of movements. The goal of our prediction is given a perturbed trajectory q and the time point when this movement is perturbed $(\mathrm{prtb\_time})$. Vector q can also be expressed by catenation of two small vectors q^1, q^2 by the same definition when using S^1, S^2 to describe S. Our goal is to estimate $q_{\mathrm{d}} = [q_d^1 \quad q_d^2]$ for this particular q with the known information q^1 and sample data matrix S. Two methods of prediction were used and compared here.

Note: $q^1 = q_d^1$.

3.4.1 Methods 1

Use all unperturbed sample movements as bases for prediction. The idea is that use all the raw data of movements as the base functions to predict the unknown movements.

$$q_d^1 = k_1 \times S^1$$

Assume if we know q_d^2, it can be expressed by $q_d^2 = k_1 \times S^2$. Cancel k_1, we have

$$q_d^2 = q_d^1 [S^1]^{-1} S^2$$

3.4.2 Method 2

The idea here is that instead of using the raw data of movements as the base functions, we use the principal components of the raw data as the base functions to predict the unknown movements. PCA is used to analyze the variability in trajectory shape. To compute the principle components, the mean vector is subtracted from each of a set of M movement vectors $\vec{m}_j$, and the covariance matrix is formed:

$$R = \sum_{j=1}^{M} \vec{m}_j \vec{m}_j^{\,T}$$

The eigenvector $\vec{c}_i$ (principle components) are then calculated using Matlab (version 5.3, Mathworks). Any movement vector among M movement vectors $\vec{m}_j$ can be exactly reconstructed as a superposition of the full set of principle components $\vec{c}_i$ according to:

$$\vec{m} = \sum_{i=1}^{N} a_i \vec{c}_i$$

where N is the number of principle components. N equals the row dimension of the data matrix which is used to calculate the principle components. If the movements are approximated with a smaller number of components $n < N$, then we can write:

$$\vec{m} \approx \sum_{i=1}^{n} a_i \vec{c}_i$$

To decide what is the number of principle components n should be used to predict the movements. All sample raw data was divided into two data sets and two data matrix was constructed. One was used as sample data based on which unknown movements were predicted and the other was used as the test data. The test data was predicted using the following method with different number of components and prediction errors were computed. Using different number of components would have different errors though the difference was not so big. The number of components with the smallest error was saved for the real prediction.

Legend:

S_{pca}, Q$_{\text{pca}}$: Matrix for principal components of raw data matrices S and Q repectively

S_{mean}, Q_{mean}: Mean for raw data matrices S and Q respectively

$$S^1 = K_1 \times S^1_{pca} \qquad Q^1 = \hat{K}_1 \times Q^1_{pca}$$

$$S^2 = K_2 \times S^2_{pca} \qquad Q^2 = \hat{K}_2 \times Q^2_{pca}$$

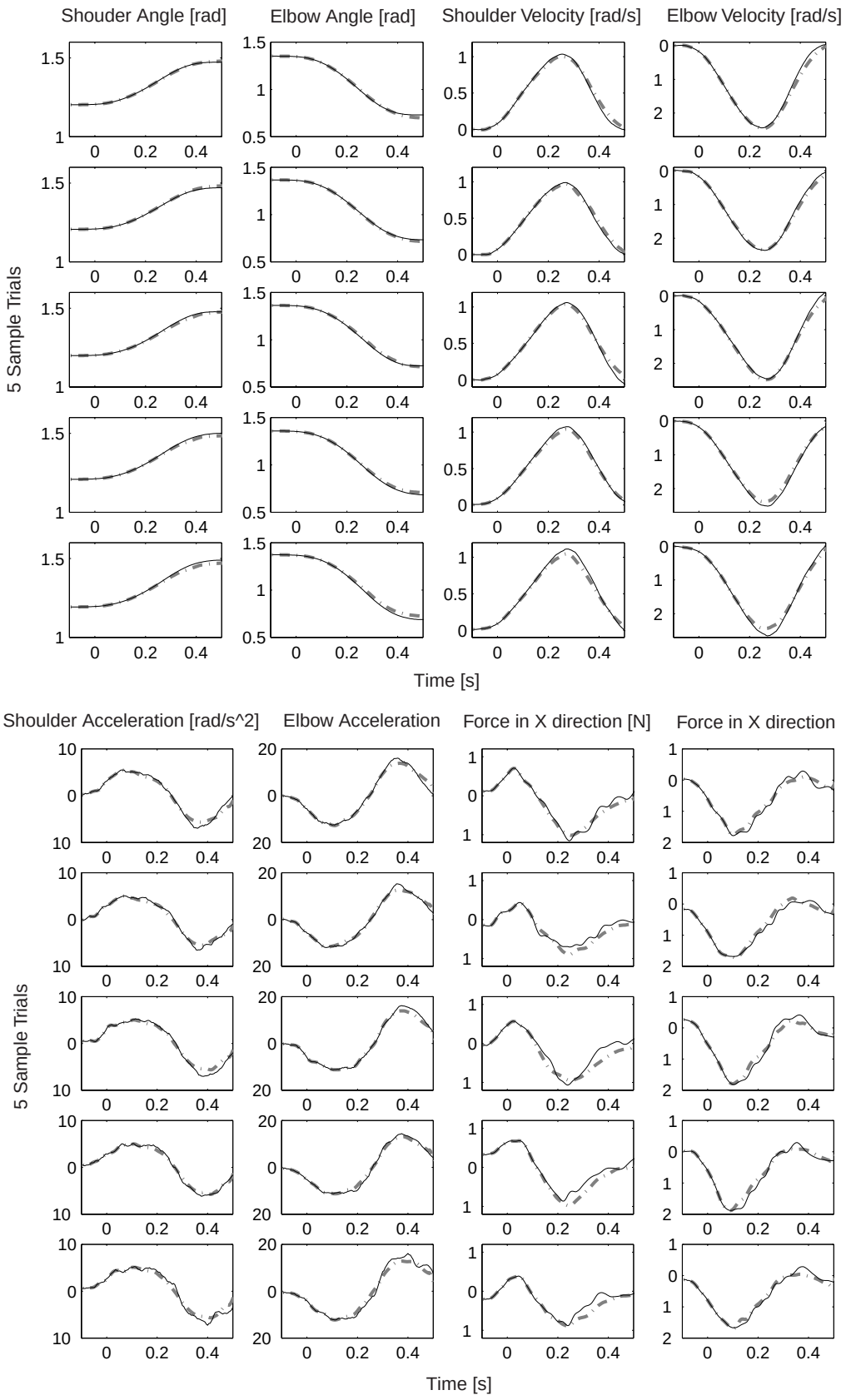
Assume that there exists the following relation for $K_1, K_2, \hat{K}_1$ *and* $\hat{K}_2$,

$$K_2 = K_1 \times C \qquad \hat{K}_2 = \hat{K}_1 \times C$$

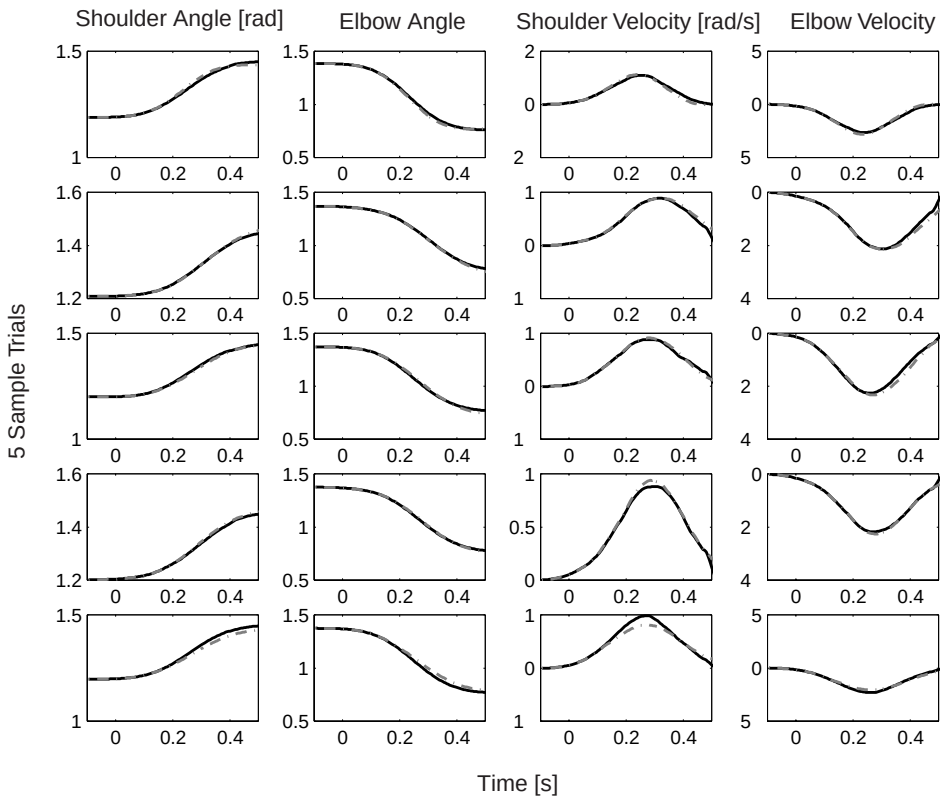
From above equations we have,

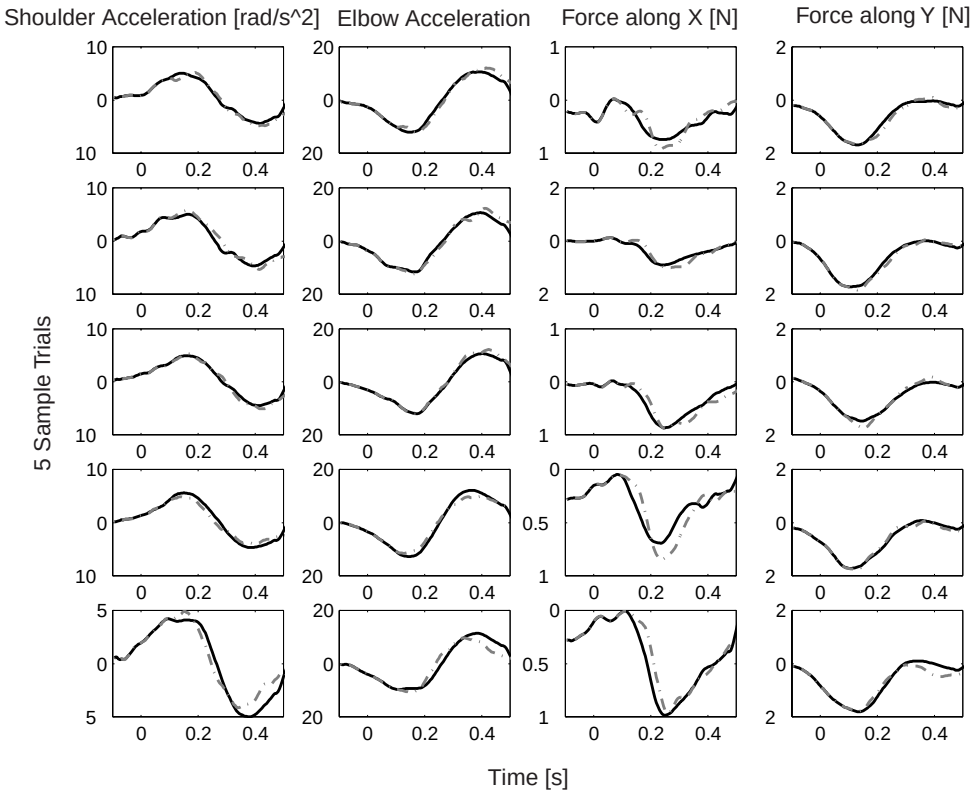
$$Q_d^2 = (Q_d^1 - Q_{mean}^1)([S_{pca}^1]^{-1} S_{pca}^1)(S^1 - S_{mean}^1)^{-1}(S^2 - S_{mean}^2)([S_{pca}^2]^{-1} S_{pca}^2) + S_{mean}^2$$

A

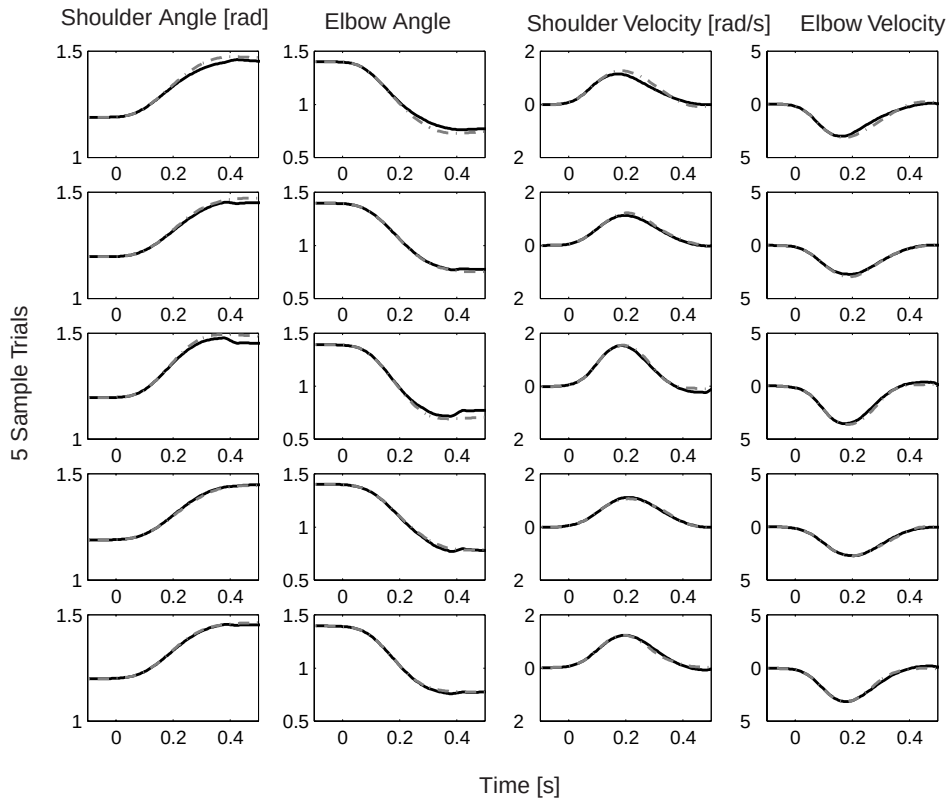

Shouder Angle [rad]
Elbow Angle [rad]
Shoulder Velocity [rad/s]
Elbow Velocity [rad/s]
5 Sample Trials
Time [s]
Shoulder Acceleration [rad/s^2]
Elbow Acceleration
Force in X direction [N]
Force in X direction
5 Sample Trials
Time [s]

B


Shoulder Angle [rad]
Elbow Angle
Shoulder Velocity [rad/s]
Elbow Velocity
5 Sample Trials
Time [s]


Shoulder Acceleration [rad/s^2]
Elbow Acceleration
Force along X [N]
Force along Y [N]
5 Sample Trials
Time [s]

C


Shoulder Angle [rad]
Elbow Angle
Shoulder Velocity [rad/s]
Elbow Velocity
5 Sample Trials
Time [s]

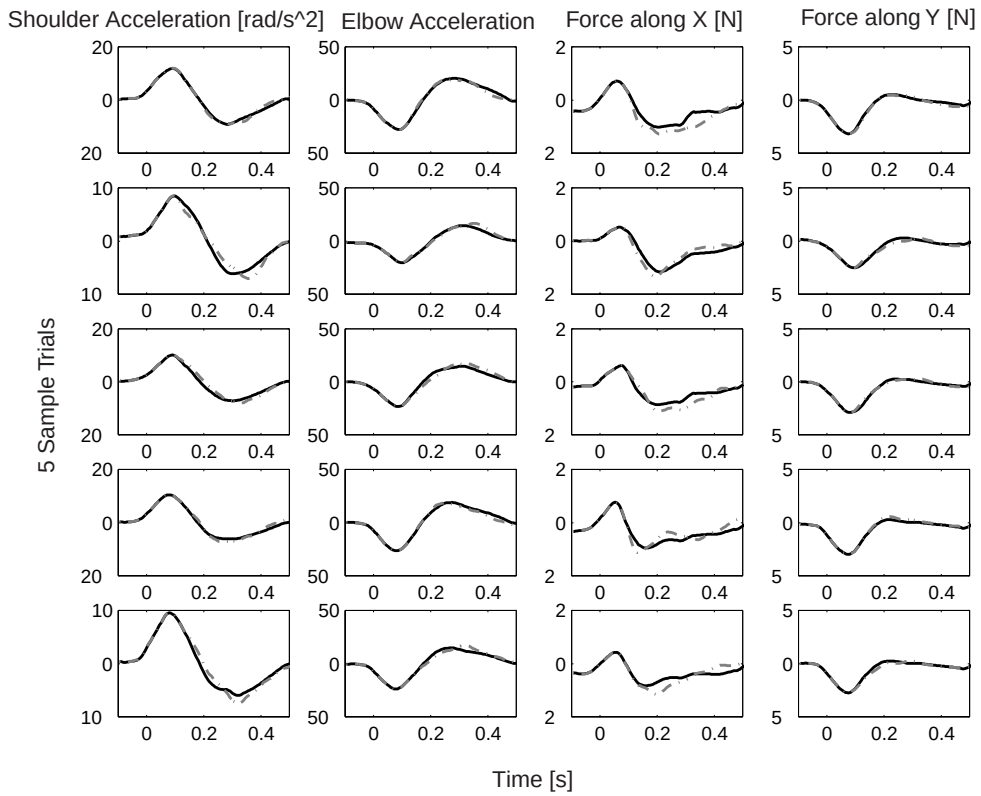
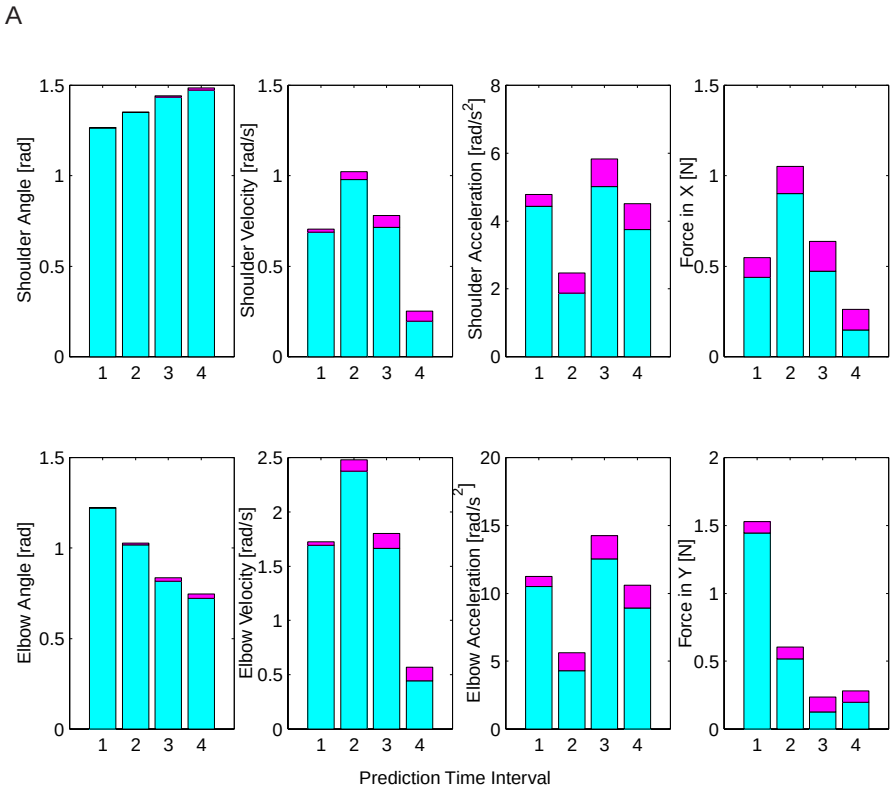
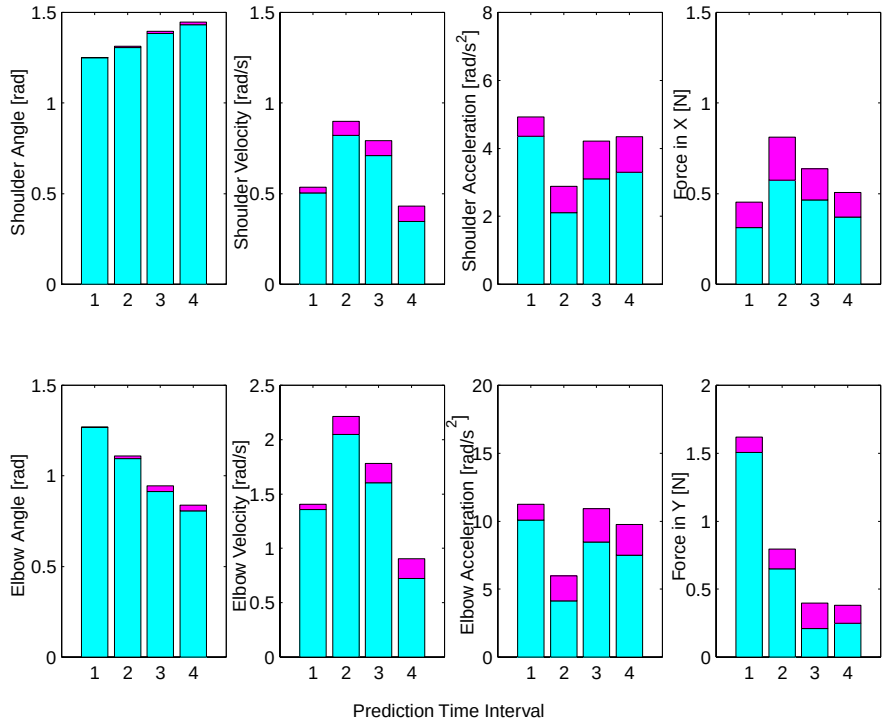

Shoulder Acceleration [rad/s^2]
Elbow Acceleration
Force along X [N]
Force along Y [N]
5 Sample Trials
Time [s]

Figure 3.9: Sample results of trajectory prediction for 3 typical subjects. Black solid line: predicted trajectory. Gray dash line: measured trajectory.

Model validation: We have measured 264 unperturbed movements for each subject. The beginning of movements is decided when hand velocity is equal or bigger that 0.03 m/sec. We also record the data 0.1 sec before the beginning of movements. We index this data with the time from –0.1 sec to 0 sec. We use the movement information from –0.1 sec to 0.1 sec to predict the information afterwards. We divide the recorded 264 movements into three sets A (132 movements), B (64 movements) and C (64 movements). We use set A as the base data for prediction. Our principal component analysis models are trained on B and then used to predict movements in C to for model validation.



B


Shoulder Angle [rad]
Shoulder Velocity [rad/s]
Shoulder Acceleration [rad/s^2]
Force in X [N]
Elbow Angle [rad]
Elbow Velocity [rad/s]
Elbow Acceleration [rad/s^2]
Force in Y [N]
Prediction Time Interval

C

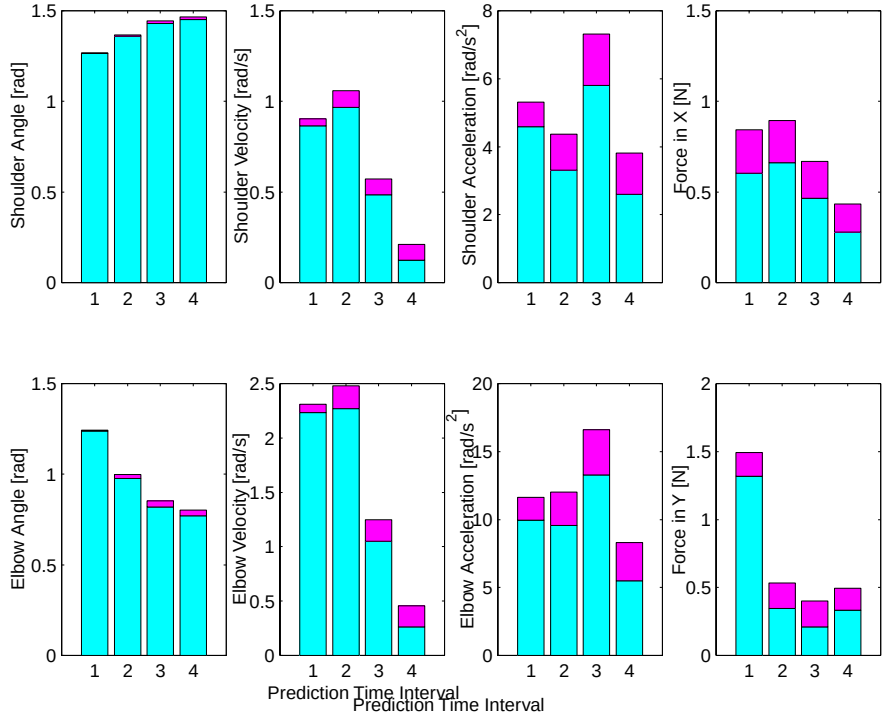

Shoulder Angle [rad]
Shoulder Velocity [rad/s]
Shoulder Acceleration [rad/s^2]
Force in X [N]
Elbow Angle [rad]
Elbow Velocity [rad/s]
Elbow Acceleration [rad/s^2]
Force in Y [N]
Prediction Time Interval
Prediction Time Interval

Figure 3.10: Average of absolute error on top of average of absolute measured value for 4 time intervals after perturbations for three subjects **A, B and C**. Interval 1: [0.1 0.2] sec, Interval 2: [0.2 0.3] sec, Interval 3: [0.3 0.4] sec and Interval 4: [0.4 0.5] sec. Perturbations are about 0.1 sec long and are imposed on the arm 0.1 sec after movements begin.

One can see from Figure 3.10 that the angle, velocity, acceleration and interaction force on the hand can be predicted quite well at least until 300 msec after the beginning of perturbations. Actually, we are interested in the change of the angle, velocity, and acceleration and interaction force caused by perturbations compared to the data if movements were not perturbed. So if for some perturbed movements we can not predict the unperturbed ones well, the predication errors are still acceptable compared to the difference caused by perturbations.

Chapter 4

Data Analysis and Results

4.1 Introduction

The experimental results from the last chapter, the human motor learning of novel dynamic force fields, indicates that human subject can adapt their internal models to predict the dynamics of the force field (Shadmehr and Mussa-Ivaldi, 1994). While there is no experimental evidence for this idea in the central nervous system, substantial evidence indicates that learning the control of arm movements involves formation of an internal model. By observing the behavior of “after-effects” after the force field was removed, investigators provided evidence in support of this theory. Internal models have been divided into two varieties: forward models and inverse models. The computational nature of such internal models, whether an inverse model or a combination of both forward inverse models, was examined by simulation (Bhushan and Shadmehr, 1999). Though forward model alone is effective for stable feedback control of movement, it is not able to exactly produce the desired trajectory. The reason for this inability is that dynamics of the plant are completely ignored in the controller, and feedback gains are not infinitely high. Remarkable similarities in instability and near path-discontinuities in the kinematics of arm movements between simulations and human behavior were observed. These results suggest that learning control of novel dynamics is accomplished with an adaptive forward model of the system. By measuring the neural activation to muscles, represented by force or torque, we tried to find more evidence for the existence of

forward models in the human motor controller and its learning and adaptation to the novel dynamic environments.

4.2 Mathematical Modeling of Human Motor Control

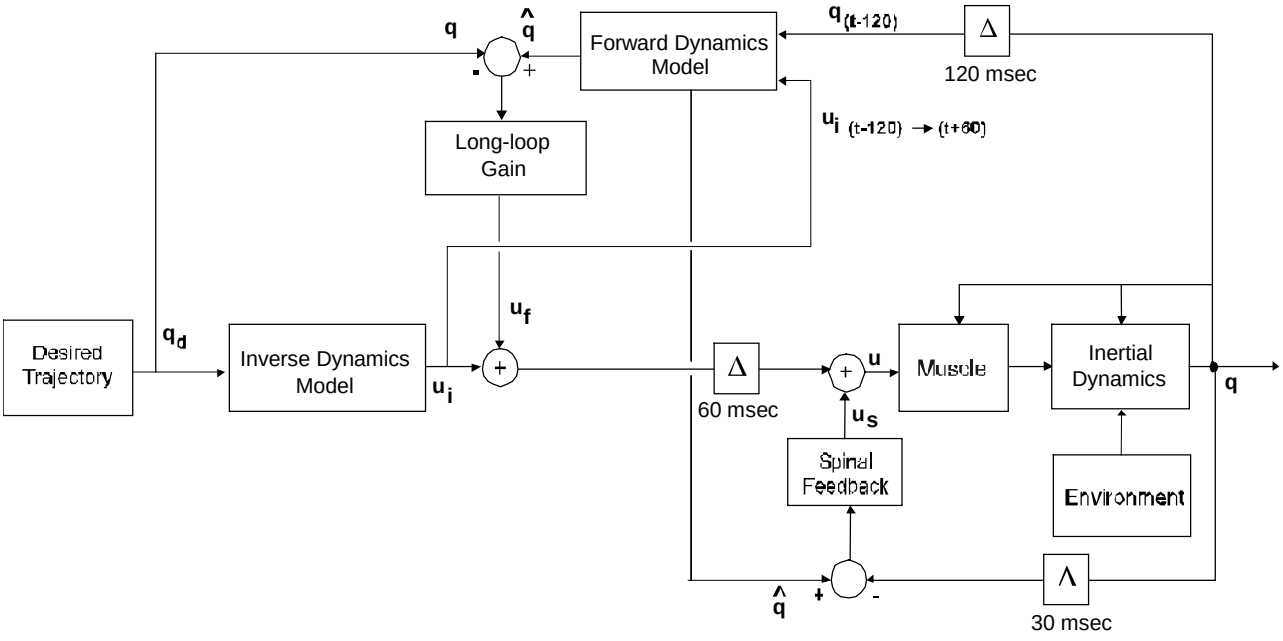


Figure 4.1: Block diagram illustrating the current understanding of the human motor control structure with both feedforward and feedback control. A control system that provides feedback control with the use of a forward and an inverse model.

The purpose of the mathematical modeling was to describe the concept and function of internal models and to understand how the adaptive controller learns an internal model of the force field produced by the robot. The adaptive controller was modeled to reasonably estimate the biomechanical behavior of the human arm. We built on ideas introduced in our previous work (Bhushan and Shadmehr, 1999). Figure 4.1 shows the diagram for the computing structure of the human motor controller.

The control system represents the arm's dynamics in the subject's joint coordinates. The variable $q(t)$ represents a point in configuration space (e.g., an array of joint angles). $\dot{q}(t)$ and $\ddot{q}(t)$ represent its first and second time derivatives respectively. $q(t)$, $\dot{q}(t)$ and

$\ddot{q}(t)$ are joint trajectory, velocity and acceleration vector respectively ($q(t) = \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix}$, where $q_1(t)$ is shoulder angle and $q_2(t)$ is elbow angle). In figure 4.1, the box labeled "Inverse Dynamics Model" is a feedforward adaptive controller. Adaptation to novel external dynamics occurs through learning of a new inverse model of the altered external environment. When the inverse model is an exact inverse of the forward plant dynamics, given the task of point-to-point reaching movements the arm can exactly track the desired trajectory q_d (input of the inverse model). We assume the desired trajectory, q_d, is a minimum jerk trajectory of the hand to the target (Flash and Hogan, 1985) with a movement period of 0.5 sec. We represent the part of neural command, output of the inverse model μ_i as

$$\mu_i = \hat{f}_p^{-1}(q_d, \dot{q}_d, \ddot{q}_d) \tag{4.1}$$

where $\hat{f}_p^{-1}$ is the estimated inverse of the forward plant dynamics. It maps the desired position q_d, velocity $\dot{q}_d$ and acceleration $\ddot{q}_d$ of joint angle into descending neural commands μ_i. When the inverse model perfect models the dynamics of the arm, $\hat{f}_p^{-1} f_p = 1$, and there is exact tracking of the desired trajectory. However, when the inverse model is not accurate, the correct torque values are not generated which in turn causes deviation from the desired trajectory q_d. The box labeled "Spinal Feedback" corrects for errors between the desired and the actual muscle state q, by producing a corrective neural activation μ_s based on a linear feedback controller with constants K_s and B_s. If the gain for a zero-delay feedback loop is infinity, then it can be easily shown that the output of the two-joint system is equal to the set-point at all times. However, the gain of the spinal reflex is limited by $\Delta = 30$ msec plus muscle activation delay in the feedback loop. The equation that relates these variables is:

$$\mu_s = K_s(\hat{q} - q(t-\Delta)) + B_s(\dot{\hat{q}} - \dot{q}(t-\Delta)) \tag{4.2}$$

The "Forward Dynamics Model" box provides this control architecture with feedback control of arm movements in addition to forward control. The feedback signal corrects for unmodeled disturbances to the system. Delays in the feedback cause instability. The forward model also estimates the state of the arm at the current time, given the delayed state at some earlier time $t-\Delta$ and a history of descending motor command $\mu_i(t)$ from $t-\Delta$ up to the current time t (Δ is the time delay in the long-loop feedback control). We forward model design is as follows:

$$\sum_0 :\ \breve{\ddot{q}} = \hat{f}_p\left[\mu_i(t), \breve{q}(t), \breve{\dot{q}}(t)\right]$$

$$\breve{\dot{q}}(t-\Delta+it_0) = \breve{\dot{q}}(t-\Delta+(i-1)t_0) + \int_{t-\Delta+(i-1)t_0}^{t-\Delta+it_0} \breve{\ddot{q}}(t)dt$$

$$\breve{q}(t-\Delta+it_0) = \breve{q}(t-\Delta+(i-1)t_0) + \int_{t-\Delta+(i-1)t_0}^{t-\Delta+it_0} \breve{\dot{q}}(t)dt$$

$$i = 1...\frac{\Delta}{t_0}$$

$$\hat{q}(t) = \breve{q}(t) \quad \hat{\dot{q}}(t) = \breve{\dot{q}}(t)$$

where $\hat{q}$ and $\hat{\dot{q}}$ the output of the forward model, $\breve{q}$ and $\breve{\dot{q}}$ are intermediate variables used by the forward model. The above equations represent the interactive solution of a non-linear differential equation $\hat{f}_p$ at time t, given initial state of the system $q(t-\Delta)$ and $\dot{q}(t-\Delta)$ and the input μ_i during the time interval $t-\Delta$ to t. t_0 is the discretized interation time interval which should be infinitely small. We represent the neural command which is from the feedback of forward model by μ_f,

$$\mu_f = K_f(\hat{q} - q_d) + B_f(\hat{\dot{q}} - \dot{q}_d) \tag{4.3}$$

where $\hat{q}$ is estimated current state and K_f and B_f are the long loop gain. This controller requires computation of muscle state through the forward model and then uses a linear controller to vary the neural activation to the muscle based on the error. The net activation μ going to the plant muscles is the sum of the activations from the brain $\mu_i + \mu_{f_s}$ and activation from the spinal reflex μ_s.

$$\mu = \mu_i(t-\tau) + \mu_f(t-\tau) + \mu_s \tag{4.4}$$

μ_i and μ_f is limited by $\tau = 60$ sec delay. The motor neural command μ, which is sent to from supraspinal central nervous system (CNS) to activate human muscles, relies on both forward and inverse model. These neural signals are then programmed to produce control force or torque, which allow for exact tracking of the desired trajectory q_d. To quantify μ, we can examine the control force or torque during arm movements, which are generated by the neural signal μ. We models the dynamics of the human motor control system coupled (in parallel) with its environment:

$$\Psi(q, \dot{q}, \ddot{q}) = M(q, \dot{q}, \mu) + E(\dot{q}) \tag{4.5}$$

where $\Psi(q, \dot{q}, \ddot{q})$, a time-invariant component, denotes a two-link arm dynamics. $E(\dot{q})$ denotes joint torque which depend on dynamics of the environment. $M(q, \dot{q}, \mu)$ represents joint torque generated by the motor neural commands, was an adaptive controller implemented by the motor system of the subject. It relies on both the inverse and forward model controls. Considering the length-tension and velocity-tension relationships of muscle force, the generated torque, $M(q, \dot{q}, \mu)$, can be represented as a function of joint trajectory, velocity, and motor command, μ, descending from the CNS.

So the neural activation, output of human motor controller, the control torque, can be represented by joint torque in joint coordinates as:

$$M(q,\dot{q},\mu) = \Psi(q,\dot{q},\ddot{q}) - E(\dot{q}) \tag{4.6}$$

Note: We model the human arm as a two joint revolute arm. To determine control force which is represented by M_F, I multiplied the control torque $M(q,\dot{q},\mu)$ around joints by $J^T(q)^{-1}$, the inverse of transposed subject's hand Jacobian.

$$M_F = J^T(q)^{-1}(\Psi(q,\dot{q},\ddot{q}) - E(\dot{q}))$$

The torque represented by Ψ is itself a sum of inertial, coriolis/centripetal, and friction forces, which is the inertia model of human arm. For movements in the null field, $E(\dot{q}) = 0$, we know that the torque required to move along the desired trajectory expressed in terms of joint angle, is given by:

$$T_d = H(q_d)\ddot{q}_d + C(q_d,\dot{q}_d)\dot{q}_d \tag{4.7}$$

where H and C were inertial and coriolis/centripetal matrix functions. With respect to the function $M(q,\dot{q},\mu)$, simulations have previously suggested a reasonable lumped model of the subject's biomechanical motor controller in the case of point-to-point movements is as follows (Shadmehr and Brashers-Krug, 1997):

$$\mu_s = K_s(\hat{q} - q(t-\Delta)) + B_s(\dot{\hat{q}} - \dot{q}(t-\Delta))$$

$$\begin{aligned} M = {} & \hat{H}(q_d)\ddot{q}_d + \hat{C}(q_d,\dot{q}_d)\dot{q}_d - E(q_d) - K_f(\hat{q} - q_d) - B_f(\dot{\hat{q}} - \dot{q}_d) \\ & - K_s(\hat{q} - q(t-\Delta)) - B_s(\dot{\hat{q}} - \dot{q}(t-\Delta)) \end{aligned} \tag{4.8}$$

where $\hat{H}$ and $\hat{C}$, which is the inverse model built by human motor controller for the dynamic of human arm, are the approximation of H and C. $\hat{q}$ and $\dot{\hat{q}}$ are the output of the forward model, the estimate of current state. q_d is the reference trajectory planned by the motor control system of the subject. $q(t-\Delta)$ and $\dot{q}(t-\Delta)$ are delayed sensory

feedback. K_f and B_f are the long loop gain and K_s and B_s are the short loop gain. This controller relies on an inverse model of dynamics of the subject's arm, represented by $\hat{H}(q_d)\ddot{q}_d + \hat{C}(q_d, \dot{q}_d)\dot{q}_d$. $\hat{H}(q_d)\ddot{q}_d + \hat{C}(q_d, \dot{q}_d)\dot{q}_d$ is the feedforward component of controller. It maps the desired position q_d, velocity $\dot{q}_d$ and acceleration $\ddot{q}_d$ of the arm joint, into descending neural commands to generate the required torque or force to move the arm. $K(q - q_d) + B(\dot{q} - \dot{q}_d)$ is the feedback component of controller. K and B are linear estimates of subject's joint stiffness and viscosity. The dynamic environment of the force field represented by $E(q_d)$. $E(q_d)$ equals zero when movements are performed in the null field.

The purpose of the computational modeling was to predict the change in the pattern of control torques or forces M that should result if adaptive control system learned to completely compensate for the dynamics of the force field. M is a sum of the desired torque (feedforward control) and corrective terms (feedback control). The corrective terms are the converging control force about the desired state of the system at time. They have zero forces only when the actual paths exactly track the desired paths. We assume that the short-loop feedback plays no role in the learning task of the human motor control. So the change in the pattern of corrective terms due to learning of the force field will mainly attribute to the forward model feedback controller. So by examining the corrective terms, we can get some knowledge the existence and characteristics of the forward model controller of CNS.

How to isolate the corrective terms (feedback control terms) from the sum of control force M (both feedforward and feedback)? We designed experiment with perturbing technique to quantify the control force only attribute to the feedback part of the human controller (See chapter 3). The assumption for the perturbed movements is that since the direction of perturbing force is randomly selected, it can not be learned by the subject. So the desired trajectory remains same for unperturbed and perturbed movements in the same force field environment.

When subject adapted to the passive dynamics of the robot manipulandum (in the null field), movements are constant and remain nearly unchanged across trials. For normal movements in the null field ($E(\dot{q}_d) = 0$) without perturbation, trajectories exactly track the desired trajectory $q_d, \dot{q}_d$. In other terms, $q = q_d$, $\dot{q} = \dot{q}_d$ and the corrective terms of the controller are zero. The control force in this situation is represented by M_0:

$$M_0 = \hat{H}(q_d)\ddot{q}_d + \hat{C}(q_d, \dot{q}_d)\dot{q}_d \tag{4.9}$$

For the movements, which are perturbed, and also in the null field, the resulted trajectories were deviated from the desired trajectory. We represent the deviation by Δq. Note that q_d remains unchanged when movements are perturbed. We name the resulting control force M_p.

$$M_p = \hat{H}(q_d)\ddot{q}_d + \hat{C}(q_d, \dot{q}_d)\dot{q}_d - K\Delta q - B\Delta\dot{q} \tag{4.10}$$

The difference between M_p and M_0 is represented by ΔM :

$$\Delta M_{q_d}(\Delta q, \Delta\dot{q}) = M_p - M_0 = -K\Delta q - B\Delta\dot{q} \tag{4.11}$$

ΔM is evaluated along the desired trajectory by the small deviation of Δq and $\Delta\dot{q}$. One can see that the control force change $\Delta M_{q_d}(\Delta q, \Delta\dot{q})$ only attributes to the feedback terms of the human motor controller and has no relation with the feedforward terms (inverse model) of the controller.

For the learned movements in the force field, we have $q = q_d$, $\dot{q} = \dot{q}_d$. $E(\dot{q}_d) \neq 0$. For movements in the force field which are not perturbed, we represent the control force during movements as $\tilde{M}_0$,

$$\tilde{M}_0 = \hat{H}(q_d)\ddot{q}_d + \hat{C}(q_d, \dot{q}_d)\dot{q}_d - E(\dot{q}_d) \tag{4.12}$$

$\tilde{H}$ and $\tilde{C}$ are the inertia model of human arm in the force field. For movements, which are perturbed in the force field, we represent the control force during movements as $\tilde{M}_p$,

$$\tilde{M}_p = \hat{H}(q_d)\ddot{q}_d + \hat{C}(q_d, \dot{q}_d)\dot{q}_d - E(q_d) - \tilde{K}(q - q_d) - \tilde{B}(\dot{q} - \dot{q}_d) \qquad (4.13)$$

(13)-(12), we have the control force change caused by perturbations in the force field,

$$\Delta\tilde{M}_{q_d}(\Delta\tilde{q}, \Delta\dot{\tilde{q}}) = \tilde{M}_p - \tilde{M}_0 = -\tilde{K}\Delta q - \tilde{B}\Delta\dot{q} \qquad (4.14)$$

Note: The control force ΔM_p, ΔM_0 and control force change $\Delta M_{q_d}(\Delta q, \Delta q)$ in the null field was adapted to $\Delta\tilde{M}_p$, $\Delta\tilde{M}_0$ and $\Delta\tilde{M}_{q_d}(\Delta\tilde{q}, \Delta\dot{\tilde{q}})$, respectively after the learning of force field. However, after subjects learned the force field, the hand trajectory in the force field converged to a path very similar to those observed in the null field. So we assume that the desired trajectories q_d in both conditions are quite similar since human motor controller builds correct internal models for the environments by practice and learning. The deviation $\Delta\tilde{q}$ and $\Delta\dot{\tilde{q}}$ from the desired trajectory caused by perturbations in the force field would be different with the ones (Δq and $\Delta\dot{q}$) in the null field since the change of dynamic environment. With the learning of novel dynamic environment and adaptation of internal model of human motor controller, subjects will use quite different stiffness and viscosity strategy to perform the point-to-point movements. By examining the control force change, we can extract the part of control force change only due to forward model

4.3 Calculation of Control Force Produced by the Human Controller

When movements are perturbed, we recorded the data for $q(t), \dot{q}(t), \ddot{q}(t)$ and the interaction force at the manipulandum $(F_x(t), F_y(t))$. Then the prediction that what the

movements would be if there were no perturbation was made for these perturbed movements. We name these predicted movements unperturbed movements. We have two groups of movements. One, perturbed movements described by $q(t), \dot{q}(t), \ddot{q}(t)$ and the other, unperturbed movements described by $q_0(t), \dot{q}_0(t), \ddot{q}_0(t)$. I used the structure and parameters of the model of the human-robot interaction to estimate the amount of force subjects create during movements, which we named control force here. Using estimates of the human arm configuration and inertia dynamic parameters, I first transformed individual subject's hand position, velocity and acceleration into the arm joint coordinates. From Chapter 3.3, I have discussed that we pre-estimate $a_i, i = 1,2,3$, the inertial dynamics of the arm of each subject. The joints information q, $\dot{q}$ and $\ddot{q}$ were transformed into estimated dynamic torque $\Psi(q, \dot{q}, \ddot{q})$ using inertial model of the subject's arm. $\Psi(q, \dot{q}, \ddot{q})$ was the net torque that moved the human arm (including both the torque produced by the subject's muscles and the interaction torque from the robot manipulandum) .

$$\Psi(q, \dot{q}, \ddot{q}) \cong H(q)\ddot{q} + C(q, \dot{q})\dot{q}$$

I then multiplied the interaction force F_{ext} on the hand by $J^T(q)$, the transposed of the subject's hand Jacobian, to determine the joint torques attributable to the interaction force. The interaction force will compensate both the passive dynamics of the robot manipulandum and the dynamics of the force field environment.

$$\tau_{ext}(t) = J^T(q)F_{ext} \tag{4.15}$$

In the null field, the robot does not produce any active force on the hand. The interaction force is the force used to only compensate the passive dynamics of the robot manipulandum. In the force field, the robot is programmed to generate active forces on the hand while a movement is being made. From the model of two-link human arm dynamics on the horizontal plane, the control force is

$$M(q,\dot{q},\mu) = \Psi(q,\dot{q},\ddot{q}) - \tau_{ext}(t) \tag{4.16}$$

$F_{ext}(t)$ denotes the external force acting on the hand from the robot handle. We name $M(q,\dot{q},\mu)$ as control force for easy understanding. J was the Jacobian matrix describing the differential transformation of coordinates from end point to joints.

Now we have the following control force from computation, M_p for perturbed trajectory in the null field, M_0 for unperturbed trajectory in the null field, $\tilde{M}_p$ for perturbed trajectory in the force field and $\tilde{M}_0$ unperturbed trajectory in the force field. *Note:* When computing control force for the perturbed trajectory, trajectory information and force were measurement data

4.4 Control Force Change Caused by Perturbations

The experiments we describe probed the relative contribution of feedback and "adaptive feedback" mechanisms in the generation of planar, voluntary reaching movements by examining the change of control torque when the perturbed trajectory q is deviated by Δq from the desired trajectory q_0. The current hypotheses of motor control provide a structure with both inverse and forward model for human arm movement control. It predicts that the input of inverse internal model is the desired trajectory, which only depends on the movement task and remains the same after perturbation since the perturbing direction is randomly selected. If we subtract the unperturbed control torque from the perturbed control torque, which we called control toque change here, is only related with the forward feedback control part (long loop and short loop) of the human arm motor control system.

When the hand is displaced from an equilibrium trajectory by an external perturbation, a force is generated to restore the original position. We developed an experimental method to measure and represent the field of elastic forces associated with movement trajectories

of the hand in the horizontal plane. When subjects conducted constant movements in a given direction, torque motors of the robot delivered small perturbations of the hand along different directions at a constant time point. We measured the corresponding restoring torque or forces $\Delta M_{q_d}(\Delta q, \Delta \dot{q})$ before the hand finally reached the targets. Since the inverse model remained unchanged when there is perturbation in the null field environment or force field environment, we assumed that the changes of control torque in $\Delta M_{q_d}(\Delta q, \Delta \dot{q})$ the null field or $\Delta \tilde{M}_{q_d}(\Delta \tilde{q}, \Delta \dot{\tilde{q}})$ in the force field were only due to the long loop delay sensory feedback through cortical structures (forward model). Restoring torque or forces $\Delta M_{q_d}(\Delta q, \Delta \dot{q})$ and $\Delta \tilde{M}_{q_d}(\Delta \tilde{q}, \Delta \dot{\tilde{q}})$ were output of the forward model. We tried to quantify the difference between $\Delta M_{q_d}(\Delta q, \Delta \dot{q})$ and $\Delta \tilde{M}_{q_d}(\Delta \tilde{q}, \Delta \dot{\tilde{q}})$ and find the evidence for the adaptation of the forward model.

Note: From observation, the $\tilde{q}_d$ in force field is nearly comparable with the q_d in the null field. When we try to predict what the path would be if the movements were not perturbed, using the sample data q_d in force field or using the sample data q_d in the null field did not make much difference. So ΔM is not very sensitive to q_d.

After the experiment both in the null field and in the force field, we have two control force changes. One is $\Delta M_{q_d}(\Delta q, \Delta \dot{q})$, the control torque change in the null field and the other is $\Delta \tilde{M}_{q_d}(\Delta \tilde{q}, \Delta \dot{\tilde{q}})$, the control toque change in the force field. . We considered a $\pm 30\%$ change in inertia of the arm and estimate control force with the varied inertia. We found that, the control force change caused by perturbation is not very sensitive to arm inertia because the effect of perturbing force is stronger than inertia for change of control force between perturbed and unperturbed movements.

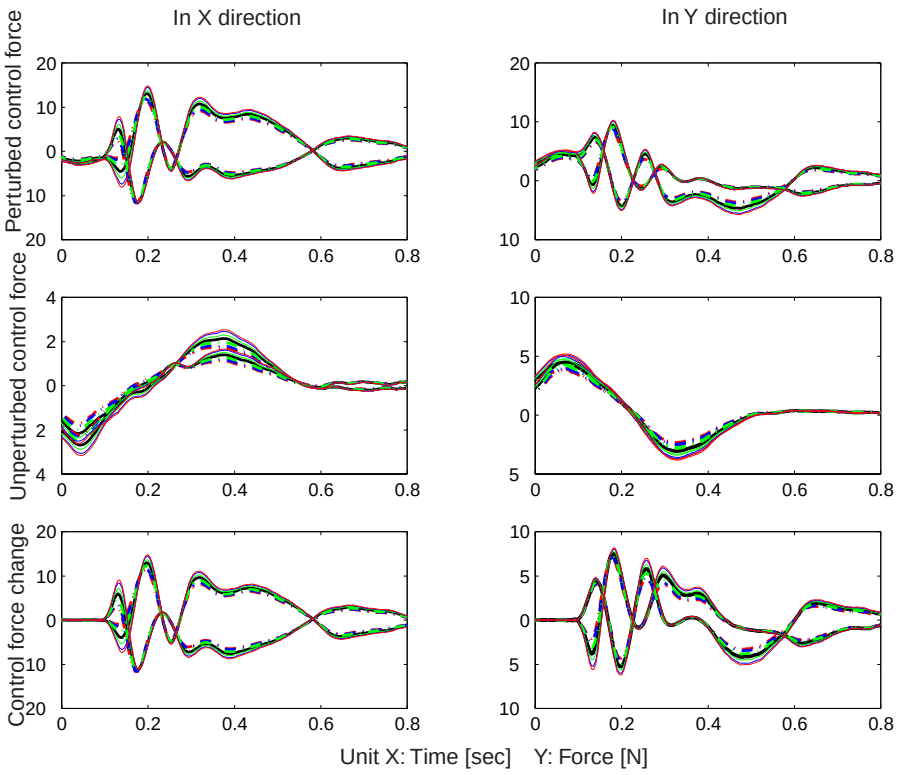


Figure 4.2: Perturbed control force, unperturbed control force and control force change with inaccurate inertial models in the null field. Inertia parameters are increased 10% , 20% and 30%, respectively(solid green, blue and red) and decreased 10% , 20% and 30%, respectively(dashed green, blue and red). Solid line: control force and change with accurate model.

One can see that the change force is not sensitive to the inertia model.

4.5 Modeling the Control Force Change in the Null Field

By using perturbing techniques and predicting what the movements would be if they are not perturbed, we measured the control force change caused by perturbation both in the null field $\Delta M_{q_d}(\Delta q, \Delta \dot{q})$ and in the force field $\Delta \tilde{M}_{q_d}(\Delta \tilde{q}, \Delta \dot{\tilde{q}})$. We want to compare ΔM_{q_d} with $\Delta \tilde{M}_{q_d}$ to see what human have learned from the new dynamic environment and whether or not the learning is performed through the forward model of human motor controller. We know that the control force change is the function of deviation from the desired trajectory. Since there is no way to express the functions in mathematical closed form, we try to quantify the change of functions (the model of human motor controller)

by comparing output of these two functions. The problem is that how can we give the same input to the two functions to compare the output signal? We solve this problem by building the model for $\Delta M_{q_d}(\Delta q, \Delta \dot{q})$ to compute $\Delta M_{q_d}(\Delta \tilde{q}, \Delta \dot{\tilde{q}})$, and then compare $\Delta M_{q_d}(\Delta \tilde{q}, \Delta \dot{\tilde{q}})$ with $\Delta \tilde{M}_{q_d}(\Delta \tilde{q}, \Delta \dot{\tilde{q}})$.

Modeling of human data can be done in many different ways. Most of them are not explicit mathematical models in closed form, but some useful function approximators (FA). Two of the most popular FAs are Neural Networks and Fuzzy Approximators. Having in mind repetitiveness of the task with some explicitly defined parameters modifying it, we chose not to use either of these two, but one other, nonlinear FA particularly suited for parametric modeling. Named Successive Approximations (Dordevic et al., 2000) it is essentially a procedure of successive (non-linear) Least Squares fitting of fitting coefficients. The model, in the form of an ordered set of coefficients, which are function of parameters used in modeling, generalizes well inside but also outside of learning domain. The other, very important, property is that these models can be addressed randomly, giving the output along some trajectories experienced never before.

This approximating procedure is successfully applied in robotics, particularly in kinematic redundancy modeling tasks (Dordevic et al., 2000). With only minor changes it is applied here. First step was a choice of parameters. According to the task definition, we adopted two parameters: kick direction p_{kd} and kick magnitude p_{km}.

The typical data acquired from target-reaching experiment with parameterized kick direction and kick magnitude are shown on following figures. These data are used for subsequent modeling.

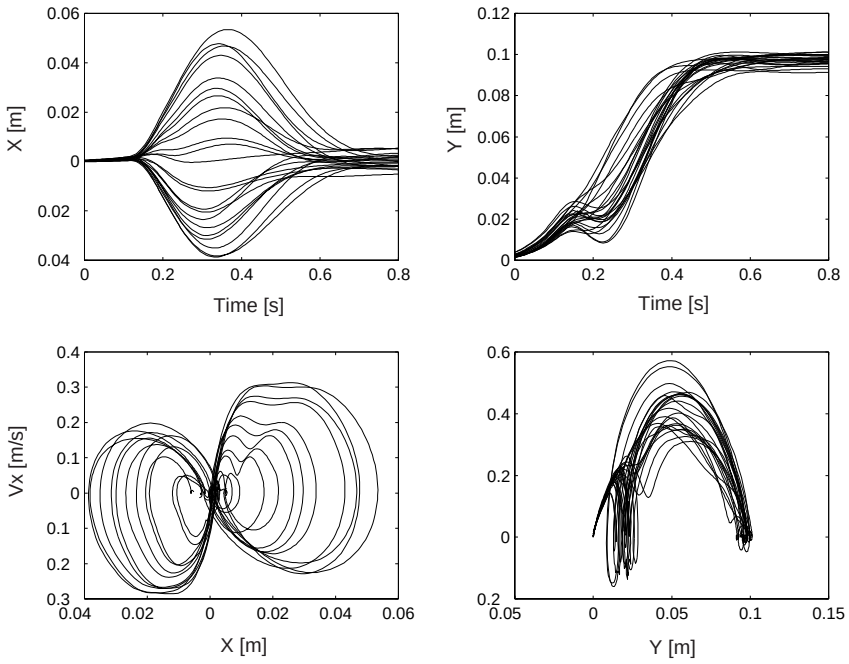


Figure 4.3: Measured perturbed hand trajectories in subject Cartesian coordinates in the null field. Movements were perturbed in 23 angle-equally-spaced directions with the same kick magnitude. We also measured the data for other 8 different kick magnitudes all of which are used to model the control force change in the null field. B: Perturbed trajectories were presented as the time series (top) and velocity-position space (bottom).

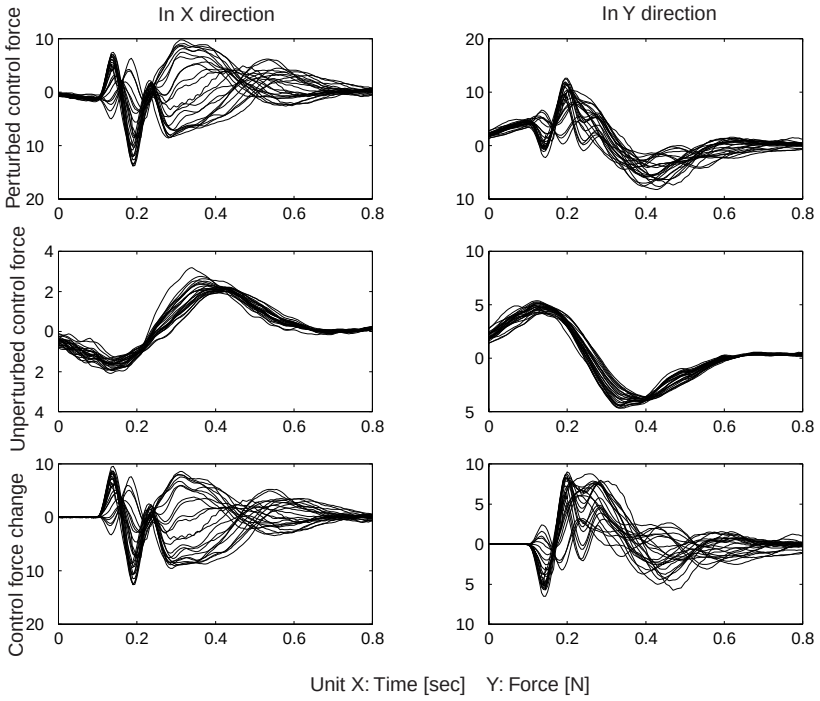


Figure 4.4: The data used for modeling. Measured control force with perturbation and predicted control force if not perturbed for one kick magnitude. Top row: Measured control force when movements were perturbed in the null field. We name this force M_p. Middle row: Predicted control force (What the control force would be if the trajectories were not perturbed in the null field). We name this force M_0. Bottom row: Control force change $M_p - M_0$.

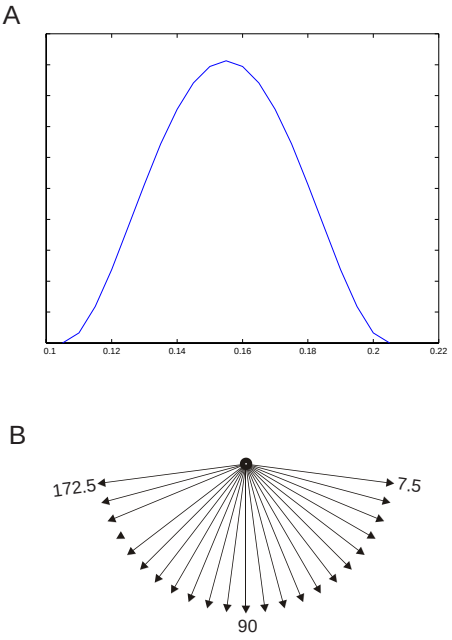


Figure A is the shape of perturbing force with the duration around 100 msec. Human arm is perturbed 100 msec after the beginning of movement. Movements are perturbed in 23 equally spaced directions as shown in Figure B.

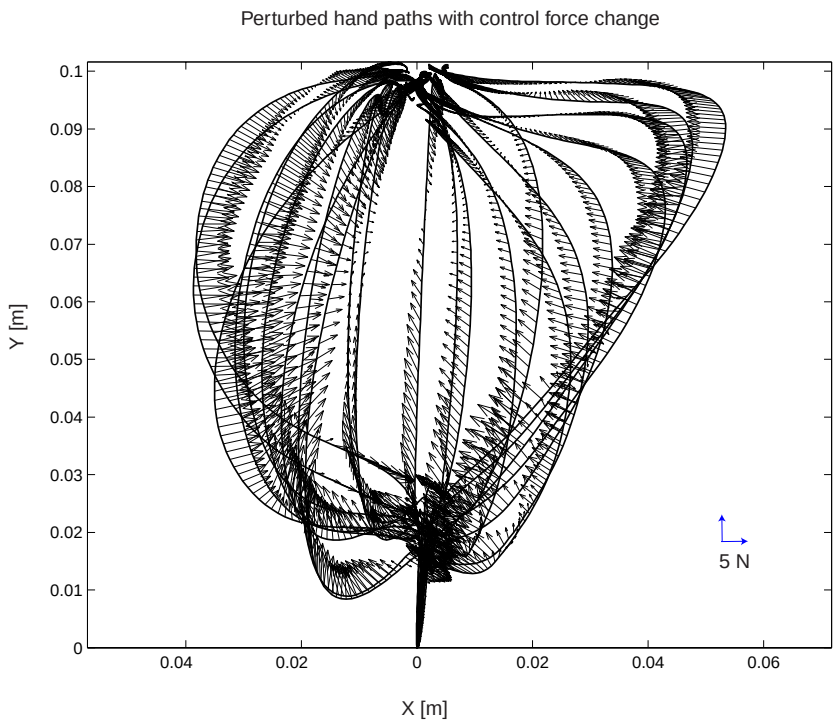
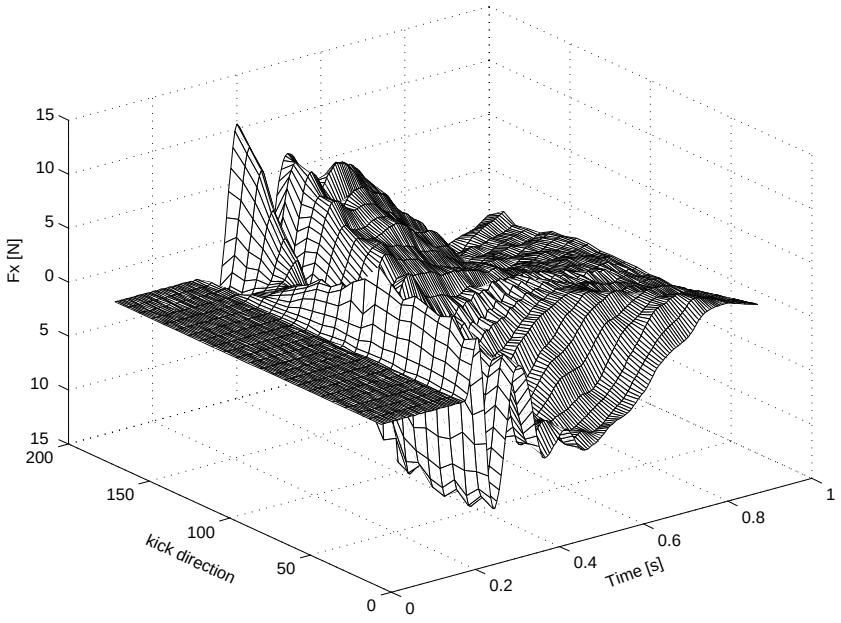
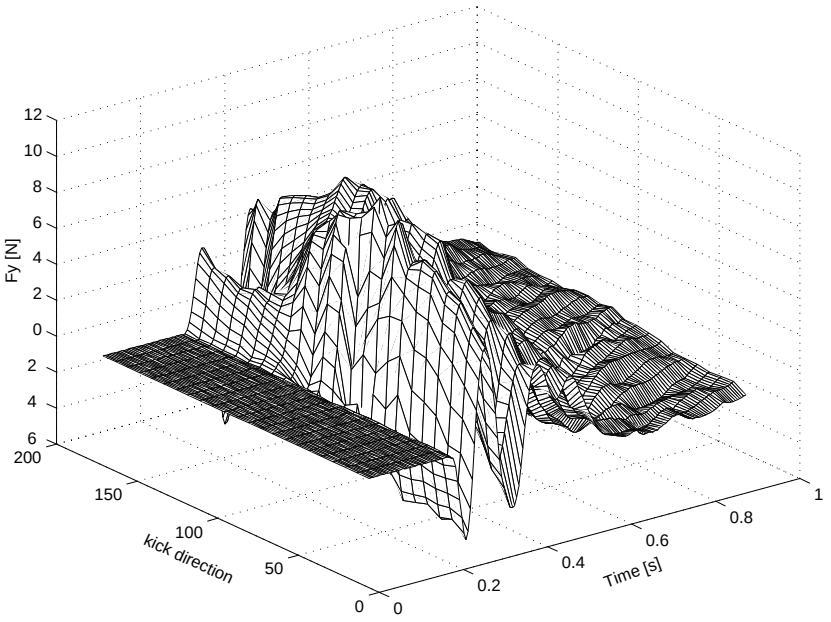
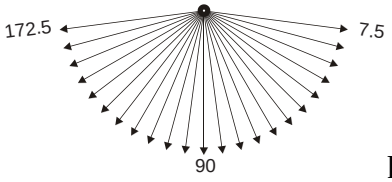


Figure 4.5: Measured control force change with the same kick magnitude in 23 different kick directions $\Delta M = M_p - M_0$ on the top of hand trajectories in the null field caused by perturbations.

A

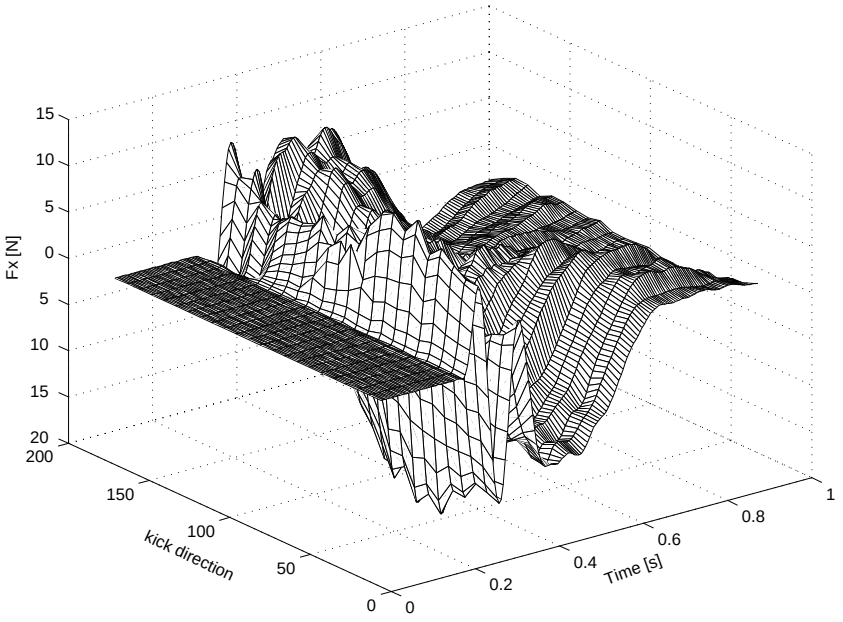


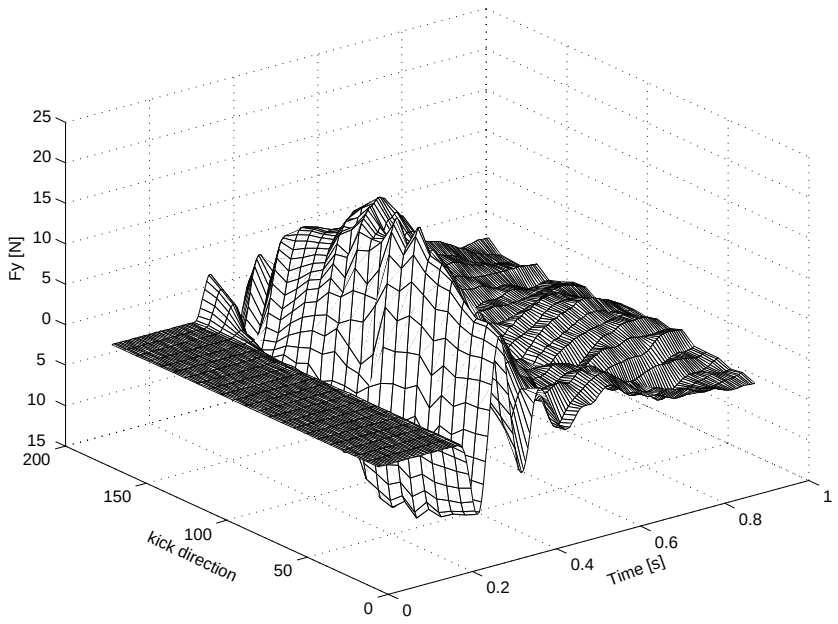




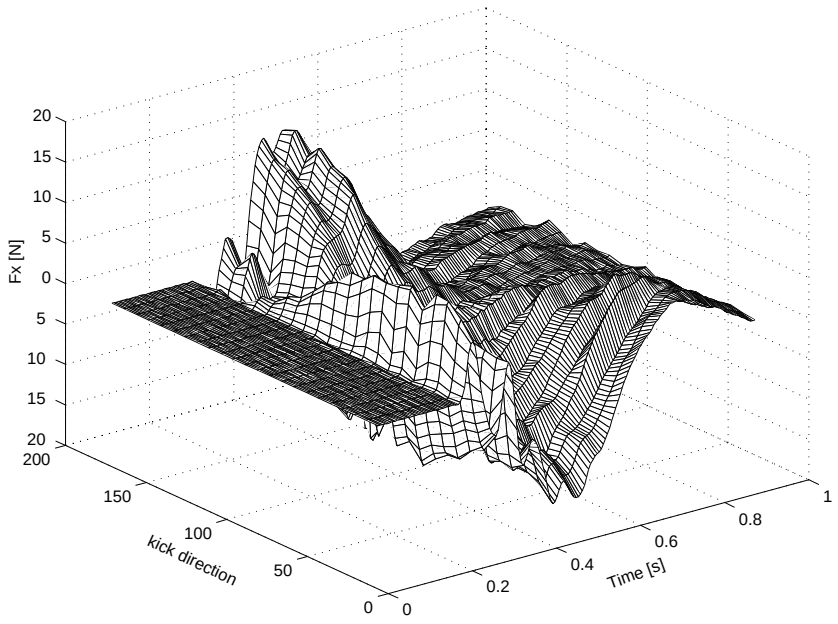
kick directions for modeling in the null field

B




Fy [N]
25
20
15
10
5
0
5
10
15
200
150
100
50
0
kick direction
0
0.2
0.4
0.6
0.8
1
Time [s]

C


Fx [N]
20
15
10
5
0
5
10
15
20
200
150
100
50
0
kick direction
0
0.2
0.4
0.6
0.8
1
Time [s]

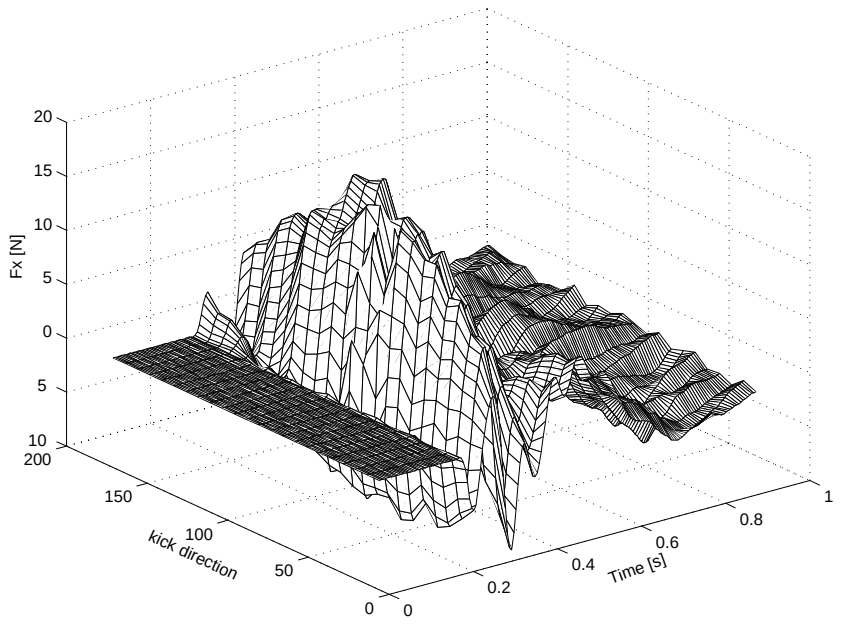


Figure 4.6: Measured control force change with the same kick magnitude in 23 angle-equally-spaced kick directions for three subjects A, B and C.

Hand trajectory, hand velocity and control force due to adaptation of human motor controller are independently modeled, giving models: $Mod_x(t, p_{kd}, p_{km})$, $Mod_{vx}(t, p_{kd}, p_{km})$, $Mod_y(t, p_{kd}, p_{km})$, $Mod_{vy}(t, p_{kd}, p_{km})$, $Mod_{\Delta M_x}(t, p_{kd}, p_{km})$ and $Mod_{\Delta M_y}(t, p_{kd}, p_{km})$. The model based on these two parameters, enables reproduction of hand trajectory, along with force exerted at the handle of manipulandum, for arbitrary values of the parameters that belong to the learning domain. Furthermore, random addressing of the model (Dordevic et al., 1999) enables addressing of a single point only, specified by a set of time instant t^*, p_{kd}^* and p_{km}^*, which lead to addressing of arbitrary trajectory.

Verification of the model is done stepwise. First, we evaluate approximation of time histories of hand positions, velocities and forces in the experiment, by keeping approximation error defined by normalized mean-squared error within 95% confidence limits. A typical trajectory of approximated motion along Ox coordinate is shown on Figure 4.8, along with 95% confidence limits. For all other kick directions, the degrees of

approximating polynomials are chosen with respect to the criteria to tighten confidence limits along the change of independent variable. In other words we tried to keep a balance between overfiting and underfiting underlaying data. Approximation in time finished for all variation of trajectories due to varied kick direction and varied kick magnitude, gives three dimensional set of coefficients of approximating polynomials that can be taken as functions of kick direction and kick magnitude variables.

Again, we performed least squares fitting with respect to the kick direction variable p_{kd}, maintaining the same balance between overfiting and underfiting which will result in acceptable confidence limits. Finally, the same is done with respect to the kick magnitude variable p_{km} (Dordevic et al., 1999).

Finally, we take a whole set of all possible trajectories from the model for kick directions as in experiment. By bootstraping, we took trajectories randomly for large number of times, and evaluated how each of trajectory correlates to the rest of them. Next six box-plots, Figure 4.7. give an information how good models of positions, velocities and forces are with respect to the change of kick direction. General conclusion is that positions are easiest to approximate, comparing to velocities and forces. Also, the quality of the model is a function of kick direction parameter, showing slight decrease of model quality within a region of kicks that produces minor motion perpendicular to the target direction. Based on this and former evaluation of modeling procedure, we take these models as a good basis for further understanding of forward model adaptation during motor learning task.

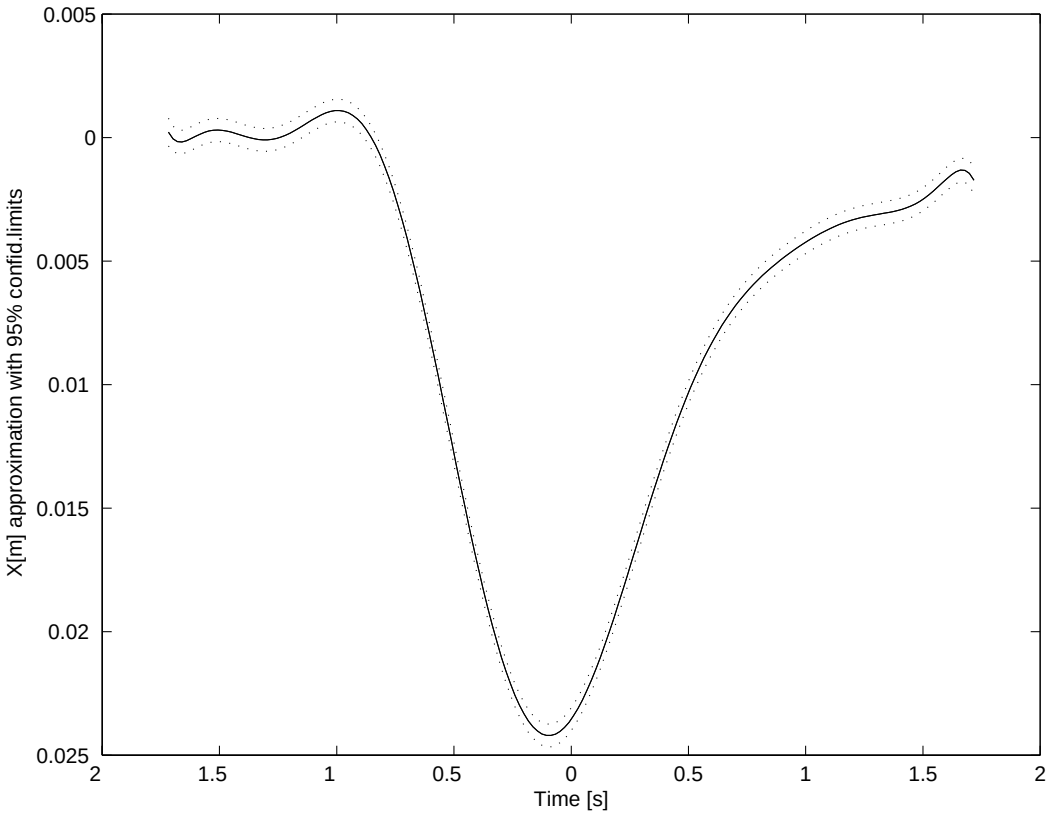


Figure 4.7: Final bootstrap testing of model quality for positions, velocites, and forces.

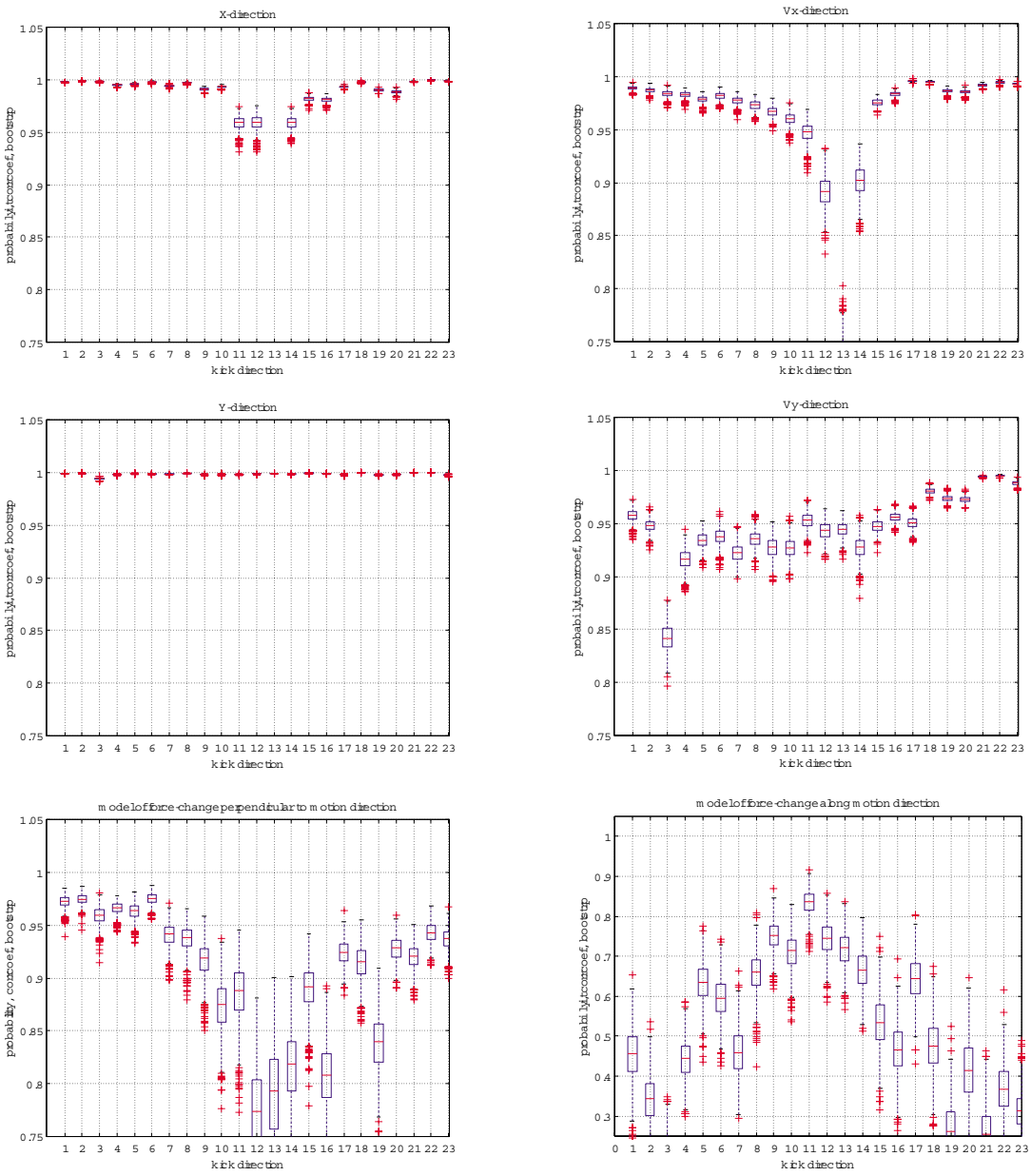


Figure 4.8: Final bootstrap testing of model quality for positions, velocites, and forces.

Having modeling done, we can use it to match arbitrary hand trajectory, assuming that it belongs to the state space regions visited by trajectories for modeling. Essentially, matching problem is a procedure of finding a nearest neighbor defined by time instant, p_{kd}^{*} and p_{km}^{*}, to the desired point of new trajectory

$$T_{new} \equiv \{x_{new}, v_{x\,new}, y_{new}, v_{y_{new}}\}:$$

$$\begin{cases} \min\limits_{t_i, p_{kd}, p_{km}} \left\| \left\{ x^i_{new} - Mod_x(t_i, p^i_{kd}, p^i_{km}) \right\} \wedge \left\{ vx^i_{new} - Mod_{vx}(t_i, p^i_{kd}, p^i_{km}) \right\} \right\|_2 \\ \min\limits_{t_i, p_{kd}, p_{km}} \left\| \left\{ y^i_{new} - Mod_y(t_i, p^i_{kd}, p^i_{km}) \right\} \wedge \left\{ vy^i_{new} - Mod_{vy}(t_i, p^i_{kd}, p^i_{km}) \right\} \right\|_2 \end{cases}$$

Now, for all points of a new trajectory $T^i_{new}, i = 1, \cdots, n$, assuming that we have the same number of points for all trajectories used in modeling, the result of matching are two vectors p^*_{kd} and p^*_{km}. These two vectors simultaneously with time-respective addressing of points can be used in addressing the two force models: $Mod_{\Delta M}(\tau, p_{kd}, p_{km})$:

$$\Delta M_{new} = Mod_{\Delta M}(t, p^i_{kd}, p^i_{km}), \;\; i = 1, \cdots, n$$

All models are three-dimensional sets of functional coefficients. Each dimension corresponds to one parameter, taking time, kick direction, and kick magnitude as parameters. Specifically, models of positions and velocities (x, vx, y, vy) are of the same size, taking 11 coefficients for approximation in time, 10 coefficients in approximation in kick direction, and 4 coefficients for approximation in kick magnitude. It practically means that approximation in time is done with polynomials of 10th degree, approximation in kick direction as a variable with polynomials of 9th degree and finally approximation in kick magnitude as variable with polynomials of 3rd degree. Slightly higher degrees of polynomials are used for force signals modeling. Also, we have modeled the force signals with two models, later blended for smoother transition.

The output of models, after addressing by desired values of time instant, kick direction and kick magnitude, is a polynomial, different for different signals (positions, velocities and forces). For example, model output for x(t) is a polynomial of 10 th degree.

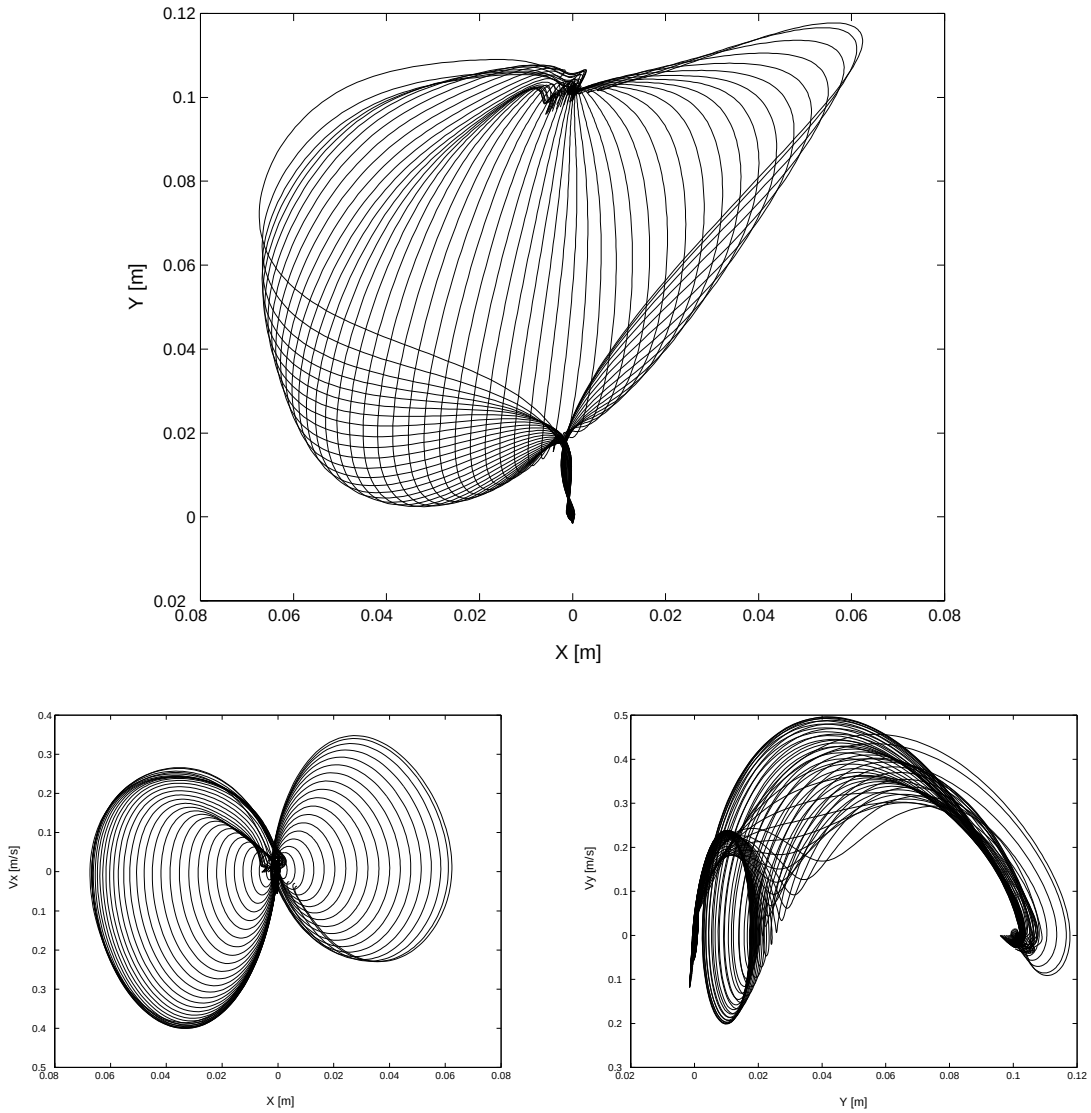
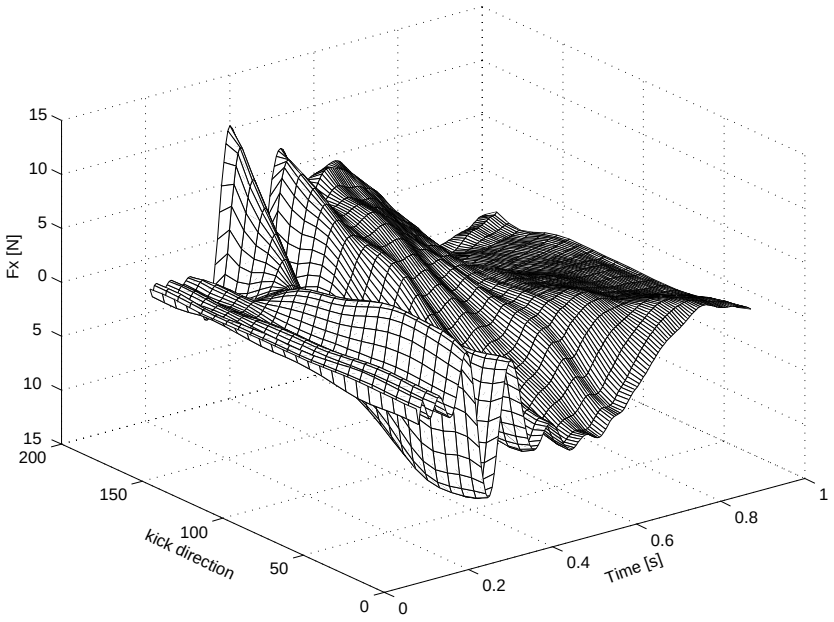
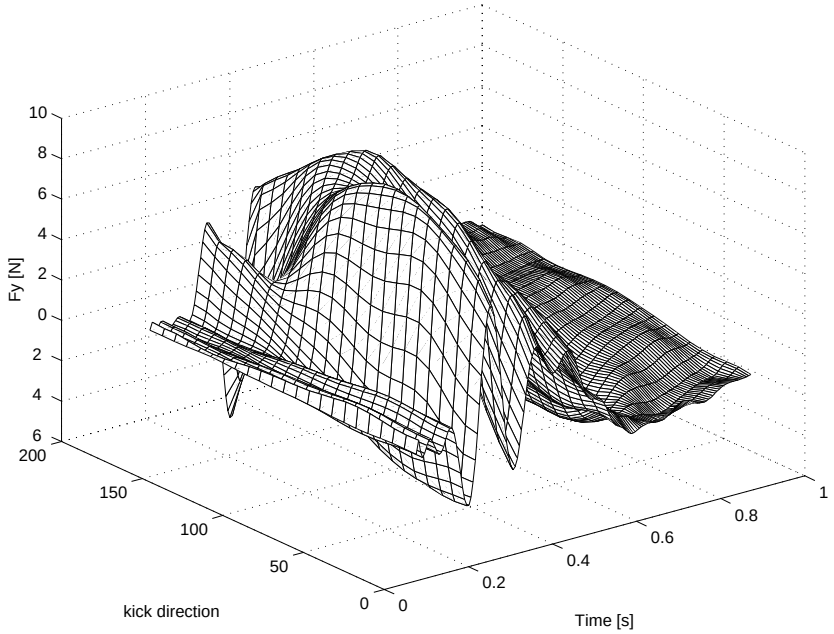


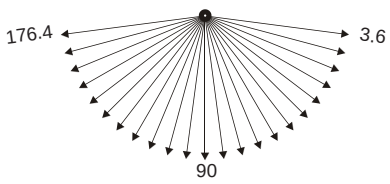
Figure 4.9: Models of perturbed hand trajectories in subject Cartesian coordinates in the null field. Top row: Modeled perturbed hand paths, which are perturbed in 50 angle-equally-spaced directions with the same, kick magnitude. Bottom row: same data are presented in velocity-position space.

A



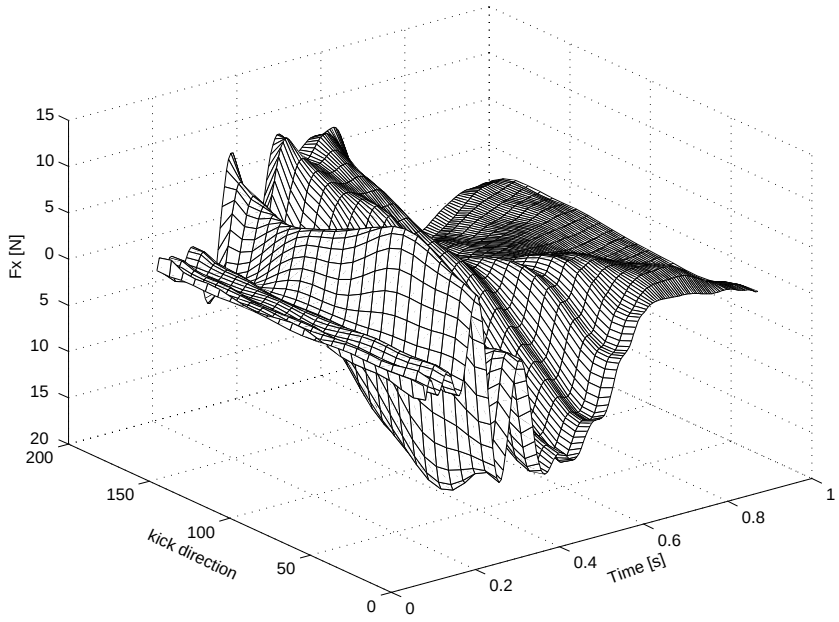
Measured forces along motion direction (in Null Field)

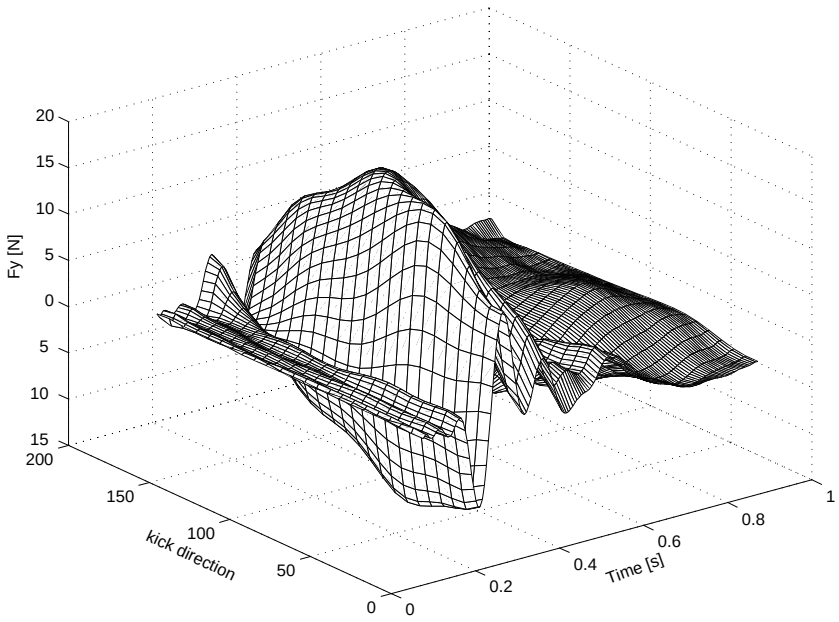




kick directions for modeling in the null field

B


Fx [N]
15
10
5
0
5
10
15
20
200
150
100
50
0
kick direction
0
0.2
0.4
0.6
0.8
1
Time [s]


Fy [N]
20
15
10
5
0
5
10
15
200
150
100
50
0
kick direction
0
0.2
0.4
0.6
0.8
1
Time [s]

C

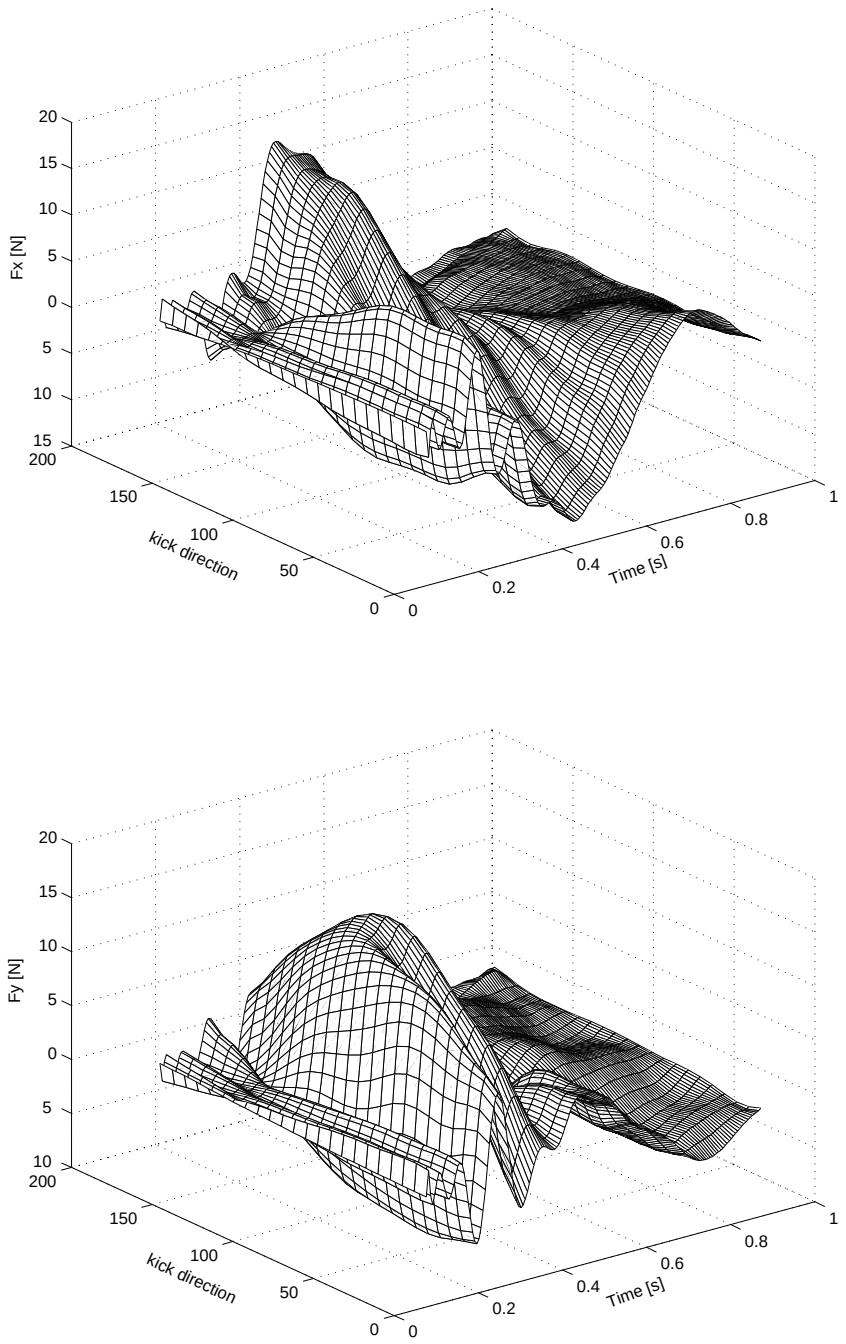


Figure 4.10: Modeled control force change with the same kick magnitude in 50 angle-equally-spaced kick directions.

4.6 Adaptation of the Control Force Change after Learning the Novel Dynamics Environment

Previous simulation indicates that human motor controller where a forward model was used in conjunction with an inverse model results in remarkable kinematic similarities to the observed human behavior. There is no doubt about motor adaptation within the changed mechanical environment. It is pretty straightforward to understand that human motor controller adapts the inverse model to predict the changed dynamic environment. For the unperturbed normal movements in two quite different dynamic environments, in this situation the deviation from the desired trajectories (which are quite similar for movements within null field and force filed) is so small that error feedback from forward model nearly plays no role in controlling movements, human motor adapts very well to both of them. But the stiffness and viscosity of human motor controller, which is adapted to one specific dynamic environment always remains same whether or not there is error in movements. The problem is that it is hard to see when error is so small. Our goal in this thesis is to go one step further. We try to find more evidence of dynamic information from arm movements, which supports the theory that human motor controller does use forward model to control movements. However, the main question is, whether or not the forward model is adapted and how the forward model contributes to the motor learning of subjects.

The reasonable lumped model of the subject's biomechanical motor controller in the case of point-to-point movements is as follows:

$$M = \hat{H}(q_d)\ddot{q}_d + \hat{C}(q_d, \dot{q}_d)\dot{q}_d - E(q_d) - K(q - q_d) - B(\dot{q} - \dot{q}_d)$$

For normal movements without disturbance from outside environment, subject performs so well (deviation from the desired trajectory $\Delta q = q - q_d$ and $\Delta\dot{q} = \dot{q} - \dot{q}_d$ are very small) that the error feedback terms, which attribute to forward model and spinal reflex, are not quite obvious. By perturbing movements in randomly selected directions, which give more deviation from the desired trajectory, we can easily measure the control force due to the feedback terms. (Note: Since perturbations cannot be learned, they have no effect to human motor controller. In other words, controller and desired trajectory are the

same for the movement perturbed and unperturbed). First, we compute the control force change ΔM_{q_d} caused by perturbations in the null field,

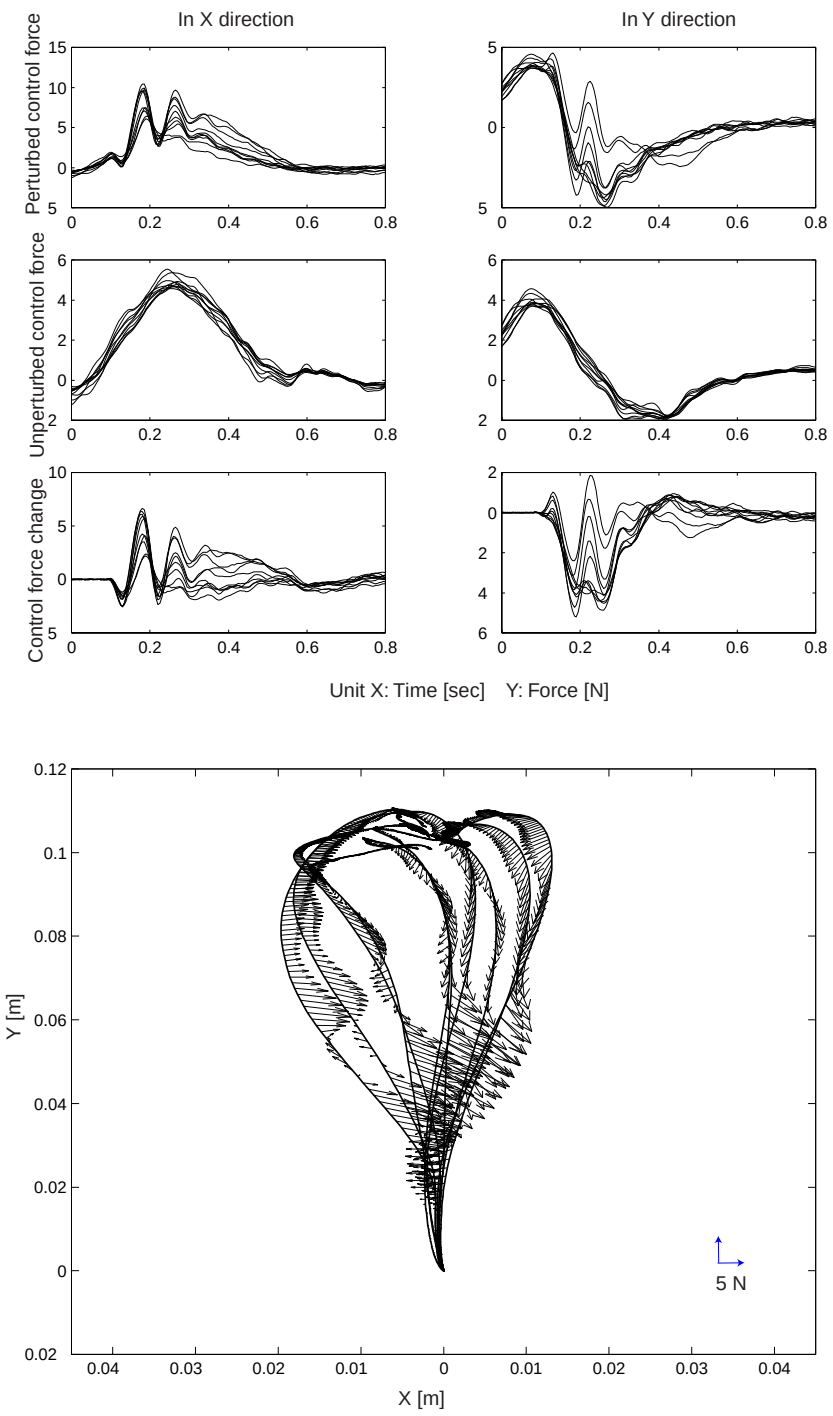
$$\Delta M_{q_d}(\Delta q, \Delta \dot{q}) = M_p - M_0 = -K\Delta q - B\Delta \dot{q}$$

where M_p and M_0 are control force during the perturbed and unperturbed movements in the null field respectively. K and B are linear estimate of subject's joint stiffness and viscosity. One can see ΔM_{q_d} is the force that only attributes the feedback control box. In Figure 4.4 control force change are plotted as arrows on top of the perturbed trajectories in the null field and one can see that the correcting force push the movements come to the desired trajectory, which is nearly a straight line upward towards the target. To provide the same state input to the forward model, we next build parameterized model for the control force change in the null field using all the movements we measured in the null field, which are perturbed in 23 directions with 9 different kick magnitudes. We can estimate the control force change for an arbitrary movement in the null field with this model.

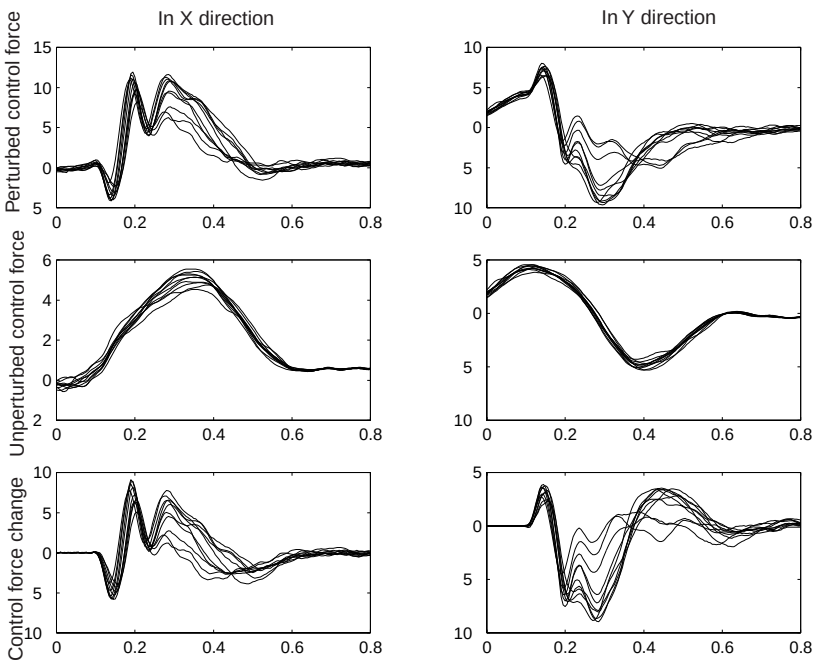
Then, we compute the control force change $\Delta \tilde{M}_{q_d}$ caused by perturbations in the force field. The torque motors of robot produce an additional force at the handle as described by the equation, $\vec{F} = B\dot{x}$, where $\dot{x}$ is the instantaneous velocity vector of the handle and $B = [0 \quad -13; 13 \quad 0]$ N sec m^{-1}.

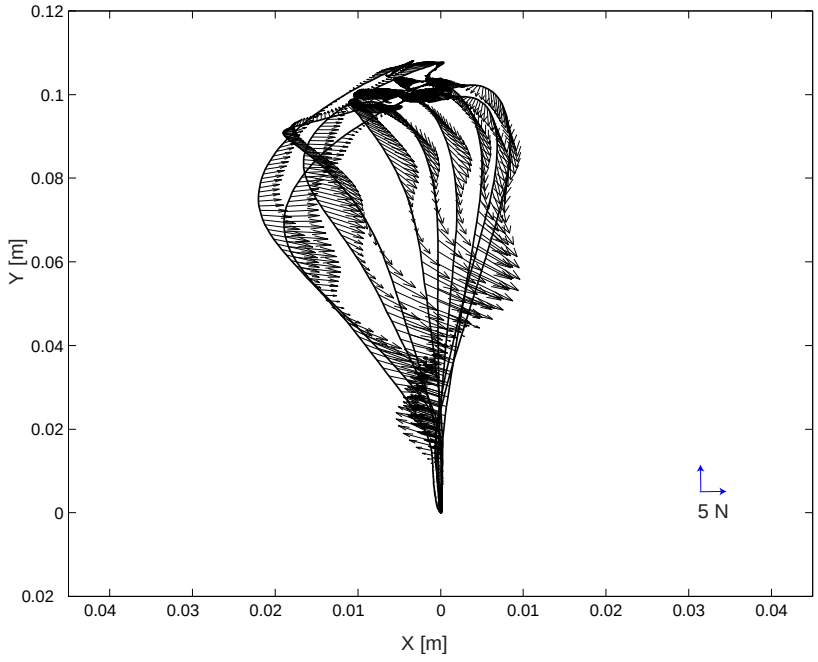
$$\Delta \tilde{M}_{q_d}(\Delta \tilde{q}, \Delta \dot{\tilde{q}}) = \tilde{M}_p - \tilde{M}_0 = -\tilde{K}\Delta \tilde{q} - \tilde{B}\Delta \dot{\tilde{q}}$$

A


In X direction
In Y direction
Perturbed control force
Unperturbed control force
Control force change
Unit X: Time [sec] Y: Force [N]
Y [m]
X [m]
5 N

B


In X direction
In Y direction
Perturbed control force
Unperturbed control force
Control force change
Unit X: Time [sec] Y: Force [N]


Y [m]
X [m]
5 N

C

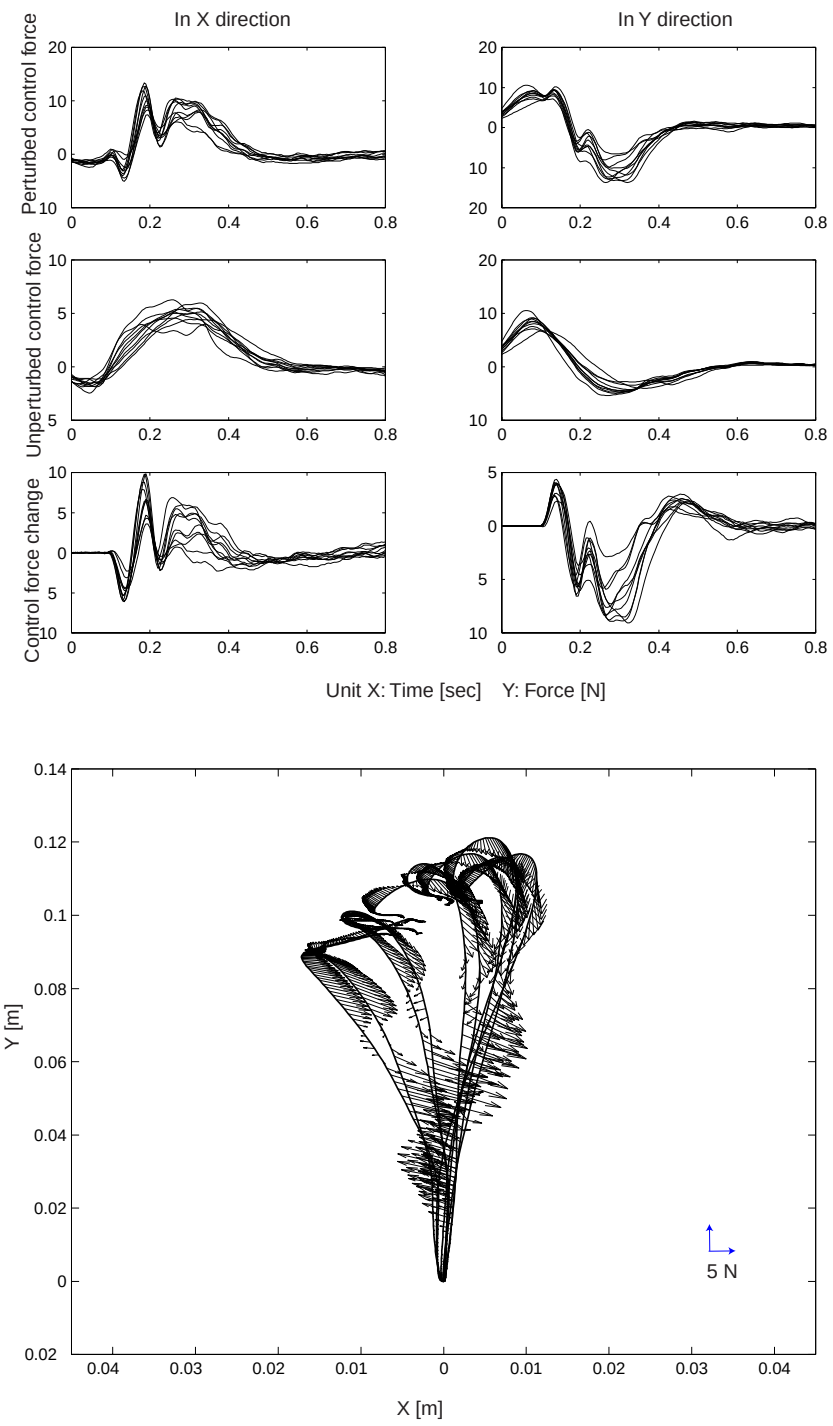
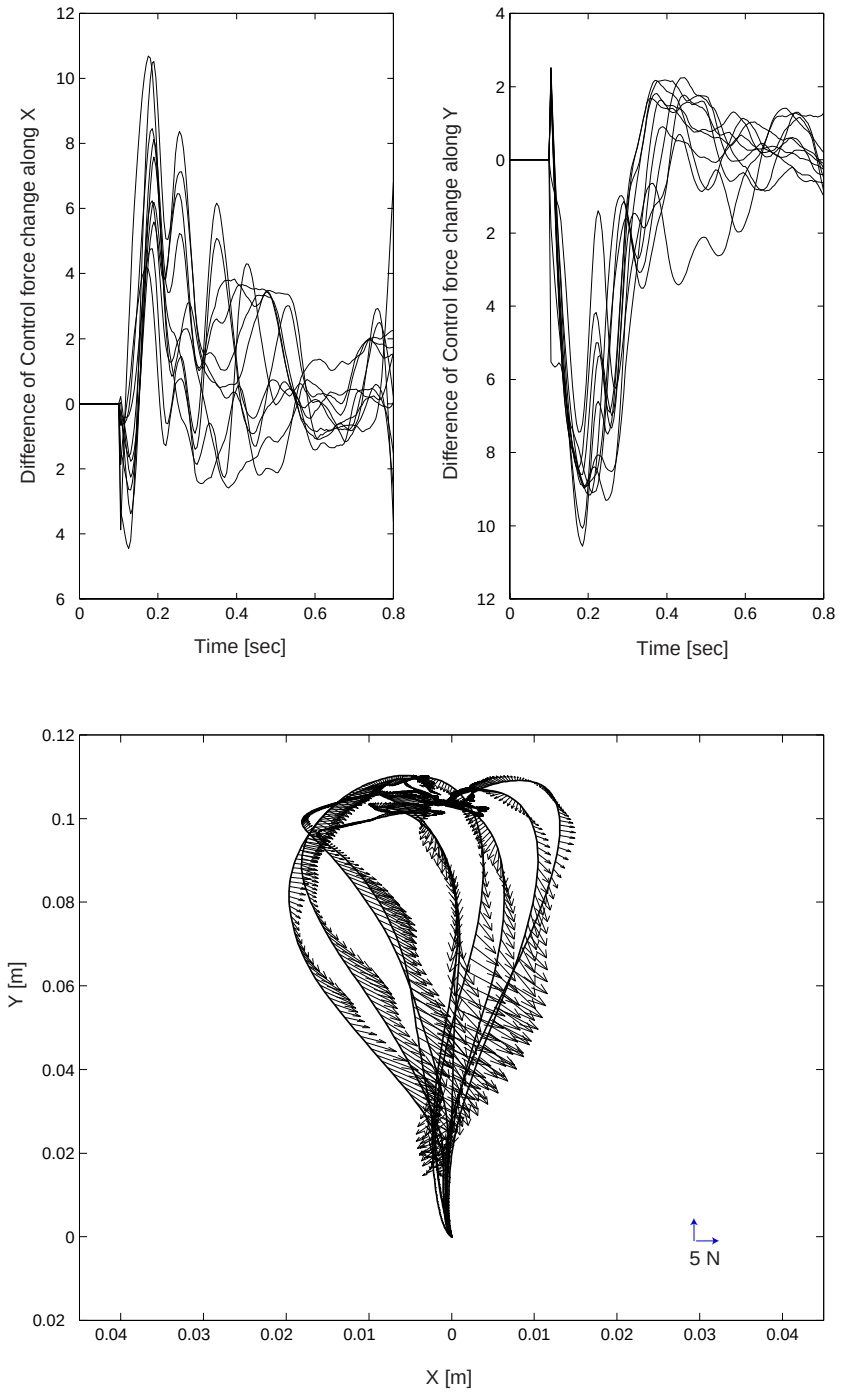
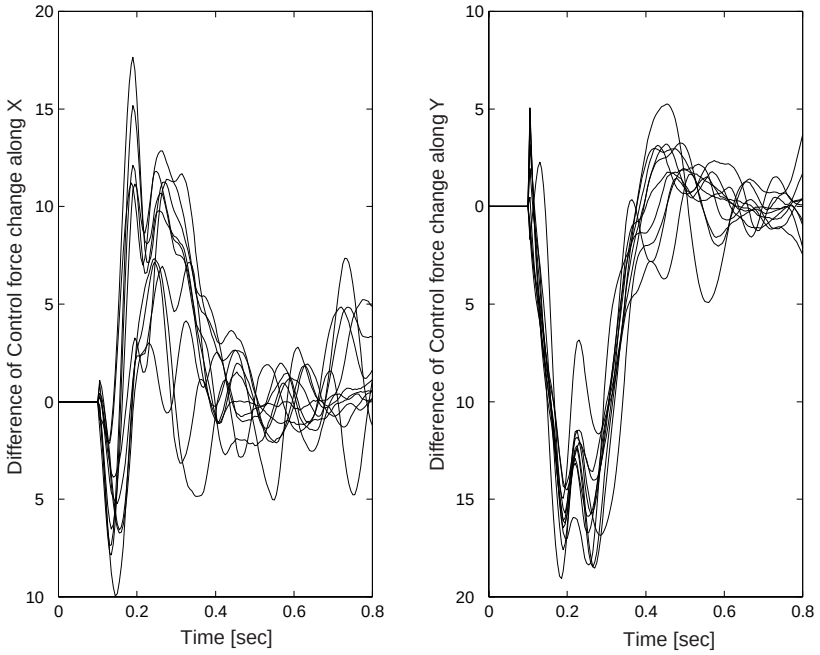


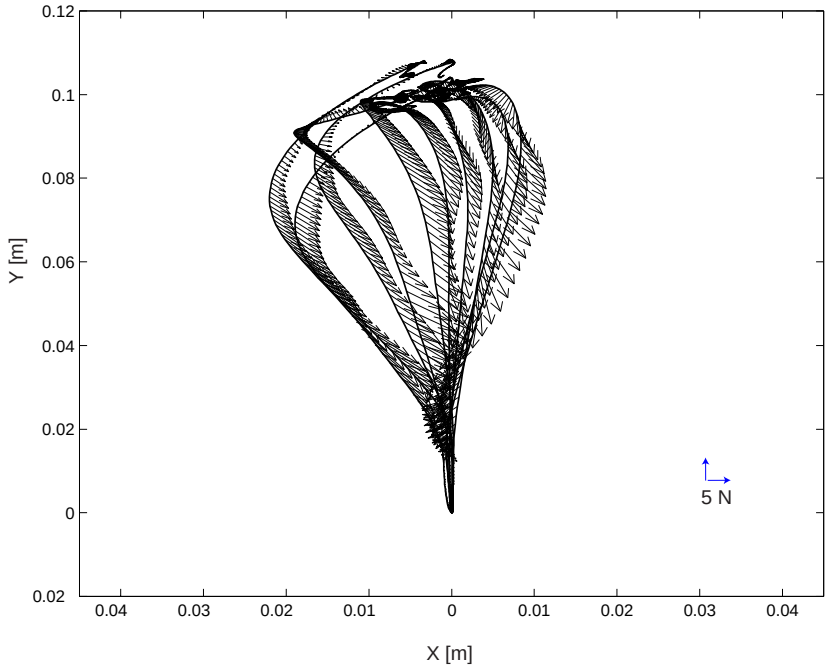
Figure 4.11: Measured control force change $\Delta\tilde{M}_{q_d}(\Delta\tilde{q}, \Delta\dot{\tilde{q}}) = \tilde{M}_p - \tilde{M}_0$ with the same kick magnitude on the top of perturbed hand trajectories in the force field caused by perturbations.

A


Difference of Control force change along X
Time [sec]
Difference of Control force change along Y
Time [sec]
Y [m]
X [m]
5 N

B


Difference of Control force change along X
Time [sec]
Difference of Control force change along Y
Time [sec]


Y [m]
X [m]
5 N

C

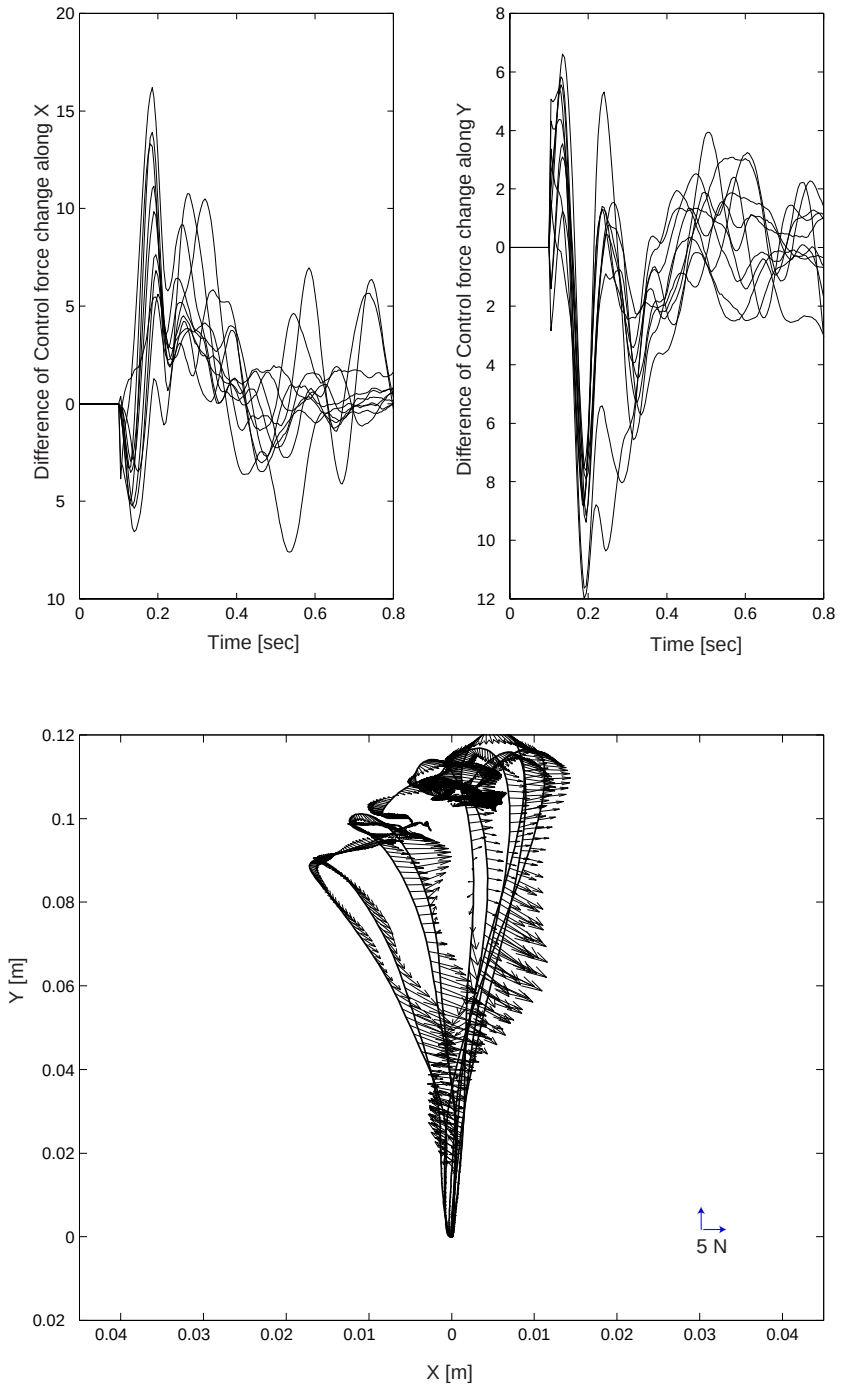


Figure 4.12: Difference between the control force changes after learning novel dynamic force field for three subjects A, B and C. The force difference is plotted by arrows on top of the perturbed trajectory $\tilde{q}, \dot{\tilde{q}}$ in the force field.

where $\tilde{M}_p$ and $\tilde{M}_0$ are control force during the perturbed and unperturbed movements in the force field respectively. By adapting K and B to $\tilde{K}$ and $\tilde{B}$, control force in the force field are adapted to predict the new dynamic environment. Using the parameterized model build for the null field, we estimate the control force change $\Delta M_{q_d}(\Delta\tilde{q}, \Delta\dot{\tilde{q}})$ for the trajectory $\tilde{q}, \dot{\tilde{q}}$ where $\tilde{q} = q_d + \Delta\tilde{q}, \dot{\tilde{q}} = \dot{q}_d + \Delta\dot{\tilde{q}}$ are perturbed trajectories in the force field. The difference between $\Delta\tilde{M}_{q_d}(\Delta\tilde{q}, \Delta\dot{\tilde{q}})$ and $\Delta M_{q_d}(\Delta\tilde{q}, \Delta\dot{\tilde{q}})$ is represented as $\delta M_{q_d}(\Delta\tilde{q}, \Delta\dot{\tilde{q}})$,

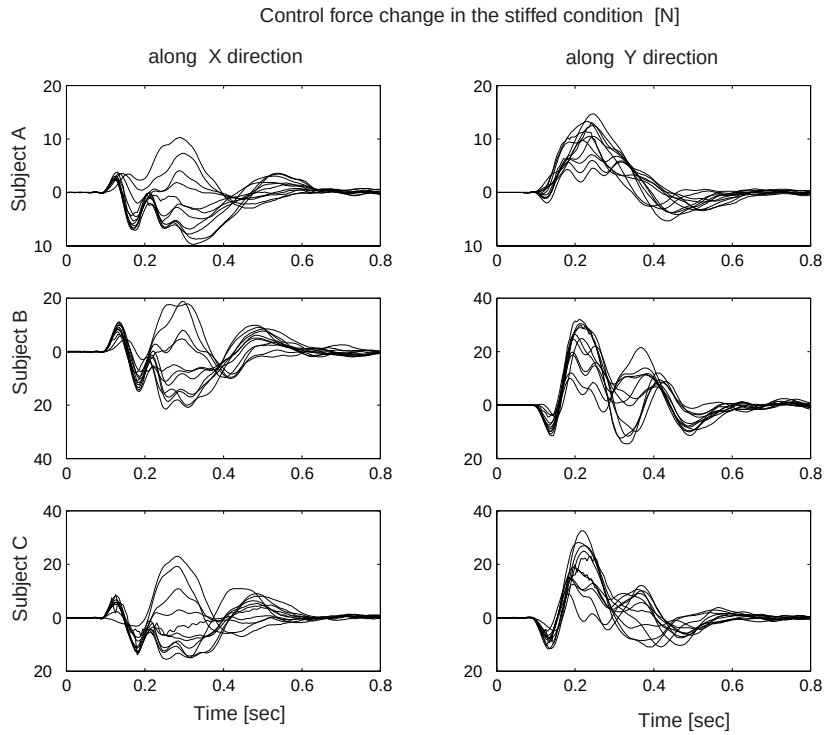
$$\delta M_{q_d}(\Delta\tilde{q}, \Delta\dot{\tilde{q}}) = \Delta\tilde{M}_{q_d}(\Delta\tilde{q}, \Delta\dot{\tilde{q}}) - \Delta M_{q_d}(\Delta\tilde{q}, \Delta\dot{\tilde{q}})$$

Figure 4.7 shows $\delta M_{q_d}(\Delta\tilde{q}, \Delta\dot{\tilde{q}})$ on top of the perturbed force field trajectories. It is easy to see from the force field equation that the effect of the force field is to push your arm to the left when arm is moving upward towards the target. Though, the initial responses to the unanticipated force field were driven-off-course from the straight line to the left. After practice and learning, subject's hand trajectories became straight and quite similar to those observed in the null field (straight line). This adaptive behavior shows that subject adjust internal models of motor controller to predict and compensate the external force presented in the forces field. One can see from the plotting $\delta M_{q_d}(\Delta\tilde{q}, \Delta\dot{\tilde{q}})$ that forward model of human motor controller also adapted to the force fields. $\delta M_{q_d}(\Delta\tilde{q}, \Delta\dot{\tilde{q}})$ represents the force generated by the forward model feedback controller, which converge the perturbed trajectory back to the desired trajectory. After practice in the force field, forward model knows there is force field always pushing the arm to the left if moving upwards. So when movement is disturbed, forward model will always expect the pushing left force and generate neural command not only push movements back to the straight desired trajectory but also push the arm to the right harder to compensate the expected forces. That is what we see from our results. The differences of control force change $\delta M_{q_d}(\Delta\tilde{q}, \Delta\dot{\tilde{q}})$ for our 3 subjects are pointing to the right due to the adaptation of the force field, which supports the theory that learning and adaptation of the new dynamic

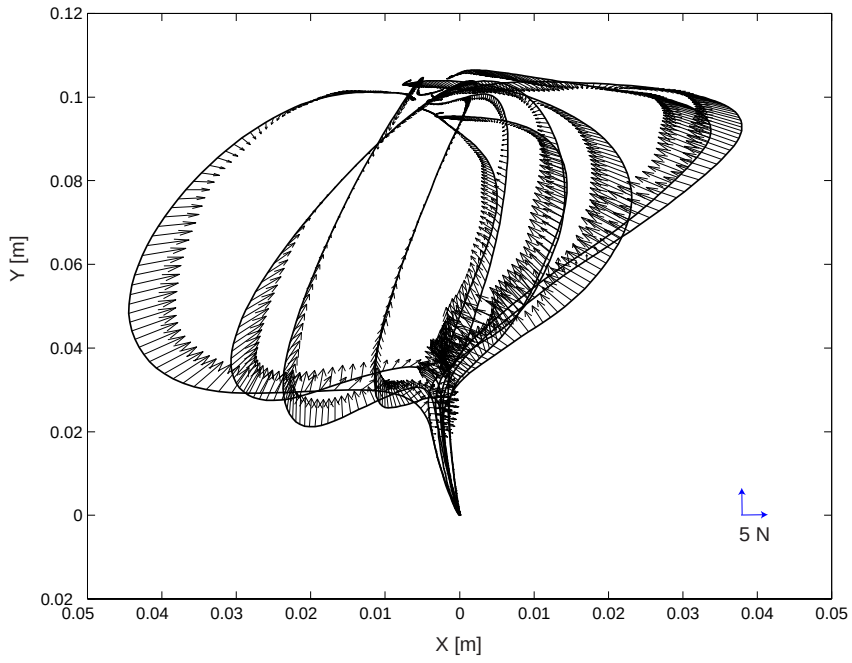
environment are also performed through the forward model of human motor controller. Results suggest that the adaptation of the forward model played a dominant role in the motor learning of subjects.

Here, we approached the system architecture of the human adaptive motor controller. The task that we considered was reaching movements in novel force field. The feature of movements we examined here are control force change, which will be only effected if the forward model has adapted to a new dynamic environment. We demonstrate that the obvious adaptation of control force change is used to compensate for the novel force field, as shown in Figure 4.7. This constant compensation of force fields could not be accounted for if the supra-spinal controller was only an open loop system composed of an inverse model and learning is only via formation of an inverse model. These characteristics point to support the architecture where descending commands were influenced by an adaptive forward model in conjunction with an inverse model. The accurate prediction of force fields provides the evidence for an adaptive forward model in the control of human arm movements in novel dynamic environments.

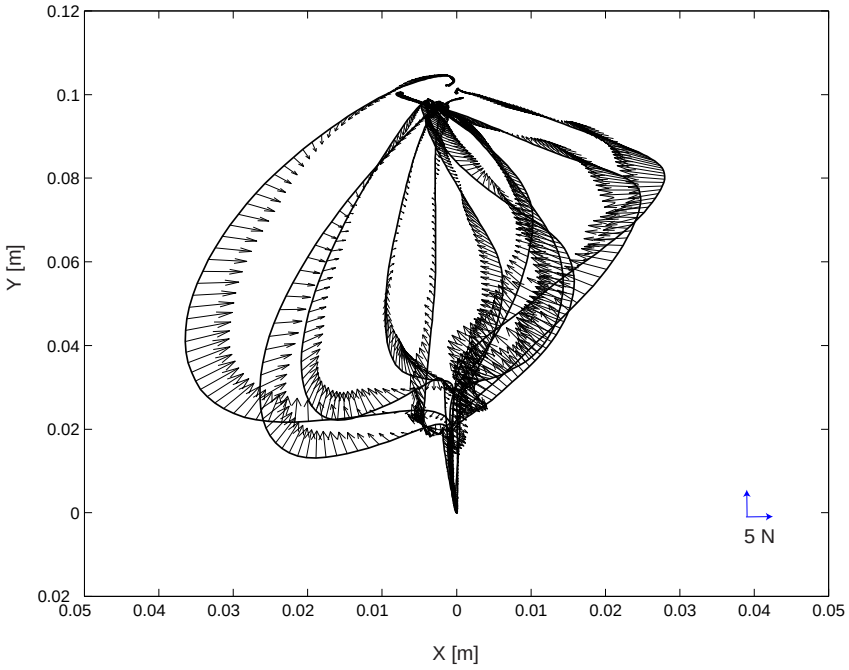
We next do a control experiment to testify that the control force change does have relationship the stiffness or viscosity of the human arm. Subjects performed the same movements in the null field with 23 equally spaced kicking directions. We instruct the subject to try her/his best to stiff her/his arm but still reach the target with the same time as before (500 msec). We measure the control force change in this situation. For these stiffed trajectories, we use the model built before to estimate what the control force change would be in the relaxed condition (All the movements we measured before this control experiment are performed in a natural and relaxed condition).


Control force change in the stiffed condition [N]
along X direction
along Y direction
Subject A
Subject B
Subject C
Time [sec]
Time [sec]

A


Y [m]
X [m]
5 N

B



C

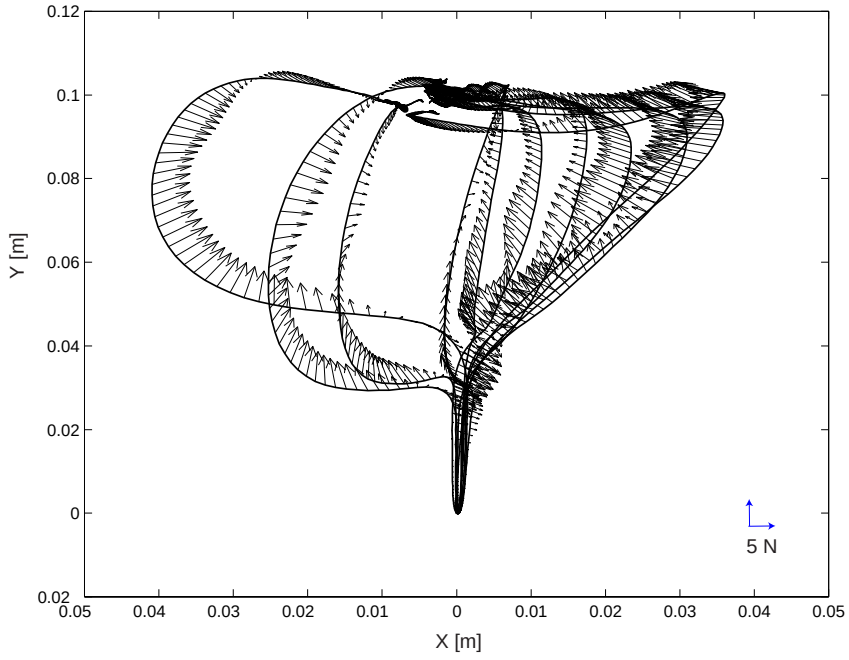
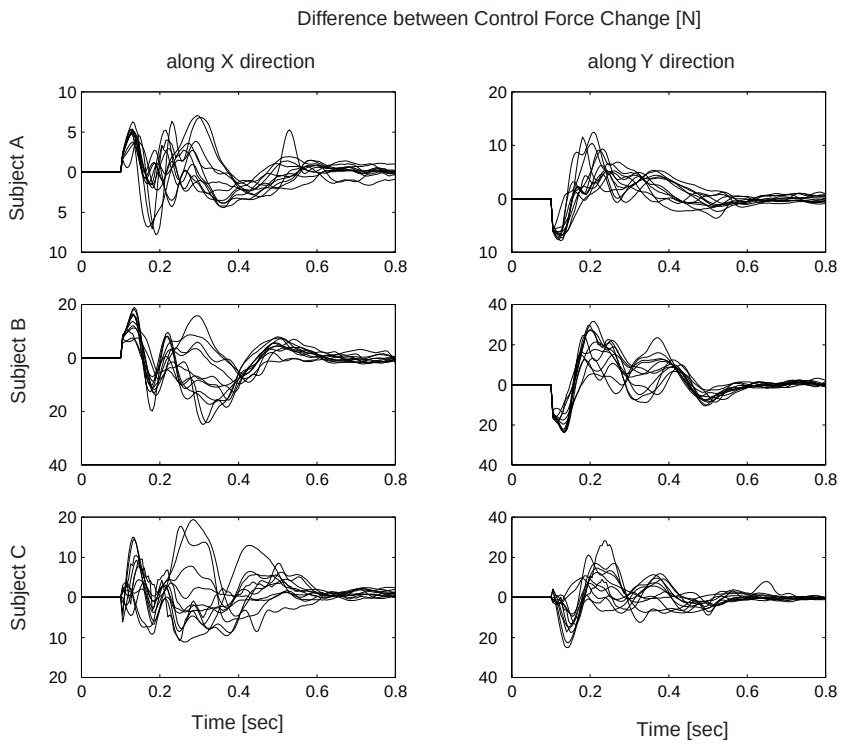
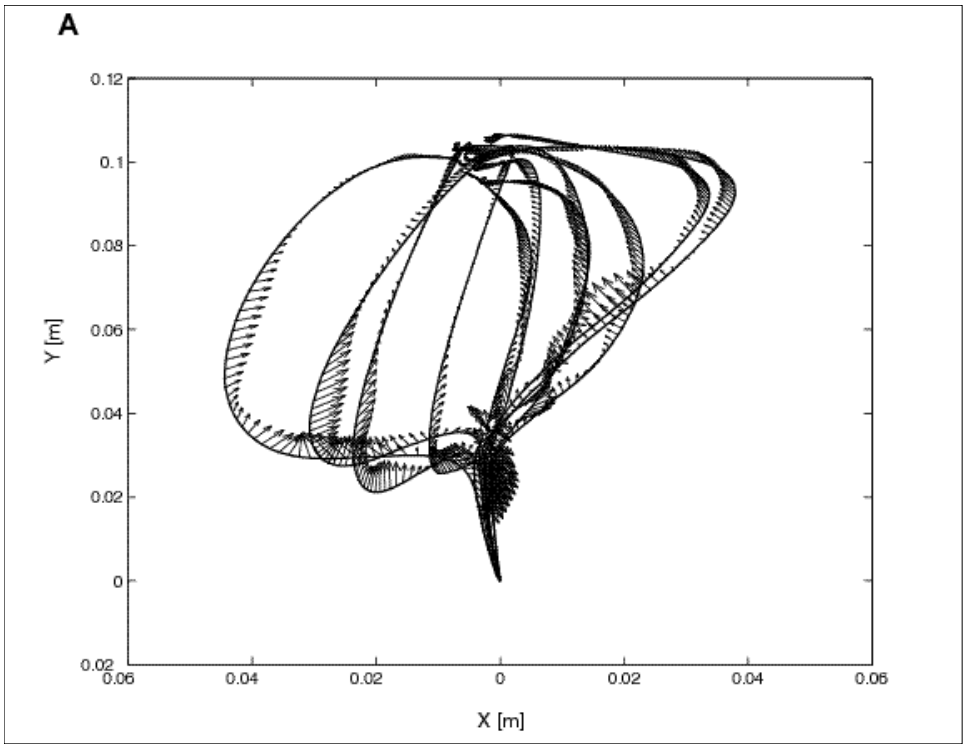
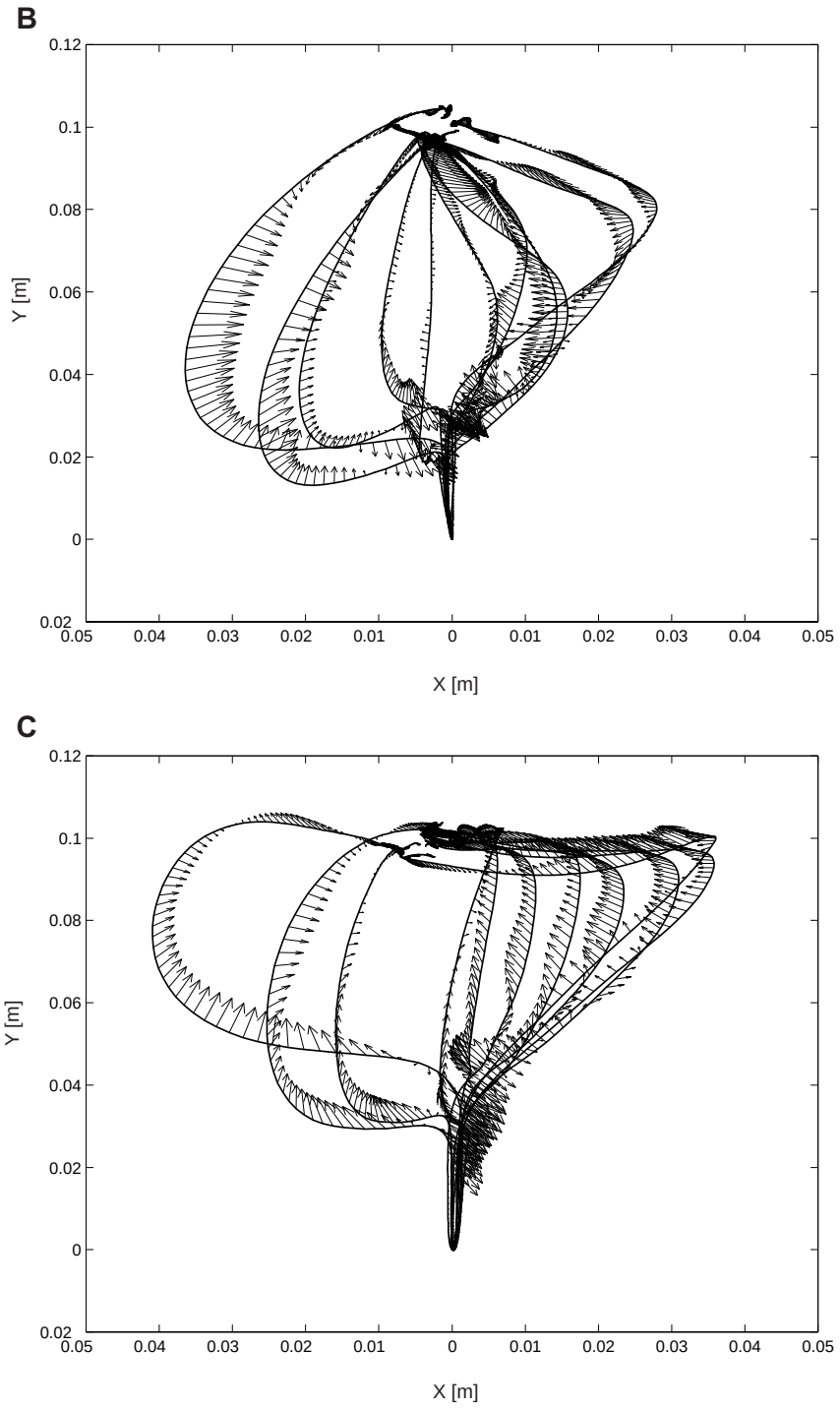


Figure 4.13: Measured control force with the same kick magnitude on the top of perturbed hand trajectories change in the stiffed situation caused by perturbations.


Difference between Control Force Change [N]
along X direction
along Y direction
Subject A
Subject B
Subject C
Time [sec]
Time [sec]


A
Y [m]
X [m]



Force scale in this figure is same as figure 4.10.

Figure 4.14: Difference between the control force changes between the stiffed movements and relaxed movements. The force difference is plotted by arrows on top of the perturbed trajectory in the stiffed situation.

Figure 4.11 shows that the change of control force change when subjects stiff their arms during the movements. One can see the change of control force change is pointing

towards the target. This means that increasing stiffness and viscosity while performing reaching movements would give more control force change toward the target. This support the idea that control force change has strong relations with feedback control strategy of human motor control, which is represented by stiffness and viscosity of the human arm.

Chapter 5

Discussion and Conclusion

In this thesis, learning to make point-to-point reaching movements in a curl force field was used as a paradigm to explore the system architecture of the human motor adaptive controller. The concept of internal model, a system for predicting behavior of the controlled movements, is divided into a forward and an inverse model. Simulation results show that the forward-inverse model feedback control seems to provide a comprehensive framework for study of computational process in the brain and adaptive human motor control. By estimating the current state on the basis of delayed information and the history neural commands through the forward model, this method provides a stable feedback control strategy for a time delayed nonlinear control system. The ability of the internal model's adaptation to the new force field provides an adaptive controller capable of learning novel dynamic environment. Simulation results provide a strong support for the existence of internal model in the brain by comparing simulated kinematic information and real human behavior. Now, in this thesis, an insight of the existence of the internal models is gained through measuring the control force, which is generated by the neural control signal to active the movement of human arm.

The existence and learning ability of the inverse model in the brain is more straightforward than forward model. With practice, the hand trajectories in the force field, though grossly distorted during the initial movements, converged to a path very similar to that observed in the null field. We assume the kinematic design, the desired movement, is independent of dynamical conditions. When subjects perform constant in the force field,

it means the desired trajectory is nearly same as the resulted trajectory and the error feedback control term is negligible in this condition. This suggested that the adaptation of the inverse model plays a dominant role in this condition. When we see the control force for unperturbed movement in the force field, a lot of control force was generated to compensate for the force field. The recovery of performance within changed mechanical environment supports the adaptation ability of the inverse model. I try to ask whether forward mode does play a role in the human control system and what happened to the forward model during the adaptation of the controller?

During trained unperturbed movements of human arm, it is hard to see the effect of forward model. Perturbing will caused more error during the otherwise stable movements. This technique gives us a toll to investigate the role of forward model in the control and adaptation process. A forward model calculates the forward process of the plant. A forward model transforms an efferent copy of descending commands into a prediction of the current state, position and velocity of the arm. Then the estimated state is compared with the desired state and the error feedback is used to modulate the demanding neural signal. Since our work has focused on an forward model of dynamics as a role of feedback control, we subtract the control force during the unperturbed movement from the control force during the perturbed movement and name it control force change (Note: The unperturbed movement we name here is actually the prediction for each perturbed movement if the movement was not perturbed). From the control architecture we provided in this thesis, the control force change is only due to computation result of forward model and spinal reflex feedback. In order to compare the control force change in the null field and in the force field in the same state (position and velocity), a parameterized model was built in the null field to compute what the control force change would be in the state of force field. We subtract the control force change in the null field from the control force change in the force field (the two control force are in the same state) to see how the control force change with respect the new dynamic environment. Results show that control force change is increased in the opposite direction of force field. This supports the idea that learning of the novel dynamic environment is also via the forward model of human motor controller. With training, the forward model

can exactly predict the force field in the mechanical environment and adapt to give more control force to compensate the force field. Control experiment is also done to support that control force change has a strong relation with forward model feedback control. Results suggested that the adaptation of the forward also played a dominant role in the motor learning of subjects. Further studies on the behavior of humans are necessary to determine the role of internal models on the full repertoire of human motor control strategy and learning of new task.

References

Blakemore Sarah J., Goodbody Susan J., and Wolpert Daniel M. (1998). Predicting the consequence of our own actions: the role of sesorimotor context esitmation

Bhushan N, Shadmehr R (1999). Computational nature of human adaptive control during learning of reaching movements in force fields. Biological Cybernetics, 81:39-60.

Bhushan N, Shadmehr R (1999). Evidence for a forward dynamics model in human adaptive motor control. Adv Neural Inform Proc Systems, 11:3-9.

Dordevic G., Rasic M., Kostic D., and Potkonjak V., Motion control skills in robotics, Submitted for publication in IEEE Transactions on System, Man and Cybernetics, Part C: Applications and Reviews (Accepted for publication for no. 2. 2000).

Dordevic G., Rasic M., Kostic D., and D. Surdilovic (1999). Learning of Inverse Kinematics Behavior of Redundant Robots, IEEE Iint. Conf. Robotics and Automation, ICRA99, Detroit, , pp. 3165-3170.

Flanagan J. Randall and Wing Alan M. (1997). The role of internal models in motion planning and control: evidence from grip force adjustments during movements of hand-held loads. J Neuroscience, February 15, 1997, 17(4): 1519-1528.

Flanagan J. Randall, Nakano Eri, Imanizu Hiroshi, Osu Rieko, Yoshioka Toshinori, and Kawato Mitsuo (1999). Composition and decomposition of internal models in motor learning under altered kinematic and dynamic environments

Jordan I. Michael and Wolpert M. Daniel (1999). Computational motor control. The cognitive neuroscience, MIT Press.

Gomi Hiroaki and Kawato Mitsuo (1997). Human arm stiffness and equilibrium-point trajectory during multi-joint movement. Biological Cybernetics, 76: 163-171.

Thoroughman KA, Shadmehr R (1999). Electromyographic correlates of learning an internal model of reaching movements. Journal of Neuroscience, 19:8573-8588.

Lacquaniti F., Carrozzo M., and Borghese N. A. (1993). Time-varying mechanical behavior of multijointed arm in man. J Neurophysiology, Vol. 69, No 5.

Mussa-Ivaldi F. A., Hogan N. and Bizzi E. (1985). Neural, mechanical, and geometric factors subserving arm posture in humans. J Neuroscience, Vol. 5, No 10, pp. 2732-2743.

Shadmehr R. and Mussa-Ivaldi F. A. (1994). Adaptive representation of dynamics during learning of a motor task. J Neurosci, 14(5):3208-3224.

Stroeve Sybert (1999). Impedance characteristics of a neuromusculoskeletal model of the human arm I. Posture control. Biological Cybernetics, 81:475-494.

Stroeve Sybert (1999). Impedance characteristics of a neuromusculoskeletal model of the human arm II. Movement control. Biological Cybernetics, 81:495-504.

Walpert Daniel M. (1994). Computational Approaches to motor control. Trends in Cognitive Sciences (1997) 1:6, 209-216.