Fall 2010

Model predictive control for adaptive digital human modeling

Katha Janak Sheth
University of Iowa

MODEL PREDICTIVE CONTROL FOR ADAPTIVE DIGITAL HUMAN MODELING

by

Katha Janak Sheth

A thesis submitted in partial fulfillment
of the requirements for the Master of
Science degree in Biomedical Engineering
in the Graduate College of
The University of Iowa

December 2010

Thesis Supervisors: Professor Soura Dasgupta
Assistant Research Engineer Rajankumar Bhatt

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

_____

MASTER'S THESIS

_____

This is to certify that the Master's thesis of

Katha Janak Sheth

has been approved by the Examining Committee for the thesis requirement for the Master of Science degree in Biomedical Engineering at the December 2010 graduation.

Thesis Committee: _____
Soura Dasgupta, Thesis Supervisor

_____
Rajankumar Bhatt, Thesis Supervisor

_____
Karim Abdel-Malek

_____
Nicole M. Grosland

_____
Jasbir Arora

I would like to dedicate my thesis to Nature, Mom - Dad, Ultimate Frisbee, Dance, Music, every single being who have helped mold my life and everything that defines freedom and creativity.

My Little ones,
you are the hope
you are the future
Keep always this youth
which is the faculty to progress
for you the phrase
"*It is impossible*"
will have no meaning.

The Mother, Seeds of Light

ACKNOWLEDGMENTS

I am heartily thankful to my academic advisor, Karim Abdel-Malek, who gave me an opportunity to be a part of a wonderful team at Virtual Soldier Research (VSR), the University Of Iowa. Virtual Soldier Research is the place that bears so much importance in my life. The learning, the fun, the understanding, everything begins from VSR. I owe my deepest gratitude to Dr. Soura Dasgupta and Rajankumar Bhatt who have supervised me throughout my research, helped me understand the concepts behind the research and marked my way towards my graduation.

The entire VSR crew, who have helped me in making the implementation of this research possible.

I would also like to show my gratitude to Dharmsinh Desai University, through which I was able to explore this opportunity at The University of Iowa under the B.E – M.S joint program. A special thanks to Dr. V.C.Patel, Dr. H.M.Desai and Dr. P. Barry Butler who made this joint program possible and also Dr. Edwin Dove, who in my initial days at UIowa, advised me academically, as well as supported me . I am heartily grateful to Prof. Saurin Shah and the entire faculty of Instrumentation and Control Engineering Department at Dharmsinh Desai University, who supported me in every possible for me to make the most out of this opportunity.

A super special thanks to Jean-Francois Veaux-Logeat and Victor Ramos, Domain leaders at Delmia, Dassault Systemes who not only made things flexible for me during the course of my internship at Delmia to work towards my thesis but supported me to the fullest amidst all my work saying “Take the time you need for your thesis and come back to us with clear mind and fresh ideas”.

My deepest thanks to Deborah Hampton and Jenny Simpson program assistants at CCAD as well as Lorena Lovetinsky and April Erickson secretaries at Biomedical

Engineering department, the University of Iowa who would help us with every possible thing in which they could

Lastly, a special thanks with love to my parents, sister, cousins, all my friends and family members who kept on saying, “Keep going, you will make it”.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Digital modeling and simulation techniques have brought about a major change in the product development cycle for all industries including engineering, defense, medicine entertainment, ergonomics and education. Real time humans with applications in the above mentioned industries show different requirements along the dimensions of appearance, function, simulation time (off-line and on-line), autonomy (autonomously interacting, reacting and making decisions) and individuality (individual personality and character) (Badler, et al. 1999).

Table 1. Requirements of representative virtual human applications.

Application	Appearance	Function	Time	Autonomy	Individuality
Cartoons	high	low	high	low	high
Games	high	low	low	medium	medium
Special Effects	high	low	high	low	medium
Medicine	high	high	medium	medium	medium
Ergonomics	medium	high	medium	medium	low
Education	medium	low	low	medium	medium
Tutoring	medium	low	medium	high	low
Military	medium	medium	low	medium	low

Table 1 Requirements for different virtual human applications

Source: (Badler, et al. 1999)

The table in Table 1 shows how virtual human application designs would differ depending upon the requirements.

For most of the applications in Table 1, digital human modeling and simulation tool serves as an evaluation tool, minimizing the cost as well as the need for physical

prototyping giving a better design or a safer product or giving intricate details thus reducing the time required to develop the products.

Considering the increased momentum of the above technological enhancement requirements in digital human industry, the day is not far when the virtual avatars would be able to interact with the material world, extending the horizons of virtual susceptibility. Its converse where the material world (including humans) would be able to interact with the virtual world has been made possible to some extent through innovations like sixth sense technology (Mistry and Maes 2009), Ada and Grace (Museum of Science, Boston), Nintendo's Wii, Microsoft's MILO and video games like Kinectimals for Xbox 360.

However, even with the advent of simulation methods, there are considerable challenges in predicting and simulating human motion and human behavior to counter actions that may resist the intent of the motion. Different methods have been tried, combined and improvised toward this end. All these methods face a considerable challenge in dynamically responding to environmental changes. The key contribution of this thesis is to use a *Model Predictive Control (MPC)* based approach to formulate strategies that are more responsive to such environmental changes as the sudden application of an inhibitory, unanticipated force when a digital human is in the middle of performing some task. Thus we wish to develop a dynamic virtual human response mechanism that simulates how a human should react to on-line unexpected environmental changes/disturbances. Once we have this capability, we can add new functionalities in making the motion as realistic as possible and having it adapt to the changes on its own. The different techniques/approaches used could be classified as data-based, non-databased, hybrid, animation and artificial intelligence. We will briefly point out the challenges faced by different techniques.

1.1 Background

The data-based approach uses *pre-existing human motion/posture data* in order to predict motions for newly given input simulation scenarios. The pre-existing motion data are typically recordings of real human performances (Park 2009). Motion Capture (MoCap) systems records-processes human motion data and translates it to virtual avatars in real time using inverse kinematics or other techniques. They are expensive with the additional requirement of a well maintained environment to capture human motion using sensors. In order to capture a motion, time intensive experiments must be conducted and large amount of data must be collected, processed and stored. The science fiction epic film Avatar, required over a petabyte of memory which equals a 32 year long MP3 with 40000 processors (Masters 2009) for creating the motion of the characters and the virtual world. In addition to these large storage, time and cost requirements, data driven approaches are generally specific to anthropometries and have difficulties with replaying the same motion for varying anthropometries and body types (i.e. have problems with motion adaptability and scalability). Additionally, some postures and motions cannot be captured by current MoCap techniques, requiring additional artificial constraints and estimation for motion reconstruction (Ausejo and Wang 2009). As dynamics (external and internal forces) play a major role in human motion, MoCap techniques fail partially in predicting non-static postures (Abdel-Malek and Arora 2009). Additional output in terms of actuation torque requirements and spine shear and compression forces are also necessary in order to employ the digital avatar for ergonomics to answer questions like whether a 50^{th} percentile female can perform a task or not. As a result, MoCap is not used as a sole technique in predicting human motion but, in many cases, is combined with physics based optimization by using the kinematic data available from MoCap (Zhang

and Chaffin 1999) or for validating the output of a predictive methodology (Rahmatalla, et al. 2008).

Lagranian formulation for manipulator dynamics dates back to the eighties. (Hollerbach 1980) derived efficiency from recurrence relations for the velocities, acceleration and the generalized forces. The number of additions and multiplications vary linearly with the number of joints, as opposed to previous Lagrangian dynamics formulation that have n^4 dependencies. It was concluded that recursive formultion based either on the Lagrangian or Newton-Euler dynamics offers the best method of dynamics calculation.

(Xiang, et al. 2007) derived sensitivity equations for the problem of optimization-based motion prediction of a mechanical system using the inverse recursive Lagrangian formulation. The simulation and sensitivity formulations were based on Denavit–Hartenberg transformation matrices. External forces and moments are taken into account in the formulation. The sensitivity information is needed in the optimization based simulation process. The formulation is demonstrated by calculating sensitivities for the optimal time trajectory planning problem of a two-link manipulator. In addition, sensitivities obtained using the proposed algorithm are compared to those obtained using the closed-form equations of motion. The two sensitivities match quite closely. The lifting motion of the two-link manipulator with external loads is also optimized by using the algorithm developed in this article.

(Marler, et al. 2009) presents a study that advances posture prediction with a multi-objective optimization (MOO) approach. With sufficient fidelity, the use of virtual humans can save time, money, and lives through improved product design, process design, and understanding of behaviour. Optimization-based posture prediction is a unique tool. MOO is used to both develop and combine the following human

performance measures: joint displacement, musculoskeletal discomfort, and a variation on potential energy. The following MOO methods are studied in the context of human modelling: objective sum, min–max, and global criterion. Using MOO yields realistic results. Of the independent performance measures, discomfort generally provides the most accurate postures. Potential energy, however, is not a significant factor in governing human posture and should be combined with other performance measures. The three MOO methods for combining performance measures yield similar results, but the objective sum provides slightly more realistic postures.

The non-databased approach is mostly physics based since it does not depend upon pre-defined/pre-generated motion. One of the successful physics based optimization techniques is the Predictive Dynamics approach (Xiang, et al. 2009b). Predictive Dynamics approach minimizes an objective function which is a function of the joint angle profiles over the entire simulation time fulfilling all the constraints. Another approach which involves musculoskeletal modeling calculates optimum values of muscle forces by minimizing the metabolic energy expenditure (Anderson and Pandy 2001). In complex models such as those of humans, solving a single optimization problem over the entire motion forms a highly non-convex minimization problem. This issue has been addressed by a controls based approach where motion is predicted/ planned over a small future interval using linearized dynamics model to track the desired motion and re-planned at regular intervals, incorporating the disturbance as well as the changes in the system states (Silva, et al. 2008).

1.2 Approach

It is this concept of receding horizon control that forms the basis for MPC. As applied to this setting, MPC proceeds as follows. It uses a three link planar model to

represent a human upper limb, in particular a 3 degrees of freedom (dof) serial-link structure with 3 revolute joints that mimics the planar upper limb model of the virtual avatar/digital human. The goal is to move the upper limb to a predetermined target position optimally. To this end it uses a set of differential equations to model the three link system, and a cost function that balances the accuracy with which the objective is met, and indirectly the time needed to meet it, against the amount of torque generated. In principle, at a given discrete time instant, it generates a torque that minimizes the cost function over several time steps, subject to the constraints imposed by the differential equation model. Since the differential equations are nonlinear, we choose the time steps to be small, and approximate the differential equation model by a set of differential equations that are more amenable to online computations. The resulting system adjusts to unanticipated disturbances in a natural way by generating torques that attempt to counter the disturbances as and when they arise. The assumption is that there is a way in which human beings can sense, and estimate disturbances as soon as they are applied based upon expected and unexpected disturbances. This approach thus helps the avatar to stabilize or produce counter-forces while performing a task, avoiding the need to recompute the motion for the whole task with the disturbances.

Our current work in this thesis is limited to predicting and controlling the motion of a planar upper limb (Sheth, et al. 2010). As such it acts as a proof of concept for applying MPC to more complicated settings. For example, one would like to extend the algorithms developed in this work and apply it to predictive dynamics based tasks like stairs climbing (Bhatt, et al. 2008) and walking (Xiang, et al. 2009a) so as to predict the motion of these tasks in the presence of unanticipated changes in the environments affecting the digital human avatar.

According to an estimate of operation engineers (OEs) exposed to whole-body vibration by (Bureau of Labor Statistics [BLS] 2003), a majority of OEs (90%) perform excavating and paving work (e.g., operating dozers, loaders, excavators), while the remaining are crane operators (10%) who are vulnerable to fatal injuries. Simulation of virtual humans under such whole body vibrations can help understand the physiological as well as preventive measures for such injuries. We use a simple example of simulating the 3 dof upper limb model with MPC under the effect of periodic disturbances of different magnitudes as body vibrations can be crudely modeled with periodic functions. The long term goal is to extend the MPC approach formulated in this thesis, to settings designed to deal with such dynamic situations as balancing on a platform or reacting to sudden external contact with an object.

1.3 Conclusion

This thesis will demonstrate an adaptive modeling and simulation of a digital human upper limb in the presence of external environmental disturbances. The organization is as follows: Chapter 2 describes the internal kinematic as well as dynamic design plant model of the digital human upper limb. Chapter 3 explores the Model Predictive Controller for the design modeled in Chapter 2. Simulation results in Chapter 4 are followed by a conclusion in Chapter 5 with some limitations and future directions.

CHAPTER 2

PLANAR DIGITAL HUMAN UPPER LIMB MODEL

2.1 Introduction

In this chapter, we discuss the kinematic and dynamic modeling of the human upper limb structure. The upper limb of a human refers to the three body segments: i) arm that goes from shoulder joint to the elbow joint, ii) forearm that goes from elbow joint to the wrist joint, and iii) the hand that refers to the region distal to the wrist joint. While the anatomical model of the upper limb has complex joints formed by various bones like Clavicle, Scapula, Humerus, Radius, Ulna, and all other hand bones, all joints in this study are modeled as revolute joints. Since the model developed in this section will be used to develop and study the new controller, only a simplified planar model of upper limb is considered for this study. Once the new controller is designed, implemented, refined, and tested on this simple planar upper limb model, a more realistic and complex upper limb model can be developed. The kinematic relationships between different segments are described by Denavit and Hartenberg notation (Denavit and Hartenberg 1955). The equations of motion are then developed using a recursive Euler-Lagrange approach.

Figure 1 shows the system block diagram. The plant is the upper limb model of the digital human avatar - Santos^{TM} developed by the Virtual Soldier Research team at the University of Iowa. In the figure,

$n(dof) = 3$

$\boldsymbol{\tau}(t) = n \times 1$ vector of joint torques,

$\boldsymbol{\theta} = n \times 1$ vector of joint angles,

$\dot{\boldsymbol{\theta}} = n \times 1$ vector of joint velocities,

$\ddot{\boldsymbol{\theta}} = n \times 1$ vector of joint accelerations,

$\mathbf{A} = n \times n$ mass-inertia matrix,

$\mathbf{B} = n \times 1$ vector representing the coriolis and centrifugal forces,

$\mathbf{C} = n \times 1$ vector of gravity forces, and

$\mathbf{D} = n \times 1$ vector representing the external disturbances acting on the system.

Since we intend to develop a simple design, the plant is a 3 link - 3 dof planar model. The Model Predictive Controller has been described in detail in the next chapter and so we would consider it as a black box to understand the plant.

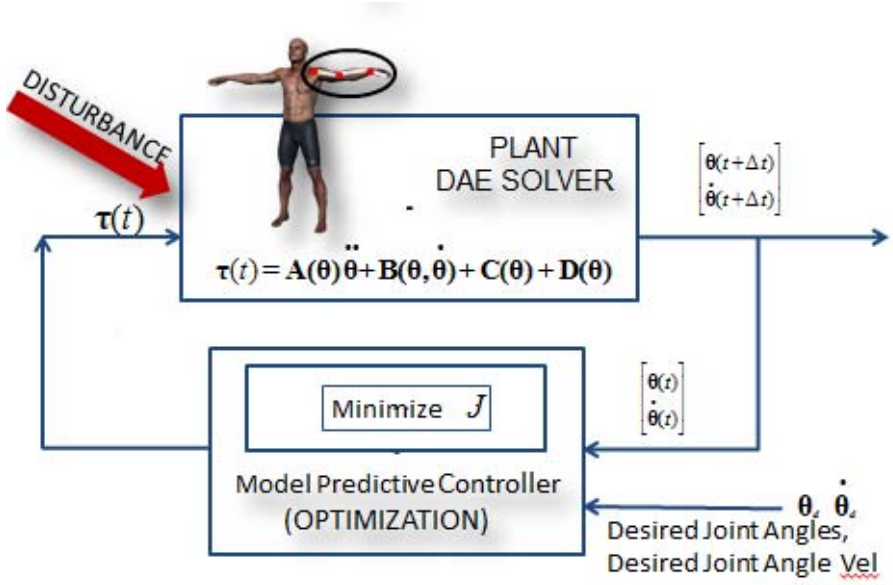

Figure 1 The digital human upper limb system block diagram

Given some controlled torque $\boldsymbol{\tau}(t)$ from the controller, the plant simulates precise movements of the joints. So, the mathematical and physics model of the plant consists of the equations of motion (EOMs) of the virtual avatar. The EOMs of a system describe the relationship between the joint angles, velocities, accelerations and the joint torques applied as a function of time. These equations are solved for joint angle positions, velocities, and accelerations using Differential Algebraic Equation (DAE) Solver from Suite of Nonlinear and Differential/Algebraic Equation Solver (Hindmarsh 2000).

2.2 DH Parameterization

The human upper limb is modeled as a 3 degrees-of-freedom (dofs) planar rigid link mechanical structure.

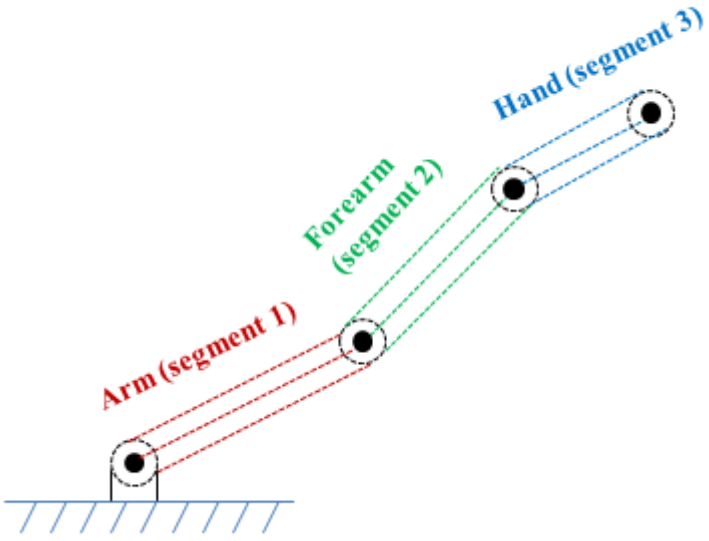

Figure 2 Planar rigid link upper limb model

Denavit and Hartenberg (DH) parameterization (Denavit and Hartenberg 1955) is used to represent the kinematic structure of the upper limb. The DH parameterization is a matrix transformation method to systematically describe the translational and rotational relationship between adjacent reference frames in an articulated chain.

In the DH representation, the relative location of the frame associated with link i and joint i with respect to previous frame is described by 4 geometric parameters θ_i , a_i , d_i , α_i . 3 of these 4 parameters remain constant for a given link, while θ_i for a revolute joint and d_i for a prismatic joint is the joint variable. Since the six dofs between two arbitrary frames must be represented by 4 parameters, a convention must be followed while defining the frames for use with DH parameterization. The ground frame (inertial frame) should be defined such that $z-axis$ is along the axis of motion of the first joint. In addition, each successive frame must follow following convention:

$z-axis$: Points along axis of motion of the next joint. If the joint is revolute, the axis of motion is the axis of rotation. If the joint is prismatic, the axis of motion is the axis of translation.

$x-axis$: Normal to previous $z-axes$ and pointing away from it.

$y-axis$: Constrained to complete right-handed coordinate frame.

For the 3 DOF human upper limb model, the three coordinate axes associated with the three segments are shown in Figure 2 and the DH parameters are defined as follows:

θ_i : joint angle from $x_{i-1}-axis$ to x_i-axis about $z_{i-1}-axis$

a_i : link length from $z_{i-1}-axis$ to z_i-axis about x_i-axis

d_i : link offset from $x_{i-1}-axis$ to x_i-axis about $z_{i-1}-axis$

α_i : link twist from $z_{i-1}-axis$ to z_i-axis about x_i-axis

These 4 parameters describe the relative location of frame i with respect to frame $i-1$.

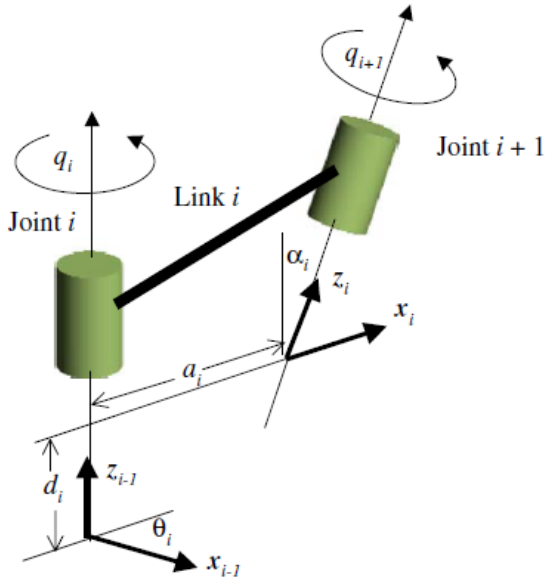

Figure 3 Joint Co-ordinates showing the DH-parameters
Source: (Xiang, et al. 2007)

$$q_i = \begin{cases} \theta_i : \text{joint } i \text{ revolute} \\ d_i : \text{joint } i \text{ prismatic} \end{cases} \qquad 2.1$$

Let ${}^{i-1}\mathbf{T}_i$ denote the homogenous transformation matrix that gives the position and orientation of frame $o_i x_i y_i z_i$ with respect to frame $o_{i-1} x_{i-1} y_{i-1} z_{i-1}$. The transformation ${}^{i-1}\mathbf{T}_i$ can be obtained by successively applying the following four transformations. First, a rotation about the $z_{i-1} - axis$ by an angle of θ_i, denoted by $\mathbf{R}_{z,\theta_i}$. Second, a translation along the new $z_{i-1} - axis$ by a distance of d_i units, denoted by $\mathbf{Trans}_{z,d_i}$. Third, a

translation along the new $x_i - axis$ by a distance of a_i units, denoted by $\mathbf{Trans}_{x,\alpha_i}$. And finally, a rotation about the $z_{i-1} - axis$ by an angle of θ_i, denoted by $\mathbf{R}_{x,\alpha_i}$.

Thus, this transformation matrix, ${}^{i-1}\mathbf{T}_i$ is expressed as a product of four basic transformations as follows:

$$
\begin{aligned}
{}^{i-1}\mathbf{T}_i &= \mathbf{T}_i(q_i) \\
&= \mathbf{R}_{z,\theta_i} \times \mathbf{Trans}_{z,d_i} \times \mathbf{Trans}_{x,\alpha_i} \times \mathbf{R}_{x,\alpha_i} \\
&= \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\qquad 2.2
$$

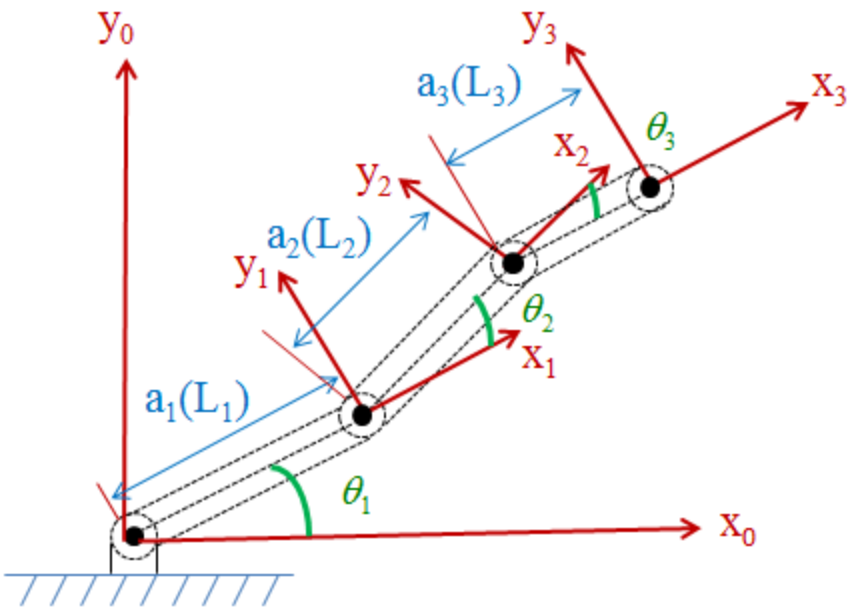

Figure 4 3-link planar upper limb manipulator model with DH

Thus the DH parameterization describes the translational and rotational relationships between adjacent links, such that through sequential transformations, the end-effector expressed in the "hand coordinates" can be transformed and expressed in the "base coordinates" which makes up the inertial frame of the dynamic system (Fu, et al. 1987). Table 2 shows the DH parameters for the upper limb planar model shown in Figure 4.

i (*joint*)	θ_i (*joint angle*)	d_i (*link offset*)	a_i (*link length*)	α_i (*link twist*)
1	θ_1	0	L_1	0
2	θ_2	0	L_2	0
3	θ_3	0	L_3	0

Table 2 DH parameters for upper limb planar model

2.3 Kinematic Model for Upper Limb

Length (m):	$L_1 = 0.29827$ $L_2 = 0.247374$ $L_3 = 0.1717$
Mass (kg):	$m_1 = 2.8$ $m_2 = 0.6$ $m_3 = 0.1$
Inertia (kg-m^2):	$I_1 = 0.233$ $I_2 = 0.00305$ $I_3 = 0.0002$
Center of Mass of Link 1:	${}^1r_1 = \left(-\frac{L_1}{2}, 0\right)$ $= (-l_1, 0)$
Center of Mass of Link 2:	${}^2r_2 = \left(-\frac{L_2}{2}, 0\right)$ $= (-l_2, \quad 0)$

Table 3 Parameters for a 3 link manipulator

The DH parameterization technique is used to represent human kinematics of the upper limb. The upper limb model under consideration for this work is planar; hence each joint is a single dof. This makes current model very well suited to DH notation. Joints with more than one dof can be represend by more than one super imposed DH joint when a spatial model of the upper limb is considered. The kinematic and dynamic relationships have been developed using the Recursive Lagrangian equations (Xiang, et al. 2007) instead of Regular Lagrangian (Goussous, et al. 2009) since the recursive method makes it possible to compute Lagrangian dynamics for larger systems in relatively less time (Hollerbach 1980). Table 3 shows the values of various kinematic and dynamic parameters used to model the human upper limb. These values are based on a 50^{th} percentile human as obtained from GEBOD software (Cheng, et al. 1996).

A detailed recursive Lagrangian formulation for a 2-link manipulator can be found in (Xiang, et al. 2007). As a first step towards development of the recursive Lagrangian dynamic equations of motion, forward recursive kinematics must be calculated to obtain the position, velocity and acceleration matrices.

The transformation matrices for each of the links can be obtained by substituting the DH parameters from Table 2 in to 2.2.

The transformation matrix $\mathbf{T}_1$ that relates the frame associated with the arm to the global frame can be derived as:

$$\mathbf{T}_1 = \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & L_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & L_1\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad 2.3$$

The transformation matrix $\mathbf{T}_2$ that relates the frame associated with the forearm to the previous frame can be derived as:

$$\mathbf{T}_2 = \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & L_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & L_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad 2.4$$

Similarly, the transformation matrix $\mathbf{T}_3$ that relates the frame associated with the hand to the previous frame can be derived as:

$$\mathbf{T}_3 = \begin{pmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & L_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad 2.5$$

The position matrix A_i for each frame can be obtained by taking the product of the position matrix of the previous frame, $\mathrm{A}_{i\text{-}1}$, with the transformation matrix of current frame, T_i. The position matrix of the ground frame, A_0 is an identity matrix.
Thus, the three position matrices, A_1, A_2, and A_3 can be calculated as:

$$\mathbf{A}_1 = \mathbf{T}_1$$

$$\mathbf{A}_2 = \mathbf{A}_1\mathbf{T}_2$$

$$\mathbf{A}_3 = \mathbf{A}_2\mathbf{T}_3 \qquad 2.6$$

The velocity matrix B_i can be calculated by taking a time derivative of the position matrix as:

$$B_i = \dot{A}_i = B_{i-1} T_i + A_{i-1} \frac{\partial T_i}{\partial q_i} \dot{q}_i \qquad 2.7$$

Thus, the velocity matrix for the ground frame will be a zero matrix. In addition, the velocity matrices B_1, B_2, and B_3 can be calculated as follows:

$$\mathbf{B}_1 = \frac{\partial \mathbf{T}_1}{\partial \theta_1} \dot{\theta}_1$$

$$\mathbf{B}_2 = \mathbf{B}_1 \mathbf{T}_2 + \mathbf{A}_1 \frac{\partial \mathbf{T}_1}{\partial \theta_2} \dot{\theta}_2$$

$$\mathbf{B}_3 = \mathbf{B}_2 \mathbf{T}_3 + \mathbf{A}_2 \frac{\partial \mathbf{T}_3}{\partial \theta_3} \dot{\theta}_3 \qquad 2.8$$

The acceleration matrix C_j can be calculated by taking a time derivative of the velocity matrix as:

$$C_j = \dot{B}_j = \ddot{A}_j = C_{j-1} T_j + 2B_{j-1} \frac{\partial T_j}{\partial q_j} \dot{q}_j + A_{j-1} \frac{\partial^2 T_j}{\partial q_j^2} \dot{q}_j^2 + A_{j-1} \frac{\partial T_j}{\partial q_j} \ddot{q}_j \qquad 2.9$$

Thus the acceleration matrix for the ground frame will also be zero. In addition, the acceleration matrices C_1, C_2, and C_3 can be calculated as follows:

$$\mathbf{C}_1 = \frac{\partial^2 \mathbf{T}_1}{\partial \theta_1^2} \dot{\theta}_1^2 + \frac{\partial \mathbf{T}_1}{\partial \theta_1} \ddot{\theta}_1$$

$$\mathbf{C}_2 = \mathbf{C}_1 \mathbf{T}_2 + 2\mathbf{B}_1 \frac{\partial \mathbf{T}_2}{\partial \theta_2} \dot{\theta}_2 + \mathbf{A}_1 \frac{\partial^2 \mathbf{T}_2}{\partial \theta_2^2} \dot{\theta}_2^2 + \mathbf{A}_1 \frac{\partial \mathbf{T}_2}{\partial \theta_2} \ddot{\theta}_2$$

$$\mathbf{C}_3 = \mathbf{C}_2\mathbf{T}_3 + 2\mathbf{B}_2\frac{\partial \mathbf{T}_3}{\partial \theta_3}\dot{\theta}_3 + \mathbf{A}_2\frac{\partial^2 \mathbf{T}_3}{\partial \theta_3^2}\dot{\theta}_3^2 + \mathbf{A}_2\frac{\partial \mathbf{T}_3}{\partial \theta_3}\ddot{\theta}_3 \qquad 2.10$$

The position, velocity, and acceleration matrices calculated above can be used to calculate the global position, velocity, and acceleration of any local point. Let r_j be the coordinates of any local point in the j^{th} coordinate system. The position vector ${}^0\mathrm{r}_j$, velocity vector ${}^0\dot{\mathrm{r}}_j$, and the acceleration vector ${}^0\ddot{\mathrm{r}}_j$, of the same point in the global reference frame can be calculated as:

$$\begin{aligned} {}^0\mathrm{r}_j &= \mathrm{A}_j\mathrm{r}_j \\ {}^0\dot{\mathrm{r}}_j &= \mathrm{B}_j\mathrm{r}_j \\ {}^0\ddot{\mathrm{r}}_j &= \mathrm{C}_j\mathrm{r}_j \end{aligned} \qquad 2.11$$

2.4 Dynamic Model for the Upper Limb

The joint actuation torques, τ_i, can be calculated for each joint using the recursive backward dynamics approach once the recursive forward kinematics results are obtained.

$$\tau_i = tr[\frac{\partial \mathbf{A}_i}{\partial q_i}\mathbf{H}_i] - \mathbf{g}^T\frac{\partial \mathbf{A}_i}{\partial q_i}\mathbf{E}_i - \mathrm{D}_i \qquad 2.12$$

where $\mathbf{H}_i = \mathrm{I}_i\mathrm{C}_i^{\mathrm{T}} + \mathrm{T}_{i+1}\mathbf{H}_{i+1}$ is a 4×4 transformation matrix.

$\mathbf{I}_i$ is the Inertia tensor matrix given by:

$$\mathbf{I}_i = \begin{pmatrix} \frac{-I_{i_{xx}} + I_{i_{yy}} + I_{i_{zz}}}{2} & I_{i_{xy}} & I_{i_{xz}} & m_i.cg_{i_x} \\ I_{i_{xy}} & \frac{I_{i_{xx}} - I_{i_{yy}} + I_{i_{zz}}}{2} & \mathrm{I}_{i_{yz}} & m_i.cg_{i_y} \\ I_{i_{xz}} & I_{i_{yz}} & \frac{I_{i_{xx}} + I_{i_{yy}} - I_{i_{zz}}}{2} & m_i.cg_{i_z} \\ m_i.cg_x & m_i.cg_y & m_i.cg_z & m_i \end{pmatrix} \qquad 2.13$$

The inertia tensor matrix can be evaluated for each link as:

$$\mathbf{I}_i = \begin{pmatrix} I_i + m_i(l_i - L_i)^2 & 0 & 0 & m_i(l_i - L_i) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ m_i(l_i - L_i) & 0 & 0 & m_i \end{pmatrix} \qquad 2.14$$

for $i = 1,2,3$

The $\mathbf{H}_i$ matrix can then be evaluated by using the acceleration matrix calculated in the previous step and with knowledge that $\mathbf{H_{n+1}} = \mathbf{H}_4$ is a zero matrix as:

$$\begin{aligned} \mathbf{H}_3 &= \mathbf{I}_3\mathbf{C}_3^{\mathrm{T}} \\ \mathbf{H}_2 &= \mathbf{I}_2\mathbf{C}_2^{\mathrm{T}} + \mathbf{T}_3\mathbf{H}_3 \\ \mathbf{H}_1 &= \mathbf{I}_1\mathbf{C}_1^{\mathrm{T}} + \mathbf{T}_2\mathbf{H}_2 \end{aligned} \qquad 2.15$$

The first term, $tr[\frac{\partial \mathbf{A}_i}{\partial q_i}\mathbf{H}_i]$, calculates the actuation torque requirement due to the inertia and Coriolis forces.

The 4×1 transformation vector E_i can be calculated as:

$$\mathrm{E}_i = m_i {}^{i}\mathrm{r}_i + \mathrm{T}_{i+1}\mathrm{E}_{i+1} \tag{2.16}$$

Substituting the values of m_i, r_i, and T_{i+1}, and with the knowledge that E_{n+1} is a zero vector, E_i can be calculated for each link as:

$$\begin{aligned}
\mathbf{E}_3 &= m_3\begin{pmatrix}(l_3 - L_3) & 0 & 0 & 1\end{pmatrix}^T \\
\mathbf{E}_2 &= m_2\begin{pmatrix}(l_2 - L_2) & 0 & 0 & 1\end{pmatrix}^T + \mathbf{T}_3\mathbf{E}_3 \\
\mathbf{E}_1 &= m_1\begin{pmatrix}(l_1 - L_1) & 0 & 0 & 1\end{pmatrix}^T + \mathbf{T}_2\mathbf{E}_2
\end{aligned} \tag{2.17}$$

The second term, $\mathbf{g}^T \frac{\partial \mathbf{A}_i}{\partial q_i}\mathbf{E}_i$, of the torque equation determines the actuation torque requirement due to the presence of gravitational forces. The gravity vector g is defined as $\mathrm{g} = \begin{pmatrix}0 & -g & 0 & 0\end{pmatrix}^T$.

The main goal behind this research is to develop a system response close to humans when exposed to disturbance like forces acting on the surface of body. So as a first step towards meeting the goal, the system is imposed upon with constant and periodic disturbance forces with different kinds of nature.

A disturbance classification is shown below:

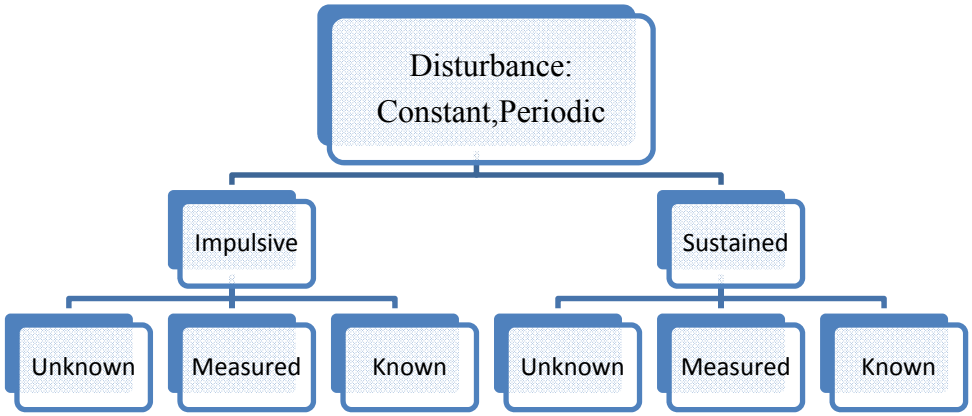

Figure 5 A disturbance classifications for the formulation

Our system can deal with different kinds of disturbance force behaviors as described in Figure 5:

i. Nature of disturbance forces:

- Constant:

  It is a constant magnitude disturbance force over a period of time.

- Periodic:

  A sinusoidal disturbance force over a period of time is periodic in nature.

ii. Duration of disturbance forces:

- Impulsive:

  A disturbance force acting for a short impulse of time is described to have an impulsive behavior.

- Sustained:

  A disturbance force acting for a large sustained amount of time is described to have a sustained behavior.

iii. Kinds of disturbance forces (The Model Predictive Control has been described in detail in Chapter 3 to give more information about predictive horizon, a term used in the description below):

- Measured:

  At the time of disturbance occurrence on the plant, with no system time lag, the disturbance information is perceived by the model predictive controller as disturbance acting in its linearized plant model. As a result, the disturbance is observed for the entire prediction horizon. This implies that it may be noticed for an extended duration in the prediction horizon, affecting the performance since the disturbance may not actually be present on the plant in the future.

  The human resemblance of *measured disturbances* can be found when he experiences unexpected disturbances which he only realizes when it occurs.

- Known:

  The controller has been fed with the disturbance information ahead of time.

  So when the predictive component is predicting motion (joint angles and joint angle velocities) over a prediction horizon, the linearized plant model of MPC takes into consideration the future disturbance which will eventually act on the plant.

  The human resemblance of *known disturbance*s can be found when he experiences expected disturbances which he realizes ahead of time and so gets the time to plan for the coming.

- Unknown:

  The system is allowed to respond on its own when the disturbance is imposed on the plant. No direct information about the disturbance is provided to the controller.

  This type of disturbance doesn't have any resemblance with the human system. But this is a good test for any controller design.

A general formulation to take into account the effect of external forces, D_i, acting on the system is discussed below:

$$\mathbf{D}_i = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \sum_{j=1}^{t} \mathbf{J}_{l_i j}{}^{\mathrm{T}} (\mathbf{-F}_{l_i j}) \qquad 2.18$$

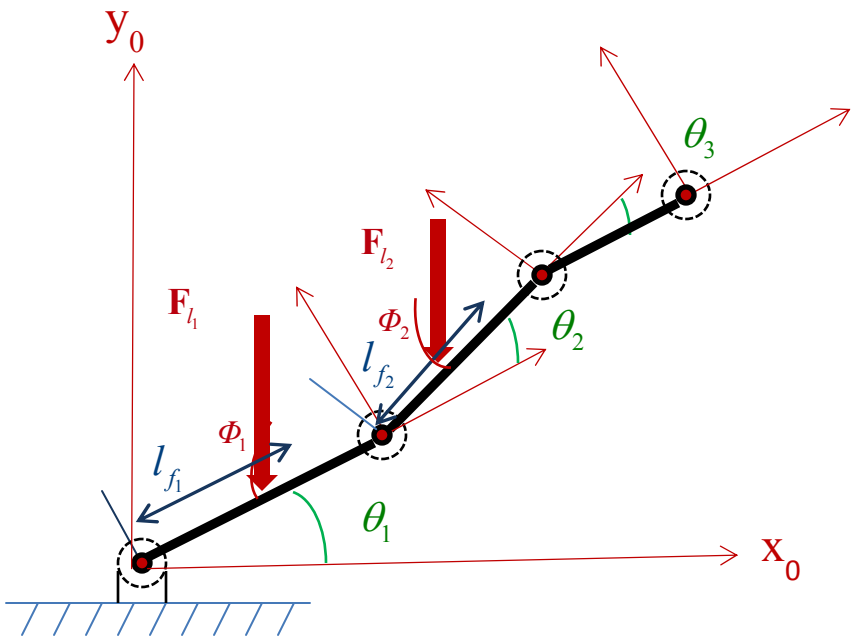

Figure 6 Parameters for disturbance formulations

where,

l_i = link on which the external force is acting,

t = total # of disturbance forces acting,

$\mathbf{F}_{l_i} = \begin{bmatrix} f_{l_i} \cos \Phi_i \\ f_{l_i} \sin \Phi_i \end{bmatrix} = 2 \times 1$ external disturbance force vector giving magnitude and direction,

$\mathbf{J}_{l_i} = 2 \times 3$ Jacobian matrix corresponding to the location of disturbance force on the surface of body,

$$= \begin{bmatrix} \frac{dx}{d\theta_1} & \frac{dx}{d\theta_2} & \frac{dx}{d\theta_3} \cdots \frac{dx}{d\theta_i} \\ \frac{dy}{d\theta_1} & \frac{dy}{d\theta_2} & \frac{dy}{d\theta_3} \cdots \frac{dy}{d\theta_i} \end{bmatrix}; \qquad 2.19$$

$i = 1, 2 \ldots \text{ndof}$

$x, y =$ location of the external disturbance on the link i ,

Note that the external forces considered in the above formulation could be forces imparted by environment or could be disturbance forces. After substituting the above equations for the 3 dof system under consideration, following equations for actuation torque requirements are obtained:

$$\tau_1 = tr[\ \frac{\partial \mathrm{A}_1}{\partial q_1}\mathbf{H}_1] - \mathrm{g}^{\mathrm{T}}\frac{\partial \mathrm{A}_1}{\partial q_1}\mathrm{E}_1 - d_1$$
$$= \left(I_1 + I_2 + m_1 l_1^2 + m_2(L_1^2 + l_2^2 + 2L_1 l_2\cos\theta_2)\right)\ddot{\theta}_1 + \left(I_2 + m_2 l_2^2 + m_2 L_1 l_2\cos\theta_2\right)\ddot{\theta}_2$$
$$- 2m_2 L_1 l_2\dot{\theta}_1\dot{\theta}_2\sin\theta_2 - m_2 L_1 l_2\dot{\theta}_2^2\sin\theta_2 + m_2 g l_2\cos(\theta_1+\theta_2) + m_1 g l_1\cos\theta_1$$
$$+ m_2 g L_1\cos\theta_1 - d_1$$

$$\tau_2 = tr[\ \frac{\partial \mathrm{A}_2}{\partial q_2}\mathbf{H}_2] - \mathrm{g}^{\mathrm{T}}\frac{\partial \mathrm{A}_2}{\partial q_2}\mathrm{E}_2 - d_2$$
$$= \left(I_2 + m_2 l_2^2\right)\ddot{\theta}_2 + \left(I_2 + m_2 l_2^2 + m_2 L_1 l_2\cos\theta_2\right)\ddot{\theta}_1 + m_2 L_1 l_2\dot{\theta}_1^2\sin\theta_2$$
$$+ m_2 g l_2\cos(\theta_1+\theta_2) - d_2$$

$$\tau_3 = tr[\ \frac{\partial \mathrm{A}_3}{\partial q_3}\mathbf{H}_3] - \mathrm{g}^{\mathrm{T}}\frac{\partial \mathrm{A}_3}{\partial q_3}\mathrm{E}_3 - d_3$$
$$= \left(I_3 + m_3 l_3^2\right)\ddot{\theta}_3 + \left(I_3 + m_3 l_3^2 + m_3 L_2 l_3\cos\theta_3\right)\ddot{\theta}_2 + \begin{pmatrix} I_3 + m_3 l_3^2 + m_3 L_2 l_3\cos\theta_3 \\ + m_3 L_1 l_3\cos(\theta_2+\theta_3)\end{pmatrix}\ddot{\theta}_1$$
$$+ \left(m_3 L_2 l_3\sin\theta_3\right)\dot{\theta}_2^2 + \left(m_3 L_2 l_3\sin\theta_3 + m_3 L_1 l_3\sin(\theta_2+\theta_3)\right)\dot{\theta}_1^2 + 2m_3 L_2 l_3\dot{\theta}_1\dot{\theta}_2\sin\theta_3$$
$$+ m_3 g l_3\cos\theta_3\cos(\theta_1+\theta_2) - m_3 g l_3\sin\theta_3\sin(\theta_1+\theta_2) - d_3$$

2.20

These EOMs are solved using the open source solver IDA obtained from Suite of Nonlinear and Differential/Algebraic Equation Solver (Hindmarsh 2000). IDA uses inexact Newton iteration methods to solve large scale non-linear system problems

modeled as differential algebraic equations. Initial-value problems are solved for a DAE system of the general form given in 2.21

$$F(t, y, \dot{y}) = 0 \tag{2.21}$$

where, the initial conditions are

$$\begin{aligned} y(t_0) &= y_0 \\ \dot{y}(t_0) &= \dot{y}_0 \end{aligned} \tag{2.22}$$

F , y , $\dot{y}$ are vectors in $\mathbf{R}^N$ and $\dot{y} = \frac{dy}{dt}$; given the initial conditions in 2.22

The integration method used is the variable co-efficient Backward Difference formula (BDF) of order q as shown in 2.23

$$\sum_{i=0}^{q} \alpha_{n,i}\, y_{n-i} = h_n \dot{y}_n \tag{2.23}$$

where y_n and $\dot{y}_n$ are the computed approximations to $y(t_n)$ and $\dot{y}(t_n)$ respectively and step size $h_n = t_n - t_{n-1}$. The coefficients $\alpha_{n,i}$ are uniquely determined by the order q .Prior to integrating the DAE initial value problem, it is necessary for y_0 and $\dot{y}_0$ to satisfy the residual as show in 2.21 with $y = y_0$ and $\dot{y} = \dot{y}_0$ at $t = t_0$

2.5 Conclusion

The Dynamics of the system is described by Regular Lagrangian equations which makes it easier while moving the system towards higher degrees of freedom. Experts would point out that an upper limb model with 3 dof is hardly any close to realism since

along with the 7 dof the clavicle, scapular movement is also to be taken into consideration. Work is in progress to move the current model with higher degrees of freedom and test it for different disturbance cases. The current framework would act as solid base for such transition

CHAPTER 3

MODEL PREDICTIVE CONTROL APPROACH

3.1 Introduction

In this chapter we provide the broad concept and the details of the Model Predictive Control (MPC) approach underlying this thesis. We begin by describing the broad concept. Consider a general system of differential equations:

$$\dot{x}(t) = f(x(t), u(t), w(t)) \tag{3.1}$$

As an example, in the setting of Figure 1, θ, $\dot{\theta}$ are the joint angle and joint angle velocity elements of x, τ represents the joint torques u, d are external disturbances represented by w, and the differential equations in the block labeled plant DAE solver, represents 3.1. Now suppose one needs to find the time function $u(t)$, that minimizes $J(x,u)$ along the solution of 3.1. In its most general form one would like to find *u(t)* that minimizes 3.2 given below along the solution of 3.1.

$$\sum_{k=0}^{\infty} J(x(k\Delta t), u(k\Delta t), w(k\Delta t)) \tag{3.2}$$

Here Δt represents a time step and *k* is an integer. In the context of our problem J would involve a balance of the objective of attaining a posture, the need to keep the torque small, and any other hard physical constraints that may be needed. One can view this as minimizing 3.2 subject to the constraints imposed by 3.1.

For a variety of reasons this is an intractable problem. MPC provides a much more tractable suboptimal solution to this problem, under the guise of Receding Horizon Control (RHC). Specifically, this approach assumes that *u* is updated only at the discrete instants of time $k\Delta t$. Further, one has access to a discrete time variant of 3.1:

$$x(t+\Delta t) = F(x(t), u(t), w(t)) \tag{3.3}$$

Then with a time horizon of length M, at each *t,* the MPC approach obtains $u(t)$, that minimizes:

$$\sum_{k=0}^{M-1} J(x(t+k\Delta t), u(t+k\Delta t), w(t+k\Delta t)) \tag{3.4}$$

Subject to 3.3, there still remains two difficulties: First even if (3.3) is available, this can still be a difficult optimization task since at every time interval $M[t, t+(M-1)\Delta t]$, u (joint torques) and w (external disturbances) keep on changing. Our simplifying assumption, is that for sufficiently small Δt and M, one assumes that u and w do not change over the interval $M[t, t+(M-1)\Delta t]$, i.e. we minimize the following:

$$\sum_{k=0}^{M-1} J(x(t+k\Delta t), u(t), w(t)) \tag{3.5}$$

Second obtaining (3.3) may itself be a daunting task since the plant model is highly non-linear. Instead we use a Taylor series approach to obtain a simplification of (3.3).

In the rest of this chapter we describe, how we obtain the approximation of (3.3), how we perform the optimization, how we select *M,* and how we deal with disturbances. We reiterate our overarching goal that the digital human simulations should be able to cope with unanticipated disturbances, introduced while the simulation is in progress.

In the sequel, we will call the generation of (3.3) and the optimization task, respectively: the predictive component and the optimization component

3.2 The Predictive Component

The predictive component of the MPC controller predicts future plant movements based on a linearized plant model approximation over a prediction horizon (a designer defined short term planning time interval). These estimated plant movements help the optimizer component to track the desired movements.

[1] Estimation of future plant movements:

For a designer defined prediction horizon, at every time step size Δt, the predictive component of the controller predicts future values of the plant movements based on a linearized approximation of the dynamic model of the plant. The actual non-linear dynamic model has been illustrated in the previous chapter.

A linear approximation is an approximation of a function/model using a linear function/model.

We have used a second order Taylor's series expansion of the form with a real function $f(x)$ differentiable at a :

$$f(x) \approx f(a) + f'(a)(x-a) + \frac{f''(a)}{2!} \times (x-a)^2 + \ldots$$

3.6

If $x = a + h$, then we obtain a Taylor's series of the form :

$$f(a+h) \approx f(a) + (h) \times f'(a) + (h)^2 \times \frac{f''(a)}{2!} + \ldots$$

3.7

With $f(a)$ as a function of joint angle $\boldsymbol{\theta}(t)$ (a column vector depicting the different dofs) and h as a time step Δt, the following Taylor series is used as an approximation for predicting the new joint angles and joint angle velocities at $x = t + \Delta t$ depending upon the joint angle, joint angle velocity and joint angle acceleration at $x = t$

$$\tilde{\boldsymbol{\theta}}(t+\Delta t) = \boldsymbol{\theta}(t) + \frac{\Delta t}{1!} \times \dot{\boldsymbol{\theta}}(t) + \frac{\Delta t \times \Delta t}{2!} \times \ddot{\boldsymbol{\theta}}(t)$$
$$\tilde{\dot{\boldsymbol{\theta}}}(t+\Delta t) = \dot{\boldsymbol{\theta}}(t) + \frac{\Delta t}{1!} \times \ddot{\boldsymbol{\theta}}(t)$$

3.8

where,

$$\ddot{\boldsymbol{\theta}}(t) = \mathbf{A}^{-1}(\boldsymbol{\theta}) \left[\boldsymbol{\tau}(t) - \mathbf{B}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) - \mathbf{C}(\boldsymbol{\theta}) - \mathbf{D}(\boldsymbol{\theta}) \right]$$

3.9

3.3 The Optimization Component

The second component that drives the MPC along with the Predictive component is the optimization component. It minimizes the objective function (the error between the desired output and the predicted output) over an optimization window of time (designer defined) optimizing the design variables (the variables controlled by the designer: joint torques τ) of the plant model at every time step Δt .

For this particular design, optimization window and prediction horizon are kept equal. The choice of the predictive horizon and optimization window (the main parameters of model predictive control) affects the performance and stability of the system. Further research can be carried out to finely tune these parameters.

The important elements of the optimization component are:

- Optimization window (Receding horizon filter)
- Design variables
- Constraints
- Objective function
- Sequential quadratic programming (optimization problem)

1) *Receding Horizon Filter (Optimization window) :*

As part of the receding horizon control, we plan optimally over small intervals of time in future using the predictive component, and re-plan at regular intervals, incorporating the changes in the system state (Silva, et al. 2008), minimizing the error between the desired motion and future predicted motion at every time step. A receding horizon has been shown in Figure 7 (Kwon and Han

2005), which gives a clear idea although in context to a simple example of investment planning. The shaded blocks are the future estimations

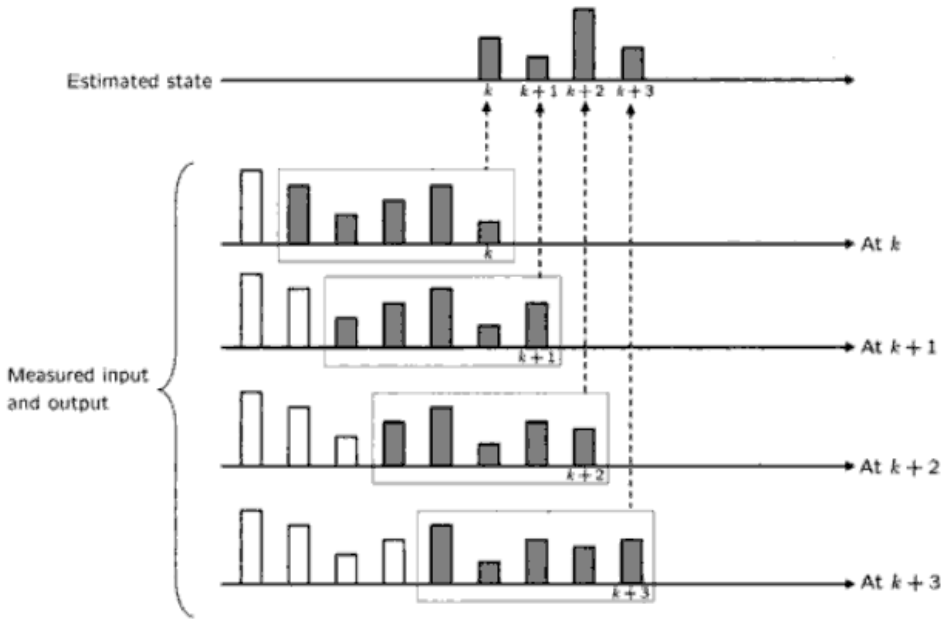

Figure 7 Receding Horizon Filter (Optimization Window)
Source: (Kwon and Han 2005)

Let's get into the details of the other components of the Optimization Component with respect to our system design:

2) *Design variables :*

The parameters that are controlled by the designer and determine the performance of a system are called the design variables. For every action to be performed by humans, joint torques are required. So torque parameter of the

motion is updated at every time step by the optimization component of the controller.

$$\boldsymbol{\tau}(t) = \begin{bmatrix} \tau_1(t) \\ \tau_2(t) \\ \tau_3(t) \end{bmatrix}$$; a column vector with 3 components corresponding to 3 dof

3) *Constraints:*

Considering the human system, the joint angles and joint torques cannot be extended beyond a certain limit. If extended can lead to pain, severe injury or even fractures

Thus there are 6 constraints at every time step.

$\theta_{low,i} \le \theta_i \le \theta_{high,i}; \tau_{low,i} \le \tau_i \le \tau_{high,i}$
where i is the dof $(=1,2,3)$

4) *Objective function:*

The optimization software program minimizes the objective function. Now this design was used to minimize the error between the desired motion and the predicted motion by minimizing the amount of energy. By motion we mean joint angles, joint angle velocity and joint angle acceleration. Also we measure the amount of energy used, by the amount of joint torques produced.

It is not desired to minimize all the parameters of motion (joint angles, joint angle velocity, joint angle acceleration and joint torques) at the same rate since the relative movement contribution is not the same. So it is important that

we penalize them differently according to a general knowledge as a proof of concept and then with further research have optimal-adaptive weights.
With the current design implementation, the desired movement is defined as a desired posture required attaining a target position to keep the design simple.

As mentioned earlier, posture prediction model gives the desired posture (desired joint angles), when given a target location.

Objective Function:

$$J_y = \sum_{y=1}^{k} \left[J_{y-1} + \sum_{i=1}^{nDOF=3} \left[W_1(\theta_{di} - \tilde{\theta}_{yi})^2 + W_2(\dot{\theta}_{di} - \tilde{\dot{\theta}}_{yi})^2 + W_3(\tau_i(t))^2 \right] \right]$$

where,

k = optimization window size = finite number of time steps Δt

i = 1, 2…ndof

W_1, W_2, W_3 = weights of the objective function

$\boldsymbol{\theta}_d$ = desired joint angle

$\tilde{\boldsymbol{\theta}}, \tilde{\dot{\boldsymbol{\theta}}}$ = second order approximation at $t = t + \Delta t$ for $y = 1 \, to \, k$

$\boldsymbol{\tau}(t)$ = joint actuation torques which are also the design variables.

Since this is a proof of concept with a simple design, ndof = 3;

Understanding the objectives:

- Joint angle error squared $(\theta_{di} - \tilde{\theta}_{yi})^2$:

  The difference between future predicted joint angles $\tilde{\theta}_{yi}$ and the desired joint angles θ_{di} is minimized.

- Joint angle velocity error squared $(\dot{\theta}_{di} - \tilde{\dot{\theta}}_{yi})^2$:

  The difference between future predicted joint angle velocities $\tilde{\dot{\theta}}_{yi}$ and the desired joint angle velocities $\dot{\theta}_{di}$ is minimized.

- Torque squared $(\tau_i(t))^2$:

  In addition, exceedingly high values of control inputs are avoided by penalizing the design variable $\boldsymbol{\tau}(t)$.

A squared value function is used for an easy numerical minimization. So minimizing a weighted $(\theta_{di} - \tilde{\theta}_{yi})^2$, $(\dot{\theta}_{di} - \dot{\tilde{\theta}}_{yi})^2$, $(\tau_i(t))^2$ at every optimization window, would yield a better performance.

5) *Sequential Quadratic Programming :*

The optimization problem:

Find optimal: design variables $\boldsymbol{\tau}(t)$
To minimize: Objective function
Subject to constraints: $\theta_{low,i} \le \theta_i \le \theta_{high,i}$; $\tau_{low,i} \le \tau_i \le \tau_{high,i}$

SNOPT software (Gill, et al. 2002) has been used to solve the optimization problem using a sequential quadratic programming algorithm. It is a gradient based approach which calculates gradients based on finite difference.

3.4 Disturbance Perturbations

Steps for MPC algorithm

1. Input to MPC at current time t : $\boldsymbol{\theta}_d$, $\dot{\boldsymbol{\theta}}_d$, $\boldsymbol{\theta(t)}$, $\dot{\boldsymbol{\theta}}\boldsymbol{(t)}$
2. Prediction of: $\tilde{\boldsymbol{\theta}}(t+\Delta t)$, $\dot{\tilde{\boldsymbol{\theta}}}(t+\Delta t)$ at next time step Δt over a prediction horizon.
3. Optimize the objective function over a time step in optimization window
4. Repeat steps 2-3 until the maximum optimization window size is reached

5. Output $\boldsymbol{\tau}(t)$

3.5 Conclusion

We talked about a linear MPC which approximates a non-linearized model of the dynamic plant (an upper limb model with 3 dofs) and predicts future movements over a receding horizon filter (an optimization window receding/moving away). The model has limited dofs since it is developed as a proof of concept. So then the first question that the design poses is, with increase in the dofs, with a more complex system, how stable or how robust the controller design would be and how natural or physically realistic the response would be. (Silva, et al. 2008) has incorporated a model predictive controller design with reference motion capture data tracking in 2D and 3D human models with 18 and 57 dofs respectively showing interactive simulations like punching, walking, jumping, marching, limping, running and many such others have been shown with MPC. Bottom line, our controller design with further research could be incorporated with physics based optimization based dynamic motions like running, stairs climbing, walking and others and extend its scope towards balancing on dynamic platforms and terrains. A Proportional-Derivative Controller could be embedded for quick actions like in (Silva, et al. 2008) to adapt to disturbances until the predictive component updates the states. The weights used in the objective function are based on an educated guess. But with further experiments and analysis, an adaptive scheme can be adopted which will greatly affect the performance while dealing with different anthropometries and scenarios.

CHAPTER 4
DIGITAL HUMAN SIMULATION

4.1 Introduction

The Model Predictive Control (MPC) approach discussed in the previous chapter is applied to simulate the behavior of any redundant system in the presence of external disturbances that act online. Since the human upper limb has redundant DOFs, the optimization-based control approach can be used to simulate human dynamic motions while optimizing some performance measures, reducing the error between desired and current position of joint angles, and considering the effect of the presence of external disturbances not known *apriori*. In this chapter, the MPC approach has been applied to control and simulate the behavior of the dynamic model of human upper limb.

A visualization plug-in was developed to allow the user to interact with the digital human upper limb of Santos™, and to visualize the results obtained by applying the developed control strategy. The "model predictive controller" module, as shown in Figure 1, was developed using C# and C++ languages and was linked by Dynamic Library (DLL) to Santos Engine™ (a human motion simulation engine developed by Virtual Soldier Research, The University Of Iowa). The back end of the program that does all of the calculations and implements the control algorithm was written in C programming language.

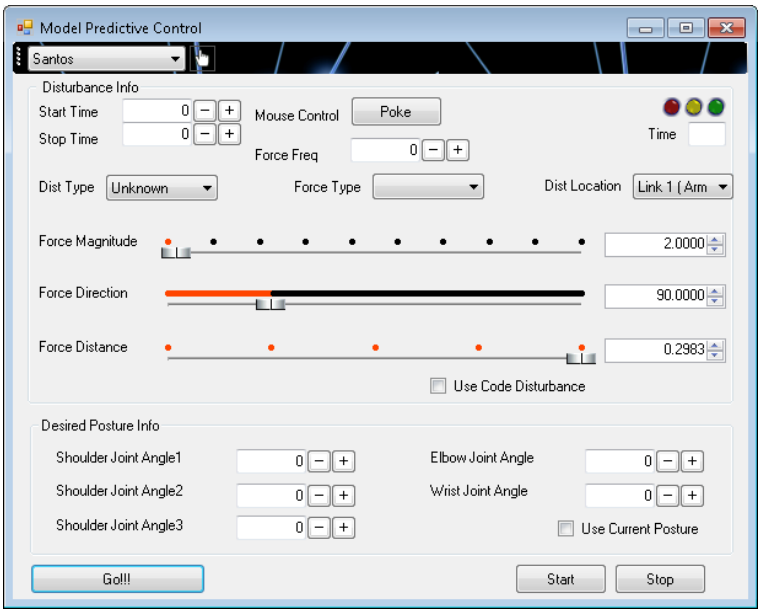

Figure 8 The Visualizer window that helps the user to interact with the upper limb of Santos™

The interface allows a user to select the avatar, if multiple avatars are present in the scene, on which the disturbances can be applied. A user can also select the start and end times for the disturbance. This feature allows the user to study the effect of on-line external disturbances even before the upper limb has reached the desired position. It also allows the user to select the force function from the force type drop down menu: i) Constant option allows the user to apply a constant force with any magnitude, ii) Sinusoidal option allows the user to apply a sinusoidal force with different magnitude and frequency. Also the Disturbance type drop down menu allows user to select any kind of disturbances from the following three options: i) Unknown, ii) Known and iii) Measured. Checking the "Use Current Posture" box, lets the user to select a desired posture generated by posture prediction (Marler, et al. 2009) when the user interactively selects a desired location for the end effector in the 3D space.

In addition, if the "Use Code Disturbance" box is checked, a user can input any number of external disturbances at any times throughout the simulation at any locations on the three segments of the upper limb.

The next section describes the inputs, MPC parameters, and different case studies for which the simulations were performed to visualize the effects of different kinds of disturbances on the upper limb model.

4.2 Simulation Results

Table 4 shows the model parameters used to simulate the upper limb motion. The lengths of the segments are based on the 50^{th} percentile human upper limb from GEBOD software (Cheng, et al. 1996). The inertia properties are obtained by assuming the segments of the upper limb to be a cylindrical rod.

Model Parameters	Arm (shoulder to elbow) Link 1	Forearm (elbow to wrist) Link 2	Hand (wrist to the tip of middle finger) Link 3
Link length (meter)	0.29827	0.247374	0.1717
Link mass (kg)	2.8	0.6	0.1
Lower joint angle limits (Rad)	-0.401	-2.58	-0.9861
Higher joint angle limits (Rad)	2.155	-0.218	1.2042
Lower torque limits (Nm)	-47	-58.7	-6
Higher torque limits (Nm)	66	60.3	12.2

Table 4 Model Parameters Used to Control and Simulate the Digital Human Upper Limb

Table 5 shows the input parameters to the controller for simulating the upper limb motion. Note that the optimization window size is twice the time step. Thus, the number of windows for receding horizon is two.

Input Parameters	Arm (sholder to elbow) Link 1	Forearm (elbow to wrist) Link 2	Hand (wrist to the tip of middle finger) Link 3
Initial joint angle position (Radians)	1.39623	-0.6108	0.26179
Initial joint angle velocity (Rad/sec)	0	0	0
Initial joint angle acceleration (Rad/sec^2)	0	0	0
Desired joint angle position (Rad)	0	0	0
Desired joint angle velocity (Rad/s)	0	0	0
Time step (sec)	0.05	0.05	0.05
Initial torque (Nm)	1	1	1
Optimization window size (sec)	0.1	0.1	0.1

Table 5 Input Parameters Used to Control and Simulate the Digital Human Upper Limb

The table below gives the external disturbance information for the first three cases for which the results are presented below:

Input Quantities	Case :1	Case : 2	Case :3
1. Disturbance type	Unknown, Known, Measured	Unknown, Known, Measured	Unknown, Known, Measured
2. Force type	Constant	Periodic - sinusoidal	Constant
3. Disturbance start time (sec)	0.4	0.4	0.3 1.3
4. Disturbance end time (sec)	4.5	4.5	1.0 2.0
5. Force magnitude (Newton)	4	4	4
6. Force freq (Hz)	-	10	-
7. Force direction (degrees)	90	90	90
8. Distance link (meter)	0.29827	0.29827	0.29827
9. Link	1	1	1

Table 6 External Disturbance Information

Case 1

This simulation case compares the system response under the effect of online constant sustained unknown, known and measured disturbances at the end of link 1. Specific information related to the system and the test case can be found in Table 4 and Table 6

Figure 9 and Figure 10 demonstrates the joint angle and torque values respectively for each joint angle. Unknown disturbance shows larger angle error than known and measured. Since the controller gets the information about known disturbance ahead of time, it generates larger counter torques to keep the system stabilized when the actual disturbance acts on the plants.

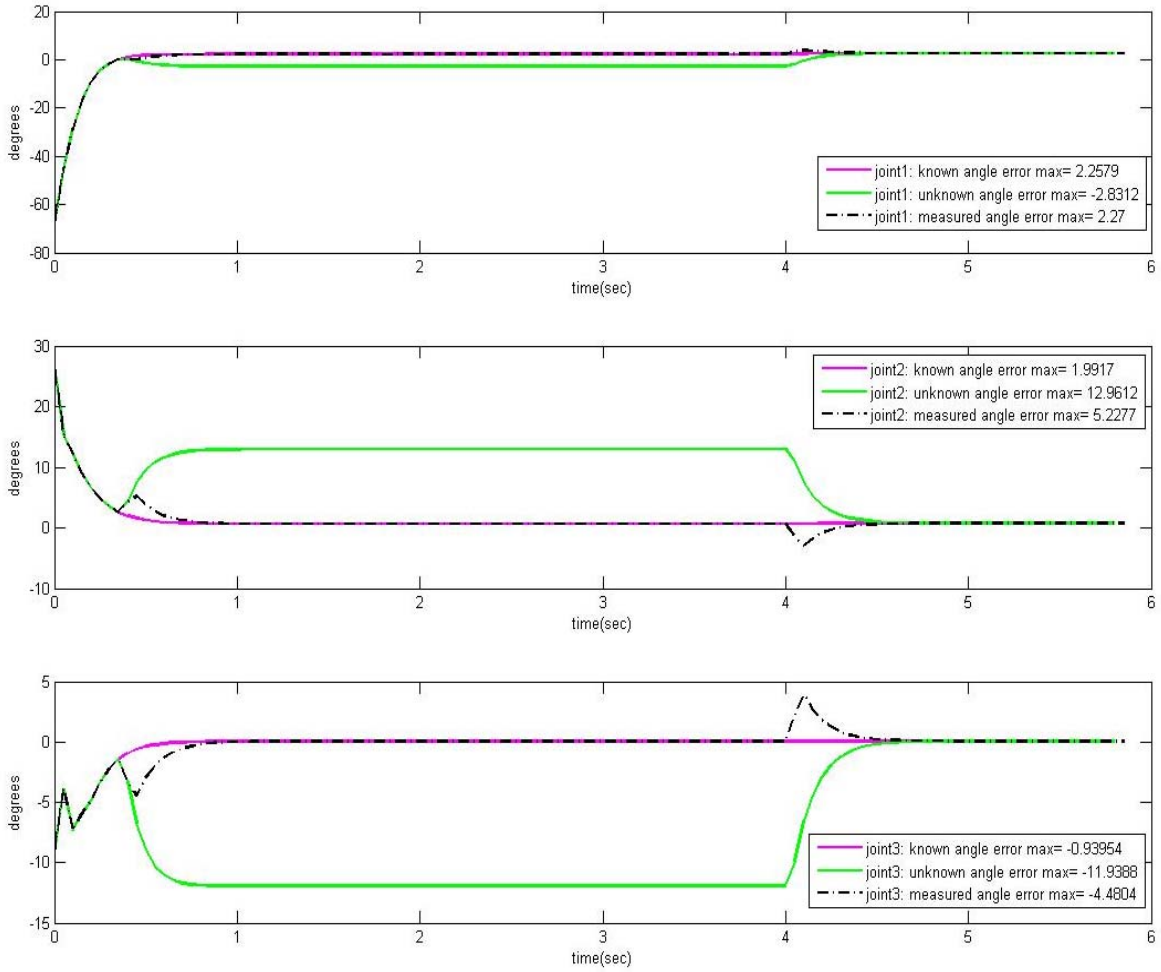

Figure 9 Comparison of individual joint angle error to compare the effect of known, unknown and measured disturbances under the effect of constant unknown disturbance

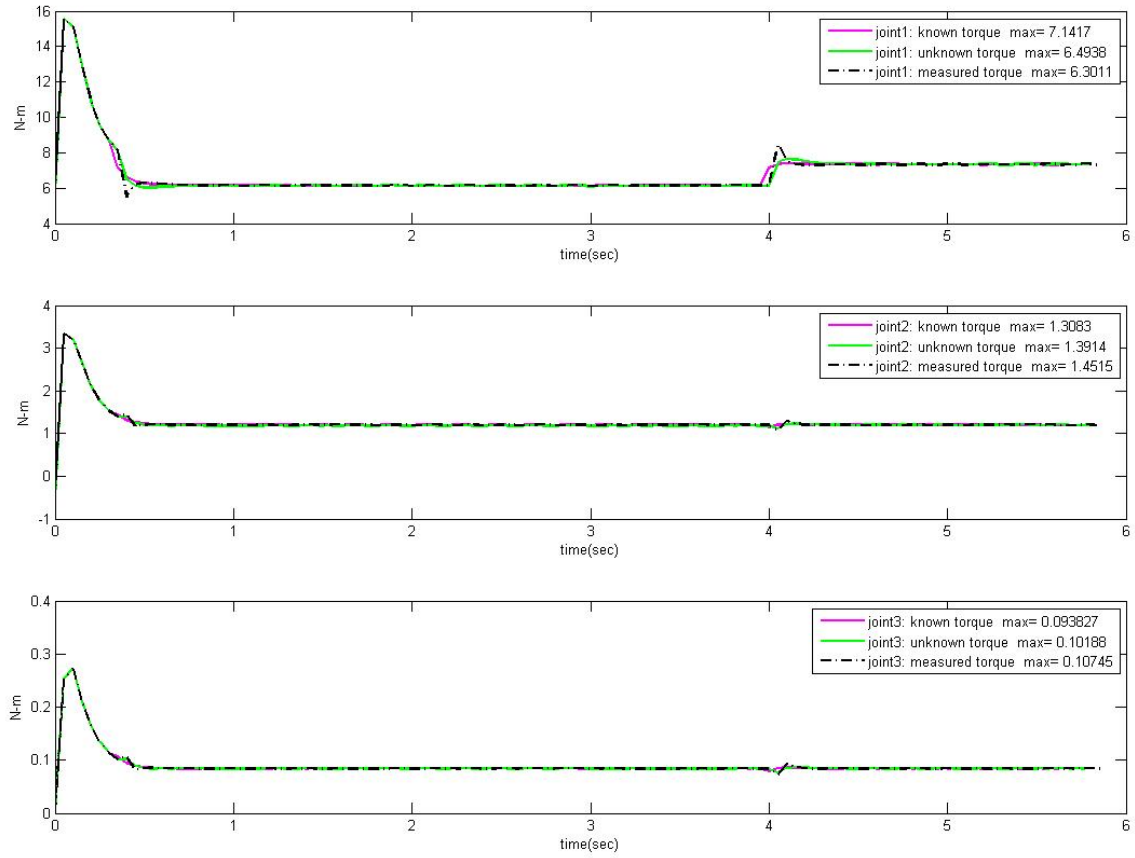

Figure 10 Comparison of individual joint torques to see the effect of known, unknown and measured disturbances under the effect of constant unknown disturbance

Case 2

Figure 11 and Figure 12 demonstrates the response of the system under periodic disturbance with a frequency of $10Hz$. The behavior of the upper limb is similar to the case 1. Under the effect on unknown disturbance, large deviations are observed from the desired joint angles Figure 11, with the controller adapting to known and measured disturbances producing large counter torques.

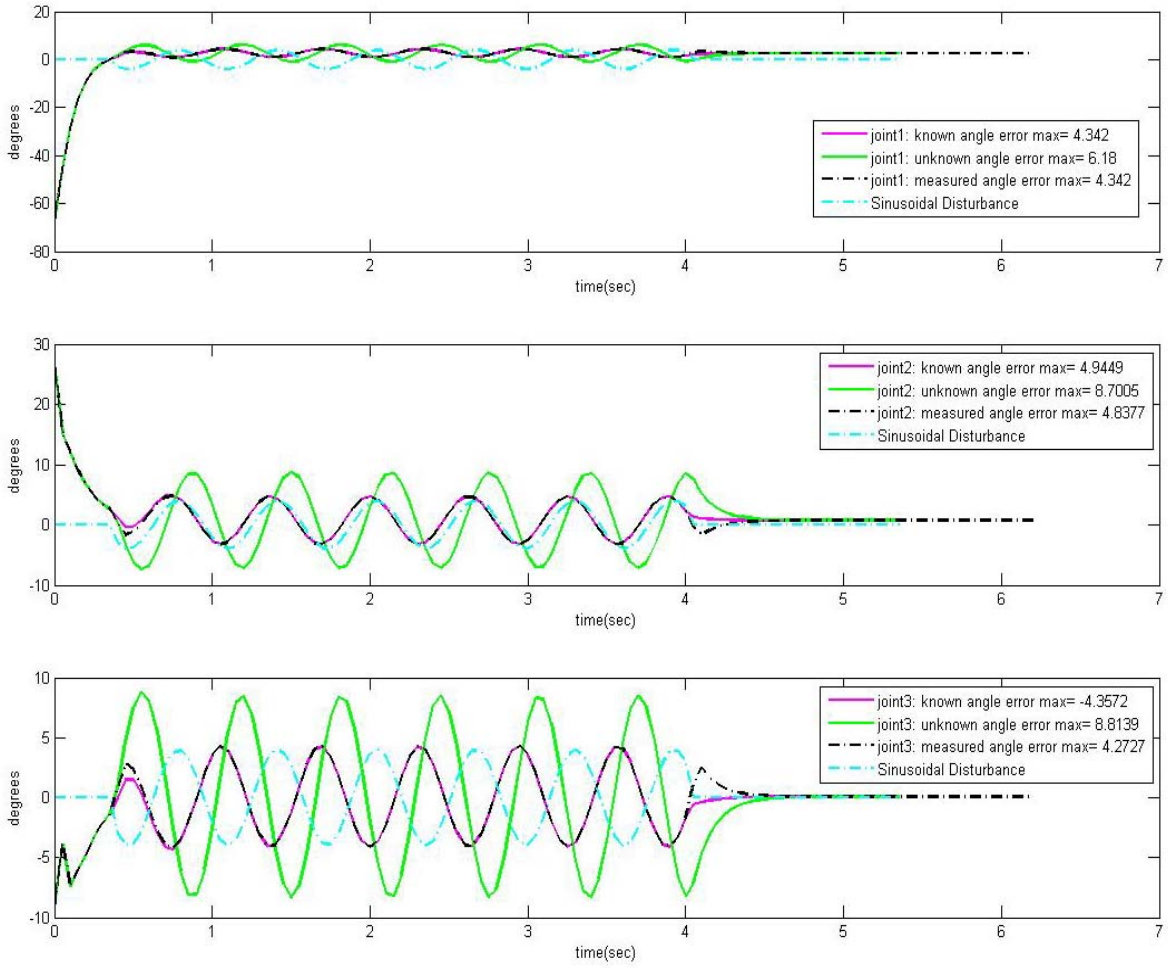

Figure 11 Comparison of individual joint angle error to see the effect of known, unknown and measured disturbances under the effect of sinusoidal disturbance

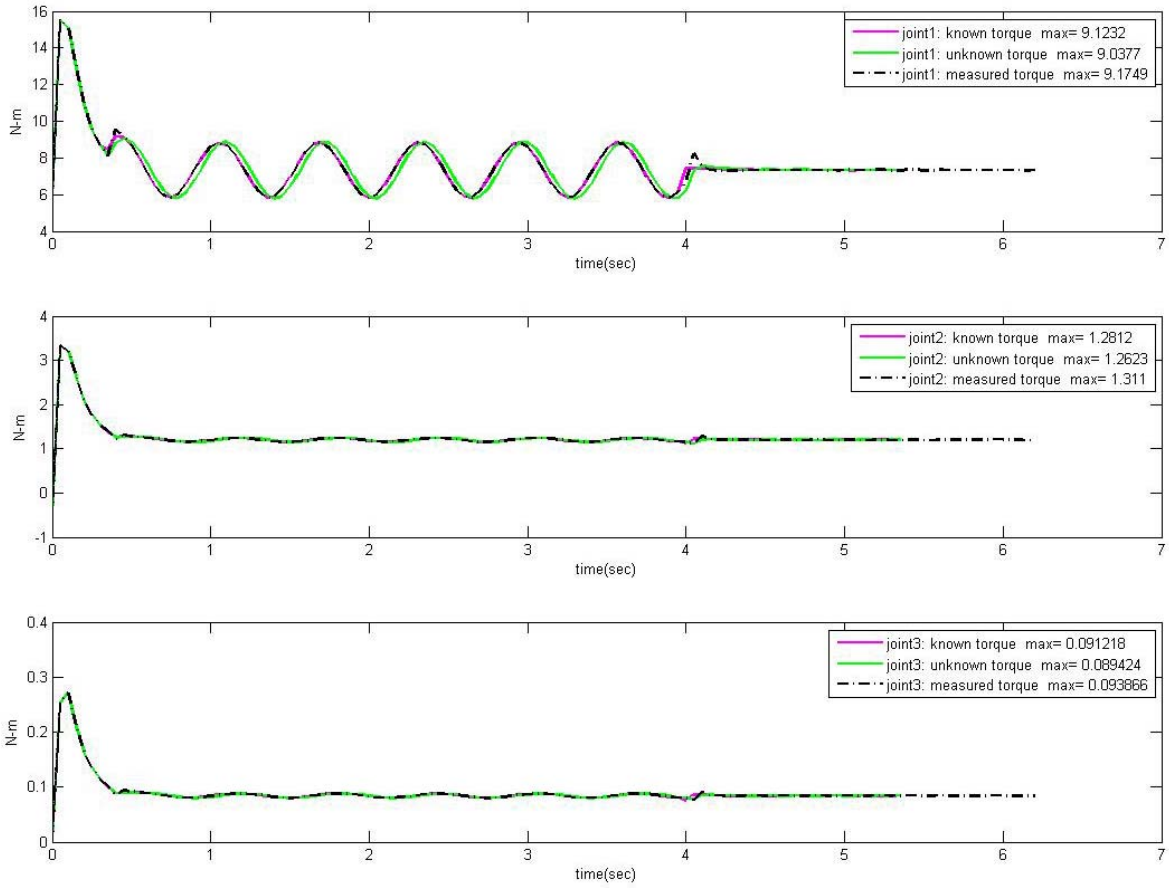

Figure 12 Comparison of individual joint torques to see the effect of known, unknown and measured disturbances under the effect of sinusoidal disturbance

Case 3

This is special simulation case to study the effect of impact disturbances before and after the system stabilizes. The known disturbance case has shown steady stability with minimum deviation of the joint angles from the desired. Figure 13 and Figure 14 show these effects

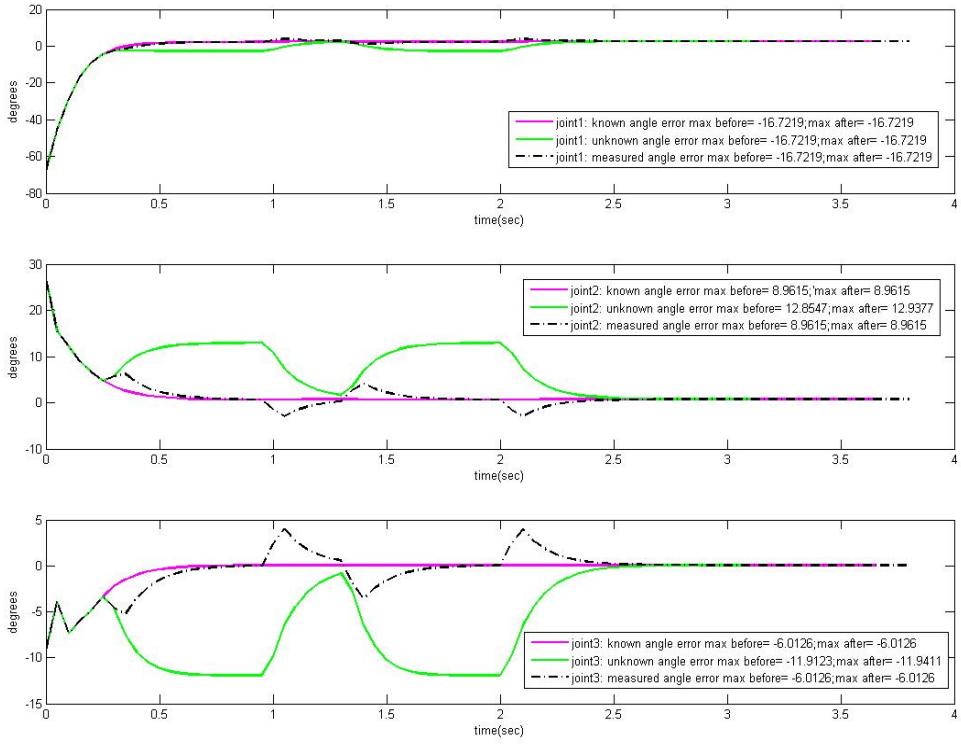

Figure 13 Impact of disturbance before and after the system stabilizes. Comparison of joint angle error under the effect of unknown, known and measured disturbances

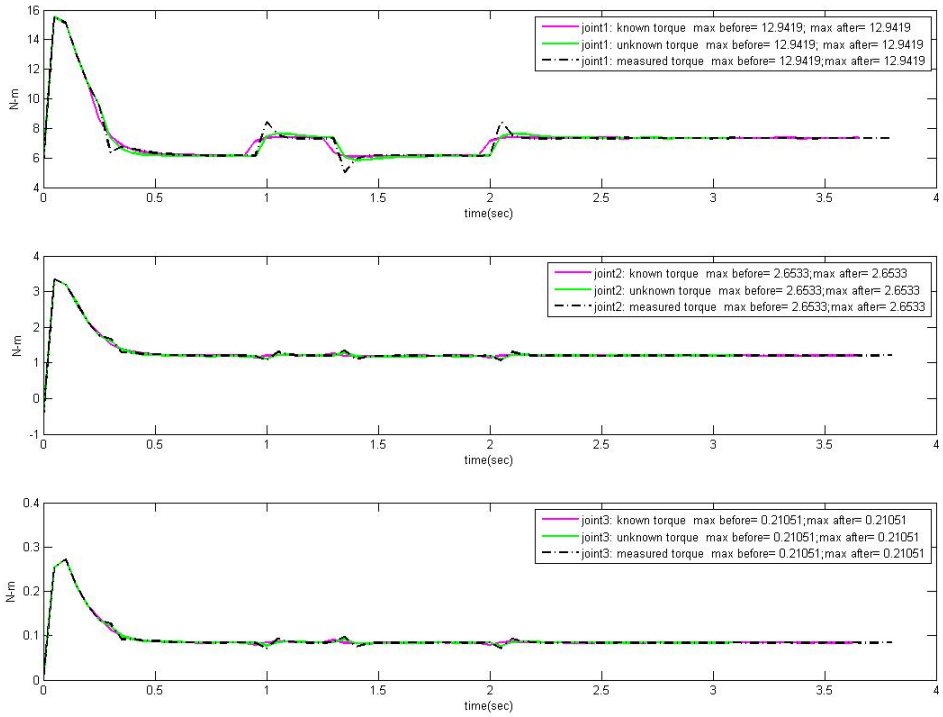

Figure 14 Impact of disturbance before and after the system stabilizes. Comparison of joint torques under the effect of unknown, known and measured disturbances

Case 4

As the predictive horizon and the optimization window size is increased, the controller gets more time to adapt to the situation precisely. As a result, with the increase in optimization window size, the controller response results in low magnitude torques to overcome reduced joint angle deviation from the desired joint angles. For an effective human system, minimizing the amount of torque becomes one of the important factors. Figure 15 and Figure 16 shows the effect of optimization window sizes one, two and three time steps long on the cost functions $(\text{sum of angle errors})^2$ and $(\text{sum of torques})^2$ respectively.

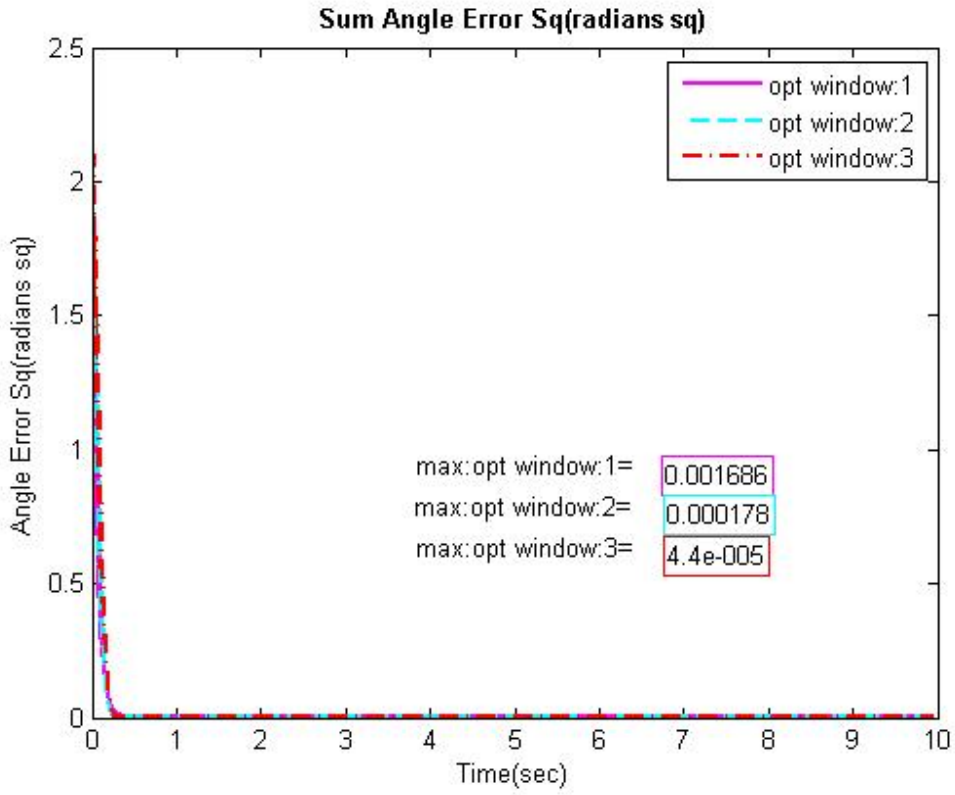

Figure 15 Comparison of the output of three window sizes for sum of angle error square (a cost function) under no disturbance case

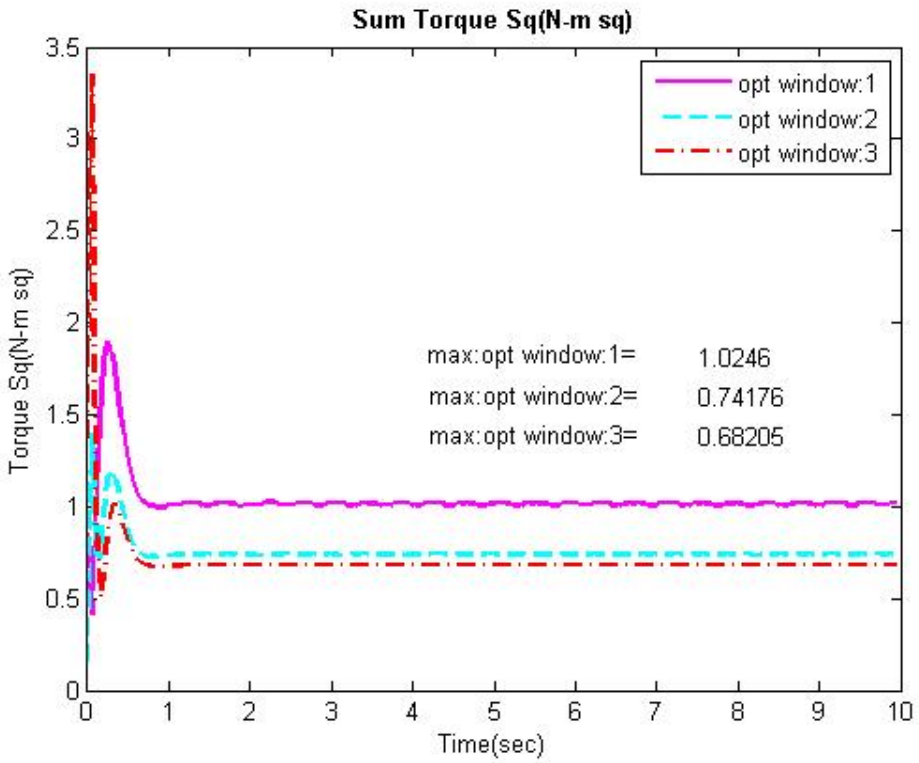

Figure 16 Comparison of the output of three window sizes for sum of torque square (a cost function) under no disturbance case

Case 5

Penalty weights of the cost functions in the optimizer component are the key parameters towards controlling or tuning the response of the Model Predictive Controller. Although we have proposed a further research on developing adaptive weights, playing with them can give some insights. The inputs for this case are same as Case 1 except the optimizing weights as described in Table 7

Simulations Compared	Objective weights			Ratios $\frac{W_1}{W_2}$
	W_1 (Sum of joint angle error)^2 $\sum^{nDOF=3} W_1\left(\theta_{di}-\tilde{\theta}_{yi}\right)^2$	W_2 (Sum of joint angle velocity error)^2 $\sum^{nDOF=3} W_2\left(\dot{\theta}_{di}-\tilde{\dot{\theta}}_{yi}\right)^2$	W_3 (Sum of torques)^2 $\sum^{nDOF=3} W_3\left(\tau_i(t)\right)^2$	
1	3282.806	10	0.1	328.2
2	3282.806	20	0.1	164
3	3282.806	40	0.1	82

Table 7 The objective weights used for simulating Case 5 in Figure 17 and Figure 18

The rate of joint angle error is increased with respect joint angle velocity error in Figure 17 and Figure 18. As a result joint angle error is seen to increase with the decrease in the weight ratio between (joint angle error)^2 and (joint angle velocity error)^2.

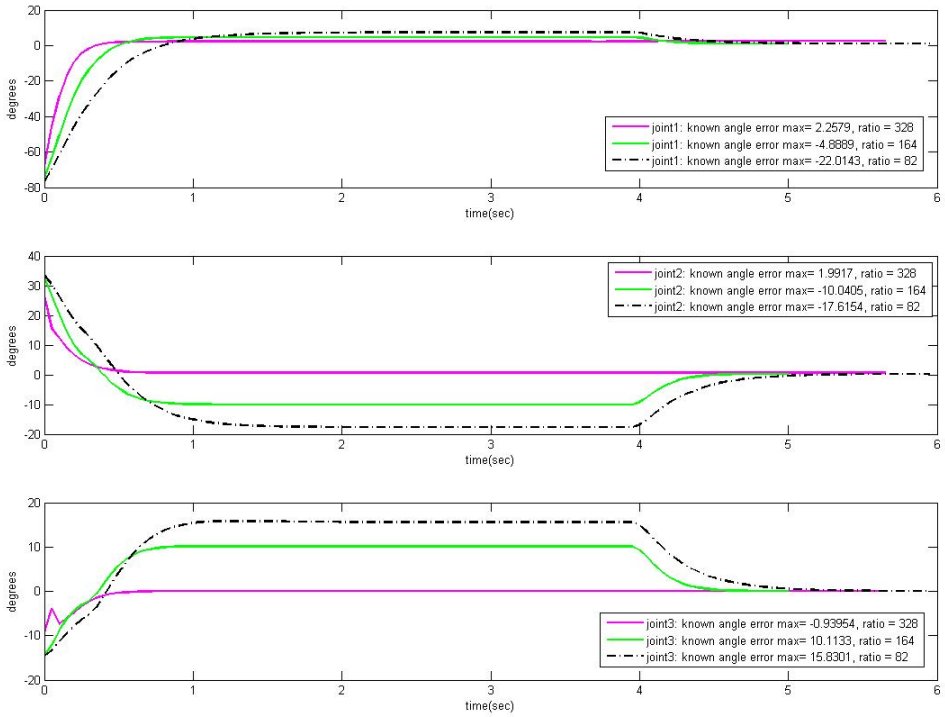

Figure 17 Comparison of joint angle error for different optimizing weight ratios between (joint angle error)^2 and (joint angle velocity error)^2 under known disturbance

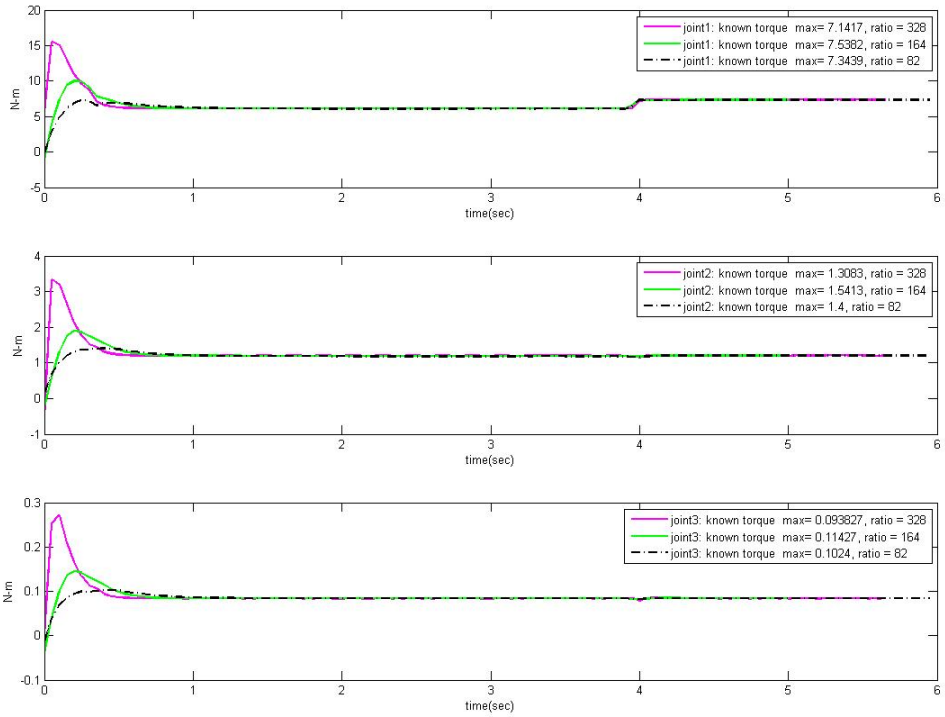

Figure 18 Comparison of torques for different optimizing weight ratios between (joint angle error)^2 and (joint angle velocity error)^2 under known disturbance

Figure 19 and Figure 20 shows the effect of different weight ratios between (joint angle error)^2 and (torque)^2 as shown in Table 8 in terms of joint angle error and torques compared for each joint under the effect of unknown disturbance. With the increase in penalty on torques, relative to the joint angle error, it was found that the system becomes oscillatory and is not able to stabilize under the effect of disturbance.

Simulations Compared	Objective weights			Ratios $\frac{W_1}{W_3}$
	W_1 (Sum of joint angle error)^2 $\sum^{nDOF=3} W_1\left(\theta_{di}-\tilde{\theta}_{yi}\right)^2$	W_2 (Sum of joint angle velocity error)^2 $\sum^{nDOF=3} W_2\left(\dot{\theta}_{di}-\dot{\tilde{\theta}}_{yi}\right)^2$	W_3 (Sum of torques)^2 $\sum^{nDOF=3} W_3\left(\tau_i(t)\right)^2$	
1	3282.806	10	0.5	6564
2	3282.806	10	0.01	328200
3	3282.806	10	0.1	32820

Table 8 The objective weights used for simulating Case 5 in Figure 19 and Figure 20

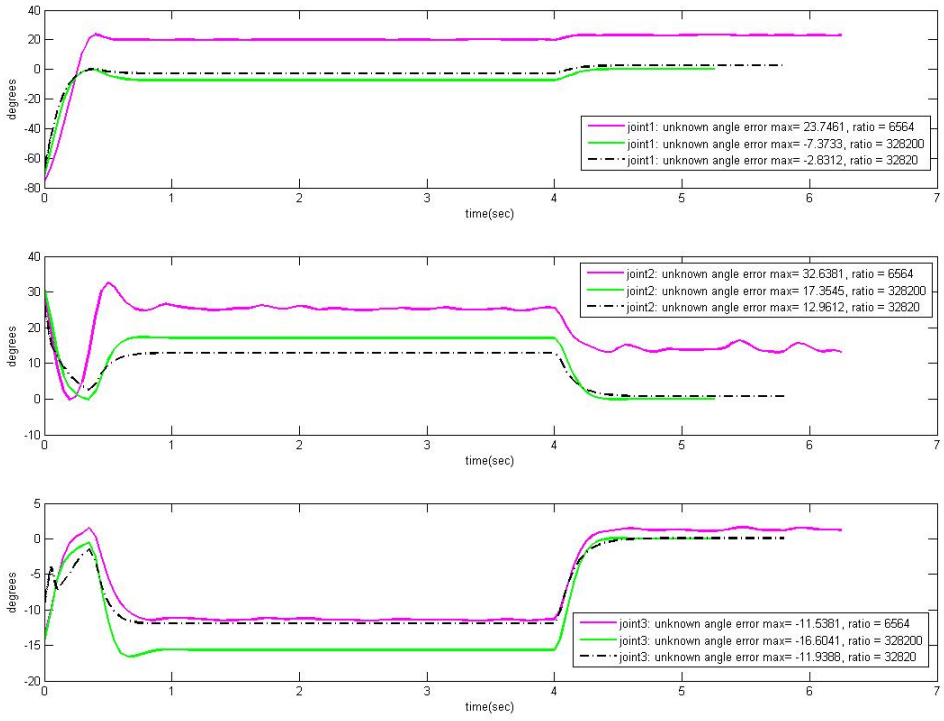

Figure 19 Comparison of joint angle error for different optimizing weight ratios between (joint angle error)^2 and (torque)^2 under unknown disturbance

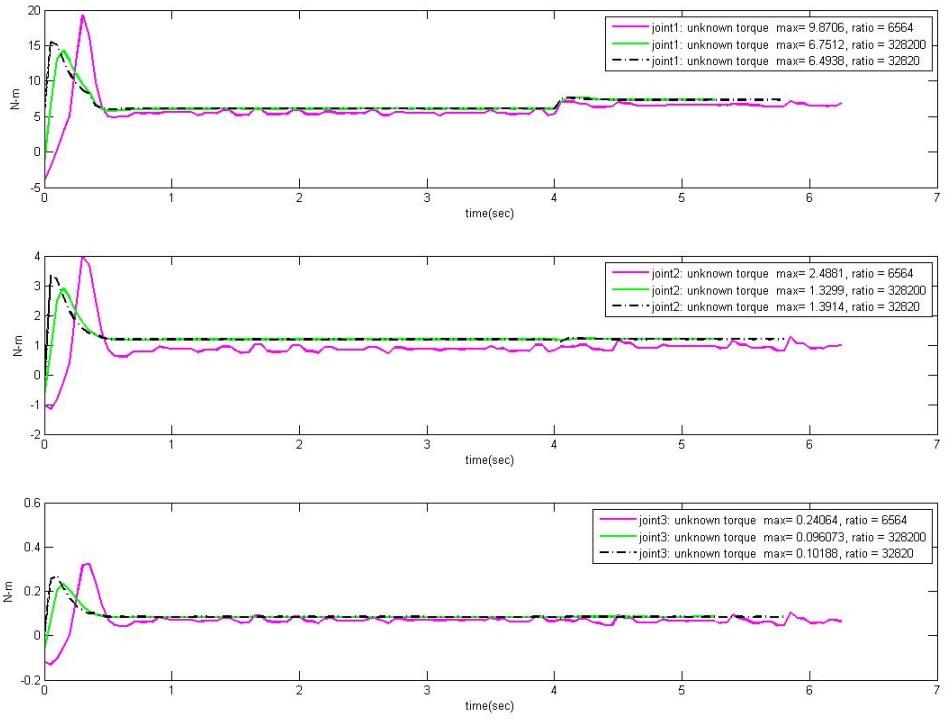

Figure 20 Comparison of torques for different optimizing weight ratios between (joint angle error)^2 and (torque)^2 under unknown disturbance

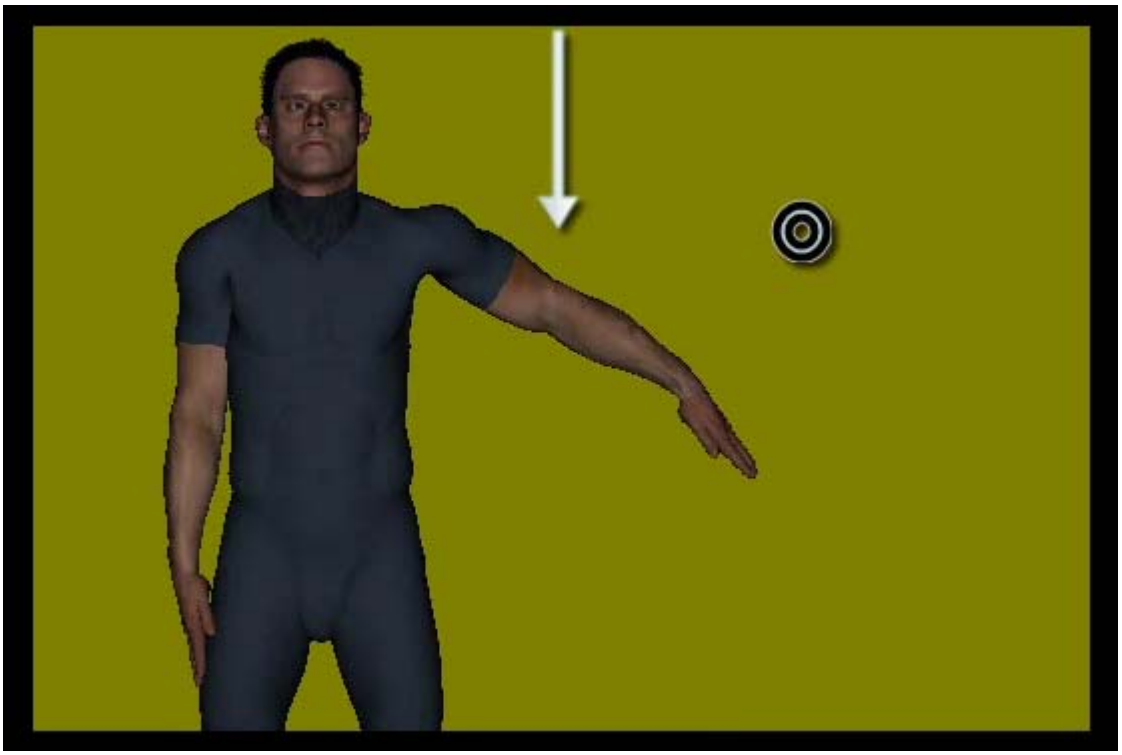

Figure 21 Snapshot of Santos™ using ISO 3411 from Santos Engine™ while simulating in the presence of external disturbance with its direction marked by the arrow and the circle depicting the goal or the desired position of end-effector for a simulation case with 17N unknown disturbance force

4.3 Conclusion

Changes in optimization window sizes, different ratios of cost function weights in the objective function - affects the performance of the system since these are the controlling paramters that define our system. With the increase in optimization window size, more precise movements are observed. But a very large increase in the optimization window size can prove to be computationally costly. Reduced penalty weights on (joint angle error)^2 relative to other cost functions can cause larger deviation of current joint angles from the desired. As far as different disturbance cases are concerned, the system is

seen to be adapting depending upon the disturbance type. For known and measured disturbance cases, the controller produces larger torques as a feature of its adaptation towards unknown to stabilize the system with reduced errors.

CHAPTER 5
CONCLUSION

Model Predictive Controller based motion prediction is built on a gradient based framework of optimization. On-line dynamic response against external environmental disturbances exposes the potential of predictive digital human modeling and simulation taking it to a level closer to "lifelike" experiences. This capability allows the user to simulate different nature of disturbances like constant or periodic over sustained duration exploiting the human perception of sensing the environmental changes through known and measured kinds of disturbances. A better approach of recursive Lagrangian formulation which determines the dynamics of the system has been incorporated lately which will increase the scalability (easy transition while adding more degrees of freedom to the model) than the regular Lagrangian formulation (Goussous, et al. 2009).

In this approach, we simulate a planar 3-link serial chain mechanism with 3 degrees of freedom to imitate upper limb motion under the effect of known, measured and unknown disturbances using linear Model Predictive Controller.

Chapter 2 describes the mathematics and dynamics of the upper limb model. It derives the recursive Lagrangian equations of motion for a 3 dof model. It also describes the disturbance formulations.

Chapter 3 describes the two main components of model predictive control: the predictive component and the optimization component.

Chapter 4 describes the test case results. It was concluded that changes in optimization window size, weights of the objective function affects the performance of a system. Moreover with unknown disturbance case, the joint angle error is seen to be

significantly higher than known and measured cases. Known case requires more counter torques ahead of time to overcome angle error.

The major limitations currently existing with this design are: planar model with 3 dof (not realistic when imitating upper limb) and constant weights penalizing the objective function. Although with recursive formulation, it is easy to step up the number of dofs for a system. Thus easy scaling (increasing the dofs) of the design, moving it into a 3D space will develop simulations close to human upper limb responses. But with increase in dof, there may be a lag in the system response, hindering its ability to react to online disturbances, which may cause a need to incorporate Proportional Derivative quick action (Silva, et al. 2008). It will also be an interesting future work to develop optimization based adaptive weights, which can adjust to different scenarios making the control more flexible.

Moreover a future blend of the Model Predictive Control based technique with physics based optimization technique: predictive dynamics (Xiang, et al. 2009b) or Empirical data-based approach using Motion Capture, will produce a more immersive and interactive simulation for different ranges of motion developing precise control of movements when exposed to environmental disturbances.

REFERENCES

Abdel-Malek, K., and Arora, J. (2009), "Physics-Based Digital Human Modeling:Predictive Dynamics," in *Handbook of Digital Human Modelling*, ed. V. G. Duffy.

Anderson, F., and Pandy, M. (2001), "Dynamic Optimization of Human Walking," *Journal of Biomechanical Engineering-Transaction of the ASME*, 123, 381-390.

Ausejo, S., and Wang, X. (2009), "Motion Capture and Human Motion Reconstruction," in *Handbook of Digital Human Modeling*, ed. V. Duffy.

Badler, N., Palmer, M., and Bindiganavale, R. (1999), "Animation Control for Real-Time Virtual Humans," *Communications of ACM*, 42, 64-73.

Bhatt, R., Xiang, Y., Kim, J., Mathai, A., Penmatsa, R., Chung, H.-J., Kwon, H.-J., Patrick, A., Rahmatalla, S., Marler, T., Beck, S., Yang, J., Arora, J., Abdel-Malek, K., and Obusek, J. (2008), "Dynamic Optimization of Human Stair-Climbing Motion."

Bureau of Labor Statistics [BLS]. (2003), "Occupational Employment and Wages, 2002."

Cheng, H., Obergefell, L., and Rizer, A. (1996), "The Development of Gebod Program," in *Biomedical Engineering Conference*, Dayton, OH, U.S.A.

Denavit, J., and Hartenberg, R. S. (1955), "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *Journal of applied mechanics*, 22, 215-221.

Fu, K., Gonzalez, R., and Lee, C. (1987), *Robotics: Control, Sensing, Vision and Intelligence*, McGrawHill.

Gill, P., Murray, W., Saunders, M., Drud, A., and Kalvelagen, E. (2002), "Snopt: An Sop Algorithm for Large-Scale Constrained Optimization," *Society for Industrial and Applied Mathematics*, 47, 99-131.

Goussous, F., Bhatt, R., Dasgupta, S., and Abdel-Malek, K. (2009), "Model Predictive Control for Human Motion Simulation," *2009 SAE International*.

Hindmarsh, A. (2000), "The Pvode and Ida Algorithms," Technical.

Hollerbach, J. (1980), "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity," *IEEE Transactions on Systems, Man, And Cybernetics*, 10.

Kwon, W., and Han, S. (2005), "Receding Horizon Control."

Marler, T., Arora, J., Yang, J., Kim, H.-J., and Abdel-Malek, K. (2009), "Use of Multi-Objective Optimization for Digital Human Posture Prediction," *Engineering Optimization*, 41, 925-943.

Masters, T. (2009), " Will Avatar Crown James Cameron ‘King of the Universe’? Bbc News," *Will Avatar crown James Cameron ‘King of the Universe’? BBC News*, memory.

Mistry, P., and Maes, P. (2009), "A Wearable Gestural Interface," in *SIGGRAPH Asia 2009*, Yokahoma, Japan.

Park, W. (2009), "Data-Based Human Motion Simulation," in *Handbook of Digital Human Modelling*, Taylor & Francis Group,LLC, p. 9.

Rahmatalla, S., Xiang, Y., Smith, R., Li, J., Meusch, J., Bhatt, R., Swan, C., Arora, J., and Abdel-Malek, K. (2008), "A Validation Protocol for Predictive Human Location," *2008 SAE International*.

Sheth, K., Goussous, F., Bhatt, R., Dasgupta, S., and Abdel-Malek, K. (2010), "Controls-Based Motion Prediction in the Presence of External Forces," in *Applied Human Factors and Ergonomics*.

Silva, M., Abe, Y., and Popovic, J. (2008), "Simulation of Human Motion Data Using Short-Horizon Model Predictive Control."

Xiang, Y., Arora, J., and Abdel-Malek, K. (2007), "Optimization-Based Motion Prediction of Mechanical Systems: Sensitivity Analysis," *Structural And Multidisciplinary Optimization*, 37.

Xiang, Y., Arora, J., Rahmatalla, S., and Abdel-Malek, K. (2009a), "Optimizaiton-Based Dynamic Human Walking Prediction: One Step Formulation," *International Journal for Numerical Methods in Engineering*.

Xiang, Y., Chung, H.-J., Kim, J., Bhatt, R., Rahamatalla, S., Yang, J., Marler, T., Arora, J., and Abdel-Malek, K. (2009b), "Predictive Dynamics: An Optimization-Based Novel Approach for Human Motion Simulation," *Structural and Multidisciplinary Optimization*.

Zhang, X., and Chaffin, D. (1999), "A Three-Dimensional Dynamic Posture Prediction Model for Simulating in-Vehicle Seated Reaching Movements: Development and Validation," *Ergonimics*, 43, 1314-1330.