Bio-mechanical Analysis of Human Joints and Extension of the Study to Robot

S. Parasuraman, and Ler Shiaw Pei

Abstract—In this paper, the bio-mechanical analysis of human joints is carried out and the study is extended to the robot manipulator. This study will first focus on the kinematics of human arm which include the movement of each joint in shoulder, wrist, elbow and finger complexes. Those analyses are then extended to the design of a human robot manipulator. A simulator is built for Direct Kinematics and Inverse Kinematics of human arm. In the simulation of Direct Kinematics, the human joint angles can be inserted, while the position and orientation of each finger tips (end-effector) are shown. Inverse Kinematics does the reverse of the Direct Kinematics. Based on previous materials obtained from kinematics analysis, the human manipulator joints can be designed to follow prescribed position trajectories.

Keywords—Kinematics, Human Joints, Robotics, Robot Dynamics, Manipulators.

I. INTRODUCTION

THIS study focuses on the kinematics analysis of human arm and extends it to the human manipulator.

The representation of the robot's end-effector position and orientation through the geometries of robots (joint and link parameters) are called Direct Kinematics. Using Direct Kinematics, the mathematical model is developed to compute the position and orientation of each fingertip (end- effector's) based on the given human joint position. Each human joint is considered as revolute joint. The homogenous transformation of the fingertip related to the base frame (arm upper limb) is formulated using Denavit-Hartenberg (D-H) method. [3]

Inverse Kinematics analysis is a formulation to compute a set of joint variables from the given end effector or tool piece pose. In the present study, an articulated finger consists of a set of rigid segments connected with joints. Each finger joint angles will be computed by the given fingertip position and orientation. Varying the angles of the joints yields an indefinite number of configurations. Geometric approach will be used to solve this problem.

Jacobians are time-varying linear transformations. In the field of robotics, Jacobians $(J(\theta))$ is relating the joint velocities to Cartesian velocities of the tip of the arm. In the present study, the procedure of Jacobians matrix of the human finger motion is formulated, each of which has three revolute joints There are several methods may be used to derive

Authors are with the Monash University Sunway Campus, Bandar Sunway, 46150, Malaysia (phone: +603 55146254, fax: +603 55146207; email: s.parasuraman@eng.monash.edu.my).

Jacobians matrix. The method used in the study is called "Velocity propagation from link to link."

Manipulator dynamics is the manipulator motion that causes the torque and moments at the joint due to the externals loads that applied at the end effector, manipulator joint velocities, and accelerations. The Newton-Euler method is used to formulate the model. The external forces acting at the fingertip, and due to this, the force and moment transferred to each joint are estimated using the Newton-Euler method.

The design of the control system will be introduced here. A primary concern of a position control system is to automatically compensate for errors in knowledge of the parameters of a system, and to suppress disturbances which tend to perturb the system from the desired trajectory.

II. THEORETICAL WORK

A. Direct Kinematics of Human Arm

First, the human arm will be modeled by using Direct Kinematics principal.
The steps to solve the problems are:[1]
1. Attach an inertial frame to the human arm joint.
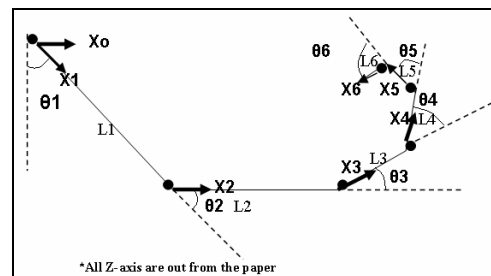2. Attach frames to links, including the finger tip.

Fig. 1 Affixing the Frames to Each Joint

3. Write out the Denavit-Hartenberg Table

TABLE I
D-H TABLE

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	$\theta 1$
2	0	L1	0	$\theta 2$
3	0	L2	0	$\theta 3$
4	0	L3	0	$\theta 4$
5	0	L4	0	$\theta 5$
6	0	L5	0	$\theta 6$
7	0	L6	0	0

4. Determine the homogenous transformation between each frame by applying the general formula.

$$
{}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

5. Apply the set of transforms sequentially to obtain a final overall transform.

$$
{}^{0}_{7}T = \begin{bmatrix} c_{1234567} & -s_{1234567} & 0 & L_1c_1 + L_2c_{12} + L_3c_{123} + L_4c_{1234} + L_5c_{12345} + L_6c_{123456} \\ s_{1234567} & c_{1234567} & 0 & L_1s_1 + L_2s_{12} + L_3s_{123} + L_4s_{1234} + L_5s_{12345} + L_6s_{123456} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

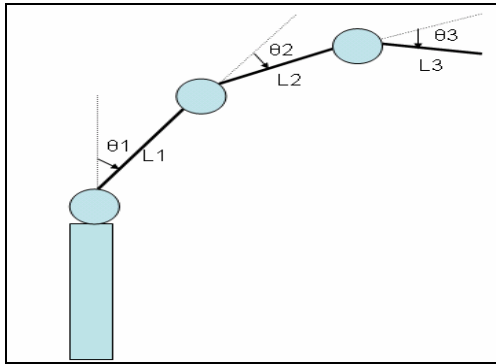### B. Inverse Kinematics of Human Fingers

Fig. 2 Flexion Angles of Fingers

First of all we previously compute forward kinematic of the fingers. [2]

$$
x = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \quad (1)
$$

$$
y = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \quad (2)
$$

And we know that

$$
\phi = (\theta_1 + \theta_2 + \theta_3) \quad (3)
$$

For inverse kinematic, we are given Cartesian coordinates, x, y, and ϕ. We will be finding the value for θ_1, θ_2, θ_3.

First step is substituting eqn (3) into (1) and (2) and θ_3 is eliminated. And we are left with two equations and two unknown θ_1, θ_2.

$$
x - l_3 \cos\phi = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \quad (4)
$$

$$
y - l_3 \sin\phi = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) \quad (5)
$$

The variables in left side are all given, while the θ_1, θ_2 are unknown in this case.

or convenience calculating purpose, we let $x' = x - l_3 \cos\phi$ and $y' = y - l_3 \sin\phi$. And we square both side for equation (4) and (5) and sum them together

$$
(x' - l_1 \cos\theta_1)^2 = (l_2 \cos(\theta_1 + \theta_2))^2 +
$$
$$
(y' - l_1 \sin\theta_1)^2 = (l_2 \sin(\theta_1 + \theta_2))^2
$$

Sum them together and we will get a single nonlinear equation in θ_1

$$
(-2l_1 x')\cos\theta_1 + (-2l_1 y')\sin\theta_1 + (x'^2 + y'^2 + l_1^2 - l_2^2) = 0 \quad (6)
$$

At first we have 3 unknowns $\theta_1, \theta_2, \theta_3$, while in equation (4), (5), we have eliminated θ_3. While in equation (6) we have further eliminate θ_2. And we are left with θ_1.

Equation (6) is in similar form of

$$
P\cos\alpha + Q\sin\alpha + R = 0 \quad (7)
$$

and by using that there are two solutions for θ_1:

$$
\theta_1 = \gamma + \sigma \cos^{-1}\left[\frac{-(x'^2 + y'^2 + l_1^2 - l_2^2)}{2l_1\sqrt{x'^2 + y^2}}\right] \quad (8)
$$

Where, $\gamma = \text{atan2}\left[\frac{-y'}{\sqrt{x'^2 + y'^2}}, \frac{x'}{\sqrt{x'^2 + y'^2}}\right]$ and $\sigma = \pm 1$.

So there are two solutions for θ_1, one is from $\sigma = -1$, while another solution is from $\sigma = +1$.

Substituting the solution back into equation (4) and (5) will give us:

$$
\cos(\theta_1 + \theta_2) = \frac{x' - l_1 \cos\theta_1}{l_2}, \quad \sin(\theta_1 + \theta_2) = \frac{y' - l_1 \sin\theta_1}{l_2}
$$

This will allows us to solve the solution for θ_2 by using atan2 function.

$$
\theta_2 = \text{atan2}\left[\frac{y' - l_1 \sin\theta_1}{l_2}, \frac{x' - l_1 \cos\theta_1}{l_2}\right] - \theta_1
$$

For each solution of θ_1, there is a solution for θ_2. And finally we will be able to calculate θ_3 by using

$$
\theta_3 = \phi - \theta_1 - \theta_2
$$

C. Jacobians of Human Fingers

A manipulator is a chain of bodies, each one capable of motion relative to its neighbors. Based on the structure, the velocities of each link can be computed in order starting from the base. The velocity of link i+1 will be that of link I, plus whatever new velocity components were added by joint i+1. Similarly, the human fingers have same theory with manipulator.[1]

Assume each joint of the finger as a rigid body with linear and angular velocity vectors describing its motion. Further,

express these velocities with respect to the link frame itself rather than with respect to the base coordinated system (palm). Rotational velocities may be added when both ω vectors are written with respect to the same frame. Therefore, the angular velocity of link i+1 is the same as that of link i plus a new component caused by rotational velocity at joint i+1.

The equations that will be used throughout this chapter are:[1]

$$^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\,\omega_i + \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} \qquad (9)$$

$$^{i+1}V_{i+1} = {}^{i+1}_{i}R({}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{i+1}) \qquad (10)$$

The rotational velocity and linear velocity at frame {4} is found out to be:

$$^{4}\omega_4 = {}^{4}_{3}R\bullet {}^{3}w_3 + \dot{\theta}_4\bullet {}^{4}\hat{Z}_4$$

$$= \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3+\dot{\theta}_4 \end{bmatrix}$$

$$^{4}V_4 = {}^{4}_{3}R\{{}^{3}V_3 + {}^{3}\omega_3 \times {}^{3}P_4\}$$

$$= \begin{bmatrix} L_1s_2c_3\,\dot{\theta}_1+L_1s_3c_2\,\dot{\theta}_1+L_2s_3(\dot{\theta}_1+\dot{\theta}_2) \\ -L_1s_2s_3\,\dot{\theta}_1+L_1c_3c_2\,\dot{\theta}_1+L_2c_3(\dot{\theta}_1+\dot{\theta}_2)+L_3(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3) \\ 1 \end{bmatrix}$$

To find the velocities with respect to the nonmoving base frame, rotate them with the rotation matrix ${}^{0}_{4}R$.

$$^{o}V_4 = {}^{0}_{4}R \bullet {}^{4}V_4$$

$$= \begin{bmatrix} c_{123} & -s_{123} & 0 \\ s_{123} & c_{123} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} (L_1s_2c_3+L_1s_3c_2+L_2s_3)\dot{\theta}_1+L_2s_3\dot{\theta}_2 \\ (-L_1s_2s_3+L_1c_3c_2+L_2c_3+L_3)\dot{\theta}_1+(L_2c_3+L_3)\dot{\theta}_2+L_3\dot{\theta}_3 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} -L_1s_1-L_2s_{12}-L_3s_{123} & -L_2s_{12}-L_3s_{123} & -L_3s_{123} \\ L_1c_1+L_2c_{12}+L_3C_{123} & L_2c_{12}+L_3c_{123} & L_3c_{123} \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

$$^{o}V = {}^{0}J \bullet \dot{\theta}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -L_1s_1-L_2s_{12}-L_3s_{123} & -L_2s_{12}-L_3s_{123} & -L_3s_{123} \\ L_1c_1+L_2c_{12}+L_3C_{123} & L_2c_{12}+L_3c_{123} & L_3c_{123} \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

$$\therefore J = \begin{bmatrix} -L_1s_1-L_2s_{12}-L_3s_{123} & -L_2s_{12}-L_3s_{123} & -L_3s_{123} \\ L_1c_1+L_2c_{12}+L_3C_{123} & L_2c_{12}+L_3c_{123} & L_3c_{123} \\ 1 & 1 & 1 \end{bmatrix}$$

Most manipulators have values of θ where the Jacobian becomes singular. Such locations are called singularities of the mechanism or singularities for short. All manipulators have singularities at the boundary of their workspace, and most have loci of singularities inside their workspace.

To find the singular points of a mechanism, the determinant of its Jacobian must be examined. This can be done by computing the determinant of J and setting it to zero.

$$DET[J(\theta)] = \begin{vmatrix} -L_1s_1-L_2s_{12}-L_3s_{123} & -L_2s_{12}-L_3s_{123} & -L_3s_{123} \\ L_1c_1+L_2c_{12}+L_3C_{123} & L_2c_{12}+L_3s_{123} & L_3c_{123} \\ 1 & 1 & 1 \end{vmatrix} = 0$$

The expression for the determinant of the Jacobian, can be simplified using trigonometric identities to:

$$|J| = L_1L_2\sin\theta_2$$

This means that the Jacobian is singular only when θ_2 is either 0 or 180 degrees. Physically, this corresponds to the middle phalanx being completely extended or complete flexed.

D. Dynamics of Human Fingers

One method of controlling a manipulator to follow a desired path involves calculating these actuator torque functions using the dynamic equations of motion of the manipulator. A second use of the dynamic equations of motion is in simulation. The dynamic equations of motion may be developed to control or simulate the motion of manipulators[9] [1].

To consider the problem of computing the torques that corresponds to a given trajectory of a manipulator. The position, velocity, and acceleration of the joints ($\theta, \dot{\theta}, \ddot{\theta}$) will be assumed to be known. With this knowledge, and with knowledge of the kinematics and mass distribution information of the robot, the joint torques required to cause this motion can be calculated [1].

The complete algorithm for computing joint torques from the motion of joints is composed of two parts. First, link velocities and accelerations are iteratively computed from link 1 out to link n and the Newton-Euler equations are applied to each link. Second, forces and torques of interaction and joint actuator torques are computed recursively from link n back to link 1. The equations are summarized below for the case of all joints rotational. [8]

Outward iterations: i: 0→5

$$^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\,{}^{i}\omega_i + \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1}$$

$$^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_{i}R\,{}^{i}\dot{\omega}_i + {}^{i+1}_{i}R\,{}^{i}\omega_i \times \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1}$$

$$^{i+1}\dot{v}_{i+1} = {}^{i+1}_{i}R({}^{i}\dot{\omega}_i\times {}^{i}P_{i+1} + {}^{i}\omega_i\times({}^{i}\omega_i\times {}^{i}P_{i+1})+ {}^{i}\dot{v}_i)$$

$$^{i+1}\dot{v}_{C_{i+1}} = {}^{i+1}\dot{\omega}_{i+1}\times {}^{i+1}P_{C_{i+1}} + {}^{i+1}\omega_{i+1}\times({}^{i+1}\omega_{i+1}\times {}^{i+1}P_{C_{i+1}})+ {}^{i+1}\dot{v}_{i+1}$$

$$^{i+1}F_{i+1} = m_{i+1}\,{}^{i+1}\dot{v}_{C_{i+1}}$$

$$^{i+1}N_{i+1} = {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1}\times {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\omega_{i+1}$$

Inward iterations: i: 6→1

$${}^{i}f_{i} = {}^{i+1}_{i}R\,{}^{i+1}f_{i+1} + {}^{i}F_{i}$$

$${}^{i}n_{i} = {}^{i}N_{i} + {}^{i+1}_{i}R\,{}^{i+1}n_{i+1} + {}^{i}P_{C_{i}} \times {}^{i}F_{i} + {}^{i}P_{i+1} \times {}^{i+1}_{i}R\,{}^{i+1}f_{i+1}$$

$$\tau_{i} = {}^{i}n_{i}^{T}\,{}^{i}\hat{Z}_{i}$$

The effect of gravity loading on the links can be included quite simply be setting ${}^{0}\dot{v}_{0}$=G, where G is the gravity vector. This is equivalent to saying that the base of the robot is accelerating upward 1 G acceleration. This fictitious upward acceleration causes exactly the same effect on the links as gravity would. Therefore, the gravity effect is calculated.

When the Newton-Euler equations are evaluated symbolically for any manipulator, they yield a dynamic equation which can be written in the form $\tau = M(\theta)\ddot{\theta} + V(\theta,\dot{\theta}) + G(\theta)$ [7]

Where $M(\theta)$ is the n x n mass matrix of the manipulator, $V(\theta,\dot{\theta})$ is an n x 1 vector of centrifugal and Coriolis terms, and $G(\theta)$ is an n x 1 vector of gravity terms. Each element of $M(\theta)$ and $G(\theta)$ is a complex function which depends on θ, the position of all the joints of the manipulator. Each element of $V(\theta,\dot{\theta})$ is a complex function of both θ and $\dot{\theta}$.
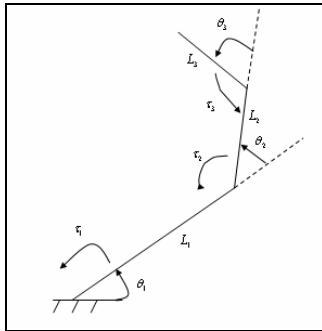

Fig. 3 Finger with Point Masses at Distal End of Joint

The assumptions for the calculation are:

1. Assume point masses at distal end of links:

$${}^{1}P_{c1} = L_{1}\hat{X}_{1}$$

$${}^{2}P_{c2} = L_{2}\hat{X}_{2}$$

$${}^{3}P_{c3} = L_{3}\hat{X}_{3}$$

2. The inertia tensor written at the center of mass for each link is the zero matrixes

$${}^{C_{1}}I_{1} = 0$$

$${}^{C_{2}}I_{2} = 0$$

$${}^{C_{3}}I_{3} = 0$$

3. There are no forces acting on the end-effector, so

$$f_{4} = 0$$

$$n_{4} = 0$$

4. The base of the finger is not rotating, so

$$\omega_{0} = 0$$

$$\dot{\omega}_{0} = 0$$

5. To include gravity forces,

$${}^{o}\dot{v}_{o} = g\hat{Y}_{0}$$

The result of the dynamic analysis of human fingers:

$$\therefore M(\theta) =$$

$$
\begin{bmatrix}
\begin{matrix} L_1^2(m_1+m_2s_2^2+c_2^2m_3^2+s_{23}^2m_3+c_2c_3c_2m_3) \\ L_2^2+L_1m_3c_{23}(1+s_2s_3)+L_2(c_2m_3)+L_3m_3 \\ +L_1L_2(c_2-c_{233}+c_3m_3+s_2s_2m_3(1+c_3)+mc_2) \\ +L_2L_3c_3+L_1L_3m_3s_{23} \end{matrix} & \begin{matrix} L_3m_3+L_2c_3m_3+2L_2^2+L_2L_3c_3 \\ +L_1L_2s_2s_3m_3(1+c_3)+L_1L_3m_3s_{23} \end{matrix} & \begin{matrix} L_3m_3+L_2L_3c_3+L_1L_3m_3s_2s_3 \end{matrix} \\[2em]
\begin{matrix} 2L_2^2+L_2m_3+L_2c_{23}m_3+L_2c_3m_3 \\ +L_2L_3c_3+L_1L_2(-c_{23}+c_2) \end{matrix} & \begin{matrix} L_3m_3+L_2c_3m_3+2L_2^2+L_2L_3c_3 \end{matrix} & L_3m_3+L_2L_3c_3 \\[1.5em]
L_3m_3+L_2c_{23}m_3+L_2c_3m_3 & L_3m_3+L_2c_3m_3 & 0
\end{bmatrix}
$$

$$V(\theta,\dot{\theta}) =$$

$$
\begin{bmatrix}
\begin{matrix} (L_2s_3m_3-L_2^2s_3c_3-L_1L_2s_2m_3+L_2^2s_3c_3)(\dot{\theta}_1+\dot{\theta}_2) \\ +[L_1m_3s_{23}(1-s_2s_3)+L_1L_2(s_2-c_{233})+L_1^2(-s_{23}c_{23}m_3+s_{23}c_2c_3m_3+s_2c_2m_2)-L_1c_2m_2g]\dot{\theta}_1^2 \\ -L_2s_2m_2(\dot{\theta}_1+\dot{\theta}_2)^2-L_2L_3s_3(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3)-L_1L_3s_{23}m_3(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3)^2 \end{matrix} \\[2em]
\begin{matrix} (L_2s_3m_3)(\dot{\theta}_1+\dot{\theta}_2)+[L_1m_3s_{23}+L_1L_2(s_2-c_{23}c_3+s_{23}c_3)]\dot{\theta}_1^2-L_2L_3s_3(\dot{\theta}_1+\dot{\theta}_2+\dot{\theta}_3) \end{matrix} \\[2em]
\begin{matrix} L_2s_3m_3(\dot{\theta}_1+\dot{\theta}_2)+L_1m_3s_{23}\dot{\theta}_1^2 \end{matrix}
\end{bmatrix}
$$

$$G(\theta) =
\begin{bmatrix}
\begin{matrix} gc_{123}m_3(1-s_2s_3)+L_2c_{12}g+L_2L_3gs_{123}+L_2gc_{123}c_3+L_1m_1gc_1 \\ +L_1gc_3c_2m_3(s_{123}+c_{123})+L_1m_2s_{12}gs_2+L_1gs_3c_2s_{123}m_3+L_1m_2c_{12}c_2g \end{matrix} \\[1.5em]
gc_{123}m_3+L_2c_{12}g+L_2L_3gs_{123}+L_2gc_{123}c_3 \\[1em]
gc_{123}m_3
\end{bmatrix}
$$

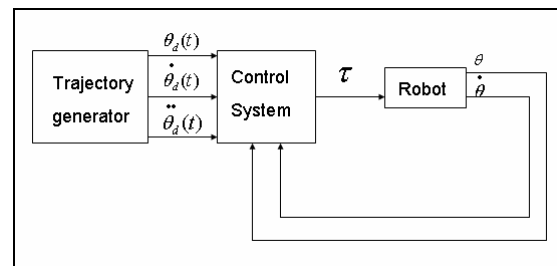### E. Controller System of Robot Manipulator

Fig. 4 High-level Block Diagram of a Robot Control System

Based on previous materials, the means to calculate joint-position time histories that correspond to desired end-effector motions through space are obtained.[5]

The manipulator will be modeled as a mechanism which is

instrumented with sensors at each joint to measure the joint angle, and an actuator at each joint to apply a torque on the neighboring link.

Since the manipulator joints have to follow prescribed position trajectories, but the actuators are commanded in terms of torque, some kind of control system must be used to compute appropriate actuator commands which will realize this desired motion. Normally, these torques are computed by using feedback from the joint sensors to compute the torque required.

Fig. 4 shows the relationship between the trajectory generator and the physical robot. The robot accepts a vector of joint torques, τ, from the control system. The manipulator's sensors allow the controller to read the vector of joint positions, θ, and joint velocities, $\dot{\theta}$. All signal lines in Fig. 4 carry N x 1 vectors [6] (where N is the number of joints in the manipulator).

III. EXPERIMENTAL WORK

A. Simulation

The Virtual Reality Toolbox is a solution for interacting with virtual reality models of dynamic systems over time. It provides a flexible MATLAB interface to virtual reality worlds. After creating MATLAB objects and associating them with a virtual world, the virtual world can be controlled by using functions and methods.

From MATLAB, the user can set the positions and properties of VRML objects, create callbacks from graphical user interfaces (GUIs), and map data to virtual objects. The user can also view the world with a VRML viewer, determine its structure, and assign new values to all available nodes and their fields.

The Virtual Reality Toolbox includes functions for retrieving and changing the virtual world properties and for saving the VRML files corresponding to the actual structure of a virtual world. MATLAB provides communication for control and manipulation of virtual reality objects using MATLAB objects.

The mathematical model formulated in previous section is implemented in the simulation. A MATLAB Graphical User Interface is created to enable the user to select the rotation axes and insert each joint angle. After the user clicks the "Calculate" button, the final position of each finger tip will be calculated and shown on the textfields. A simulation of the human arm motion will then be shown on the screen. Refer to Fig. 5 and Fig. 6.
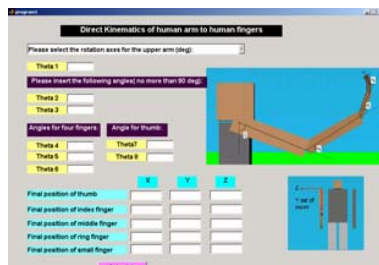

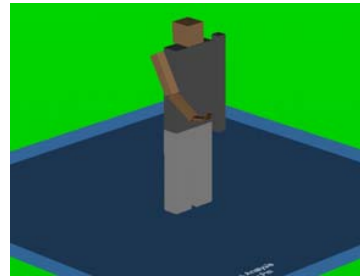
Fig. 5 A GUI for Direct Kinematics Calculation

Fig. 6 VRML for Direct Kinematics Simulation

The mathematical models formulated in previous section are implemented in the simulation here. The user can select different diameter of cylinder range from 30mm up to 70mm. After press the "Calculate" button, the joint angles for each finger will then be calculated and shown on the GUI. The appearance of each finger will be shown in the VRML. Refer to Fig. 7, Fig. 8 and Fig. 9.

Fig. 7 A GUI for Inverse Kinematics Calculation

Fig. 8 VRML for Inverse Kinematics Calculation

Fig. 9 VRML for Inverse Kinematics Calculation

IV. CONCLUSION

This paper presents the bio-mechanical analysis of human joints. The discussion is limited to the human shoulder, elbow,

wrist and fingers. The research is then extended to the study of human manipulator.

Kinematics is the science of motion. In human movement, it is the study of the positions, angles, velocities, and accelerations of body segments and joints during motion. The body segments are considered to be rigid bodies for the purposes of describing the motion of the body.

Forward kinematics is to compute the position and orientation of finger tip (end-effector) by using given a set of joint angles. The position and orientation computed is relative to the arm (base frame). The simulation of human arm forward kinematics is performed through MATLAB Graphical User Interface and VRML.

Inverse kinematics does the reverse of forward kinematics. It is used to compute all possible sets of joint angles by using given position and orientation of the finger tip. In the thesis, first joint angle θ_1 is also assumed to be known, the angles to be computed are θ_1 and θ_3. The simulation of human arm inverse kinematics is performed through MATLAB Graphical User Interface and VRML.

In addition to dealing with static positioning problems, the human fingers in motion are analyzed. In performing velocity analysis of a mechanism, a matrix quantity will be defined. It is called the Jacobian of the manipulator. The Jacobian specifies a mapping from velocities in joint space to velocities in Cartesian space. The nature of this mapping change as the configuration of the manipulator varies. A certain points which are called singularities will be found out. At this point, the mapping is not invertible.

Dynamics is to study the forces required to cause motion. A complex set of torque which are applied to the finger joints are computed. By calculating these torque functions, the desired path followed by the human fingers can be found out.

REFERENCES

[1] John J.Craig. *Introduction to Robotics*. 2nd ed. Pearson Education International.

[2] V.Kumar. *Introduction to Robot Geometry and Kinematics*.[online]. Available from: http://www.seas.upenn.edu/~meam520/notes02/IntroRobotKinematics5.pdf

[3] *Manipulator Kinematics* [online]. Available from: http://users.rsise.anu.edu.au/~chen/teaching/Robotics_ENGN4627_2005/lectureNotes/engn4627-Part03.pdf.

[4] National Institute of Biomedical Imaging and Bioengineering, "Wearable Robots Help with Stroke Rehabilitation", last accessed on 22 Novemeber 2007, http://www.nibib.nih.gov/nibib/File/Health%20and%20Education/Eadvances/PDFs/Wearable%20Robots%20Help%20with%20Stroke%20Rehabilitation.pdf http://www.immersion.com/

[5] Bouzit M., Popescu G., Burdea G., and Boian R, "The Rutgers Master II-ND Force Feedback Glove", IEEE VR 2002 Haptics Symposium, March 2002.

[6] Johnson M. J., "Recent Trends in Robot-Assisted Therapy Environments to Improve Real-Life Functional Performance After Stroke", Journal of NeuroEngineering and Rehabilitation, 18 December 2006.

[7] Lum P. S., Burgar C. G. Shor P. C. Majmundar M., Van der Loos M., "Robot-Assisted Movement Training Compared with Conventional Therapy Techniques for the Rehabilitation of Upper-Limb Motor Function After Stroke" American Congress of Rehabilitation Medicine

and the American Academy of Physical Medicine and Rehabilitation, 2002.

[8] Loureiro R. C. V., Collin C. F., and Harwin W. S., "Robot Aided Therapy: Challenges Ahead for Upper Limb Stroke Rehabilitation", International Conference Disability, Virtual Reality & Assoc. Tech, 2004.

[9] Tognetti A., Lorussi F., Bartalesi R., Quaglini S., Tesconi M., Zupone G., and De Rossi D., "Wearable Kinesthetic System for Capturing and Classifying Upper Limb Gesture in Post-Stroke Rehabilitation", Journal of NeuroEngineering and Rehabilitation, 2 March 2005.

S. Parasuraman has received a BE in Mechanical Engineering from the University of Madras, India and an ME in Manufacturing Automation from Anna University, India. He received a PhD in Mechanical Engineering (Robotics) from Monash University, Australia. He had worked in various capacities as Research and Development Engineer, Research and Development Manager, Lecturer and Senior Lecturer in various Industries in Malaysia and Monash University. His research interests are robotics, artificial intelligence, mobile robot navigation, sensors data fusion for mobile robots and robot kinematics, dynamics and controls.