Modeling and Analysis for Anthropomorphic Motion in Redundant Robot Arms

Manohar Anumolu

December 18, 2017

University of Maryland, College Park

Abstract

The modeling and analysis of a seven degree-of-freedom robot arm is researched for anthropomorphic configurations, i.e., those configurations that resemble human characteristics. Firstly, the Denavit-Hartenberg Parameters are computed for a given 'Right-Arm' Robot arm. For visualization purposes, an interactive model is first generated using Peter Corke's Robotics Toolbox following which a Simulink Model is generated using the above parameters. The Forward and Inverse Kinematics are computed, as well as the End-Effector Velocity using the Manipulator Jacobian. Due to the complications found in a conventional human hand, a gripper constructed with two prismatic joints is used.

LIST OF FIGURES

INDEX

I. INTRODUCTION

As Robots are increasingly becoming an integral part of day-to-day life, there is a need for robots designed to move and act in environments designed for humans. There are various reasons to build humanoid robots as they result in good engineering; to build a humanoid robot requires a varied blend of fields like biology, cognitive science, engineering, computational neuroscience, A.I, Speech Recognition, Navigation, Image-Processing and their total integration into a single unit.

Humanoid Robots prove useful in a wide variety of applications. Studies have shown that autistic children respond favorably to such robots, which is a promising use of such technology. [1] Additionally, their capabilities (or rather their limits) promise a safe and efficient work environment for people. One such example is NASA's Robonaut II, a dexterous humanoid robot built to help humans explore and work in space. [2]

II. NOTIONS AND PRELIMINARIES

In this project, a humanoid robot arm is being modeled and Analyzed. It is a Seven Degree-of-Freedom Robot, with an Extra 'Dof' for the end-effector (I have taken a prismatic joint, I have not considered it in my calculations as the sole function is to 'clasp' an object, so the end effector is technically the end-point of the lower arm).

Furthermore, since only the anthropomorphic configurations are considered, there are certain joint limits for every joint which the robot must follow.

As we can see, the human arm is redundant, since it has seven degrees of freedom, while only six degrees of freedom are required for a given position and orientation of the arm-endpoint. This is a challenge for the inverse kinematics as we see later. It is shown that it is possible to associate a cost-function for each human joint, related to dynamics, neuro-physiological aspects etc. [3]. The arm performs movements that optimize these cost functions. However, a variety of these cost-functions are computationally expensive and not suitable for real-time implementation. In this project, I have not used any cost function, but rather directly solved for the angles required to drive the arm to a certain point in space. Figure [1] is from [3] as well.

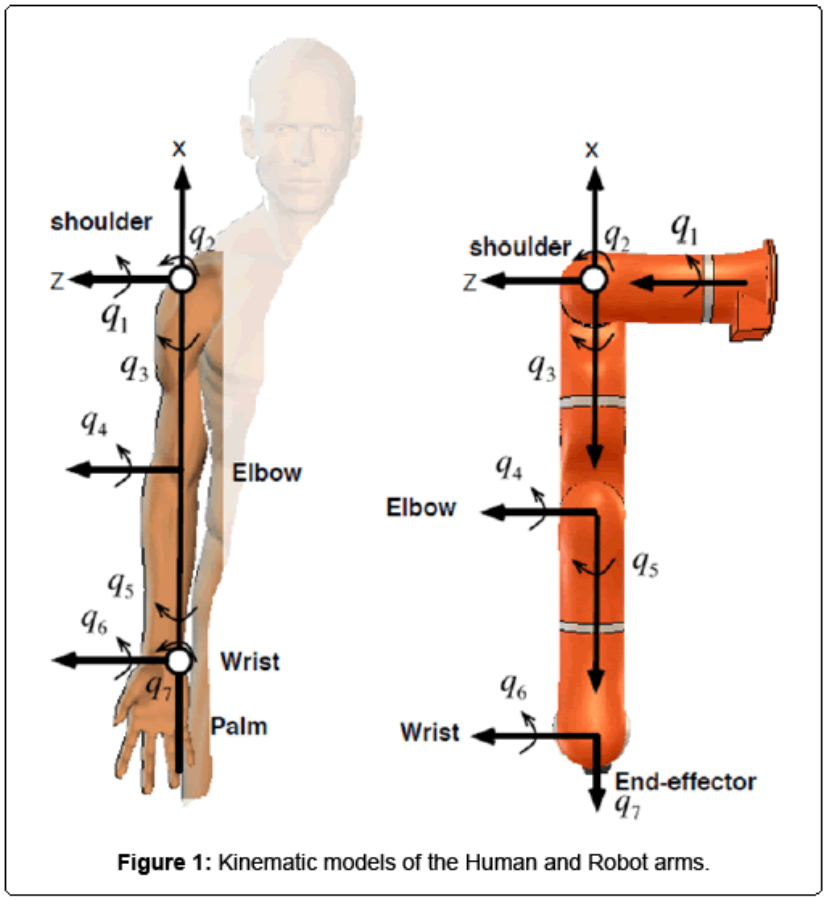


Figure 1: Kinematic models of the Human and Robot arms.

As we can see, this model follows a 3-2-2 configuration, where there are three joints present in the shoulder to represent a ball-and-socket articulation which is given by three revolute joints. The elbow consists

of two revolute joints as well, and the wrist has a minimal representation with two revolute joints. (The joint at the elbow compensates for the range of motion in the wrist).

The joint restrictions are shown as follows, and the implementation of the inverse kinematics MATLAB script was done with a loose approximation of the joint constraints (within a tolerance of 10-15 degrees) of the following figures. Although you can still go beyond the limits in the Simulink model, the script warns you of possible over-extensions in a joint. The following joint-limit information was taken from [3] as well, and shows the minimum and maximum range of all the 7 joints in the arm. For this project I have used a 'Right Arm' Model to do the implementation. The Following are Figure [2], Figure [3]

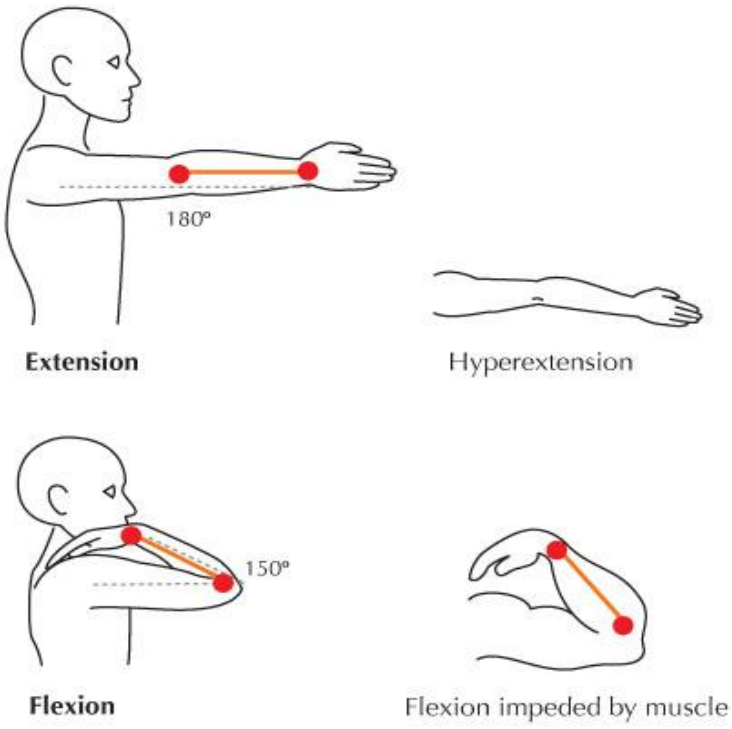
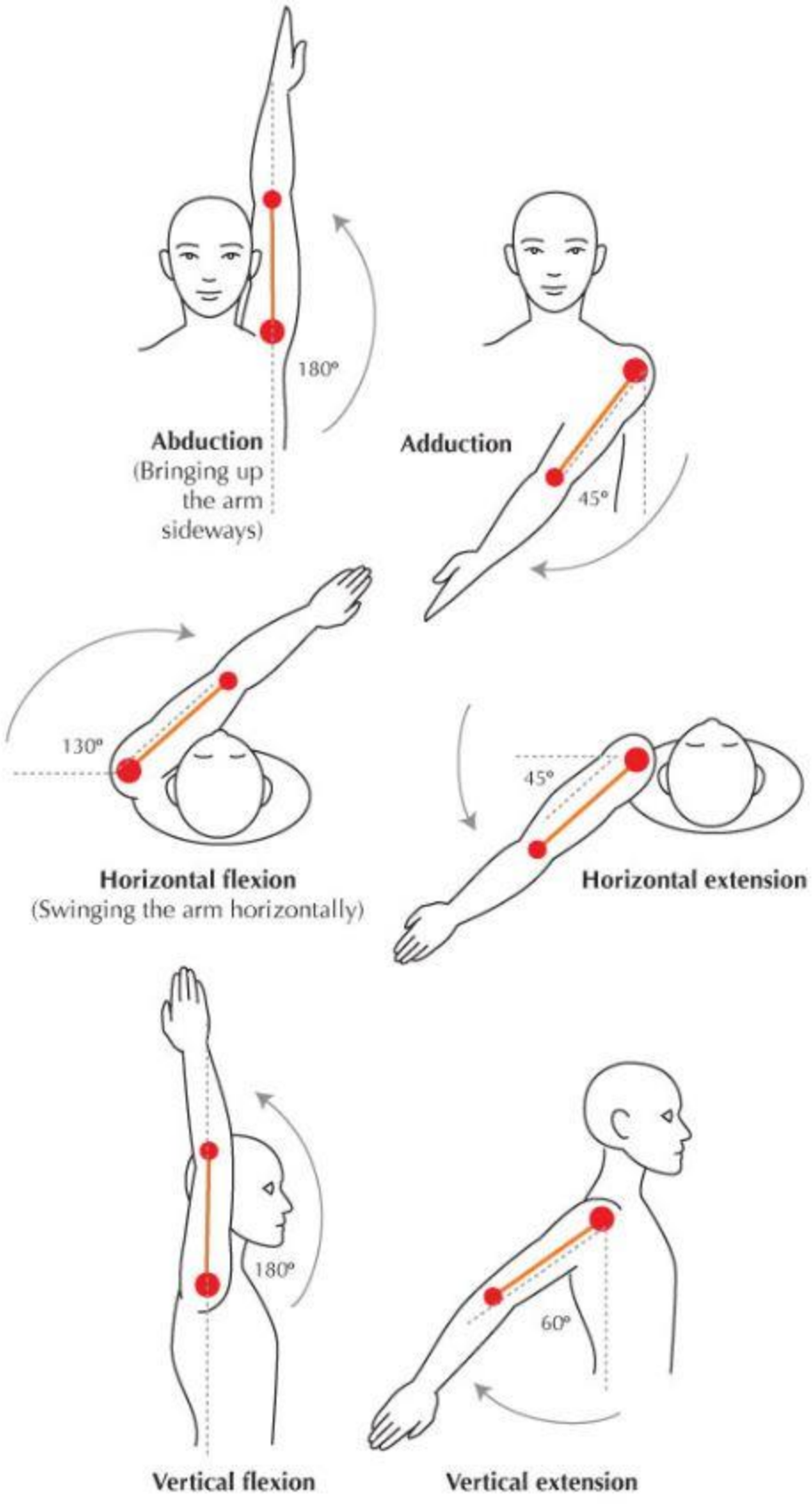


Figure [2] Elbow Extension/Flexion Limits


180°
Abduction
(Bringing up
the arm
sideways)
Adduction
45°
130°
Horizontal flexion
(Swinging the arm horizontally)
45°
Horizontal extension
180°
Vertical flexion
60°
Vertical extension

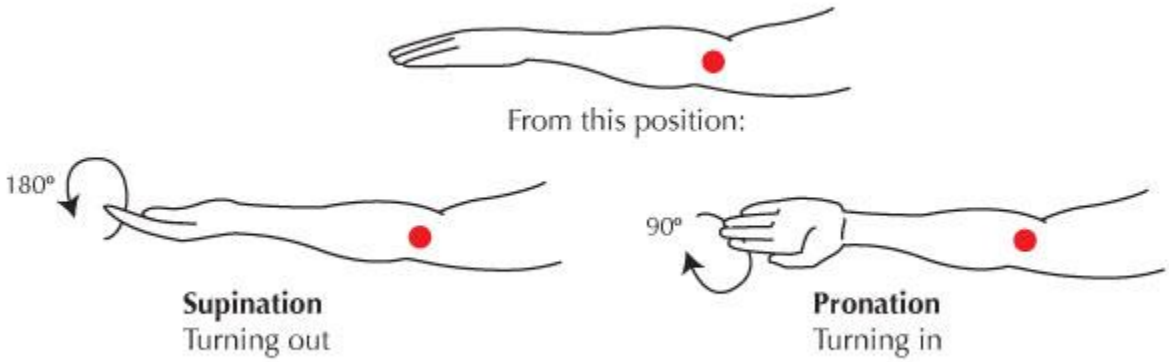


Figure [4] Elbow Supination/Pronation Joint Limits

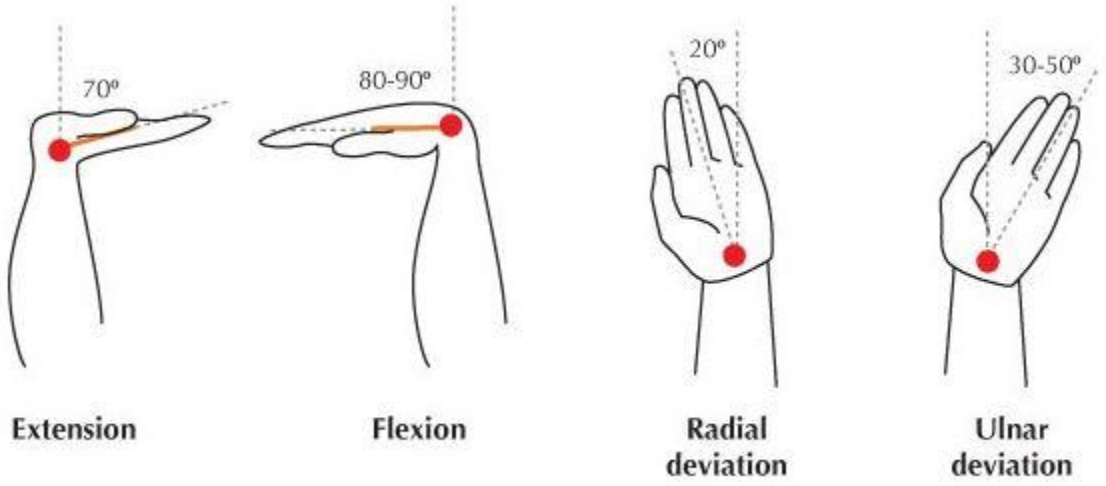


Figure [5] Wrist Extension/Flexion, Radial/Ulnar Deviations

Figure [2] depicts the joint-limits for the elbow, and Figure [3] depicts the Joint-Limits for the Shoulder Joint. All the four figures in this section are taken from [3]

III. RIGID MANIPULATOR DESIGN

First the Denavit-Hartenberg Table is derived for the above Robot Model.

Joint	θ_i	d_i	a_i	α_i
1	θ_1	0	0	90
2	$\theta_2 + 90$	0	0	90
3	$\theta_3 + 90$	-40cm	0	90
4	$\theta_4 + 180$	0	0	90
5	$\theta_5 + 180$	-50cm	0	90
6	$\theta_i + 90$	0	-15cm	90
7	θ_7	0	0	-90

The lengths of the Upper Arm, Lower Arm, and the Palm are 40cm, 50cm and 15cm respectively. Due to the nature of the axes however, the parity is changed for being proper DH Parameters. The Forward Kinematics can then be derived, with the position and orientation of the end-effector as a 4x4 Functional matrix of the seven joint parameters.

For Visualization Purposes, I have constructed both a Rigid Body Tree from MATLAB's Robotics Toolbox, as well as a Model using Peter Corke's Robotics Toolbox [4]. It is observed that the Rigid Body Tree Model was slightly erroneous, possibly due to implementation errors.

The following is the visualization from Peter Corke's Robotics Toolbox, in MATLAB.

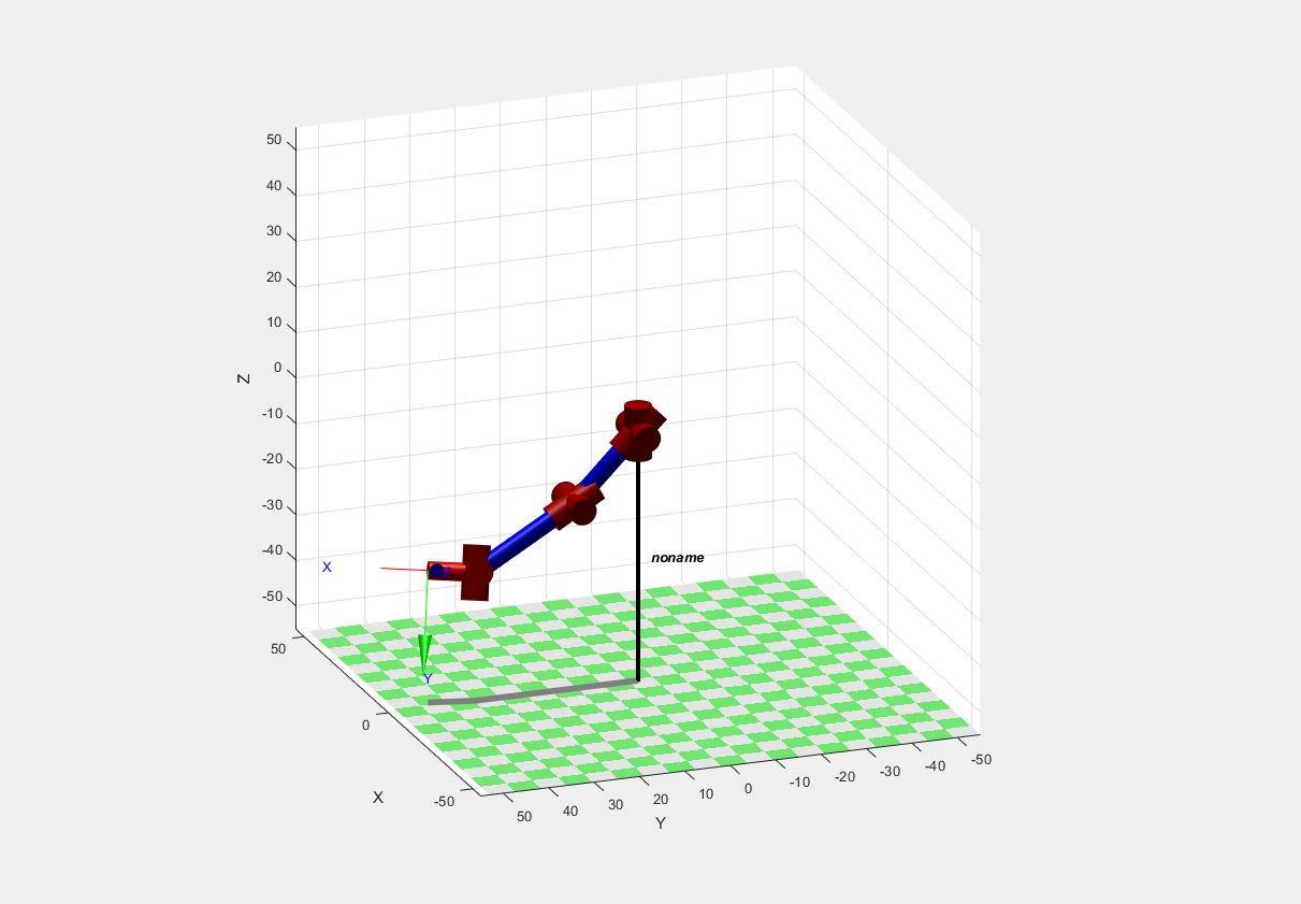


Figure [6] Robot Arm Model in MATLAB

IV. SIMULINK MODEL

In this project, the "Right-Arm" Robot arm is modeled in Simulink, primarily with Primitive Solid Blocks for Links and Cylinders depicted as revolute joints. During the simulation, Parameters such as joint velocities are observed through the Scope Block. The rotation of the blocks is done using Rigid Transforms and fully realize the DH Parameters shown above. The model studied is the 'Right Arm' configuration, while I have also designed a "Left Arm" it is not analyzed completely and can be commented out before simulation.

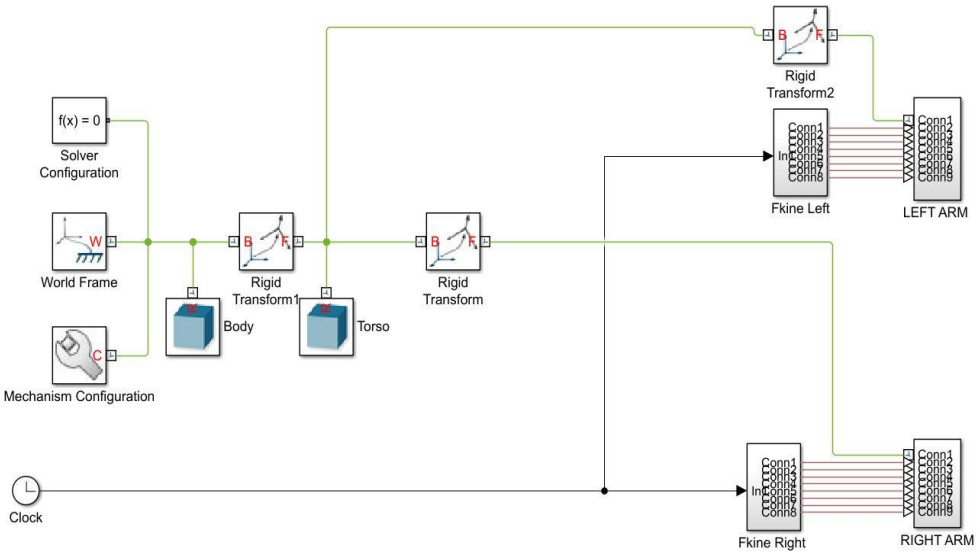


Figure [7] The Complete Anthropomorphic Robot Arm System

There are primarily two blocks for the entire “Right Arm” system. One is the Forward Kinematics Block, which gives the input to drive the actuators into moving during the simulation. The other block is the actual constructed model of the right arm shown as follows:

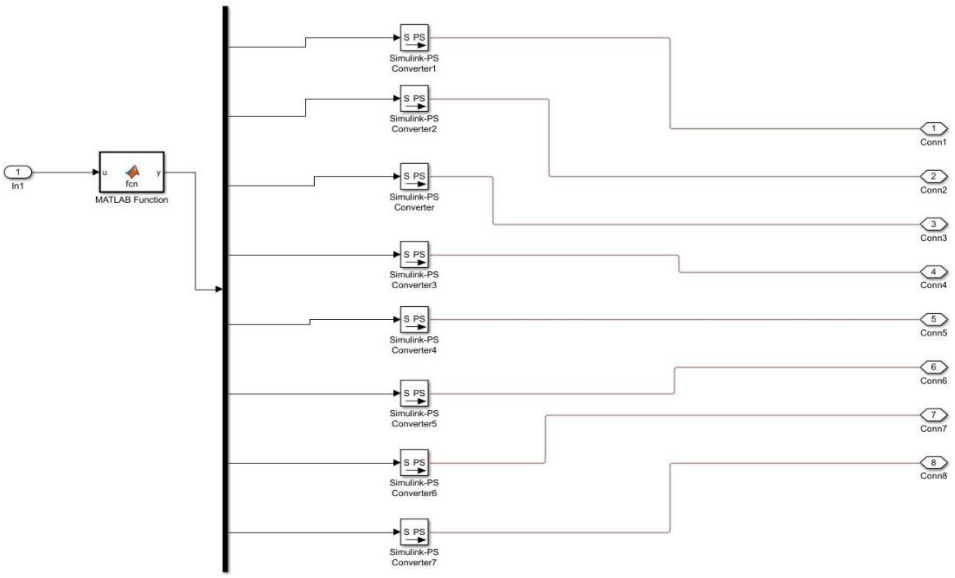


Figure [8] Forward Kinematics Subsystem for “Right Arm” Robot Arm

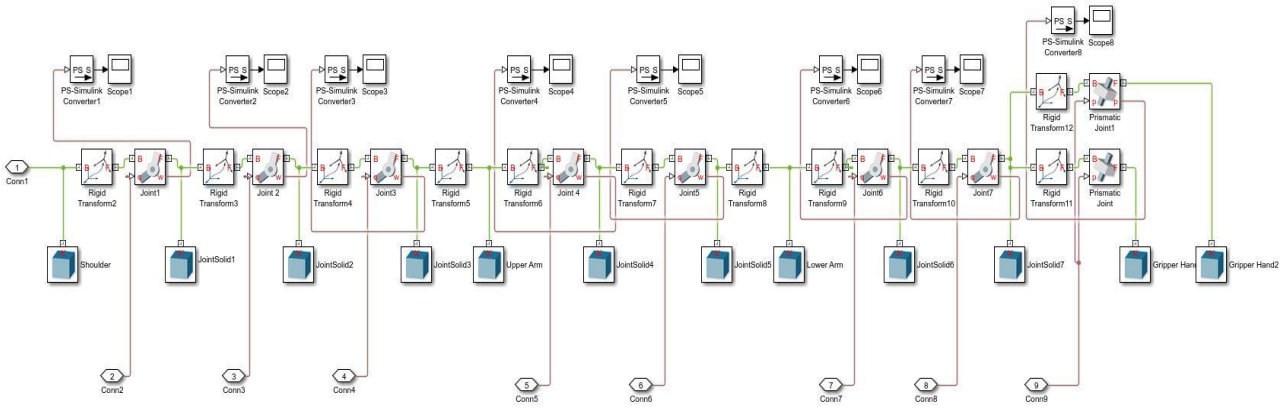


Figure [9] “Right Arm” Robot Arm Model Subsystem which contains the Constructed Model

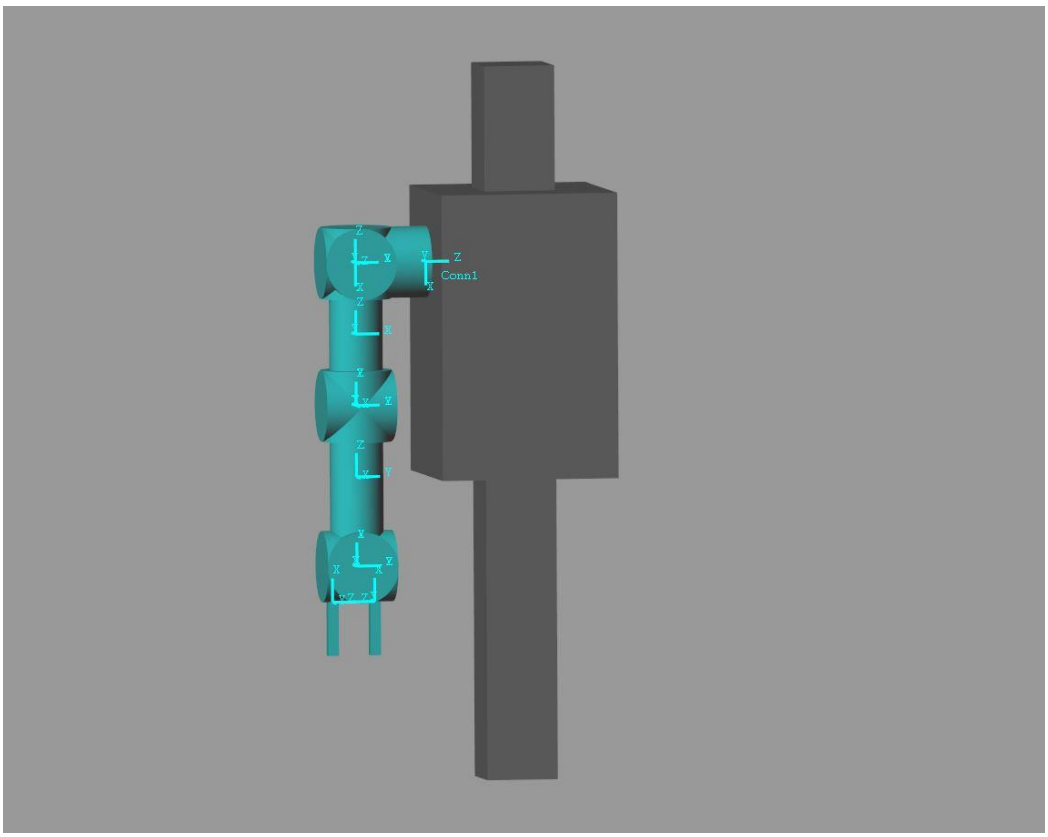


Figure [10] “Right Arm” Robot Arm Model during Simulation.

V. FORWARD KINEMATICS

The forward kinematics is computed entirely in MATLAB. All the code is commented, and instructions given to run it. Firstly, the 4x4 Transformation matrices are computed for every transformation and then the final transformation matrix is T_7^0 in this case, as there are seven joint Parameters. Due to the symbolic computation of the Transformation being very large, I have pasted the link to the Program in the 'Code' Section of the Project.

The symbolic computation of the forward kinematics is not only necessary to specify the end-effector pose and position, but is also important for the computation of the inverse kinematics.

VI. INVERSE KINEMATICS

To compute the inverse kinematics, I first need a desired position and pose to which I drive the Manipulator from its "Home" configuration (All angles are set to 0). Most the of the times in real life, finding the orientation is a very difficult task, so only the position is required as input, and we must derive the orientation. Adding on to the fact that the manipulator is kinematically redundant, and there might be many (if not infinite) poses possible for the same point

The mathematical approach I have used is as follows:

1.)Find the Position Vector for the Home Configuration. Set the normalized vector as Default "Home".

2.)Find the Position Vector for the Desired Point. Set the Normalized Vector as "New Point".
3.)Find the Rotation Matrix between the two vectors to get a pose.
4.)Update the *known* Transformation Matrix with the pose. This Transformation matrix is completely numerical; the pose is now known, and the end-effector points *x, y, z* are the desired points of the end-effector.
5.)Equate with the Symbolic Transformation Matrix found during the Forward Kinematics.
6.)There are Twelve Equations in Seven Variables to be solved. A Nonlinear Simultaneous Functional Equation solver can be used to get the parameter vector "theta" which includes all the desired angles for every joint (θ_1 ... ... θ_7).
7.) After solving, check that the parameter vector "theta" isn't out of bounds by comparing with the desired joint constraints. If it is out of bounds, the end-effector cannot be reached with that pose.

With this approach, I have met with partial success.The Code is given at the end.

VII. MANIPULATOR JACOBIAN

The Manipulator Jacobian is found to mathematically model the end-effector velocity. After finding the Forward Transformation Matrices, a little manipulation leads to the Jacobian for the 7-Dof Robot Manipulator Arm.

The Code is given at the end and is properly commented. The Manipulator Jacobian of the 'Right-Arm" Anthropo-morphic

Human Arm is as follows (As it is completely made with revolute joints):

$$Jacobian = \begin{bmatrix} J_{v1} & J_{v2} & J_{v3} & J_{v4} & J_{v5} & J_{v6} & J_{v7} \\ Z_0 & Z_1 & Z_2 & Z_3 & Z_4 & Z_5 & Z_6 \end{bmatrix}$$

where $J_{v(i)} = (Z_i - 1)\, x(O_n - O_i - 1),\ n = 7, i = (1,7)$

and $J_{v(i)}$ and Z_i are both *3x1* matrices.

VIII. DEMONSTRATIONS

1. Testing the Forward Kinematics with the following Angles:
   `[-3.14/6 0 3.14/6 -3.14/2 0 -3.14/6 -3.14/6 0.03]`
   https://www.youtube.com/watch?v=yVphvLmn5FY&feature=youtu.be

2. Testing the Inverse Kinematics for the following Point:
   `Point (-21.5, 64, 35.5)`
   https://www.youtube.com/watch?v=II-URnMx6oI&feature=youtu.be

3. Future Scope: Forward Kinematics for a 2-Armed Body
   https://www.youtube.com/watch?v=EJhMVJWveY0&feature=youtu.be

4. Drive Link for **Code**: -
   https://drive.google.com/drive/folders/1w5l5kHBWqtwYPhdffzMlgYgPCN8w36Qt?usp=sharing
   https://drive.google.com/open?id=1BPgvuhHa3oukuLLIuPPi1VXfVyS-AWBr

IX. CONCLUSION AND FUTURE WORK

The Robotic Arm is accurately modeled according to the DH-Parameters as given above. The forward and inverse kinematics are calculated, and implemented.

Since some of the functions used in the inverse kinematics are not supported for code generation in Simulink, I wasn't able to establish a Functional Block to directly give the angles as inputs to the Robot Arm. It is to be run as a '.m' file, the theta values manually taken and plugged into the Forward Kinematic Block to take us to the desired point.

For the end-effector velocities, the data can be collected from the scopes attached to every joint. Due to the fact that each graph exhibits a constant velocity, I have given only one for reference (The rest can be found by running the program):

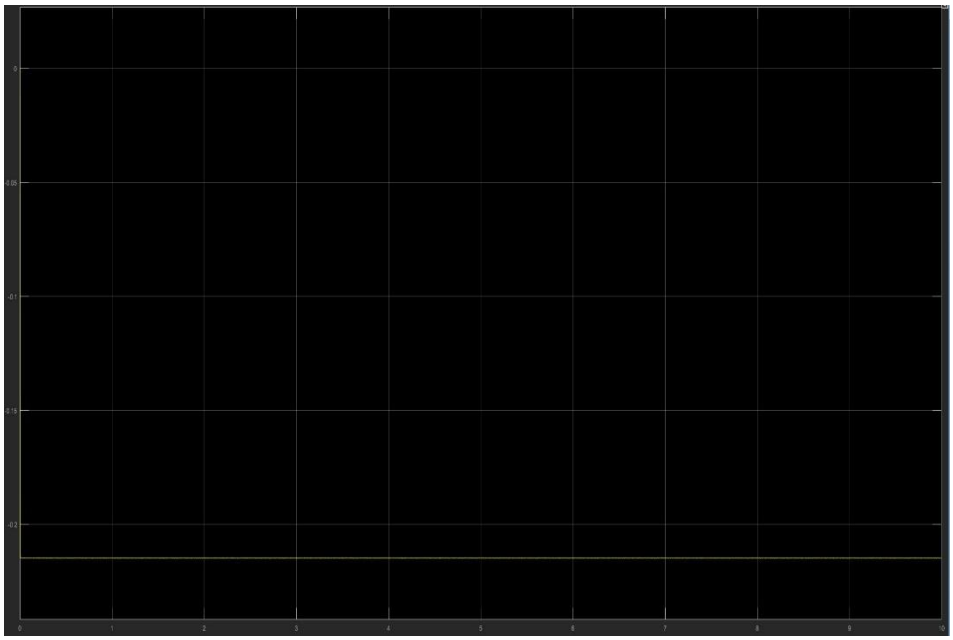


Figure [11] Joint 1 Scope Data- Constant Vel. At approx. 0.17

Future work is to make a fully-functional 2-armed robot with a mobile platform at the bottom containing a differential drive to give it mobility, navigating terrain with path planning, and trajectory planning aspects.

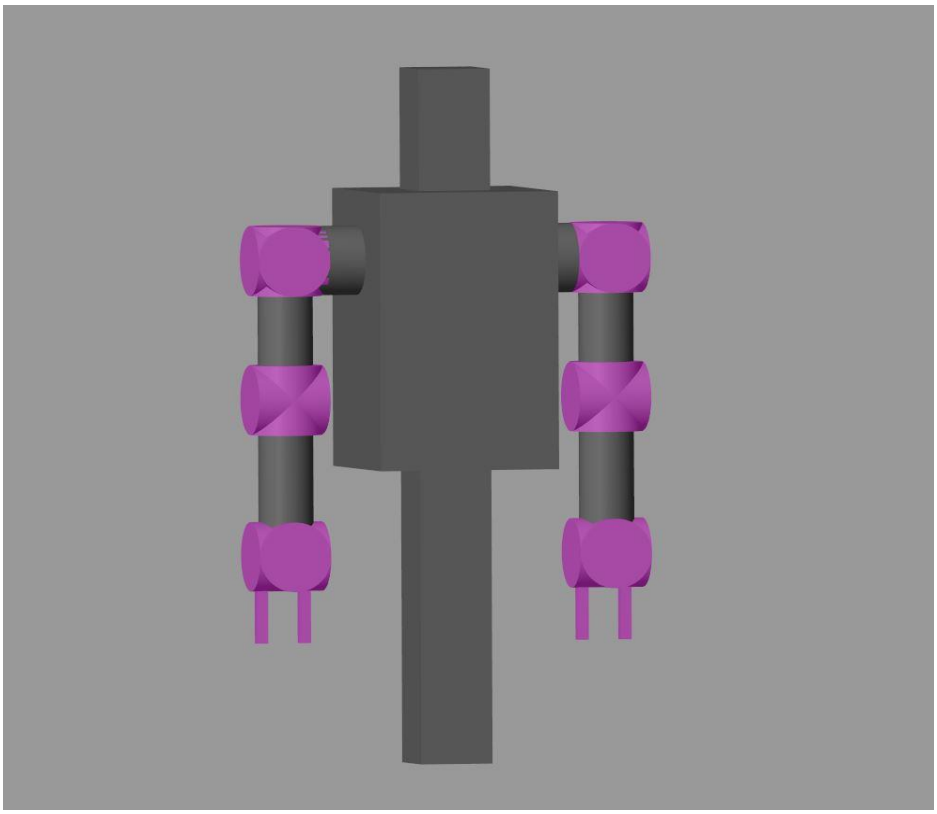


Figure [12] Two-Armed Primitive Block Humanoid

Other major aspirations include attaching vision sensors, and give the model some sort of learning aspects. Further research is required to even decide what kind of goals I would like it to accomplish.
There are many humanoid robots manufactured, but the general idea is to design a cost-effective general-purpose robot.

A highly optimistic analogy of what I would like to design is something like the following:

Figure[13] Mitra from Invento Robotics [5]

References

[1] Ismail et al "Robot-Based Intervention Program for Autistic Children with Humanoid Robot NAO: Initial Response in Stereotyped Behavior", Procedia Engineering, Volume 41, Pg 1441-1447, 2012

[2] Diftler et al "Robonaut 2- the first humanoid robot in space" Robotics and Automation (ICRA), 2011 IEEE International Conference Pgs. 2178-2183, 2011/5/9

[3] Wang Y, Artemiadis P (2013) Closed-Form Inverse Kinematic Solution for Anthropomorphic Motion in Redundant Robot Arms. Adv Robot Autom 2:110. doi:10.4172/advances-robotics-automation

[4] http://petercorke.com/wordpress/toolboxes/robotics-toolbox

[5] https://www.mitrarobot.com/

[6] MATLAB Documentation

[7] Robot Dynamics and Control, Spong. M, Vidyasagar. M