University at Buffalo

MAE 493/593

Math Methods in Robotics

Analysis of a 4RPR Manipulator

Students:
P. Davis
R. Nagasamudram

Professor:
Dr. V. Krovi

Contents

1 Introduction

In this project, our goal is to simulate the dynamics of a planar 4RPR Parallel-chain Manipulator computationally using the help of MATLAB. The user would be able to choose the 3 active joints of the system and specifies the size of the rectangular platform of which the midpoint is the end-effector. The user also inputs the joint constraints and can analyze the work space and manipulability of a given configuration. Furthermore, a control scheme is employed which tracks a circle or ellipse around the point (0,0) from a point of user's choice. The user can specify the gain and simulation time of the control scheme. In the real world, such a system can be used for a broad variety of applications including CNC machines, robotic test simulators and other industrial equipment.

2 Background and Theory

In this report, we study the planar parallel kinematic manipulator shown below. Each kinematic chain has a revolute, prismatic and revolute joint. We consider the situation where we have four kinematic chains (a 4RPR system).

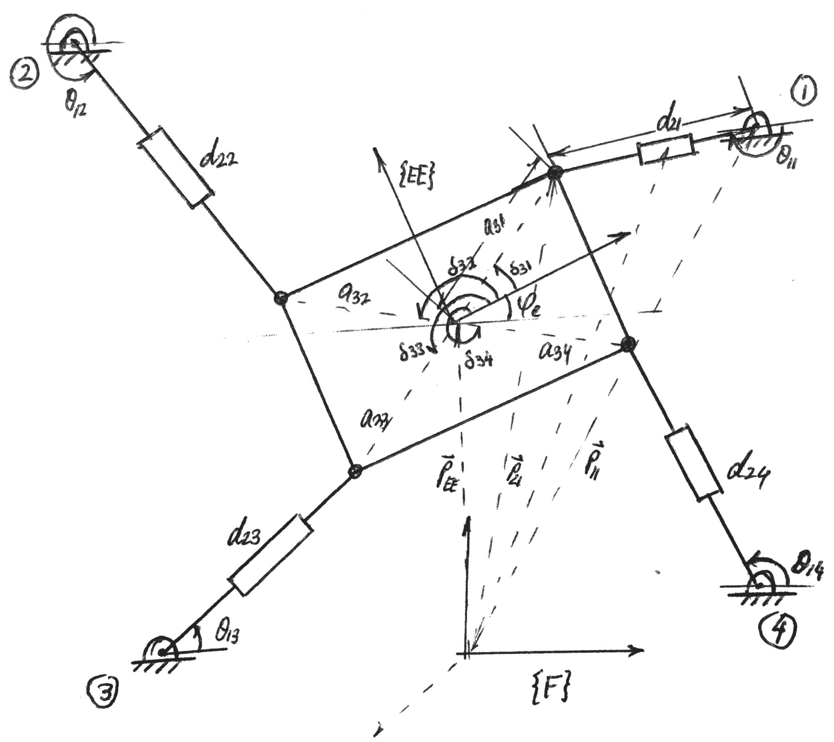


Figure 1: 4RPR Manipulator

{F} is any fixed reference frame. There are four grounded joints, one for each chain (numbered 1, 2, 3, and 4) and the position vectors to the first chain from the fixed frame {F} are shown. The end effector frame of reference is the frame {EE}. φ_e is the orientation of the end effector platform. The position vectors from the fixed frame, {F} are shown in the picture. The various joint variables in the system are,

Revolute	:	$\theta_{11}\ \theta_{12}\ \theta_{13}\ \theta_{14}$
Prismatic	:	$d_{21}\ d_{22}\ d_{23}\ d_{24}$

3 Mathematical Methods

We first consider the degrees of freedom of the 4RPR manipulator. The degrees of freedom for a planar manipulator is given by

$$M = 3(n-1) - 2j_1 - j_2 \tag{1}$$

For the case of the 4RPR manipulator, we have,

$$M = 3(9) - 2(12) = 3\text{d.o.f} \tag{2}$$

In order to have full control of the manipulator, we need to have 3 *active* joints. Any more would lead to redundancy in the system and any less would lead to a situation where the system cannot be fully controlled.

3.1 Kinematic Modeling

Modeling the kinematics of such a planar manipulator can be complex.

3.1.1 Forward Pose Kinematics

The solution to the forward requires a mapping,

$$\psi \colon \boldsymbol{J} \mapsto \boldsymbol{T}$$

where $\boldsymbol{J}$ is the joint space of the manipulator and $\boldsymbol{T}$ is the task space of the manipulator.
We use the loop closure method to derive the equations for the forward kinematic analysis. Consider the point $\vec{P_{21}}$
We have the equality,

$$\vec{P_{21}} = \vec{P_{11}} + \begin{bmatrix} d_{21}\cos\theta_{11} \\ d_{21}\sin\theta_{11} \end{bmatrix} = \vec{P_{ee}} + \begin{bmatrix} a_{31}\cos(\varphi_e + \delta_{31}) \\ a_{31}\sin(\varphi_e + \delta_{31}) \end{bmatrix} \tag{3}$$

where,

$$\vec{P_{11}} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

and

$$\vec{P_{ee}} = \begin{bmatrix} x_e \\ y_e \end{bmatrix}$$

In general, for the j^{th} chain, we have,

$$\begin{bmatrix} x_j \\ y_j \end{bmatrix} + \begin{bmatrix} d_{2j}\cos\theta_{1j} \\ d_{2j}\sin\theta_{1j} \end{bmatrix} = \begin{bmatrix} x_e \\ y_e \end{bmatrix} + \begin{bmatrix} a_{3j}\cos(\varphi_e + \delta_{3j}) \\ a_{3j}\sin(\varphi_e + \delta_{3j}) \end{bmatrix} \tag{4}$$

The full set of equations we arrive at are,

$$
\begin{aligned}
x_1 + d_{21}\cos(\theta_{11}) &= x_e + a_{31}\cos(\varphi_e + \delta_{31}) \\
y_1 + d_{21}\sin(\theta_{11}) &= y_e + a_{31}\cos(\varphi_e + \delta_{31}) \\
x_2 + d_{22}\cos(\theta_{12}) &= x_e + a_{32}\cos(\varphi_e + \delta_{32}) \\
y_2 + d_{22}\sin(\theta_{12}) &= y_e + a_{32}\cos(\varphi_e + \delta_{32}) \\
x_3 + d_{23}\cos(\theta_{13}) &= x_e + a_{33}\cos(\varphi_e + \delta_{33}) \\
y_3 + d_{23}\sin(\theta_{13}) &= y_e + a_{33}\cos(\varphi_e + \delta_{33}) \\
x_4 + d_{24}\cos(\theta_{14}) &= x_e + a_{34}\cos(\varphi_e + \delta_{34}) \\
y_4 + d_{24}\sin(\theta_{14}) &= y_e + a_{34}\cos(\varphi_e + \delta_{34})
\end{aligned}
$$

These equations are highly non-linear and arriving at an analytical solution can be complicated. Numerical methods are used to arrive at solutions for these equations. The `MATLAB` command, `fsolve` will be used to solve for the forward kinematics problem.

3.1.2 Inverse Pose Kinematics

The solution to inverse kinematics problem requires a mapping,

$$\vartheta : \boldsymbol{T} \mapsto \boldsymbol{J}$$

where $\boldsymbol{T}$ is the task space of the manipulator and $\boldsymbol{J}$ is the joint space of the manipulator.
The equations for the inverse kinematics problem can be derived analytically, as will be demonstrated in this section
In general, the equations for the j^{th} joint are,

$$
\begin{aligned}
x_j + d_{2j}\cos(\theta_{1j}) &= x_e + a_{3j}\cos(\varphi_e + \delta_{3j}) \qquad (5) \\
y_j + d_{2j}\sin(\theta_{1j}) &= y_e + a_{3j}\sin(\varphi_e + \delta_{3j}) \qquad (6)
\end{aligned}
$$

These can be written in the form,

$$
\begin{aligned}
d_{2j}\cos(\theta_{1j}) &= x_e + a_{3j}\cos(\varphi_e + \delta_{3j}) - x_j \\
d_{2j}\sin(\theta_{1j}) &= y_e + a_{3j}\sin(\varphi_e + \delta_{3j}) - y_j
\end{aligned}
$$

Dividing one by the other, we have,

$$\theta_{1j} = \tan^{-1}\left(\frac{y_e + a_{3j}\sin(\varphi_e + \delta_{3j}) - y_j}{x_e + a_{3j}\cos(\varphi_e + \delta_{3j}) - x_j}\right) \qquad (7)$$

Squaring both equations and adding we have,

$$d_{2j} = \pm\sqrt{(x_e + a_{3j}\cos(\varphi_e + \delta_{3j}) - x_j)^2 + (y_e + a_{3j}\sin(\varphi_e + \delta_{3j}) - y_j)^2} \qquad (8)$$

3.2 Jacobian Matrix Evaluation

Consider the equations derived from the forward kinematic analysis.

$$\begin{bmatrix} x_e \\ y_e \end{bmatrix} + \begin{bmatrix} a_{3j}\cos(\varphi_e + \delta_{3j}) \\ a_{3j}\sin(\varphi_e + \delta_{3j}) \end{bmatrix} = \begin{bmatrix} x_j \\ y_j \end{bmatrix} + \begin{bmatrix} d_{2j}\cos\theta_{1j} \\ d_{2j}\sin\theta_{1j} \end{bmatrix}$$

Differentiating the set of equations above, we have,

$$\begin{bmatrix} A \end{bmatrix} \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\varphi}_e \end{bmatrix} = \begin{bmatrix} B \end{bmatrix} \begin{bmatrix} \dot{\theta_{11}} \\ \dot{\theta_{12}} \\ \dot{\theta_{13}} \\ \dot{\theta_{14}} \\ \dot{d_{21}} \\ \dot{d_{22}} \\ \dot{d_{23}} \\ \dot{d_{24}} \end{bmatrix} \tag{9}$$

where,

$$\begin{bmatrix} A \end{bmatrix} = \begin{bmatrix} 1 & 0 & -a_{31}\sin(\varphi_e + \delta_{31}) \\ 1 & 0 & -a_{32}\sin(\varphi_e + \delta_{32}) \\ 1 & 0 & -a_{33}\sin(\varphi_e + \delta_{33}) \\ 1 & 0 & -a_{34}\sin(\varphi_e + \delta_{34}) \\ 0 & 1 & a_{31}\cos(\varphi_e + \delta_{31}) \\ 0 & 1 & a_{32}\cos(\varphi_e + \delta_{32}) \\ 0 & 1 & a_{33}\cos(\varphi_e + \delta_{33}) \\ 0 & 1 & a_{34}\cos(\varphi_e + \delta_{34}) \end{bmatrix}$$

and,

$$\begin{bmatrix} B \end{bmatrix} = \begin{bmatrix} -d_{21}s_{11} & 0 & 0 & 0 & c_{11} & 0 & 0 & 0 \\ 0 & -d_{22}s_{12} & 0 & 0 & 0 & c_{12} & 0 & 0 \\ 0 & 0 & -d_{23}s_{13} & 0 & 0 & 0 & c_{13} & 0 \\ 0 & 0 & 0 & -d_{24}s_{14} & 0 & 0 & 0 & c_{14} \\ d_{21}c_{11} & 0 & 0 & 0 & s_{11} & 0 & 0 & 0 \\ 0 & d_{22}c_{12} & 0 & 0 & 0 & s_{12} & 0 & 0 \\ 0 & 0 & d_{23}c_{13} & 0 & 0 & 0 & s_{13} & 0 \\ 0 & 0 & 0 & d_{24}c_{14} & 0 & 0 & 0 & s_{14} \end{bmatrix}$$

Here, s_{1j} is $\sin\theta_{1j}$ and c_{1j} is $\cos\theta_{1j}$
The equation mentioned previously is of the form,

$$A\dot{\underset{\sim}{X}} = B\dot{\underset{\sim}{\theta}} \tag{10}$$

This gives,

$$\dot{\underset{\sim}{X}} = A^{-1}B\dot{\underset{\sim}{\theta}} \tag{11}$$

where,

$$J = A^{\#}B \tag{12}$$

is the Jacobian of the manipulator. This isn't solved for analytically at this point though it can be done. The resulting solution for J is too large to be displayed here. Note that $A^{\#}$ is the pseudo inverse of A.

3.3 Manipulability

The manipulability of a kinematic structure is the degree to which the structure can be manipulated. Manipulability measures provide information regarding the quality of the kinematic structure. In this report we provide measures of manipulability by using Yoshikawa's Index for manipulability and by using the Condition number Index for manipulability.
Recall the Jacobian of the manipulator.

$$\dot{\underset{\sim}{X}} = J\dot{\underset{\sim}{\theta}}$$

Note that the Jacobian of the manipulator is not a constant, but depends on the joint variables of the manipulator at a given position in the task space. Thus,

$$J = J(\theta)$$

3.3.1 Yoshikawa Manipulability

The manipulability of the kinematic structure at state θ, w is given by,

$$w = \sqrt{|J(\theta)J^T(\theta)|} \tag{13}$$

One way to visualize the manipulability across all states of the kinematic structure would be construct a surface plot of the manipulability variable w across the entire workspace of the kinematic structure. (The workspace of the kinematic structure or the manipulator in this case, is set of all points in the task space of the manipulator)

3.3.2 Isotropic Manipulability

The manipulability of the kinematic structure at state θ, w is given by,

$$w = \sqrt{1/cond(J(\theta))} \tag{14}$$

where σ is the condition number of the matrix $J(\theta)$. The condition number, σ of a matrix, M, is the ratio of the largest to the smallest singular value in the singular value decomposition of the matrix, M.

3.4 Redundancy

The problem requires us to constrain the manipulator to possess only two degrees of freedom instead of three. One way to reduce the degrees of freedom in the 4RPR manipulator would be to fix the orientation of the end effector platform to a constant value. Given the scope of this analysis we have chosen to fix the orientation of the platform to 0^o w.r.t the horizontal. This is done in the velocity level by changing a few of the terms in the Jacobian of the manipulator. Recall that the Jacobian of the manipulator is,

$$J = A^{\#}B$$

where, $A^{\#}$ is the pseudoinverse of A. We have,

$$\dot{\underset{\sim}{X}} = J\dot{\underset{\sim}{\theta}} \implies \dot{\underset{\sim}{\theta}} = J^{-1}\dot{\underset{\sim}{X}} = B^{-1}A\dot{\underset{\sim}{X}}$$

J^{-1} is an 8x3 matrix. Now recall that $\dot{\underset{\sim}{X}} = [x_e,\ y_e,\ \varphi_e]^T$. Setting the third column of J^{-1} to 0 would constrain the end effector platform to a constant orientation of 0^o.

3.5 Manipulator Control

There are a few ways a kinematic structure can be manipulated and controlled. We look at open loop and closed loop control schemes that will be implemented on the 4RPR planar manipulator.

3.5.1 Open Loop Control

Open loop control has no error term and takes in no feedback from the system. The output is entirely based on the current state of the system. Open loop control for this manipulator is given by the equation

$$\dot{\underset{\sim}{\theta}} = J^{-1}\dot{\underset{\sim}{X}} \tag{15}$$

3.5.2 Closed Loop Control

Closed loop control involves feedback from the system and requires an error term to be added to the equations. We look at closed loop control in the task space and joint space of the manipulator.
The equation for task space based closed loop control is,

$$\dot{\underset{\sim}{\theta}}^{task} = J^{-1}\left(\dot{\underset{\sim}{X}} + K\left(X^d - X^a\right)\right) \tag{16}$$

$\dot{\underset{\sim}{\theta}}^{task}$ is the values for $\dot{\underset{\sim}{\theta}}$ under task space based control. K is the gain of the controller, X^d is a vector containing the desired values of position and X^a is a vector containing the actual values of position.
The equation for joint space based closed loop control is,

$$\dot{\underset{\sim}{\theta}}^{joint} = J^{-1}\dot{\underset{\sim}{X}} + K\left(\theta^d - \theta^a\right) \tag{17}$$

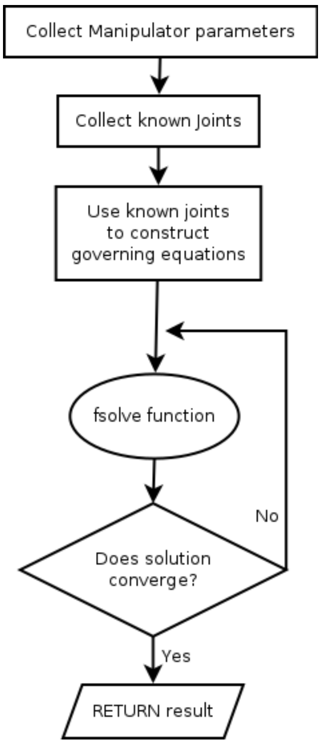
$\dot{\underset{\sim}{\theta}}^{joint}$ is the values for $\dot{\underset{\sim}{\theta}}$ under joint space based control. K is the gain of the controller, θ^d is a vector containing the desired values of all the joint variables and θ^a is a vector containing the actual values of the joint variables.

4 Implementation

4.1 Algorithms

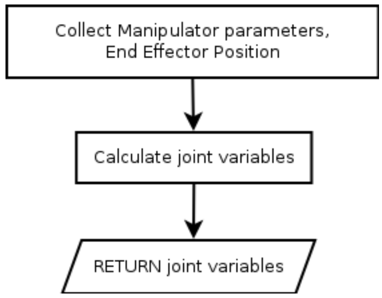
4.1.1 Forward Kinematics

`MATLAB` function - `fKinSolve`



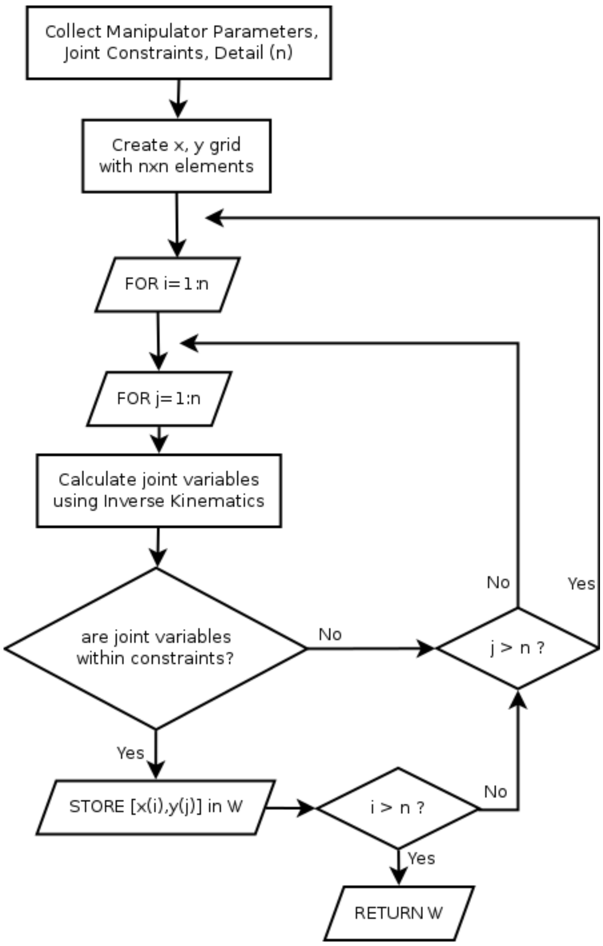
4.1.2 Inverse Kinematics

`MATLAB` function - `iKinSolve`



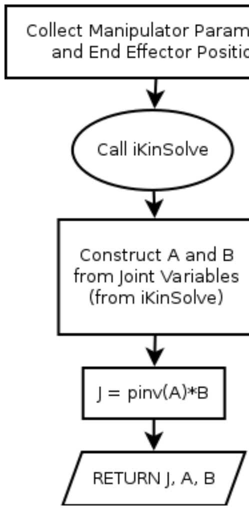
4.1.3 Workspace construction

`MATLAB` function - `makeWorkspace`



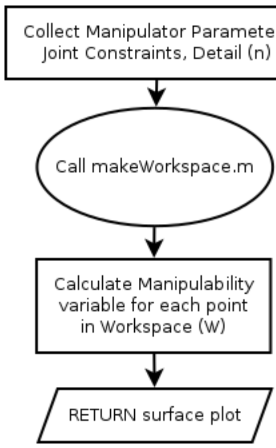
4.1.4 Jacobian Construction

`MATLAB` function - `giveJacobian`



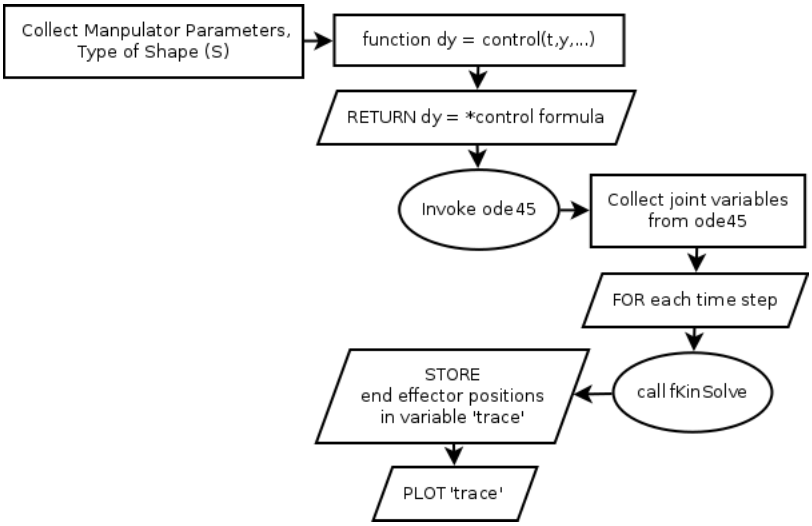
4.1.5 Yoshikawa's/Condition Number Manipulability Index

`MATLAB` function - `yoshikawaManipulability, isentropicManipulability`



4.1.6 Control Schemes

`MATLAB` function - `doControl`



4.2 Development of `MATLAB` program and GUI

`GUIDE` was used to develop a GUI application. The `MATLAB` GUI application helps to visualize various facets of the kinematic manipulator discussed in this report.

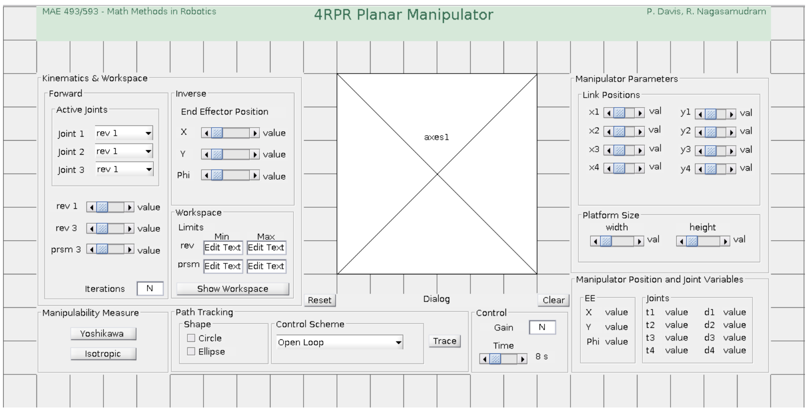


Figure 2: GUIDE GUI Layout - 4RPR Manpiulator

5 Results

The figure below shows the start up screen of the GUI.

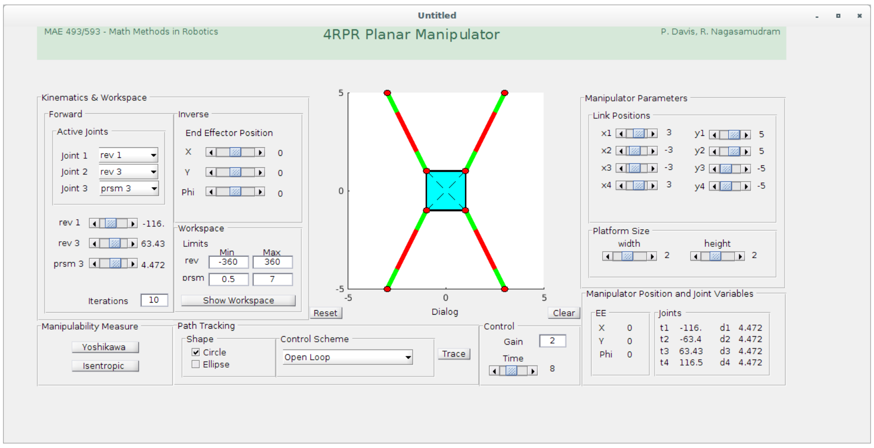


Figure 3: GUI - Start Up; Default Parameters

The layout of the GUI is relatively straightforward. The axes in the middle of the GUI module shows an image of the 4RPR manipulator. The module on the right side controls the manipulator parameters and the module on the bottom right displays relevant information regarding the joint variables and the end effector position. The user is free to change the parameters as he/she wishes. The modules on the left deal with the forward, and inverse kinematics problem. The module right below deals with the workspace analysis part of this undertaking. There are radio buttons - 'Yoshikawa' and 'Isotropic' that show a surface plot of the manipulability variable w.r.t workspace. The control module gives the user the ability to compare open loop and closed loop control schemes. The figure below shows the workspace of the manipulator (default orienation)

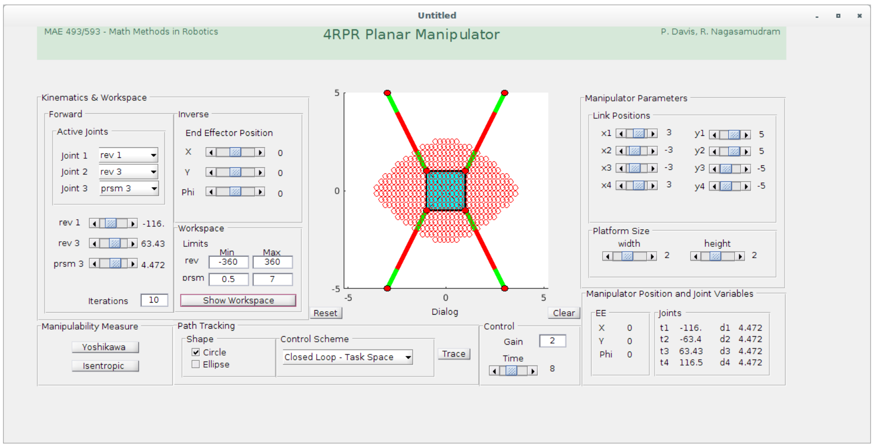
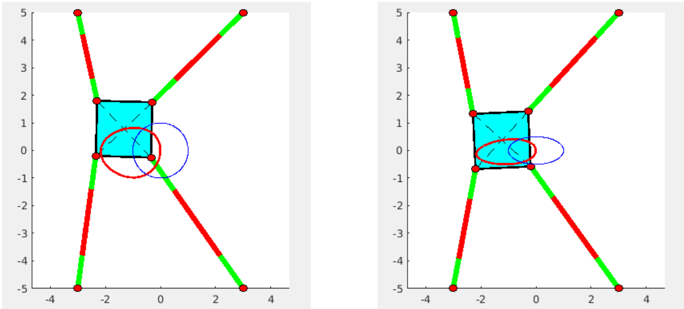


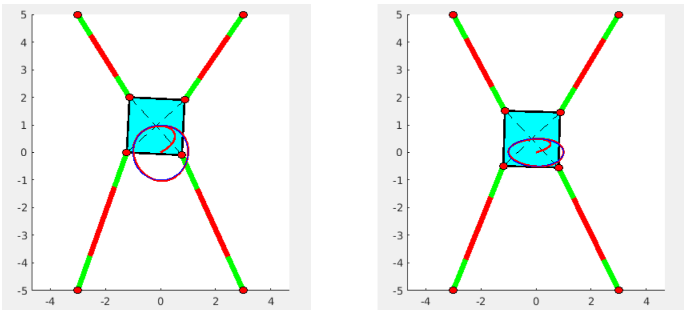
Figure 4: Workspace of Manipulator - Default Parameters

The next few figures display the output of the open loop and closed loop control schemes implemented for the path tracking problem on the 4RPR manipulator.



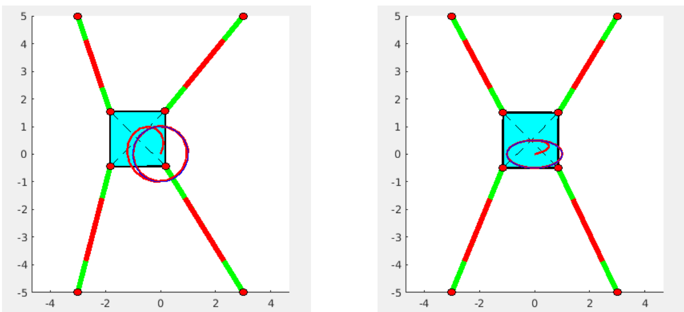
(a) Circle *(b) Ellipse*

Figure 5: Open Loop Control



(a) Circle, Gain = 2 *(b) Ellipse, Gain = 2*

Figure 6: Closed Loop Control - Joint Space



(a) Circle, Gain = 0.5 *(b) Ellipse, Gain = 2*

Figure 7: Closed Loop Control - Task Space

6 Discussion

All the functionalities of the simulation are explained above. The possible improvements for the future version would be:

- Path Tracking - The controller tracks a random path specified by the user.
- Path Control - User gets to specify the properties of different geometric shapes including where the center is and what the sizes would be
- Expansion to spatial case - Remove the constraint that the manipulator has to be planar
- Better user interface