Report

Assignment 1

Mikhail Ostanin, Stanislav Mikhel

Robot description

3-RRR is a planar parallel manipulator with 3 degrees of freedom (DOF). It consists of a movable platform with 3 legs. Each leg has 3 links and 3 joints, two of them are passive and one contains actuator. There are different locations of actuators, we are going to consider the case when active joints are connected to the base.

Kinematic scheme

Robot scheme is represented on figure 1. There are 2 coordinate frames, first one is global and second one is local, associated with movable platform (point C).

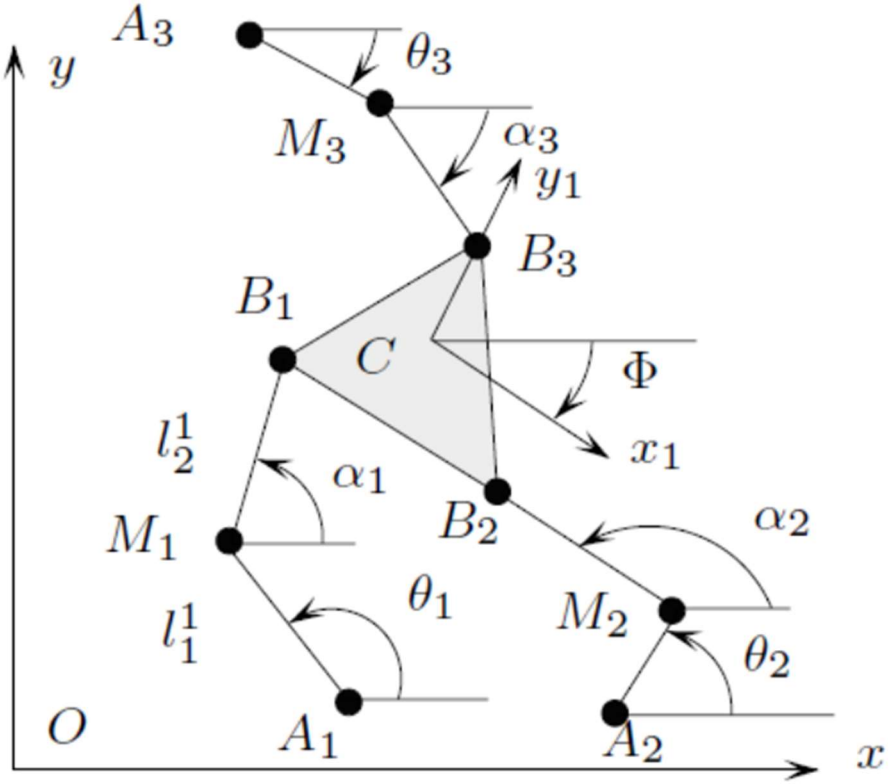


Figure 1. Robot scheme.

Inverse kinematics

Since this robot is parallel, inverse kinematics calculations are quiet simple. In local coordinate system of the platform position of each angle (point B_i) can be found as

$$x_i^C = CB_i \cdot cos(\gamma_i),\ y_i^C = CB_i \cdot sin(\gamma_i),$$

where $CB_i$and γ_i- radius-vector and orientation of the point i relatively to the local coordinate frame. In global coordinate frame position of each point B could be found using transformation matrix

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = T \cdot \begin{pmatrix} x^C \\ y^C \\ 1 \end{pmatrix}$$

Here we assume that matrix T contains both rotational and translational parts.

Each leg is a simple 2-link manipulator. Angle of the second joint αcan be found from equation

$$\alpha = acos\left(\frac{p_x^2 + p_y^2 - l_1^2 - l_2^2}{2 * l_1 * l_2}\right)$$

Here p_x, p_y- coordinates of the vector A_iB_i, l_1, l_2- length of the joints A_iM_i, $M_iB_i$correspondingly. Then actuator angle is

$$\theta = atan2\left(\frac{p_y}{p_x}\right) \pm acos\left(\frac{p_x^2 + p_y^2 - l_1^2 - l_2^2}{2 \cdot l_1\sqrt{p_x^2 + p_y^2}}\right)$$

Forward kinematics

In order to solve forward kinematics problem first of all we should find coordinates of the platform local frame in global coordinate system. For serial chain we get:

$$x = x_b + l_1cos(\theta) + l_2cos(\theta + \alpha) + l_ccos(\gamma + \phi)$$
$$y = y_b + l_1sin(\theta) + l_2sin(\theta + \alpha) + l_csin(\gamma + \phi)$$

Here l_c- length of vector BC, ϕ- orientation of the platform. Length $l_2$can be represented in the following form

$$l_2^2 = \left(x - x_b - l_1cos(\theta) - l_ccos(\gamma + \phi)\right)^2 + \left(y - y_b - l_1sin(\theta) - l_csin(\gamma + \phi)\right)^2$$

We can define function F:

$$F(x, y, \phi) = \left(x - x_b - l_1cos(\theta) - l_ccos(\gamma + \phi)\right)^2 + \left(y - y_b - l_1sin(\theta) - l_csin(\gamma + \phi)\right)^2 - l_2^2$$

which is equal to 0 when the configuration is correct. For given robot we need in 3 functions, one for each leg, so $F = [F_1F_2F_3]^T$. Platform position for the given joint angles can be found from the following algorithm.

1. Set initial position $Z_{old} = [x, y, \phi]^T$.
2. Find F.
3. Find Jacobian
$$J_F = \begin{bmatrix} dF_1/dx & dF_1/dy & dF_1/d\,\phi \\ dF_2/dx & dF_2/dy & dF_2/d\,\phi \\ dF_3/dx & dF_3/dy & dF_3/d\,\phi \end{bmatrix}$$
4. Find new position $Z_{new} = Z_{old} - J_F^{-1}F$.
5. If difference between $Z_{old}$and Z_{new}more then predefined tolerance, then go to step 2.

Computation of Jacobian

In order to find Jacobian, let's find derivatives of the equations of x and y.

$$\dot{x} = -l_1 sin(\theta)\dot{\theta} - l_2 sin(\theta + \alpha)\left(\dot{\theta} + \dot{\alpha}\right) - l_c sin(\gamma + \phi)\dot{\phi}$$
$$\dot{y} = l_1 cos(\theta)\dot{\theta} + l_2 cos(\theta + \alpha)\left(\dot{\theta} + \dot{\alpha}\right) + l_c cos(\gamma + \phi)\dot{\phi}$$

If we multiply both equations to trigonometrical functions from $\theta + \alpha$, add them and subtract one term, then the following equation could be obtained.

$$cos(\theta + \alpha)\dot{x} + sin(\theta + \alpha)\dot{y} - l_c sin\big((\theta + \alpha) - (\gamma + \phi)\big)\dot{\phi} = l_1 sin(\alpha)\dot{\theta}$$

Left part of this equation depends on Cartesian coordinates, while right – form joint values. In matrix form it can be written as

$$J_z \dot{Z} = J_\theta \dot{\Theta}$$

where

$$J_z = \begin{pmatrix} cos(\beta_1) & sin(\beta_1) & -l_c sin\big(\beta_1 - (\gamma_1 + \phi)\big) \\ cos(\beta_2) & sin(\beta_2) & -l_c sin\big(\beta_2 - (\gamma_2 + \phi)\big) \\ cos(\beta_3) & sin(\beta_3) & -l_c sin\big(\beta_3 - (\gamma + \phi)\big) \end{pmatrix},$$
$$J_\theta = \begin{pmatrix} l_1 sin(\alpha_1) & 0 & 0 \\ 0 & l_1 sin(\alpha_2) & 0 \\ 0 & 0 & l_1 sin(\alpha_3) \end{pmatrix},$$
$$\beta_i = \theta_i + \alpha_i$$

Since $\dot{\Theta} = J\dot{Z}$, the Jacobian is $J = J_\theta^{-1} J_z$.

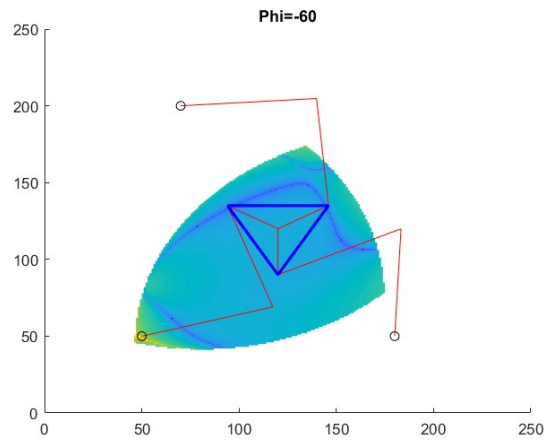
Singularity maps for robot workspace

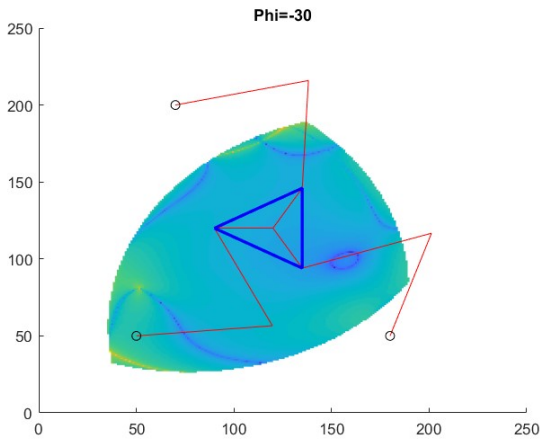
Singularity analyses were based on Jacobian analysis in specific end-effector position and orientation, formula:

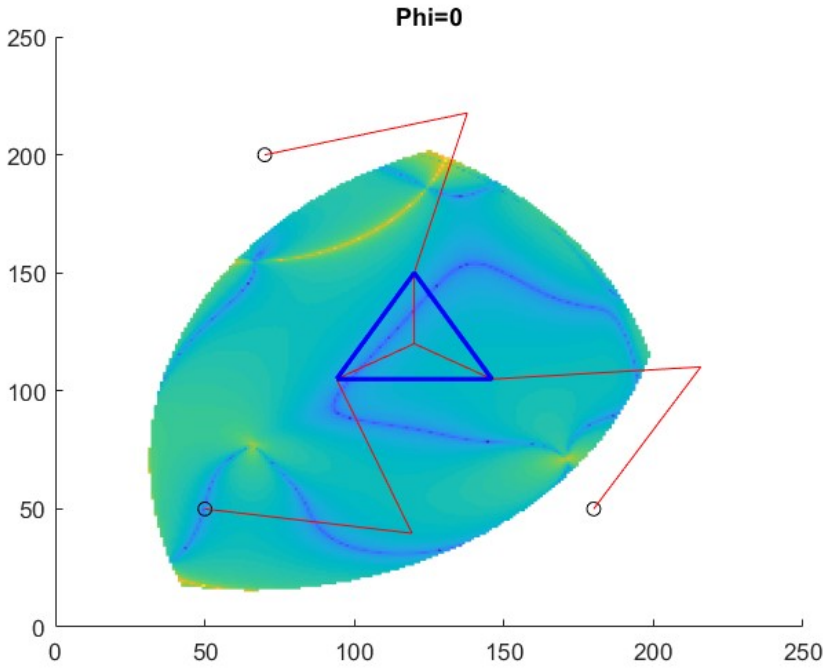
$$m = \sqrt{J \cdot J^T}$$

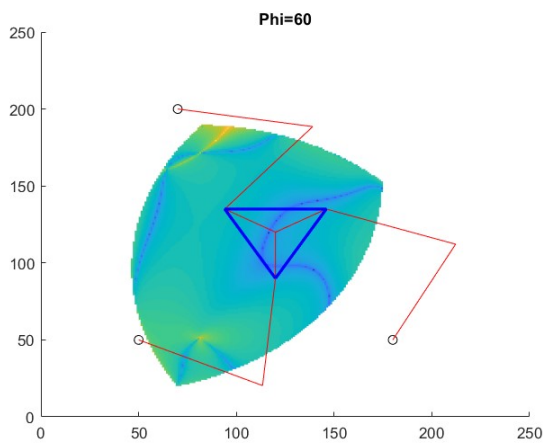
where m is manipulability index.

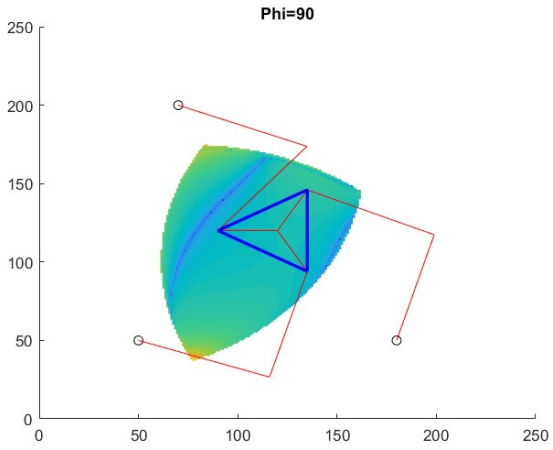
Figures below show singular maps for phi = (-60, -30, 0, 30, 60, 90), yellow parts correspond to singularity area.











Analysis of obtained results

Inverse kinematics of the 3RRR robot is quiet simple, since solution for each leg can be reduced to inverse kinematics of a 2 link manipulator. But the total number of possible solutions is 6, because each leg can be presented in 2 variants.

Forward kinematics doesn't have a closed form solution, but it can be obtained with the help of iteration algorithm. The result depends on initial conditions of the platform location. Different start positions can converge to the different robot configurations.

Jacobian can be obtained as a product of two matrices. First one performs mapping for Cartesian space, and second – for joint space.

As can be seen from the singularity map, workspace of the robot is close to ellipse. Main regions of singularity are located near the edges. Orientation of the platform also has influence on the size of workspace and locations of singularities.

Summary

In this work was obtained a model of 3RRR. This model allows to solve forward and inverse kinematic problems. Jacobian of the robot was calculated as well. Singularity map was built for different platform positions and orientations.