# Homework Project 2

# Kinematics, Dynamic and Control of Robots

## Table of Contents

# 1. Inverse Kinematics

The 3DOF Parallel robot given in Figure 1, has the joint variables vector denoted by $\boldsymbol{q} = [d_1, d_2, d_3]^T$, which resembles the position of prismatic joints, and a task vector denoted by $\boldsymbol{x} = [x, y, \theta]^T$, which indicates the position and orientation of the triangular platform in the plane.

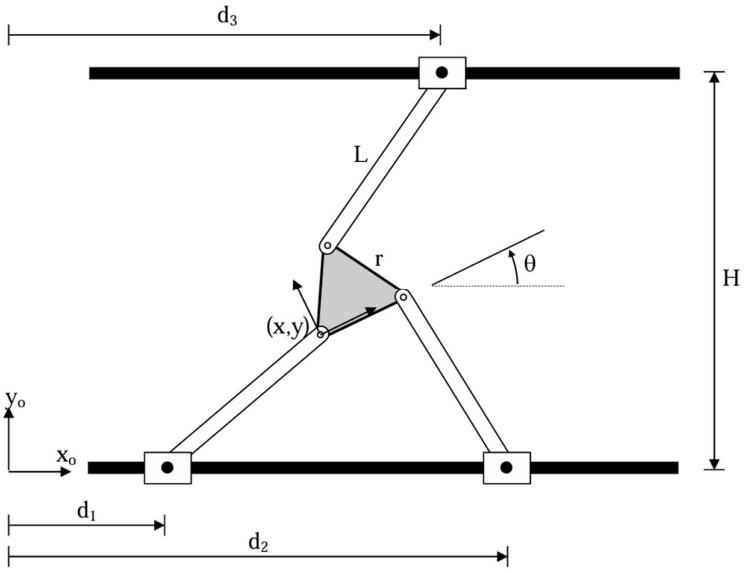



*Figure 1 - Drawing of a 3DOF parallel robot analyzed in this project*

The Inverse Kinematics problem is to find the joint vector $\boldsymbol{q}$, given a task vector $\boldsymbol{x}$.

Given the position and orientation of the platform, it would be wise to begin by finding the vertices of the platform. We will denote the three vertices by $\{(x_i, y_i) \mid i = 1,2,3\}$, where the reference frame is positioned at $(x_1, y_1)$ and the right vertex is $(x_2, y_2)$. They are given as follows

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}; \quad \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} x + r\cos\theta \\ y + r\sin\theta \end{pmatrix}; \quad \begin{pmatrix} x_3 \\ y_3 \end{pmatrix} = \begin{pmatrix} x + r\cos(\theta + \pi/3) \\ y + r\sin(\theta + \pi/3) \end{pmatrix}; \tag{1.1}$$

Applying the Pythagorean Theorem between the prismatic joint positions and the given points in (1.1) yield the equations:

$$\begin{aligned} L^2 &= (x_1 - d_1)^2 + y_1^2 \\ L^2 &= (x_2 - d_2)^2 + y_2^2 \\ L^2 &= (x_3 - d_3)^2 + (y_3 - H)^2 \end{aligned} \tag{1.2}$$

Where $r$, $L$ and $H$ are given lengths of the robot as shown in Figure 1.

Substituting the relations in (1.1) into (1.2) results in:

$$\begin{aligned} L^2 &= (x - d_1)^2 + y^2 \\ L^2 &= (x + r\cos\theta - d_2)^2 + (y + r\sin\theta)^2 \\ L^2 &= (x + r\cos(\theta + \pi/3) - d_3)^2 + (y + r\sin(\theta + \pi/3) - H)^2 \end{aligned} \tag{1.3}$$

Solving equations (1.3) for $\boldsymbol{q}$ yields the following solutions:

$$d_1 = \begin{cases} x + \sqrt{L^2 - \mathrm{y}^2} \\ x - \sqrt{L^2 - \mathrm{y}^2} \end{cases} \tag{1.4}$$

$$d_2 = \begin{cases} x + r\cos(\theta) + \sqrt{L^2 - \left(\mathrm{y} + r\sin(\theta)\right)^2} \\ x + r\cos(\theta) - \sqrt{L^2 - \left(y + r\sin(\theta)\right)^2} \end{cases} \tag{1.5}$$

$$d_1 = \begin{cases} x + \sqrt{L^2 - \left(y - H + r\sin\left(\theta + \frac{\pi}{3}\right)\right)^2} + r\cos\left(\theta + \frac{\pi}{3}\right) \\ x - \sqrt{L^2 - \left(y - H + r\sin\left(\theta + \frac{\pi}{3}\right)\right)^2} + r\cos\left(\theta + \frac{\pi}{3}\right) \end{cases} \quad (1.6)$$

There is a total of eight solutions as each joint parameter has two solutions. The solution tree is given in Figure 2:

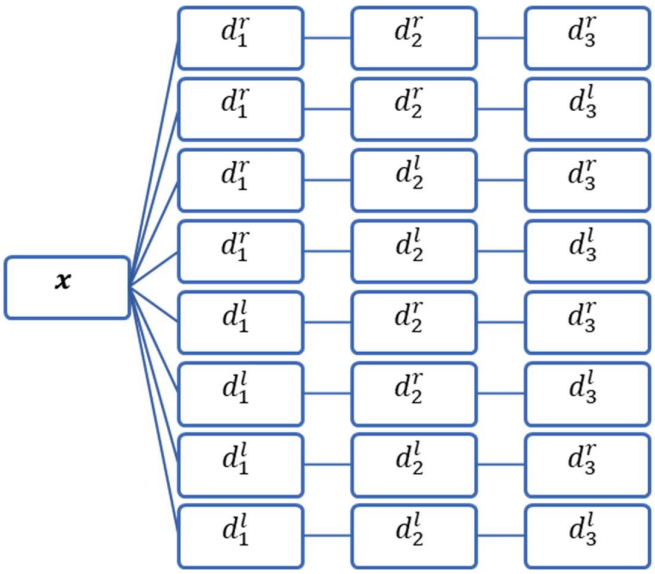



*Figure 2 - Solution tree of the Inverse Kinematics problem. All three parameters can be found at the same time, so there is no meaning for the order in the tree. It indicates that there are eight solutions of joint values for a given task.*

Following are the solutions for the given task $\boldsymbol{x}_0 = [2, 1, 45°]^T$, by using (1.4), (1.5), and (1.6) in MATLAB.

$$\boldsymbol{q}_i = \begin{bmatrix} 3.7321 & 3.7321 & 3.7321 & 3.7321 & 0.2679 & 0.2679 & 0.2679 & 0.2679 \\ 3.7491 & 3.7491 & 1.6651 & 1.6651 & 3.7491 & 3.7491 & 1.6651 & 1.6651 \\ 3.4531 & 0.0293 & 3.4531 & 0.0293 & 3.4531 & 0.0293 & 3.4531 & 0.0293 \end{bmatrix} \quad (1.7)$$

Where every column in (1.7) corresponds to a single solution in the solution tree. The robot has been plotted for every solution for $\boldsymbol{q}$ given the same task $\boldsymbol{x}_0$, and the solutions are displayed in Figure 3 in the next page. The larger values for $d_i$ correspond the "right" solutions, and the lower values correspond to the "left" solutions. The platform seems to remain fixed in its task $\boldsymbol{x}_0$ for all joint solutions as expected.

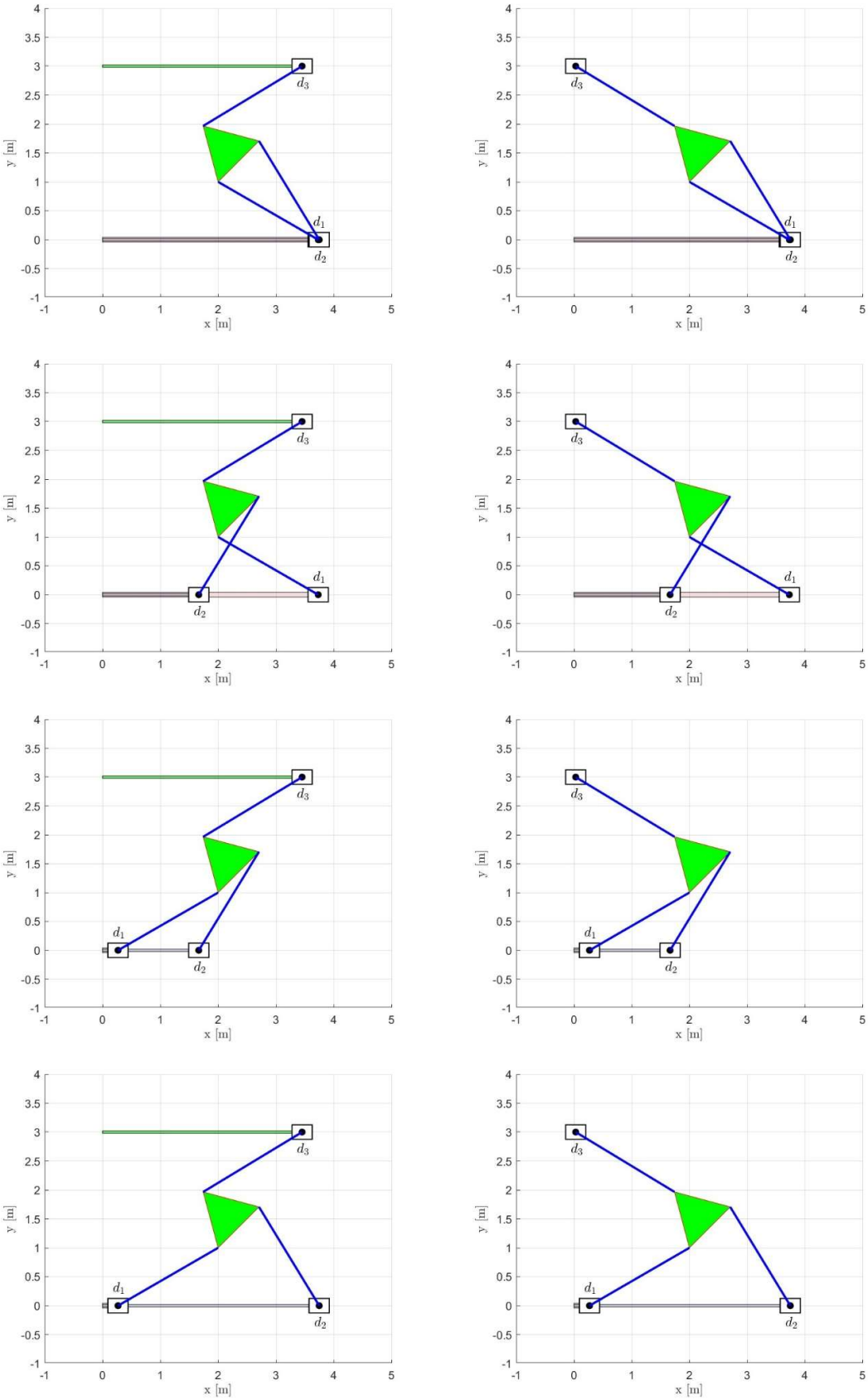



*Figure 3 - Multiple Solutions for the Inverse Kinematics problem. All solutions result in the same task vector as the position and orientation of the triangular platform is fixed. In addition, the joint variables come in paires, which means that there are only two values per joint displayed in the eight solutions.*

## 2. Forward Kinematics

The Forward Kinematics problem is to find the task vector $\boldsymbol{x}$, given the joint values vector $\boldsymbol{q}$. This problem turns out to be quite cumbersome for parallel robots in general, and for this robot as well, so an exact analytical solution will not be presented. However, the steps that were taken to solve this problem are given as follows.

Expanding equations (1.3), result in the following equations:

$$L^2 + 2\,d_1 x = {d_1}^2 + x^2 + y^2 \tag{2.1}$$

$$L^2 + 2\,d_2 x + 2\,d_2\,r\cos(\theta) = {d_2}^2 + r^2 + 2\cos(\theta)\,r\,x + 2\sin(\theta)\,r\,y + x^2 + y^2 \tag{2.2}$$

$$H^2 + {d_3}^2 + r^2 + x^2 + y^2 + r\,x\cos(\theta) + r\,y\sin(\theta) + \sqrt{3}\,d_3\,r\sin(\theta) + \sqrt{3}\,r\,y\cos(\theta) = L^2 + 2\,H\,y + 2\,d_3\,x_1 + H\,r\sin(\theta) + d_3\,r\cos(\theta) + \sqrt{3}\,H\,r\cos(\theta) + \sqrt{3}\,r\,x\sin(\theta) \tag{2.3}$$

Subtracting (2.1) from (2.2), and adding (2.3) to (2.1) yields:

$$2\,d_2\,x - 2\,d_1 x + 2\,d_2\,r\cos(\theta) = {d_2}^2 - {d_1}^2 + r^2 + 2\,r\,x\cos(\theta) + 2ry\sin(\theta) \tag{2.4}$$

$$2d_1 x + H^2 + {d_3}^2 + r^2 + rx\cos(\theta) + ry\sin(\theta) + \sqrt{3}d_3 r\sin(\theta) + \sqrt{3}ry\cos(\theta) = {d_1}^2 + 2Hy + 2d_3 x + Hr\sin(\theta) + d_3 r\cos(\theta) + \sqrt{3}Hr\cos(\theta) + \sqrt{3}rx\sin(\theta) \tag{2.5}$$

By doing so, we removed the expressions of $x^2$ and $y^2$, and we can now solve for $x$ and $y$ using linear algebra. We will define a partial task vector of $[x, y]^T$ and write the corresponding matrix form of (2.4) and (2.5) as such:

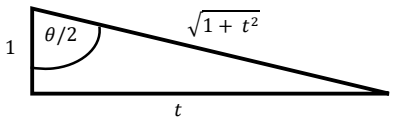
$$\begin{bmatrix} 2d_2 - 2d_1 - 2r\cos(\theta) & -2r\sin(\theta) \\ 2d_1 - 2d_3 + r\cos(\theta) - \sqrt{3}r\sin(\theta) & r\sin(\theta) - 2H + \sqrt{3}r\cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = -\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{2.6}$$

Where the coefficients $[c_1 \quad c_2]^T$ are given by:

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} {d_1}^2 - {d_2}^2 + 2\cos(\theta)\,d_2 r - r^2 \\ H^2 - {d_1}^2 + {d_3}^2 + r^2 + r\sin(\theta)\left(-H + \sqrt{3}d_3\right) + r\cos(\theta)\left(-d_3 - \sqrt{3}H\right) \end{bmatrix} \tag{2.7}$$

We solved equation (2.6) for $[x, y]^T$ as functions of joint values $\boldsymbol{q}$ and the task angle $\theta$ using MATLAB. Given the solutions for $[x, y]^T$ we could substitute them into (2.1), resulting in a single constraint equation involving only the joint values $\boldsymbol{q}$ and the task angle $\theta$. Since the dependance in $\theta$ is both with sine and cosine, we introduce a half angle tangent substitution (see Figure 4 below) which result in sine and cosine of the task angle as follows:

$$\tan\left(\frac{\theta}{2}\right) = t \Rightarrow \sin(\theta) = 2\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\theta}{2}\right) = \frac{2t}{1+t^2}\,;\cos(\theta) = \cos^2\left(\frac{\theta}{2}\right) - \sin^2\left(\frac{\theta}{2}\right) = \frac{1-t^2}{1+t^2} \tag{2.8}$$




$$\sin\left(\frac{\theta}{2}\right) = \frac{t}{\sqrt{1+t^2}} \qquad \cos\left(\frac{\theta}{2}\right) = \frac{1}{\sqrt{1+t^2}}$$

*Figure 4 - Half angle tangent substitution. Both sine and cosine are substituted with a quotient function in the parameter t.*

This substitution, after some MATLAB symbolic simplifications, allowed the constraint equation to be in the form:

$$\frac{num(d_1,d_2,d_3,H,r,L,t)}{den(d_1,d_2,d_3,H,r,L,t)} = \frac{P_{num}(t)}{P_{den}(t)} = 0 \tag{2.9}$$

Where $num$ is the numerator given by a polynomial function of t, and $den$ is the denominator (polynomial of $t$), because the joint values and geometric parameters are given in the Forward Kinematics problem.

The solution for (2.9) is given by equating the numerator to zero, and its roots are the solutions for $t$. These roots were found using the "roots" function in MATLAB in a designated FK function, and any imaginary roots were neglected. Once the solutions of $t$ were found, they were substituted into (2.8), yielding a single angle of $\theta$ via the atan2 function.

Once the task angle $\theta$ was found, it was substituted into the solved solutions for $[x, y]^T$. Thus, given the joint values $\boldsymbol{q} = [d_1, d_2, d_3]^T$, the task vector $\boldsymbol{x} = [x, y, \theta]^T$ was found and the Forward Kinematics is solved.

Following are the solutions for $\boldsymbol{x}$, given a single choice of a joint vector $\boldsymbol{q}_0 = [3.7321, 1.6651, 3.4531]^T$, using the forward kinematics described above. The single choice is the third solution from the IK problem $\boldsymbol{q}_3$.

$$\boldsymbol{x}_i = \begin{bmatrix} 3.3893 & 3.7658 & 3.7312 & 2.0000 \\ 1.9704 & 1.9997 & 2.0000 & 1.0000 \\ -3.0329 & -2.7491 & -1.9600 & 0.7854 \end{bmatrix} \quad (2.10)$$

Where every column in (2.10) corresponds to a different solution found. The robot has been plotted for every solution for $\boldsymbol{x}_i$ given the same joint vector $\boldsymbol{q}_0$, and the solutions are displayed in Figure 5 below. It is shown that the solution for the task $\boldsymbol{x}_0$ is given in the last column, which corresponds to the task vector used in the IK section. The solution in the bottom right corner of Figure 5 is identical to the 3rd solution in Figure 3 as expected. Note that the polynomial in $t$ is of 6th order, so in general we would expect to get six solutions, however in this specific choice of $\boldsymbol{q}_0$, there were two imaginary roots that were neglected, thus resulting in only four solutions.

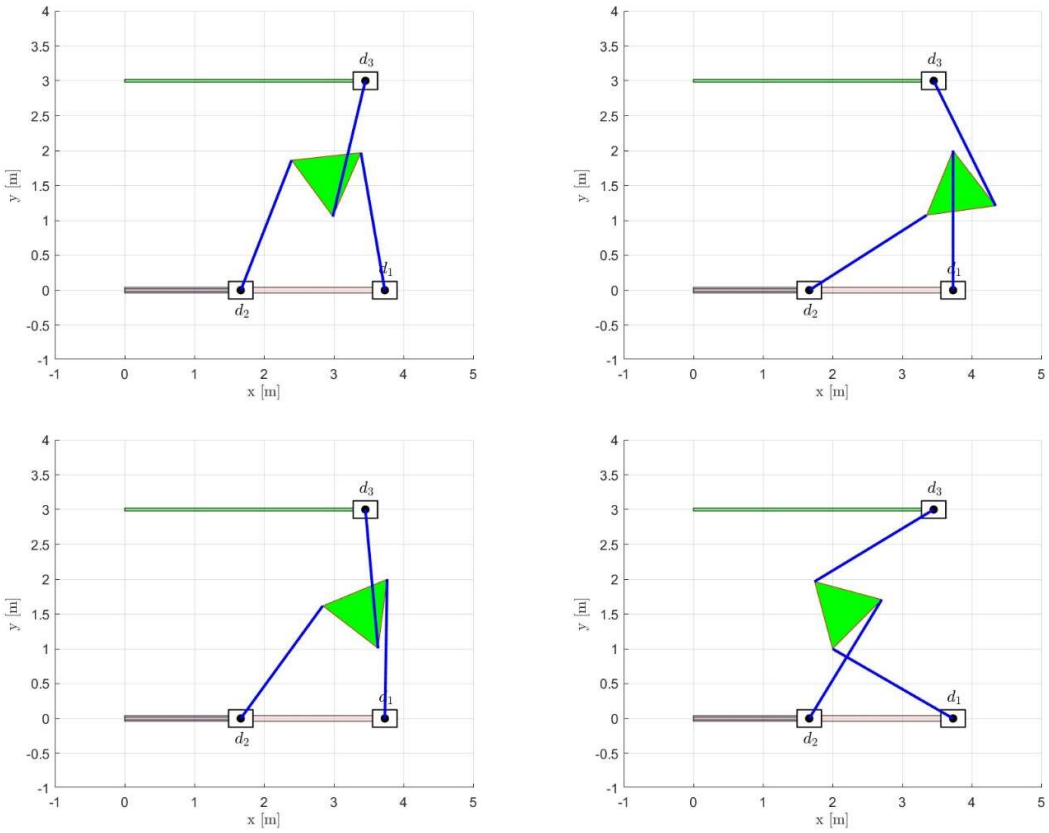



*Figure 5 - Multiple Solutions for the Forward Kinematics problem. All solutions include the same joints vector as the values of $d_i$ are identical. In addition, the 4th plotted solution corresponds to the same solution found in the Inverse Kinematics problem when given the same task vector.*

To test the validity of the FK and IK calculations, we calculated the numerical errors by nesting the IK into the FK as follows. We entered a specific task vector $\boldsymbol{x}_0$, and found its Inverse Kinematics solutions $\boldsymbol{q}_i$. For each solution $\boldsymbol{q}_i$, we computed $j$ different Forward Kinematics solutions $\boldsymbol{x}_j$. Then, using the MATLAB function "ismembertol" we searched in $\boldsymbol{x}_j$ for the corresponding input $\boldsymbol{x}_0$, which we denote by $\boldsymbol{x}_j$. Then we calculated the errors by finding the norm of the differences in the task vector for each solution $i$.

$$\boldsymbol{e} = \|\boldsymbol{x}_0 - \boldsymbol{x}_j\| = [0.3086, 0.1870, 0.0175, 0.0071, 0.0182, 0.0619, 0.0225, 0.0223] \cdot 10^{-13} \quad (2.11)$$

The corresponding errors were in the order of magnitude of $10^{-1}$ and relatively small compared to the task values $\boldsymbol{x}_0$. This is possibly due to the numerical solution for the polynomial in $t$, and its substitution.

## 3. Jacobian

In parallel robots, the Jacobian can be found by the following two matrices:

$$\boldsymbol{J}_x = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} \quad , \quad \boldsymbol{J}_q = -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{q}} \tag{3.1}$$

Where the function $\boldsymbol{F}$ was found by three implicit constraints given in (1.3) and moving the $L^2$ terms to the right-hand side. The Matrices were found using the symbolic function in MATLAB called "jacobian":

$$\boldsymbol{J}_q = -\begin{bmatrix} 2x - 2\,d_1 & 0 & 0 \\ 0 & 2x - 2\,d_2 + 2\,r\cos(\theta) & 0 \\ 0 & 0 & 2\,d_3 - 2x - r\cos(\theta) + \sqrt{3}\,r\sin(\theta) \end{bmatrix} \tag{3.2}$$

The FK Jacobian had large expressions, so it is displayed in columns:

$$\begin{aligned} [\boldsymbol{J}_x]_{i,j=1} &= \begin{bmatrix} 2\,d_1 - 2x \\ 2\,d_2 - 2x - 2\,r\,\cos(\theta) \\ 2x - 2\,d_3 + r\,\cos(\theta) - \sqrt{3}\,r\sin(\theta) \end{bmatrix} \\ [\boldsymbol{J}_x]_{i,j=2} &= \begin{bmatrix} -2\,y \\ -2\,y - 2\,r\,\sin(\theta) \\ 2\,y - 2\,H + r\,\sin(\theta) + \sqrt{3}\,r\,\cos(\theta) \end{bmatrix} \\ [\boldsymbol{J}_x]_{i,j=3} &= \begin{bmatrix} 0 \\ 2\,rx\,\sin(\theta) - 2\,ry\,\cos(\theta) - 2\,d_2\,r\,\sin(\theta) \\ d_3 r\sin(\theta) + ry\cos(\theta) - rx\sin(\theta) - Hr\cos(\theta)\ \dots \\ \dots + \sqrt{3}Hr\sin(\theta) + \sqrt{3}d_3 r\cos(\theta) - \sqrt{3}rx\cos(\theta) - \sqrt{3}ry\sin(\theta) \end{bmatrix} \end{aligned} \tag{3.3}$$

## 4. Singular States

The singular states occur when the Jacobian loses its full rank. In the parallel robot, the criterion is extended to the IK and FK problems as there are two Jacobian matrices, and the singular states are found by comparing the Jacobian matrices determinants to zero (This is because the Jacobian Matrices are 3x3, so the loss of rank is equivalent to equating the determinant to zero).

The requirement on the task angle was given as $\theta = 0$, resulting in the simplified Jacobian Matrices:

$$[\boldsymbol{J}_q]_{\theta=0} = -\begin{bmatrix} 2x - 2\,d_1 & 0 & 0 \\ 0 & 2x - 2\,d_2 + 2\,r & 0 \\ 0 & 0 & 2\,d_3 - 2x - r \end{bmatrix} \tag{4.1}$$

$$[\boldsymbol{J}_x]_{\theta=0} = \begin{bmatrix} 2\,d_1 - 2x & -2y & 0 \\ 2\,d_2 - 2\,r - 2x & -2y & -2\,ry \\ r - 2\,d_3 + 2x & 2y - 2\,H + \sqrt{3}\,r & ry - H\,r + \sqrt{3}\,d_3\,r - \sqrt{3}\,rx \end{bmatrix} \tag{4.2}$$

The determinants are given respectively:

$$|\boldsymbol{J}_q|_{\theta=0} = -4\,(d_1 - x)\,(r - d_2 + x)\,(r - 2\,d_3 + 2x) \tag{4.3}$$

$$|\boldsymbol{J}_x|_{\theta=0} = 4\,d_1\,r\,y^2 + 4\,d_2\,r\,y^2 - 8\,d_3\,r\,y^2 + 4\,H\,r^2\,y + 8\,H\,rx\,y + 4\sqrt{3}\,d_1\,r^2\,y - 4\sqrt{3}\,d_3\,r^2\,y - 4\,H\,d_1\,r\,y - 4\,H\,d_2\,r\,y - 4\sqrt{3}\,d_1\,d_3\,r\,y + 4\sqrt{3}\,d_2\,d_3\,r\,y + 4\sqrt{3}\,d_1\,r\,x\,y - 4\sqrt{3}\,d_2\,rxy \tag{4.4}$$

Since the joint vector can be increased in any fixed value $\tilde{\boldsymbol{d}}$ and it will just shift the robot towards the right, we concluded that to find qualitatively different singular states, the value of $x$ can be chosen arbitrarily as it will just shift the solution. We arbitrarily chosen it as $x = 1$.

### Inverse Kinematics Singularity

Given the choice for $x$, the IK singularities are given by the following three solutions of (4.3):

$$\begin{bmatrix} d_1 \\ d_2 - r \\ d_3 - \frac{1}{2}r \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 + r \\ 1 + \frac{1}{2}r \end{bmatrix}_{r=1} = \begin{bmatrix} 1 \\ 2 \\ 3/2 \end{bmatrix} \tag{4.5}$$

However, since the task value for $y$ is unknown, it can be found using the constraints in (1.3) for each of the solutions in (4.5). When solving (1.3) for $y$ for each solution in (4.5), we get respectively:

$$\begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3/2 \end{bmatrix} \Rightarrow y = \begin{cases} \pm L \\ \pm L \\ H - \frac{\sqrt{3}}{2} r \pm L \end{cases}_{r=1, L=2, H=3} = \begin{cases} \pm 2 \\ \pm 2 \\ 3 - \frac{\sqrt{3}}{2} \pm 2 \end{cases} \tag{4.6}$$

The only case for $y$ given above that satisfies the limitation: $0 < y < 2$ is when: $y = 3 - \frac{\sqrt{3}}{2} - 2 \approx 0.134$. Therefore, the only relevant IK singularity is given at the task state $\boldsymbol{x}^* = [1, 0.134, 0]^T$, with the constraint on the third joint: $d_3 = 3/2$. Since the general IK solution includes 8 states, and in singularity we are expecting two solution branches to converge, we could expect only four solutions. This agrees to us receiving only four solutions as shown below. The resulting Inverse Kinematics solutions are given by:

$$\boldsymbol{q}_i = \begin{bmatrix} 2.9955 & 2.9955 & 2.9955 & 2.9955 & -0.9955 & -0.9955 & -0.9955 & -0.9955 \\ 3.9955 & 3.9955 & 0.0045 & 0.0045 & 3.9955 & 3.9955 & 0.0045 & 0.0045 \\ 1.5000 & 1.5000 & 1.5000 & 1.5000 & 1.5000 & 1.5000 & 1.5000 & 1.5000 \end{bmatrix} \tag{4.7}$$

Two solution branches have converged as there are duplicates in (4.7). These singular states are plotted below:

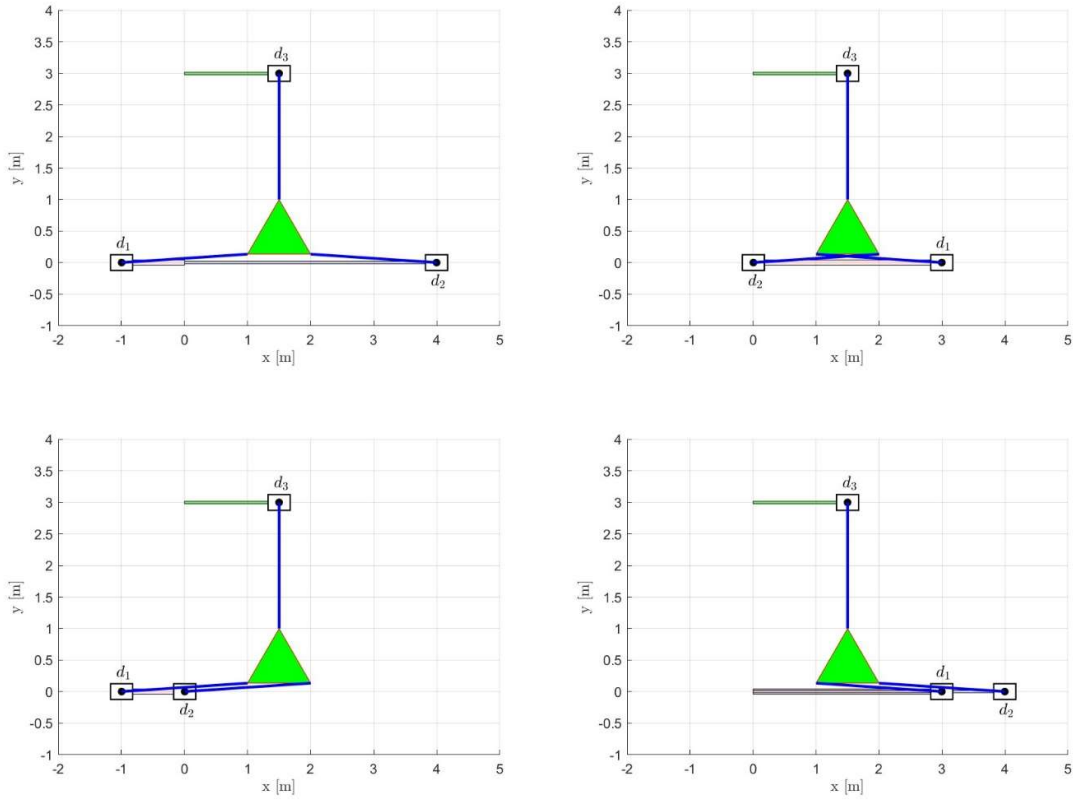



*Figure 6 - The Inverse Kinematics Singular States within the limits provided. At these states there is loss of DOF because there is a task velocity direction that cannot be obtained. In addition, two IK solution branches converged as they came out to be identical states.*

## Forward Kinematics Singularities

Given the arbitrary choice for $x$, the FK singularities are found by solving (4.4). Since this equation is dependent on four variables $(d_1, d_2, d_3, y)$, it is necessary to find three more equations to properly solve for $y$. For this reason, we had to use the three constraint equations presented in (1.3) in addition to (4.4).

The solutions were found numerically using "vpasolve" function in MATLAB, while assuming the solution for $y$ is limited to the range: $0 < y < 2$. The solved states are presented below, along with their corresponding plots in Figure 7.

$$\boldsymbol{x}_i = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1.732 & 1.732 & 1.067 & 1.067 & 0.134 & 0.134 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.8}$$

In (4.8), it is shown that two solution branches of the FK converge, since there are duplicates, which is indicative of FK singular states.

The directions in which $J_x \dot{\boldsymbol{x}} = 0$ holds were found by computing the null space of $J_x$ evaluated at these states and normalizing the vectors.

$$\hat{\boldsymbol{x}}_i = \begin{bmatrix} 0.6124 & 0.6124 & -0.5335 & -0.5335 & -0.0300 & 0.0300 \\ -0.3536 & -0.3536 & -0.8458 & 0.8458 & -0.4470 & -0.4470 \\ 0.7071 & 0.7071 & 0 & 0 & 0.8940 & 0.8940 \end{bmatrix} \tag{4.9}$$

Where the $i^{th}$ direction corresponds to the $i^{th}$ solution in (4.8), and in Figure 7 respectively.

Forward Kinematics singularities exhibit an interesting phenomenon where the continuation of the links connecting between the joints and the platform intersect at some point (or all being parallel). This is visualized in Figure 7. This means that there is a direction of velocity in the task space that locally does not change the velocity of the joints. When dashed lines are parallel, the platform can move in the perpendicular direction without changing the joint velocities as indicated by the red vector. When the dashed lines intersect at some point, the platform can locally rotate about that point without changing the joint velocities, which corresponds to the red vector pointing in the perpendicular direction with respect to the intersection point.

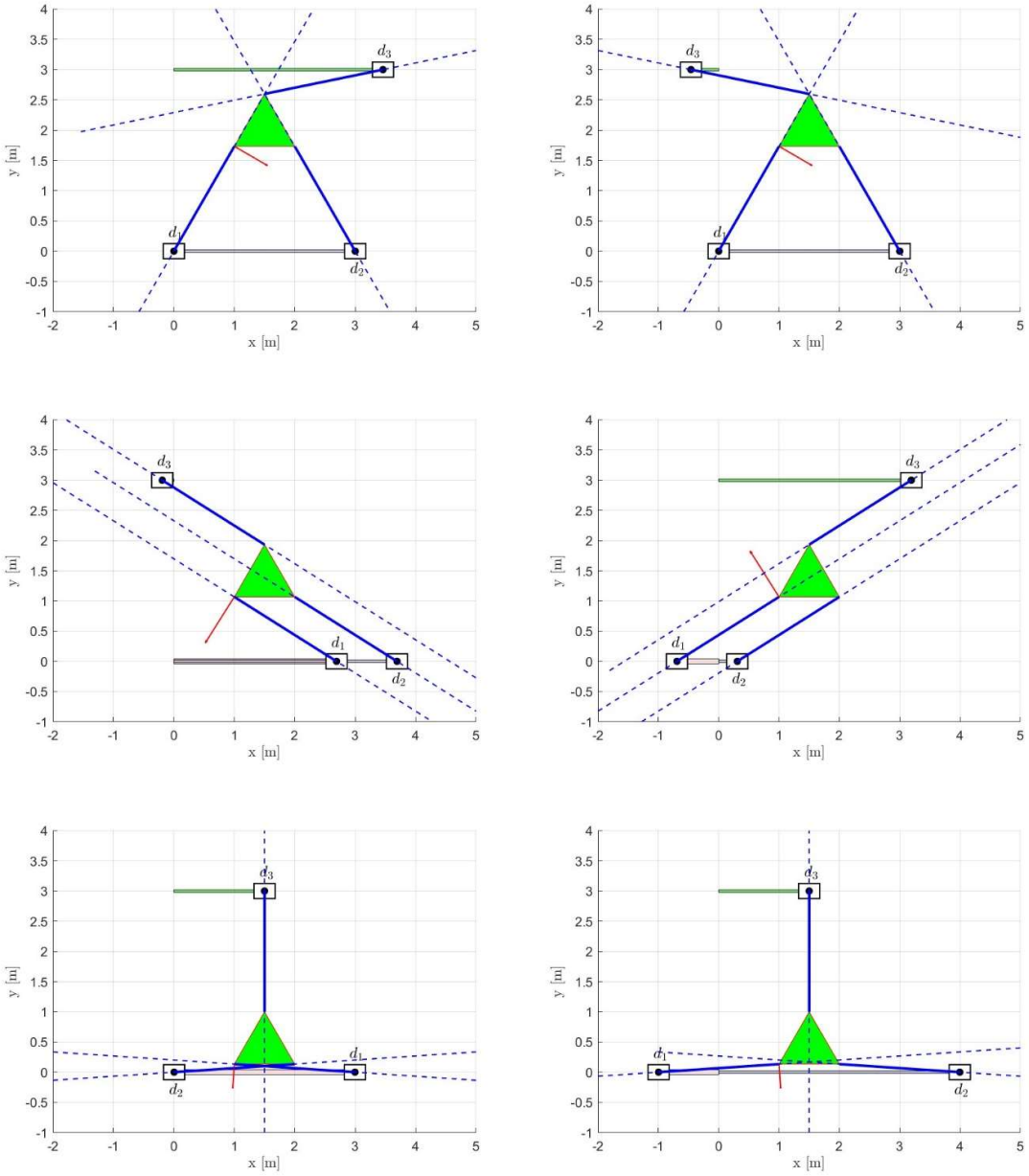



*Figure 7 - The Forward Kinematics Singular states under the limitation of the task angle to be zero, an arbitrary value for x and 0 < y < 2. At these (six) qualitatievly different states there is a gain in a DOF. It is shown that for each state there is a specific point where all continuation lines of the links connecting between the platform and the joints intersect. This phenomenon agrees with the theory learned in the Lecture on Singularities. In addition, there are only three task vectors corresponding to six joint values, which indicates that two solution branches of FK converged, as expected in FK singular states.*

## 5. Summary

In this project, we analyzed the kinematics of a parallel robot. We found that Inverse Kinematics was relatively simpler to solve than Forward Kinematics, and unlike serial robots, Forward Kinematics had multiple solutions. The Singularities were divided into FK and IK, and we saw that in an IK singular state, two solution branches of the IK converged (see Figure 6 and result (4.7)). In the FK singular states, we saw that two FK solution branches converged (see result (4.8)). In addition, there was a gain in a DOF, with a local task space velocity direction that does not change the joint velocities, and a visualization of a phenomenon of intersection of lines (see Figure 7). On a final note, this project was challenging, yet rewarding, as we learned new concepts and skills along the way.