ENPM 662-INTRO TO ROBOT MODELLING

FINAL PROJECT REPORT

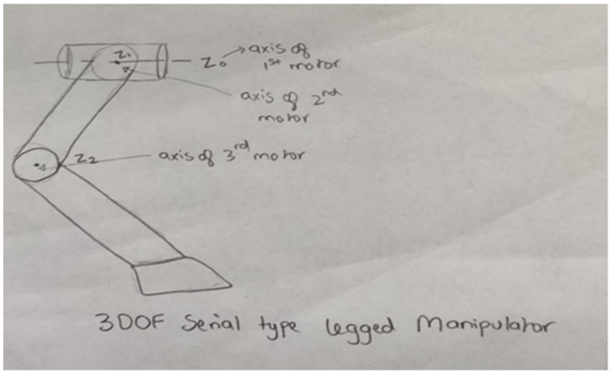
**Motivation-**

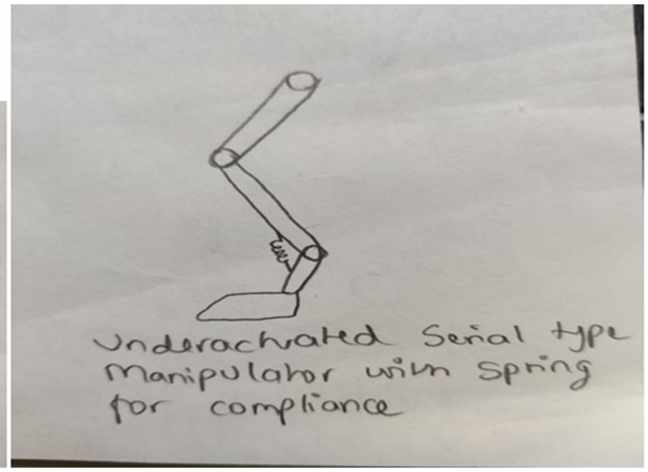
At most places where locomotion is needed, wheeled mechanisms (machines) are used. This is generally because of the high efficiency we get while rotating a wheel as rolling friction is less than sliding friction. Wheels generally suffer from a big disadvantage that is they need flat roads to run smoothly; even with the suspension technology we currently have wheeled locomotion pretty much cannot work on rough, steep and unknown terrains.

Now when we look at mammals we are gifted with different kind of locomotion mechanisms which we call legs, which enable us to traverse both flat and rough terrain. Legs have the ability to precisely position themselves and thus balance the entire structure hanging on them hence giving them the ability to traverse rough terrain, climb very steep hills, and also intelligently move in unknown terrains. Legs give the user the ability to create massive accelerations and ground contact forces which are very useful in tasks like jumping and running fast. Now while making robots we have generally come to a max innovation point when it comes to wheeled robots, where we are confident enough to send them to different planets. But this does not apply to robot legs, where a lot of R&D is still going on to achieve animal like locomotion in the most efficient manner.

When we think of building a quadruped (4 legged robot) we generally think of mechanisms that are serial link based that is one motor link connected on the top of another getting 2 or 3 dofs. These robots suffer from problems like high moment of inertia of each leg, center of mass a little lower than required(without dead weight alignment), each actuator has different speed and torque requirements as the actuator placed on top should be most powerful as it has to support and actuate the weight of the robot as well as the actuators lying below it. Also these types of robot cannot move omni-directionally, hence in tightly spaced environments they may need to turn standing in the same place to move forward which is inefficient and also impossible in certain cases. Sometimes even redundantly actuated mechanisms are used so that some compliance can be achieved (which is very difficult to achieve in serial based leg manipulators) but these robots suffer from low position accuracy because of the un-actuated link. In this project I propose a different kind of robot leg mechanism that attempts to solve the problems faced by all above mentioned leg manipulators.

Generic Mechanisms used in legged manipulators




Here all the 3 links are actuated

Here only the upper two links are actuated.

**Literature Survey**

A lot of research has been carried out on legged robots and some really world class robots have been fabricated in the past. Some of the very well known and accomplished 4 legged robots are RHex, Little Dog, Sand Flea, MIT Cheetah, Spot of Boston Dynamics etc. RHex is shown to have performed jumps and fast runs using active compliance mechanisms but its legs are constrained to 1 plane hence leading to low controllability. Little dog has shown to have performed dynamic maneuvers but its high impedance actuators, slow leg speed and leg mechanisms complicate force vectoring(control) and hence limit the magnitude of dynamic activities like jumping and running. SandFlea from Boston Dynamics uses wheels on the foot for executing Omni directional movement and jump using compressed gas to launch itself as high as 30feet in the air. MIT Cheetah uses low impedance actuators to generate high speeds and high force motions with excellent proprioceptive sensing but its dynamic mobility is constrained by its limited work envelope of the leg. This proposal concentrates on building a robot capable of very high omni directional mobility and traverse on a variety of rough, steep and challenging terrains with some dynamic capability. Traversal over rough, steep and challenging terrain is more of a control problem than a design problem and hence we won’t dive much into that, we will just ensure that the robot is capable of doing so in its design limits.

Images of different well known legged robots

Figure 2.1: The current state of art in dynamic legged machines in 2016: 1) Boston Dynamics (BDI) SpotMini, 2) Penn/Ghost Robotics Minitaur, 3) ATRIAS, 4) RHex and Canid, 5) StarlETH, 6) BDI Spot, 7) MIT Cheetah, 8) BDI Atlas.

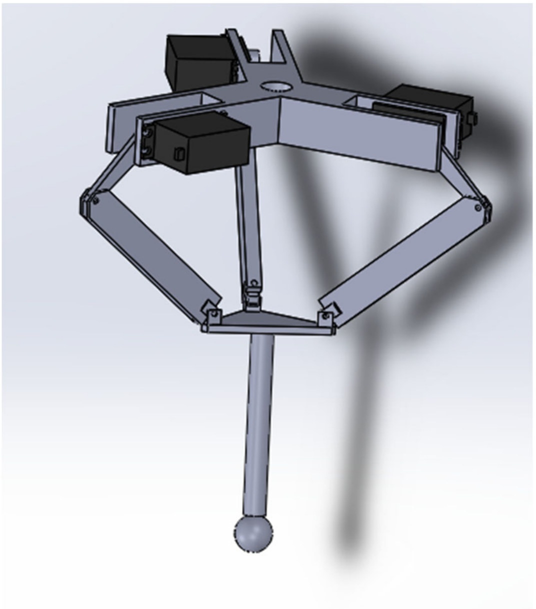
**Robot Description (Hypothesis):**

Herein I propose a model of a robotic leg that tries to solve all the problems mentioned above. Here we use a 3 DOF RRP parallel manipulator. On the platform of the manipulator a fixed link is attached to extend the range of the mechanism.

- This type of mechanism allows the movement of the feet(end effector) in 3 dimensions which is the sole requirement of this project which enables omni directional movement. We don't consider orientation of the feet as we attach a sphere at the end of the link which makes the orientation an obsolete feature.
- The typical parallel manipulators like delta robots can also be used in this application but it will become very difficult to apply 3d forces in diagonal directions. Hence we choose a RRP manipulator instead of a PPP parallel manipulator.
- Also the work envelope required for a robotic leg is fully satisfied by using a parallel manipulator which makes all mountings and connections very concise. This project will only concentrate on the mathematical modelling of the robot and not on the control element.

**Design of Robot**

The design of the robot consists of 4 modules which are assembled first and then attached to the base frame. These modules are individual leg mechanisms and contain 3 actuators (Servo Motor) each. The link directly connected to the servo motor arm is connected using a revolute joint and the link connected using spherical joints (for simulation purposes) and ball joints when modelling the robot in real life. A rigid link with a sphere is connected to the platform as mentioned above. Features of the robot have been discussed below and all of them have been taken into account while designing the system



Design of the Robot Design of a single leg Manipulator

The robot has been designed in such a way that it minimizes the formation of excessive triangles making it easier for the simulation software to handle

In the single leg manipulator design, the black part is the servo motor, the link attached to it is the servo arm and the link attached to the servo arm is the link. All the links are connected to a small triangular platform and the triangular platform has a rigid link attached to it that is referenced to as the leg or the rigid link. A spherical ball is connected to the leg which is referenced to as the foot. In the entire report all components are referenced as mentioned above.

Here we use 3 actuators to get the motion of the platform of the leg in 3DOF. The leg-platform has 1 translational DOF as explained above, and it has 2 angular DOFs which are used to position the end effectors position. For this project we do not consider the orientation of the foot as the foot is a ball and its orientation is not of much use, we instead consider the 3 translational co-ordinates to completely specify the position of each foot and from that derive the three parameters (2 angles and 1 translation) using trigonometry and some intuition and hence derive the joint angles using these parameters from the IK equations. So to conclude the point of the feet which is in contact to the ground has 3 translational DOF which we control using the 3 joint variables.

When in full upright position the robot reaches to upto a height of 645 mm (64.5cm) .Each leg can move in the direction perpendicular to the base plane of the robot by 35cm at max. The length of the robot is about 750mm and the breadth is about 400mm. The Legs of the robot never come in the way of each other. When designed materials for the robot are uploaded in the software and most of the solid components are made from PLA and all moving parts like links, joints are made from aluminium. After this the weight of the robot comes up to 9.9kg. The base is made of soft material like wood which reduces the overall weight of the robot and the robot is designed for dynamic movements like jumping and running hence payload is not given a lot of importance.

**Scope:**

Herein we look to design a robot that is agile enough so that it can jump distances equal to atleast half its height. If this is achieved the robot will be agile enough to run at high speeds and climb steep hills.

It seems odd to think that this mechanism can apply forces on the ground in any direction but it actually can. Once the leg of the robot touches the ground we should consider it as a single link coming out of the body of the robot. Now we have control over the exact endpoints where the robot touches the ground and once it does any further actuation just changes the orientation of the whole robot. As we have control over the endpoints, we can program the robot to move the endpoint touching the ground in any direction in the permissible workspace, and hence we can apply a force in any direction by manipulation of its feet position. If the robot is touching the ground and all its feet move in the same direction while touching the ground, the robot as a whole will move in that particular direction. This can be seen in the video I have uploaded where the robot just stands there and dances (changes its orientation) by manipulating where the endpoints of the feet should be. This can be purely done by only the IK alone. Now if we want to have control over the magnitude of force we fall into the dynamics part which is discussed further or we can use force sensors in a closed loop feedback mechanism to control the magnitude of the force.

I myself have an idea to achieve force control in legs without using inaccurate force sensors and the super complex yet inaccurate dynamics calculation. What we can do is add a passive joint over the active joint and connect both the passive and active joints with springs. Now to control force, we can measure the exact compression of the spring and use actuation in combination with braking of the actuators to achieve very accurate force control. I have attached a diagram for this idea to make it clearer. This idea is out of the scope for this project and hence it is not discussed thoroughly.

Due to the parallel manipulator arrangement accuracy is always very high so that does not need to be worried about hence traversing of very rough terrain becomes possible and this problem is more of a control problem. Ahead in the proposal we discuss about feedback of normal reaction on each foot which is very important when traversing challenging terrains. The scope of the robot is further discussed in motor sizing and other features where all parameters of the robot are fixed. .

**Model Assumptions:**

I have made this design for simulation purposes only (The design is made such that it can be broken down into the least no of triangles), if fabrication has to be considered the design will have to be changed considerably. I have used spherical wrist type joint instead of ball joint because most simulation softwares do not include a 3dof rotary joint and hence consider it as a fixed joint. For the simulation I have not considered compliance and errors in joints in my mathematical models. Also it is assumed that ground geometry is going to be very flat and hence not bringing in complexity into the mathematical models. Also all the math done is wrt only 1 leg and it can be expanded during implementation. Also in my calculations I have not considered friction with ground as I have selected actuators with a factor of safety of 1.5 which will easily compensate for friction, But during implementation of dynamic maneuers friction cannot be ignored and has to be considered in the mathematical model which is out of scope for this project.

**Mathematical Modelling of the Mechanism**

I have concentrated mainly on the Inverse Kineamatics, Inverse velocity Kinematics, Workspace analysis and Inverse dynamics of each leg alone. The highest focus is on the Inverse Kinematics (IK) as it is possible to control the robot with only IK but when complex dynamic manuevers have to be performed all the other math too becomes very important.

In Parallel Manipulators it is easier to derive the IK than the FK as opposed to serial Manipulators. IK is the one we have to use the most hence it is fine if FK is not performed.

In Parallel Manipulators the IK is performed by using vector math to make it easier, as doing it with geometry alone becomes very difficult to visualize and derive. Also the PMs form a closed loop structure and the addition property of vectors can be used as we can see many triangles of vectors forming and hence we can get the relations between these vectors.

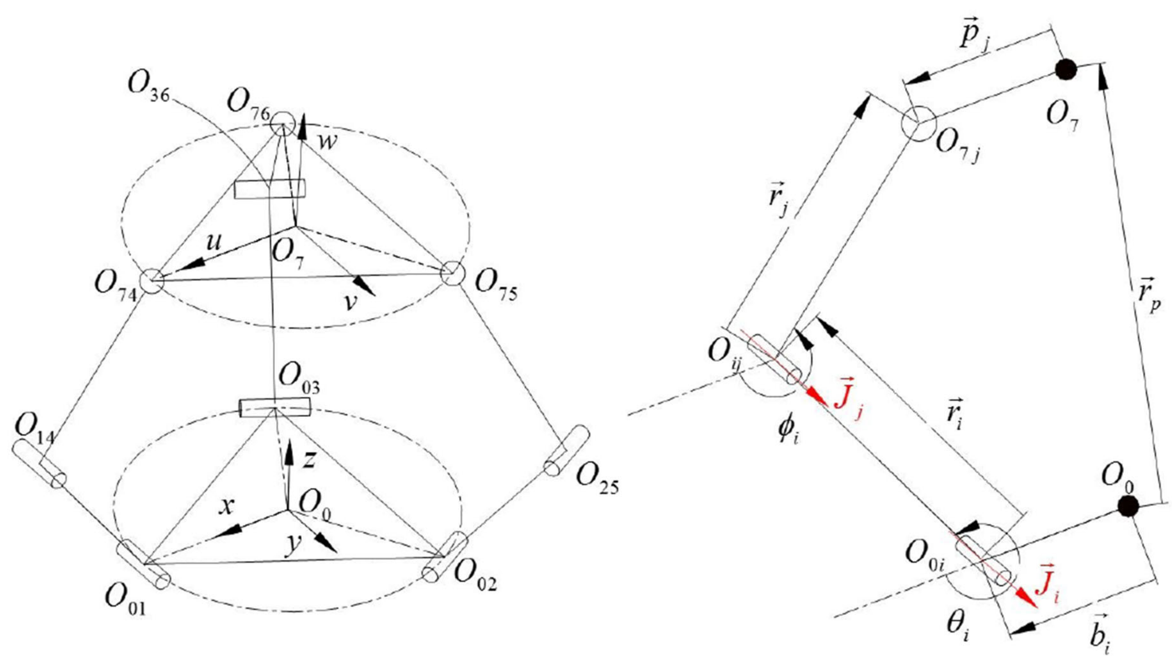
Each length in the manipulator is considered a vector (as they keep on changing when the PM moves) and we use the property of addition and subtraction of vectors to form triangles and relations between these vectors which we can use to derive the final joint variables.

2. **Forward Kinematics of the PM_leg:**

The forward position level kinematic analysis problem of PMs is more complex
than the inverse kinematics task .The common methodology to obtain the pose of the moving platform is as follows: the coordinates of the S joints are formulated in terms of active and passive joint variables and then the fixed distance in between the S joints is used as constraint to formulate the constraint equations.
The obtained equations are non-linear in terms of active and passive joint variables. Making use of tangent of the half angle substitution, some mathematical manipulations are applied to the non-linear constraint equations and at the end a 16th degree polynomial is determined in terms of one of the passive joint variables. Now to solve this equation numerical methods have to be used as it is not possible to solve a $16^{th}$ degree polynomial with conventional methods

1. **Inverse Kinematics of the PM_leg:**

For the IK problem, we have the 2 angular orientations of the platform and its vertical displacements and we should find the joint variables




The PM consists of a fixed base, a moving platform and 3 identical links connecting the base and the platform. The links lie on separate planes each at an angle of 120 deg from each another. Each links plane is normal to the parallel revolute joint axes and passes through the ball joint centers associated with the link. Each link is composed of three joints:

- an active revolute joint connected to the base represented by points $O_{0i}$ on the link plane,
- a passive revolute joint between upper and lower links represented by points $O_{ij}$ on the link plane,
- a ball joint between upper limb and platform represented by points for i = 1; 2; 3 and j = i + 3.

In the figure above, a fixed coordinate frame $O_{0xyz}$ is attached on the base, where the origin $O_0$ is the chosen as the center of the circle which is tangent to the three fixed revolute joints. The radius of the base circle is b. The circle coincides with $O_{0i}$ for i = 1; 2; 3. The vectors from $O_0$ to $O_{0i}$ are $=b_i$ and the x-axis is chosen to be along $b_1$. The angle between $b_1$and $b_2$ is $alpha_{12}$ = 120. The angle between $b_1$ and $b_3$ is $alpha_{13}$ = 240. A coordinate frame .$O_7$ uvw is attached on the moving platform, where the origin $O_7$ is the center of the circle which passes through the three S joint centers $O_{7j}$ for j = 4; 5; 6. The radius of the platform circle is p. The vectors from $O_7$ to $O_{7j}$ are $p_j$ . The u-axis is chosen to be along $p_4$ .The angle between $p_4$ and $p_5$ is $alpha_{45}$ = 120.The angle between $p_4$ and $p_6$ is $alpha_{46}$ = 240. The position vector that defines the location of the moving platform with respect to the fixed coordinate frame is
$r_P = O_0O_7$.
$r_i = O_{0i}O_{ij}$ are the lower link vectors and $r_j = O_{ij}O_{7j}$ are the upper link vectors. The axes of the active revolute joints are along $J_i$ unit vectors and passive R joints are along $J_j$ unit vectors. $J_1$ and $J_4$ are along the y-axes of the fixed coordinate frame. All these notations are used throughout the report as the same and the diagram and the notations are taken form a paper referenced below in the report. From the figure,

$r_p=[O_{7,x}\ O_{7,y}\ O_{7,z}]^T$

Let R be a rotation matrix that defines the orientation of the platform

R=[$u_x$ $v_x$ $w_x$ ]
[$u_y$ $v_y$ $w_y$ ]
[$u_z$ $v_z$ $w_z$ ]

u, v, w are column vectors of R which are mutually perpendicular to each another

Since $O_{74}$ is constrained on $xz$ plane:

$$\overrightarrow{O_0O_{74}} = \vec{r}_P + \vec{p}_1 \Rightarrow \begin{bmatrix} O_{74,x} \\ O_{74,y} \\ O_{74,z} \end{bmatrix} = \begin{bmatrix} O_{74,x} \\ 0 \\ O_{74,z} \end{bmatrix} = \begin{bmatrix} O_{7,x} \\ O_{7,y} \\ O_{7,z} \end{bmatrix} + \mathbf{R} \begin{bmatrix} p \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} O_{7,x} + pu_x \\ O_{7,y} + pu_y \\ O_{7,z} + pu_z \end{bmatrix} \quad (3.4)$$

Since $O_{75}$ is constrained on the $y = \tan(120°)\, x$ plane:

$$\overrightarrow{O_0O_{75}} = \vec{r}_P + \vec{p}_2 \Rightarrow \begin{bmatrix} O_{75,x} \\ O_{75,y} \\ O_{75,z} \end{bmatrix} = \begin{bmatrix} O_{75,x} \\ -\sqrt{3}O_{75,x} \\ O_{75,z} \end{bmatrix} = \begin{bmatrix} O_{7,x} \\ O_{7,y} \\ O_{7,z} \end{bmatrix} + \mathbf{R} \cdot \mathbf{Z}(\alpha_{45}) \cdot \begin{bmatrix} p \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} O_{7,x} - \frac{pu_x}{2} + \frac{\sqrt{3}pv_x}{2} \\ O_{7,y} - \frac{pu_y}{2} + \frac{\sqrt{3}pv_y}{2} \\ O_{7,z} - \frac{pu_z}{2} + \frac{\sqrt{3}pv_z}{2} \end{bmatrix} \quad (3.5)$$

where $\mathbf{Z}(\cdot)$ represents the rotation matrix around the z-axis. Since $O_{76}$ is constrained on the $y = \tan(240°)\, x$ plane:

And similarly equations for [$O_{74}$ $O_{75}$ $O_{76}$] is calculated. Now $O_{7,x}$ $O_{7,y}$ $O_{7,z}$ are calculated from the above written equations

$$O_{7,y} = -u_y p$$

$$O_{7,y} = \frac{pu_y}{2} - \frac{\sqrt{3}pv_y}{2} - \sqrt{3}\left(O_{7,x} - \frac{pu_x}{2} + \frac{\sqrt{3}pv_x}{2}\right)$$

$$O_{7,y} = \frac{pu_y}{2} + \frac{\sqrt{3}pv_y}{2} + \sqrt{3}\left(O_{7,x} - \frac{pu_x}{2} - \frac{\sqrt{3}pv_x}{2}\right)$$

Now from these equations, we subtract the 2 equations and we obtain $u_x$=$u_y$ and equating both these equations gives us

$$O7x = p\frac{ux - vy}{2}$$

Now the components of the Rotation matrix are obtained by using the Euler rotation sequence.

$$\mathbf{R} = \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{bmatrix} =$$

$$\begin{bmatrix} c\psi_y c\psi_z & -c\psi_y s\psi_z & s\psi_y \\ c\psi_z s\psi_x s\psi_y + c\psi_x s\psi_z & c\psi_x c\psi_z - s\psi_x s\psi_y s\psi_z & -s\psi_x c\psi_y \\ s\psi_x s\psi_z - c\psi_x c\psi_z s\psi_y & s\psi_x c\psi_z + c\psi_x s\psi_y s\psi_z & c\psi_x c\psi_y \end{bmatrix}$$

Now using the orthogonality relations we find

$$\psi_z = \tan^{-1}\left(\frac{-\,s\,\psi_x\,s\,\psi_y}{c\,\psi_x + c\,\psi_y}\right)$$

Now, from the figure above using vector addition we have

$$\vec{r}_P + \vec{p}_j = \vec{b}_i + \vec{r}_i + \vec{r}_j$$

First, the positions of $O_{7j}$ points in terms of given pose parameters are determined by using the left-hand side of Equation 3.14. Then using the right-hand side of the Equation 3.14 and resolving into $x$, $y$ and $z$ components:

$$x: \quad l_2 \mathrm{c}\phi_i \mathrm{c}\alpha_{1i} = O_{7j,x} - \mathrm{c}\alpha_{1i}\left(b + l_1 \mathrm{c}\theta_i\right) \tag{3.15}$$

$$y: \quad l_2 \mathrm{c}\phi_i \mathrm{s}\alpha_{1i} = O_{7j,y} - \mathrm{s}\alpha_{1i}\left(b + l_1 \mathrm{c}\theta_i\right) \tag{3.16}$$

$$z: \quad l_2 \mathrm{s}\phi_i = -O_{7j,z} - l_1 \mathrm{s}\theta_i \tag{3.17}$$

Multiplying Equation 3.17 with $\mathrm{c}\,\alpha_{1i}$ and adding up the square of Equation 3.17 with the square of Equation 3.15:

$$\begin{array}{rc} & \{l_2 \mathrm{c}\alpha_{1_i} \mathrm{c}\phi_i = O_{7j,x} - \mathrm{c}\alpha_{1i}\left(b + l_1 \mathrm{c}\theta\right)\}^2 \\ + & \underline{\{l_2 \mathrm{c}\alpha_{1_i} \mathrm{s}\phi_i = \mathrm{c}\alpha_{1i}\left(-O_{7j,z} - l_1 \mathrm{s}\theta_i\right)\}^2} \\ & A_i \mathrm{c}\theta_i + B_i \mathrm{s}\theta_i + C_i = 0 \end{array} \tag{3.18}$$

where

$$A_i = 2 l_1 \mathrm{c}\alpha_{1i}\left(b \mathrm{c}\alpha_{1i} - O_{7j,x}\right)$$

$$B_i = 2 l_1 O_{7j,z} \mathrm{c}^2 \alpha_{1i}$$

$$C_i = {O_{7j,x}}^2 - 2b O_{7j,x} \mathrm{c}\alpha_{1i} + \mathrm{c}^2 \alpha_{1i}\left(b^2 + {l_1}^2 - {l_2}^2 + {O_{7j,z}}^2\right)$$

for $C_i - A_i \neq 0$, $i = 1, 2, 3$ and $j = i + 3$. Applying tangent of the half angle substitution to Equation 3.18 and solving for $\theta_i$:

$$\theta_i = 2\tan^{-1}\left(\frac{-B_i \pm \sqrt{{A_i}^2 + {B_i}^2 - {C_i}^2}}{C_i - A_i}\right) \tag{3.19}$$

These equations are used in the simulation to simulate a walking gait for the robot. Also the passive joint variables are computed as

$$\mathrm{c}\phi_i = \frac{O_{7j,x} - \mathrm{c}\alpha_{1i}(b + l_1 \mathrm{c}\theta_i)}{l_2 \mathrm{c}\alpha_{1i}} \quad \mathrm{s}\phi_i = -\frac{O_{7j,z} + l_1 \mathrm{s}\theta_i}{l_2}$$

$$\phi_i = \mathrm{atan2}\left(\mathrm{c}\phi_i, \mathrm{s}\phi_i\right)$$

Now we have the joint variables in terms of O7x, $\Psi_x$ and $\Psi_y$. But we have the position of the end co-ordinate of the feet (end effector). Assume we have (X,Y,Z) and assume the length of the foot coming out of the platform is L

$$O7x = Z - L$$

$$\Psi x = atan(Y/(Z-L))$$

$$\Psi y = atan(X/(Z-L))$$

We get these equations by mainly intuitions of the geometry and I have validated these equation from solidworks.

This IK part is the most important part of the project as the simulation totally depends upon this calculation. To verify this, I have written a MATLAB code that inputs the (X,Y,Z) co-ordinates and outputs the joint variables to get the feet to that position. This code too is validated by using Solidworks, where we choose some arbitrary position of the feet and insert them into the MATLAB code to find joint variables. Then we check what are the value of the joint variables in Solidworks and they match the values we get in MATLAB.

## 2. Velocity and Acceleration Analysis:

We derive most of the components of the velocity analysis by differentiating the equations we found in the IK analysis. These components are used extensively in the dynamics of the system.

Differentiate equation 3.18

$$\dot{A}_i \,\mathrm{c}\,\theta_i + \dot{B}_i \,\mathrm{s}\,\theta_i + \dot{C}_i + (B_i \,\mathrm{c}\,\theta_i - A_i \,\mathrm{s}\,\theta_i)\,\dot{\theta}_i = \overline{D}_i \dot{\overline{O}}_{7j} + E_i \dot{\theta}_i = 0$$

$$\overline{D}_i = \left[-2\,\mathrm{c}\,\alpha_{1i}\,(b + l_1\,\mathrm{c}\,\theta_i) + 2O_{7j,x} \quad 0 \quad 2\,\mathrm{c}\,\alpha_{1i}{}^2\,(O_{7j,z} + l_1\,\mathrm{s}\,\theta_i)\right]$$
$$E_i = 2l_1\,\mathrm{c}\,\alpha_{1i}\,(\mathrm{c}\,\alpha_{1i}\,\mathrm{c}\,\theta_i O_{7j,z} + (-b\,\mathrm{c}\,\alpha_{1i} + O_{7j,x})\,\mathrm{s}\,\theta_i)$$

D and E are defined for further convenience

We define some more matrix variables like,

$$\dot{\Theta} = \begin{bmatrix} \dot{\Theta}_1 & \dot{\Theta}_2 & \dot{\Theta}_3 \end{bmatrix}$$
$$\dot{s} = \begin{bmatrix} \dot{O_{74}} & \dot{O_{75}} & \dot{O_{76}} \end{bmatrix}$$
$$\dot{t} = \begin{bmatrix} \dot{O_{7x}} & \dot{O_{7y}} & \dot{O_{7z}} & \dot{\varphi_x} & \dot{\varphi_y} & \dot{\varphi_z} \end{bmatrix}$$
$$\dot{x}_i = \begin{bmatrix} \dot{O_{7z}} & \dot{\varphi_x} & \dot{\varphi_y} \end{bmatrix}$$
$$\dot{O}_{7x} = J_{O_{7x}} \dot{x}_i$$
$$\dot{O}_{7y} = J_{O_{7y}} \dot{x}_i$$
$$\dot{\varphi_z} = J_{O_{7y}} \dot{x}_i$$

The values of the Jacobians are

$$\mathbf{J}_{O_{7,x}} = \begin{bmatrix} 0 & -\frac{p \,\mathrm{c}\,\psi_y \,\mathrm{s}\,\psi_y (\mathrm{c}\,\psi_x + \mathrm{c}\,\psi_y \,\mathrm{s}\,\psi_x \,\mathrm{s}\,\psi_y)}{2(1+\mathrm{c}\,\psi_x \,\mathrm{c}\,\psi_y)^2} & -\frac{p \,\mathrm{c}\,\psi_y \,\mathrm{s}\,\psi_x (\mathrm{c}\,\psi_x + \mathrm{c}\,\psi_y + \mathrm{s}\,\psi_x \,\mathrm{s}\,\psi_y)}{2(1+\mathrm{c}\,\psi_x \,\mathrm{c}\,\psi_y)^2} \end{bmatrix}$$

$$\mathbf{J}_{O_{7y}} = \begin{bmatrix} 0 & -\frac{p \,\mathrm{c}\,\psi_y \,\mathrm{s}\,\psi_y (\mathrm{c}\,\psi_x + \mathrm{c}\,\psi_y)}{(1+\mathrm{c}\,\psi_x \,\mathrm{c}\,\psi_y)^2} & \frac{p \,\mathrm{s}\,\psi_x \begin{pmatrix} \mathrm{s}^2\psi_x \mathrm{s}^4\psi_y - \mathrm{c}^4\psi_y - \mathrm{c}^2\psi_x \,\mathrm{c}\,2\psi_y - \\ \frac{1}{4}\,\mathrm{c}\,\psi_x \left(5\,\mathrm{c}\,\psi_y + 3\,\mathrm{c}\,3\psi_y\right) \end{pmatrix}}{(1+\mathrm{c}\,\psi_x \,\mathrm{c}\,\psi_y)^3} \end{bmatrix}$$

$$\mathbf{J}_{\psi_z} = \begin{bmatrix} 0 & \frac{-\mathrm{s}\,\psi_y}{1+\mathrm{c}\,\psi_x \,\mathrm{c}\,\psi_y} & \frac{-\mathrm{s}\,\psi_x}{1+\mathrm{c}\,\psi_x \,\mathrm{c}\,\psi_y} \end{bmatrix}$$

$$\dot{t} = \mathbf{J}_{x_i,t} \dot{x}_i$$

$$\mathbf{J}_{x_i,t} = \begin{bmatrix} \mathbf{J}_{O_{7,x}} & \mathbf{J}_{O_{7,y}} & \mathbf{I}_{3\times 3} & \mathbf{J}_z \end{bmatrix}^T$$


$$\mathbf{J}_{O_{7,x}} = \begin{bmatrix} 0 & -\frac{p \,\mathrm{c}\,\psi_y \,\mathrm{s}\,\psi_y (\mathrm{c}\,\psi_x + \mathrm{c}\,\psi_y \,\mathrm{s}\,\psi_x \,\mathrm{s}\,\psi_y)}{2(1+\mathrm{c}\,\psi_x \,\mathrm{c}\,\psi_y)^2} & -\frac{p \,\mathrm{c}\,\psi_y \,\mathrm{s}\,\psi_x (\mathrm{c}\,\psi_x + \mathrm{c}\,\psi_y + \mathrm{s}\,\psi_x \,\mathrm{s}\,\psi_y)}{2(1+\mathrm{c}\,\psi_x \,\mathrm{c}\,\psi_y)^2} \end{bmatrix}$$

$$\mathbf{J}_{O_{7y}} = \begin{bmatrix} 0 & -\frac{p \,\mathrm{c}\,\psi_y \,\mathrm{s}\,\psi_y (\mathrm{c}\,\psi_x + \mathrm{c}\,\psi_y)}{(1+\mathrm{c}\,\psi_x \,\mathrm{c}\,\psi_y)^2} & \frac{p \,\mathrm{s}\,\psi_x \begin{pmatrix} \mathrm{s}^2\psi_x \mathrm{s}^4\psi_y - \mathrm{c}^4\psi_y - \mathrm{c}^2\psi_x \,\mathrm{c}\,2\psi_y - \\ \frac{1}{4}\,\mathrm{c}\,\psi_x \left(5\,\mathrm{c}\,\psi_y + 3\,\mathrm{c}\,3\psi_y\right) \end{pmatrix}}{(1+\mathrm{c}\,\psi_x \,\mathrm{c}\,\psi_y)^3} \end{bmatrix}$$

$$\mathbf{J}_{\psi_z} = \begin{bmatrix} 0 & \frac{-\mathrm{s}\,\psi_y}{1+\mathrm{c}\,\psi_x \,\mathrm{c}\,\psi_y} & \frac{-\mathrm{s}\,\psi_x}{1+\mathrm{c}\,\psi_x \,\mathrm{c}\,\psi_y} \end{bmatrix}$$

$$\dot{\bar{S}} = \mathbf{J}_{t,S}\dot{\bar{t}}$$

$$\text{where } \mathbf{J}_{t,S} = \begin{bmatrix} \mathbf{J}_{t,O_{74}} \\ \mathbf{J}_{t,O_{75}} \\ \mathbf{J}_{t,O_{76}} \end{bmatrix} \text{ and}$$

$$\mathbf{J}_{t,O_{7j}} = \begin{bmatrix} 1 & 0 & 0 & 0 & \begin{pmatrix} -p\,\mathrm{s}\,\psi_y \\ \mathrm{c}\,(\alpha_{4j}+\psi_z) \end{pmatrix} & -p\,\mathrm{c}\,\psi_y\,\mathrm{s}\,(\alpha_{4j}+\psi_z) \\ 0 & 1 & 0 & p\begin{pmatrix} \begin{pmatrix} \mathrm{c}\,\psi_x\,\mathrm{s}\,\psi_y \\ \mathrm{c}\,(\alpha_{4j}+\psi_z) \end{pmatrix} \\ -\,\mathrm{s}\,\psi_x\,\mathrm{s}\,(\alpha_{4j}+\psi_z) \end{pmatrix} & \begin{pmatrix} p\,\mathrm{s}\,\psi_x\,\mathrm{c}\,\psi_y \\ \mathrm{c}\,(\alpha_{4j}+\psi_z) \end{pmatrix} & p\begin{pmatrix} \psi_x\,\mathrm{c}\,(\alpha_{4j}+\psi_z) \\ -\begin{pmatrix} \mathrm{s}\,\psi_x\,\mathrm{s}\,\psi_y \\ \mathrm{s}\,(\alpha_{4j}+\psi_z) \end{pmatrix} \end{pmatrix} \\ 0 & 0 & 1 & p\begin{pmatrix} \begin{pmatrix} \mathrm{s}\,\psi_x\,\mathrm{s}\,\psi_y \\ \mathrm{c}\,(\alpha_{4j}+\psi_z) \end{pmatrix} \\ +\,\mathrm{c}\,\psi_x\,\mathrm{s}\,(\alpha_{4j}+\psi_z) \end{pmatrix} & \begin{pmatrix} -p\,\mathrm{c}\,\psi_x\,\mathrm{c}\,\psi_y \\ \mathrm{c}\,(\alpha_{4j}+\psi_z) \end{pmatrix} & p\begin{pmatrix} \mathrm{s}\,\psi_x\,\mathrm{c}\,(\alpha_{4j}+\psi_z) \\ +\begin{pmatrix} \mathrm{c}\,\psi_x\,\mathrm{s}\,\psi_y \\ \mathrm{s}\,(\alpha_{4j}+\psi_z) \end{pmatrix} \end{pmatrix} \end{bmatrix}$$

Also,

$$\begin{pmatrix} \overline{D}_1 & 0_{1x3} & 0_{1x3} \\ 0_{1x3} & \overline{D}_2 & 0_{1x3} \\ 0_{1x3} & 0_{1x3} & \overline{D}_3 \end{pmatrix} \dot{\bar{S}} + \begin{pmatrix} E_1 & 0 & 0 \\ 0 & E_2 & 0 \\ 0 & 0 & E_3 \end{pmatrix} \dot{\bar{\theta}} = \mathbf{J}_S\dot{\bar{S}} + \mathbf{J}_\theta\dot{\bar{\theta}} = \overline{0}$$

So $J_s$ is known and $J_\theta$ is also known from the equation at the start of the velocity analysis

We have calculated $\dot{\bar{S}}$ above and hence this formula becomes

$$\dot{\bar{\theta}} = -\mathbf{J}_\theta^{-1}\mathbf{J}_S\mathbf{J}_{t,S}\mathbf{J}_{x_i,t}\dot{\bar{x}}_i = \mathbf{J}_{x_i,\theta}\dot{\bar{x}}_i$$

And by taking derivative of the equation we get

$$\ddot{\bar{\theta}} = \dot{\mathbf{J}}_{x_i,\theta}\dot{\bar{x}}_i + \mathbf{J}_{x_i,\theta}\ddot{\bar{x}}_i$$

In a similar fashion we can also find the angles at the passive joint i.e. $O_{14}$ ,$O_{25,}$ ,$O_{26}$ i.e.$\dot{\phi}\iota$ which will be useful in the dynamic analysis

Here I have directly written the equations for the above mentioned

$$\dot{\bar{\phi}} = \left(\mathbf{J}_{S,\phi}\mathbf{J}_{x_i,S} + \mathbf{J}_{\theta,\phi}\mathbf{J}_{x_i,\theta}\right)\dot{\bar{x}}_i = \mathbf{J}_{x_i,\phi}\dot{\bar{x}}_i$$

$$\ddot{\bar{\phi}} = \dot{\mathbf{J}}_{x_i,\phi}\dot{\bar{x}} + \mathbf{J}_{x_i,\phi}\ddot{\bar{x}}$$

Where

$$\mathbf{J}_{S_i,\phi_i} = \begin{bmatrix} 0 & 0 & \frac{-1}{l_2 c\phi_i} \end{bmatrix}$$

$$J_{\theta_i,\phi_i} = -\frac{l_1 \, \mathrm{c}\,\theta_i}{\mathrm{c}\,\phi_i l_2}$$

Similarly like in the IK analysis we can make this in the form of end point joint co-ordinates velocities [$\dot{X}$ $\dot{Y}$ $\dot{Z}$] by finding the $\dot{x\iota}$ by taking derivative of equations

$$O7x = Z - L$$

$$\Psi x = atan(Y/(Z - L))$$

$$\Psi y = atan(X/(Z - L))$$

These calculations are not extensively used in any simulation of this project, but it can be used to track the trajectory of any path given both the position and velocity of the end effector.

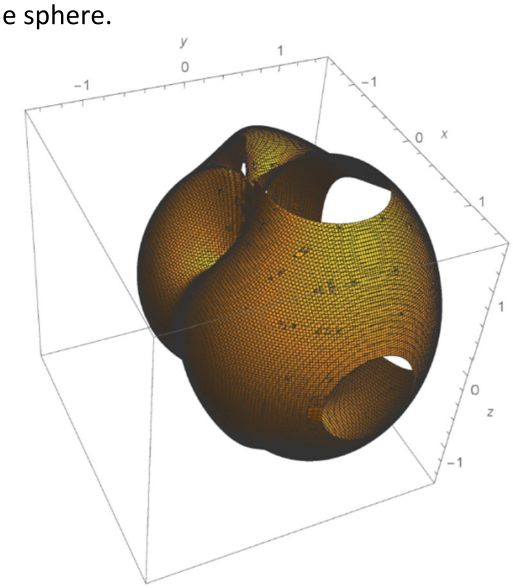
### 3. Singularity Analysis

There are 3 types of singularities for PMs. In order to investigate the singular configurations of this PM, the Jacobian matrices given are used. The 1st type singularities, also known as inverse kinematics singularities, occur when Det [$J_\theta$] = 0. This is a surface in the joint space, called as the singularity surface. Formulating the singularity surface equation from Det [$J_\theta$] = 0 is computationally challenging. Instead, notice that inverse kinematic singularities occur when either of the links are in extended or folded configurations, where the two links of the limbs become collinear. So to speak, the singularity conditions can be analyzed separately for each link. Since the links of the PM are identical and the base and the platform are symmetrical, the conditions for each links are equivalent to each other. Without loss of generality, link 1 can be analyzed. Link 1 is in extended or folded configuration the square root term in IK Equation is equal to zero. The 1$^{st}$ type singularities are obtained:

$$A_1^2 + B_1^2 - C_1^2 =$$

$$\left\{ - \begin{pmatrix} -4b^2 + 4{l_1}^2 - 8l_1l_2 + 4{l_2}^2 - \\ 4{O_{7,z}}^2 - 4b^2{\mathrm{c}_x}^2 - b^2{\mathrm{c}_x}^4 \\ +8{l_1}^2\mathrm{c}_x\mathrm{c}_y - 16l_1l_2\mathrm{c}_x\mathrm{c}_y + \\ 8{l_2}^2\mathrm{c}_x\mathrm{c}_y - 8{O_{7,z}}^2\mathrm{c}_x\mathrm{c}_y + \\ 6b^2{\mathrm{c}_x}^2{\mathrm{c}_y}^2 + 4{l_1}^2{\mathrm{c}_x}^2{\mathrm{c}_y}^2 - \\ 8l_1l_2{\mathrm{c}_x}^2{\mathrm{c}_y}^2 + 4{l_2}^2{\mathrm{c}_x}^2{\mathrm{c}_y}^2 - \\ 4{O_{7,z}}^2{\mathrm{c}_x}^2{\mathrm{c}_y}^2 - 9b^2{\mathrm{c}_y}^4 + \\ 8bO_{7,z}\mathrm{s}_y + 16bO_{7,z}\mathrm{c}_x\mathrm{c}_y\mathrm{s}_y + \\ 8bO_{7,z}{\mathrm{c}_x}^2{\mathrm{c}_y}^2\mathrm{s}_y - \\ 4b^2{\mathrm{s}_y}^2 - 8b^2\mathrm{c}_x\mathrm{c}_y{\mathrm{s}_y}^2 - \\ 4b^2{\mathrm{c}_x}^2{\mathrm{c}_y}^2{\mathrm{s}_y}^2 - 4b^2{\mathrm{s}_x}^2{\mathrm{s}_y}^2 - \\ b^2{\mathrm{s}_x}^4{\mathrm{s}_y}^4 - 2b^2{\mathrm{c}_x}^2{\mathrm{s}_x}^2{\mathrm{s}_y}^2 + \\ 6b^2{\mathrm{c}_y}^2{\mathrm{s}_x}^2{\mathrm{s}_y}^2 + 12b^2{\mathrm{c}_y}^2 \end{pmatrix} \begin{pmatrix} -4b^2 + 4{l_1}^2 + 8l_1l_2 + 4{l_2}^2 - \\ 4{O_{7,z}}^2 - 4b^2{\mathrm{c}_x}^2 - b^2{\mathrm{c}_x}^4 + \\ 8{l_1}^2\mathrm{c}_x\mathrm{c}_y + 16l_1l_2\mathrm{c}_x\mathrm{c}_y + \\ 8{l_2}^2\mathrm{c}_x\mathrm{c}_y - 8{O_{7,z}}^2\mathrm{c}_x\mathrm{c}_y + \\ 12b^2{\mathrm{c}_y}^2 - b^2{\mathrm{s}_x}^4{\mathrm{s}_y}^4 + \\ 6b^2{\mathrm{c}_x}^2{\mathrm{c}_y}^2 + 4{l_1}^2{\mathrm{c}_x}^2{\mathrm{c}_y}^2 + \\ 8l_1l_2{\mathrm{c}_x}^2{\mathrm{c}_y}^2 + 4{l_2}^2{\mathrm{c}_x}^2{\mathrm{c}_y}^2 - \\ 4{O_{7,z}}^2{\mathrm{c}_x}^2{\mathrm{c}_y}^2 - 9b^2{\mathrm{c}_y}^4 + \\ 8bO_{7,z}\mathrm{s}_y + 16bO_{7,z}\mathrm{c}_x\mathrm{c}_y\mathrm{s}_y + \\ 8bO_{7,z}{\mathrm{c}_x}^2{\mathrm{c}_y}^2\mathrm{s}_y - \\ 4b^2{\mathrm{s}_y}^2 - 8b^2\mathrm{c}_x\mathrm{c}_y{\mathrm{s}_y}^2 - \\ 4b^2{\mathrm{c}_x}^2{\mathrm{c}_y}^2{\mathrm{s}_y}^2 - 4b^2{\mathrm{s}_x}^2{\mathrm{s}_y}^2 - \\ 2b^2{\mathrm{c}_x}^2{\mathrm{s}_x}^2{\mathrm{s}_y}^2 + 6b^2{\mathrm{c}_y}^2{\mathrm{s}_x}^2{\mathrm{s}_y}^2 \end{pmatrix} \right\} = 0$$

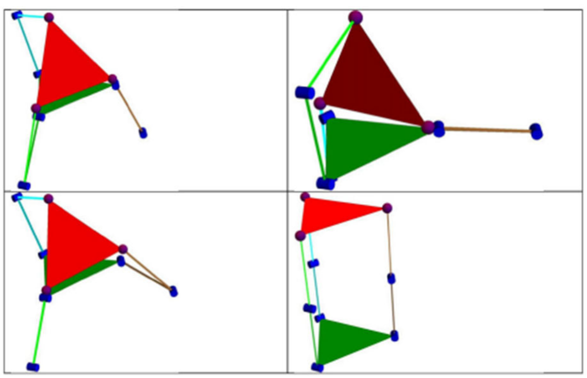
Here subscripts x and y refers to $\Psi$x and $\Psi$y. When the condition given in the above Equation is satisfied, the upper and lower links at any of the limbs are collinear. As stated above, 1st type singularities represent the configurations where the mechanism is at the inner or outer boundaries of its workspace. Therefore, it is more convenient to represent the singularity surface in the task space, rather than the joint space. Figure 3.5 represents the singularity surface of the 1st limb and Figure 3.6 represents several inverse kinematic singular configurations of the PM. The singularity surface is obtained in spherical coordinates. The heave of the moving platform, $O_{7z}$, is considered as the radius of a sphere and rotations of the moving platform, $\Psi$x and $\Psi$y, are considered as azimuth and polar angles of the sphere.



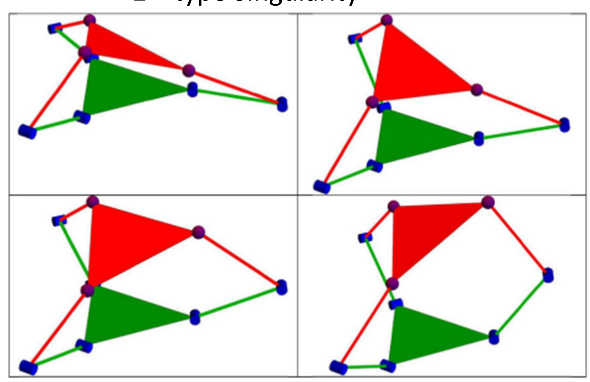

The 2nd type singularities, also known as forward kinematic singularities occur when Det $[J_{xi}] = 0$.

The 2nd type singularities refer to the configurations of the PM when any of the upper links lie on the plane of the moving platform. Unlike the 1st type of singularities, the 2nd type of singularities cannot be evaluated per link. As a result, the singularity condition given in Equation 3.52 is a very long and highly non-linear equation in terms of the independent task space parameters. For this reason a singularity surface for this type is not plotted. Figure 3.7 represents several configurations that has this type of singularities. The 3rd type of the singularities are obtained when
Det $[J_{xi}]$ = Det $[J_\theta]$ =0. These matrices can be taken from the velocity kinematics that were previously derived.
This type of singularities occur platform possesses finite motions. This happens when the actuator can still move but the platform remains stationary. The 3rd type of singularities are not observed for the PM studied here because Det $[J_\theta]$ and Det $[J_{xi}]$ are not simultaneously equal to zero. The singularity is not simulated extensively but is borrowed from research papers and journals available online arnd are sourced at the end.

1st Type Singularity



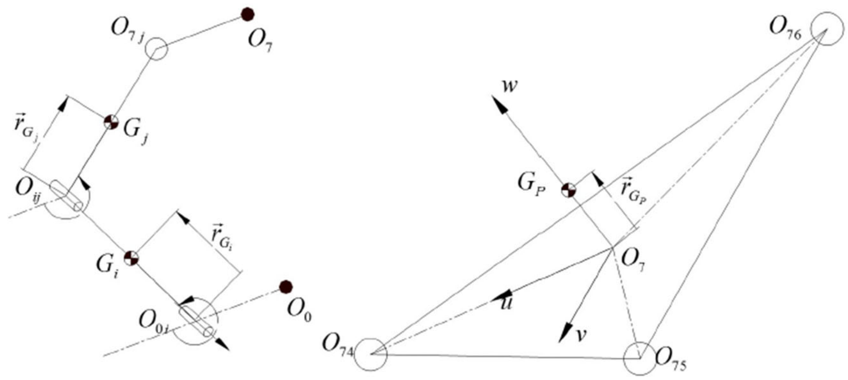
2nd type Singularity



4. **Dynamics of the System:**

Dynamic analysis consists of obtaining the required input torques in order to track a desired trajectory (inverse dynamics) or obtaining the trajectory of the moving platform with given input torques (forward dynamics). In this project, only inverse dynamics of the PM is formulated. In order to formulate the inverse dynamics of the 3-RRS PM, firstly velocities and accelerations of mass centers are determined. Then the forces on the system are defined.

**6.1 Determining mass center velocities and accelerations:**




Here throughout the subscript I denotes the lower(actuated) link and the subscript j denotes the lower(passive) link. The subscript p denotes the platform.Starting from the lower link at $i_{th}$ base, the distance of the center of mass(COM) $G_i$ from point $O_{1i}$ is shown with $d_1$ and its mass is $m_1$. The distance of the COM $G_j$ of the upper link at $i_{th}$ chain from point $O_{ij}$ is denoted by d2 and mass is shown with m2. Due to symmetry of the chains, $G_i$ and $G_j$ are assumed to lie on $O_{0i}O_{ij}$ and $O_{ij}O_{7j}$, respectively. Since the cross section of the moving platform is an equilateral triangle, the center of mass lies on the w-axis. The distance of the COM of the platform from point O7 along w-axis is shown with dp and its mass is called mp as presented in the figure above. The position, velocity and acceleration of $G_i$ can be formulated as:

$$\bar{p}_{G_i} = \overline{O_0 G_i} = \begin{bmatrix} \mathrm{c}\,\alpha_{1i}\,(b + d_1\,\mathrm{c}\,\theta_i) \\ \mathrm{s}\,\alpha_{1i}\,(b + d_1\,\mathrm{c}\,\theta_i) \\ -d_1\,\mathrm{s}\,\theta_i \end{bmatrix}$$

$$\bar{v}_{G_i} = \dot{\bar{p}}_{G_i} = \mathbf{J}_{\theta_i,G_i}\dot{\theta}_i$$

where

$$\mathbf{J}_{\theta_i,G_i} = -\begin{bmatrix} d_1\,\mathrm{c}\,\alpha_{1i}\,\mathrm{s}\,\theta_i \\ d_1\,\mathrm{s}\,\alpha_{1i}\,\mathrm{s}\,\theta_i \\ d_1\,\mathrm{c}\,\theta_i \end{bmatrix}$$

$$\bar{a}_{G_i} = \ddot{\bar{p}}_{G_i} = \dot{\mathbf{J}}_{\theta_i,G_i}\dot{\theta}_i + \mathbf{J}_{\theta_i,G_i}\ddot{\theta}_i$$

where

$$\dot{\mathbf{J}}_{\theta_i,G_i} = \begin{bmatrix} -d_1\,\mathrm{c}\,\alpha_{1i}\,\mathrm{c}\,\theta_i \\ -d_1\,\mathrm{s}\,\alpha_{1i}\,\mathrm{c}\,\theta_i \\ d_1\,\mathrm{s}\,\theta_i \end{bmatrix}\dot{\theta}_i$$

The position, velocity and acceleration of $G_j$ are formulated as:

$$\bar{p}_{G_j} = \overline{O_0 G_j} = \begin{bmatrix} \mathrm{c}\,\alpha_{1i}\,(b + l_1\,\mathrm{c}\,\theta_i + d_2\,\mathrm{c}\,\phi_i) \\ \mathrm{s}\,\alpha_{1i}\,(b + l_1\,\mathrm{c}\,\theta_i + d_2\,\mathrm{c}\,\phi_i) \\ -l_1\,\mathrm{s}\,\theta_i - d_2\,\mathrm{s}\,\phi_i \end{bmatrix}$$

$$\bar{v}_{G_j} = \dot{\bar{p}}_{G_j} = \mathbf{J}_{\theta_i,G_j}\dot{\theta}_i + \mathbf{J}_{\phi_i,G_j}\dot{\phi}_i$$

$$\mathbf{J}_{\theta_i,G_j} = -\begin{bmatrix} l_1\,\mathrm{c}\,\alpha_{1i}\,\mathrm{s}\,\theta_i \\ l_1\,\mathrm{s}\,\alpha_{1i}\,\mathrm{s}\,\theta_i \\ l_1\,\mathrm{c}\,\theta_i \end{bmatrix}$$

and

$$\mathbf{J}_{\phi_i,G_j} = -\begin{bmatrix} d_2\,\mathrm{c}\,\alpha_{1i}\,\mathrm{s}\,\phi_i \\ d_2\,\mathrm{s}\,\alpha_{1i}\,\mathrm{s}\,\phi_i \\ d_2\,\mathrm{c}\,\phi_i \end{bmatrix}$$

$$\bar{a}_{G_j} = \ddot{\bar{p}}_{G_j} = \dot{\mathbf{J}}_{\theta_i,G_j}\dot{\theta}_i + \mathbf{J}_{\theta_i,G_j}\ddot{\theta}_i + \dot{\mathbf{J}}_{\phi_i,G_j}\dot{\phi}_i + \mathbf{J}_{\phi_i,G_j}\ddot{\phi}_i$$

where

$$\dot{\mathbf{J}}_{\theta_i,G_j} = \begin{bmatrix} -l_1\,\mathrm{c}\,\alpha_{1i}\,\mathrm{c}\,\theta_i \\ -l_1\,\mathrm{s}\,\alpha_{1i}\,\mathrm{c}\,\theta_i \\ l_1\,\mathrm{s}\,\theta_i \end{bmatrix} \dot{\theta}_i$$

and

$$\dot{\mathbf{J}}_{\phi_i,G_j} = \begin{bmatrix} l_1\,\mathrm{c}\,\alpha_{1i}\,\mathrm{s}\,\theta_i \\ l_1\,\mathrm{s}\,\alpha_{1i}\,\mathrm{s}\,\theta_i \\ l_1\,\mathrm{c}\,\theta_i \end{bmatrix} \dot{\phi}_i$$

The mass center of the platform:

$$\bar{p}_{G_P} = \overline{O_0 G_P} = \vec{r}_P + \begin{bmatrix} d_p\,\mathrm{s}\,\psi_y \\ -d_p\,\mathrm{s}\,\psi_x\,\mathrm{c}\,\psi_y \\ d_p\,\mathrm{c}\,\psi_x\,\mathrm{c}\,\psi_y \end{bmatrix}$$

$$\bar{v}_{G_P} = \dot{\bar{p}}_{G_P} = \dot{\vec{r}}_P + \mathbf{J}_{\psi,G_P}\dot{\psi}$$

where

$$\mathbf{J}_{\psi,G_P} = \begin{bmatrix} 0 & d_p\,\mathrm{c}\,\psi_y & 0 \\ -d_p\,\mathrm{c}\,\psi_x\,\mathrm{c}\,\psi_y & d_p\,\mathrm{s}\,\psi_x\,\mathrm{s}\,\psi_y & 0 \\ -d_p\,\mathrm{c}\,\psi_y\,\mathrm{s}\,\psi_x & -d_p\,\mathrm{c}\,\psi_x\,\mathrm{s}\,\psi_y & 0 \end{bmatrix}$$

$$\bar{a}_{G_P} = \ddot{\bar{p}}_{G_P} = \ddot{\vec{r}}_P + \dot{\mathbf{J}}_{\psi,G_P}\dot{\psi} + \mathbf{J}_{\psi,G_P}\ddot{\psi}$$

where

$$\dot{\mathbf{J}}_{\psi,G_P} = d_p \begin{bmatrix} 0 & -\,\mathrm{s}\,\psi_y\dot{\psi}_y & 0 \\ \left(\mathrm{s}\,\psi_x\,\mathrm{c}\,\psi_y\dot{\psi}_x + \mathrm{c}\,\psi_x\,\mathrm{s}\,\psi_y\dot{\psi}_y\right) & \left(\mathrm{c}\,\psi_x\,\mathrm{s}\,\psi_y\dot{\psi}_x + \mathrm{s}\,\psi_x\,\mathrm{c}\,\psi_y\dot{\psi}_y\right) & 0 \\ \left(\mathrm{s}\,\psi_x\,\mathrm{s}\,\psi_y\dot{\psi}_y - \mathrm{c}\,\psi_x\,\mathrm{c}\,\psi_y\dot{\psi}_x\right) & \left(\mathrm{s}\,\psi_x\,\mathrm{s}\,\psi_y\dot{\psi}_x - \mathrm{c}\,\psi_x\,\mathrm{c}\,\psi_y\dot{\psi}_y\right) & 0 \end{bmatrix}$$

**6.2 Calculating the Inertial, Gravitational, External Forces and Moments**

Here throughout the subscript I denotes the lower(actuated) link and the subscript j denotes the lower(passive) link. The subscript p denotes the platform. Here m1 is the mass of the lower link(actuated) and m2 is the mass of the upper link(unactuated). Here throughout the subscript I denotes the lower(actuated) link and the subscript j denotes the lower(passive) link. The subscript p denotes the platform

The internal forces (comprising of gravitational and inertial forces) acting on the lower links, upper links and the moving platform can be formulated as:

$$\bar{F}_i^{\mathrm{int}} = m_1\left(\bar{g} + \bar{a}_{G_i}\right)$$

$$\bar{F}_j^{\text{int}} = m_2\left(\bar{g} + \bar{a}_{G_j}\right)$$

$$\bar{F}_P^{\text{int}} = m_P\left(\bar{g} + \bar{a}_{G_P}\right)$$

Where $g = [0\ 0\ 9.81]$.
The inertial moments occurring on the links and platform can be expressed as:

$$\bar{M}_1^{\text{in}} = I_{1,yy}\begin{bmatrix}\ddot{\theta}_1 & \ddot{\theta}_2 & \ddot{\theta}_3\end{bmatrix}^T$$

$$\bar{M}_2^{\text{in}} = I_{2,yy}\begin{bmatrix}\ddot{\phi}_1 & \ddot{\phi}_2 & \ddot{\phi}_3\end{bmatrix}^T$$

$$\bar{M}_P^{\text{in}} = \mathbf{I}_P\bar{\alpha}_P$$

where $I_{1;yy}$ and $I_{2;yy}$ are the yy component of the moments of inertia of the lower and upper links about their COM, IP is the moment of inertia matrix of the mobile platform and $\alpha_P$ is the angular acceleration array of the moving platform with respect to the fixed frame. $\alpha_p$ is calculated as:

$$\tilde{\omega}_P = \dot{\mathbf{R}}\mathbf{R}^T = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \Rightarrow \bar{\omega}_P = \begin{bmatrix}\omega_x \\ \omega_y \\ \omega_z\end{bmatrix} = \mathbf{J}_{t,\omega}\dot{\bar{t}}$$
$$\Rightarrow \bar{\alpha}_P = \dot{\mathbf{J}}_{t,\omega}\dot{\bar{t}} + \mathbf{J}_{t,\omega}\ddot{\bar{t}}$$

where

$$\mathbf{J}_{t,\omega} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & \mathrm{s}\,\psi_y \\ 0 & 0 & 0 & 0 & \mathrm{c}\,\psi_x & -\mathrm{s}\,\psi_x\,\mathrm{c}\,\psi_y \\ 0 & 0 & 0 & 0 & \mathrm{s}\,\psi_x & \mathrm{c}\,\psi_x\,\mathrm{c}\,\psi_y \end{bmatrix}$$

and

$$\dot{\mathbf{J}}_{t,\omega} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \mathrm{c}\,\psi_y\dot{\psi}_y \\ 0 & 0 & 0 & 0 & -\mathrm{s}\,\psi_x\dot{\psi}_x & \mathrm{s}\,\psi_x\,\mathrm{s}\,\psi_y\dot{\psi}_y - \mathrm{c}\,\psi_x\,\mathrm{c}\,\psi_y\dot{\psi}_x \\ 0 & 0 & 0 & 0 & \mathrm{c}\,\psi_x\dot{\psi}_x & -\mathrm{s}\,\psi_x\,\mathrm{c}\,\psi_y\dot{\psi}_x - \mathrm{c}\,\psi_x\,\mathrm{s}\,\psi_y\dot{\psi}_y \end{bmatrix}$$

$$\bar{F}_P^{ext} = \begin{bmatrix}F_x^{ext} & F_y^{ext} & F_z^{ext}\end{bmatrix}^T$$

The input torques applied on the active revolute joints are formulated as:

$$\bar{\tau}^a = \begin{bmatrix}\tau_1 & \tau_2 & \tau_3\end{bmatrix}^T$$

6.3 **Lagrange's Method to derive Dynamics**
The dynamic analysis of the PMs have been a complicated task due to their close chain structure and closed loop chains with a large number of passive variables where this PM has 3 1dof joints and 3 3dof

joints as passive variables. Three main methods can be applied to perform the dynamic analysis of PMs: Newton-Euler classical procedure, application of Lagrange's equations and multipliers, and finally virtual work principle. Here I have shown the Lagrange's method to derive the dynamics of the PM
The $n_{th}$ Lagrange's equation for the dynamics of the PM is written as

$$\sum_{k=1}^{9} \lambda_k \frac{\partial \Gamma_k}{\partial q_n} = L_n - Q_n^* \qquad \text{………………(6.3.1)}$$

where λk are Lagrange's multipliers, $\Gamma_k$ are constraint equations, $q_n$ is $n_{th}$ generalized coordinates. Here throughout the subscript I denotes the lower(actuated) link and the subscript j denotes the lower(passive) link. The subscript p denotes the platform
L is the Lagrange function,

$$L_n = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_n}\right) - \frac{\partial L}{\partial q_n}$$

Here $Q_n$ is the corresponding generalized force. In the inverse dynamic analysis, generalized coordinates ($\bar{q}$) include all of the joint and task space parameters

$$\bar{q} = \begin{bmatrix} \bar{\theta}^T & \bar{\phi}^T & \bar{O}_7^T & \bar{\psi}^T \end{bmatrix}^T$$

In order to formulate the Lagrangian function, the kinetic and potential energies of the moving platform and links has to be calculated. The kinetic energies of the $i_{th}$
lower and upper links and the moving platform can be formulated as

$$KE_i = \tfrac{1}{2}\left(I_{1,yy} + m_1 {d_1}^2\right)\dot{\theta}_i^2$$

$$KE_j = \tfrac{1}{2}\left[m_2 {l_1}^2 \dot{\theta}_i^2 + \left(I_{2,yy} + m_2 {d_2}^2\right)\dot{\phi}_i^2 + 2 m_2 l_1 d_2 \,\mathrm{c}\left(\theta_i - \phi_i\right)\dot{\phi}_i \dot{\theta}_i\right]$$

$$KE_P = \tfrac{1}{2}\left[m_p V_{G_P}^2 + \bar{\omega}_P^T \mathbf{I}_P \bar{\omega}_P\right]$$

The potential energy expressions of ith the lower and upper links and the moving platform can be formulated as:

$$PE_i = -g d_1 \,\mathrm{s}\,\theta_i$$

$$PE_j = -g\left(l_1 \,\mathrm{s}\,\theta_i + d_2 \,\mathrm{s}\,\phi_i\right)$$

$$PE_P = g\left(O_{7,z} + d_p \,\mathrm{c}\,\psi_x \,\mathrm{c}\,\psi_y\right)$$

Then the Lagrangian function L =ΣKE - ΣPE can be formulated as:

$$L = \left\{ \begin{array}{c}
-g m_P\left(d_P \,\mathrm{c}\,\psi_x \,\mathrm{c}\,\psi_y + O_{7,z}\right) + d_1 g m_1 \,\mathrm{s}\,\theta_1 + d_1 g m_1 \,\mathrm{s}\,\theta_2 + d_1 g m_1 \,\mathrm{s}\,\theta_3 \\
-g m_2\left(-l_1 \,\mathrm{s}\,\theta_1 - d_2 \,\mathrm{s}\,\phi_1\right) - g m_2\left(-l_1 \,\mathrm{s}\,\theta_2 - d_2 \,\mathrm{s}\,\phi_2\right) \\
-g m_2\left(-l_1 \,\mathrm{s}\,\theta_3 - d_2 \,\mathrm{s}\,\phi_3\right) + 0.5 m_P \dot{O}_{7,x}^2 + 0.5 m_P \dot{O}_{7,y}^2 + 0.5 m_P \dot{O}_{7,z}^2 \\
+0.5\left(I_{G_1} + {d_1}^2 m_1 + {l_1}^2 m_2\right)\left(\dot{\theta}_1^2 + \dot{\theta}_2^2 + \dot{\theta}_3^2\right) \\
+d_2 l_1 m_2 \,\mathrm{c}\left(\theta_1 - \phi_1\right)\dot{\theta}_1\dot{\phi}_1 + d_2 l_1 m_2 \,\mathrm{c}\left(\theta_2 - \phi_2\right)\dot{\theta}_2\dot{\phi}_2 \\
+d_2 l_1 m_2 \,\mathrm{c}\left(\theta_3 - \phi_3\right)\dot{\theta}_3\dot{\phi}_3 + 0.5\left(I_{G_2} + {d_2}^2 m_2\right)\left(\dot{\phi}_1^2 + \dot{\phi}_2^2 + \dot{\phi}_3^2\right) \\
+d_P m_P \,\mathrm{c}\,\psi_y\left(-\,\mathrm{c}\,\psi_x \dot{O}_{7,y} - \mathrm{s}\,\psi_x \dot{O}_{7,z}\right)\dot{\psi}_x \\
+\left(0.5 I_{P,x} + 0.25 {d_P}^2 m_P + 0.25 {d_P}^2 m_P \,\mathrm{c}\,2\psi_y\right)\dot{\psi}_x^2 \\
+\left(0.25 I_{P,y} + 0.25 I_{P,z} + 0.5 {d_P}^2 m_P + \left(0.25 I_{P,y} - 0.25 I_{P,z}\right)\mathrm{c}\,2\psi_x\right)\dot{\psi}_y^2 \\
+d_P m_P\left(\mathrm{c}\,\psi_y \dot{O}_{7,x} + \mathrm{s}\,\psi_y\left(\mathrm{s}\,\psi_x \dot{O}_{7,y} - \mathrm{s}\,\psi_x \dot{O}_{7,z}\right)\right)\dot{\psi}_y \\
+\left(I_{P,x} \,\mathrm{s}\,\psi_y \dot{\psi}_x + \left(-I_{P,y} + I_{P,z}\right)\mathrm{c}\,\psi_x \,\mathrm{s}\,\psi_x \,\mathrm{c}\,\psi_y \dot{\psi}_y\right)\dot{\psi}_z \\
+0.5\left(I_{P,x}\mathrm{s}^2\psi_y + \mathrm{c}^2\psi_y\left(I_{P,z}\mathrm{c}^2\psi_x + I_{P,y}\mathrm{s}^2\psi_x\right)\right)\dot{\psi}_z^2
\end{array} \right\}$$

Then $L_n$ term in Equation 3.75 is formulated for n = 1; 2; …, 12 as:

$$L_1 = \left\{ \begin{array}{c} g\left(-d_1 m_1 - l_2 m_2\right) c\theta_1 + d_2 l_2 m_2 s\left(\theta_1 - \phi_1\right)\dot{\phi}_1^2 \\ + \left(I_{1,yy} + {d_1}^2 m_1 + {l_2}^2 m_2\right)\ddot{\theta}_1 + d_2 l_2 m_2 c\left(\theta_1 - \phi_1\right)\ddot{\phi}_1 \end{array} \right\}$$

$$L_2 = \left\{ \begin{array}{c} g\left(-d_1 m_1 - l_2 m_2\right) \mathrm{c}\,\theta_2 + d_2 l_2 m_2 \,\mathrm{s}\left(\theta_2 - \phi_2\right)\dot{\phi}_2^2 \\ + \left(I_{1,yy} + {d_1}^2 m_1 + {l_2}^2 m_2\right)\ddot{\theta}_2 + d_2 l_2 m_2 \,\mathrm{c}\left(\theta_2 - \phi_2\right)\ddot{\phi}_2 \end{array} \right\}$$

$$L_3 = \left\{ \begin{array}{c} g\left(-d_1 m_1 - l_2 m_2\right) \mathrm{c}\,\theta_3 + d_2 l_2 m_2 \,\mathrm{s}\left(\theta_3 - \phi_3\right)\dot{\phi}_3^2 \\ + \left(I_{1,yy} + {d_1}^2 m_1 + {l_2}^2 m_2\right)\ddot{\theta}_3 + d_2 l_2 m_2 \,\mathrm{c}\left(\theta_3 - \phi_3\right)\ddot{\phi}_3 \end{array} \right\}$$

$$L_4 = -d_2 g m_2 \,\mathrm{c}\,\phi_1 + I_{2,yy}\ddot{\phi}_1 + d_2 m_2 \left( \begin{array}{c} -l_2 \,\mathrm{s}\left(\theta_1 - \phi_1\right)\dot{\theta}_1^2 + \\ l_2 \,\mathrm{c}\left(\theta_1 - \phi_1\right)\ddot{\theta}_1 + d_2\ddot{\phi}_1 \end{array} \right)$$

$$L_5 = -d_2 g m_2 \,\mathrm{c}\,\phi_2 + I_{2,yy}\ddot{\phi}_2 + d_2 m_2 \left( \begin{array}{c} -l_2 \,\mathrm{s}\left(\theta_2 - \phi_2\right)\dot{\theta}_2^2 + \\ l_2 \,\mathrm{c}\left(\theta_2 - \phi_2\right)\ddot{\theta}_2 + d_2\ddot{\phi}_2 \end{array} \right)$$

$$L_6 = -d_2 g m_2 \,\mathrm{c}\,\phi_3 + I_{2,yy}\ddot{\phi}_3 + d_2 m_2 \left( \begin{array}{c} -l_2 \,\mathrm{s}\left(\theta_3 - \phi_3\right)\dot{\theta}_3^2 + \\ l_2 \,\mathrm{c}\left(\theta_3 - \phi_3\right)\ddot{\theta}_3 + d_2\ddot{\phi}_3 \end{array} \right)$$

$$L_7 = m_P\left(-d_P \,\mathrm{s}\,\psi_y \dot{\psi}_y^2 + \ddot{O}_{7,x} + d_P \,\mathrm{c}\,\psi_y \ddot{\psi}_y\right)$$

$$L_8 = m_P\left(\ddot{O}_{7,y} + d_P\left( \begin{array}{c} \mathrm{s}\,\psi_x \,\mathrm{c}\,\psi_y \dot{\psi}_x^2 + 2\,\mathrm{c}\,\psi_x \,\mathrm{s}\,\psi_y \dot{\psi}_x \dot{\psi}_y + \\ \mathrm{s}\,\psi_x \,\mathrm{c}\,\psi_y \dot{\psi}_y^2 - \mathrm{c}\,\psi_x \,\mathrm{c}\,\psi_y \ddot{\psi}_x + \mathrm{s}\,\psi_x \,\mathrm{s}\,\psi_y \ddot{\psi}_y \end{array} \right)\right)$$

$$L_9 = m_P\left(g + \ddot{O}_{7,z} + d_P\left( \begin{array}{c} -\,\mathrm{c}\,\psi_x \,\mathrm{c}\,\psi_y \dot{\psi}_x^2 + 2\,\mathrm{s}\,\psi_x \,\mathrm{s}\,\psi_y \dot{\psi}_x \dot{\psi}_y - \\ \mathrm{c}\,\psi_x \,\mathrm{c}\,\psi_y \dot{\psi}_y^2 - \mathrm{s}\,\psi_x \,\mathrm{c}\,\psi_y \ddot{\psi}_x - \mathrm{c}\,\psi_x \,\mathrm{s}\,\psi_y \ddot{\psi}_y \end{array} \right)\right)$$

$$L_{10} = \left\{ \begin{array}{c} -d_P g m_P \,\mathrm{s}\,\psi_x \,\mathrm{c}\,\psi_y - {d_P}^2 m_P \,\mathrm{s}\,2\psi_y \dot{\psi}_x \dot{\psi}_y + 0.5\left(I_{P,y} - I_{P,z}\right)\mathrm{s}\,2\psi_x \dot{\psi}_y^2 - \\ d_P m_P \,\mathrm{c}\,\psi_x \,\mathrm{c}\,\psi_y \ddot{O}_{7,y} + \left(I_{P,x} + \left(I_{P,y} - I_{P,z}\right)\mathrm{c}\,2\psi_x\right)\mathrm{c}\,\psi_y \dot{\psi}_y \dot{\psi}_z - \\ I_{P,y}\,\mathrm{c}\,\psi_x \mathrm{s}\psi_x \mathrm{c}^2\psi_y \dot{\psi}_z^2 + 0.5 I_{P,z} \mathrm{s}2\psi_x \mathrm{c}^2\psi_y \dot{\psi}_z^2 - d_P m_P \,\mathrm{s}\,\psi_x \,\mathrm{c}\,\psi_y \ddot{O}_{7,z} + \\ I_{P,x}\ddot{\psi}_x + 0.5 {d_P}^2 m_P \ddot{\psi}_x + 0.5 {d_P}^2 m_P \,\mathrm{c}\,2\psi_y \ddot{\psi}_x + I_{P,x}\,\mathrm{s}\,\psi_y \ddot{\psi}_z \end{array} \right\}$$

$$L_{11}=\left\{\begin{gathered}-d_P g m_P \,\mathrm{c}\,\psi_x \,\mathrm{s}\,\psi_y+0.5 d_P{}^2 m_P \,\mathrm{s}\,2\psi_y \dot{\psi}_x^2 \\ +0.5\left(I_{P,z}\mathrm{c}^2\psi_x \,\mathrm{s}\,2\psi_y+\left(-I_{P,x}+I_{P,y}\mathrm{s}^2\psi_x\right) s2\psi_y\right)\dot{\psi}_z^2 \\ +\dot{\psi}_x\left(-I_{P,x}\,\mathrm{c}\,\psi_y\dot{\psi}_z+\left(-I_{P,y}+I_{P,z}\right)\left(\mathrm{s}\,2\psi_x\dot{\psi}_y+\mathrm{c}\,2\psi_x\,\mathrm{c}\,\psi_y\dot{\psi}_z\right)\right) \\ +d_P m_P\,\mathrm{c}\,\psi_y\ddot{O}_{7,x}+d_P m_P\,\mathrm{s}\,\psi_x\,\mathrm{s}\,\psi_y\ddot{O}_{7,y} \\ -d_P m_P\,\mathrm{c}\,\psi_x\,\mathrm{s}\,\psi_y\ddot{O}_{7,z}+0.5\left(I_{P,y}+I_{P,z}\right)\ddot{\psi}_y \\ +d_P{}^2 m_P\ddot{\psi}_y+0.5\,\mathrm{c}\,2\psi_x\left(I_{P,y}-I_{P,z}\right)\ddot{\psi}_y \\ -0.5\,\mathrm{s}\,2\psi_x\,\mathrm{c}\,\psi_y\left(I_{P,y}-I_{P,z}\right)\ddot{\psi}_z\end{gathered}\right\}$$

$$L_{12}=\left\{\begin{gathered}0.5\left(I_{P,y}-I_{P,z}\right)\mathrm{s}\,2\psi_x\,\mathrm{s}\,\psi_y\dot{\psi}_y^2+ \\ \left(-I_{P,z}\mathrm{c}^2\psi_x+\left(I_{P,x}-I_{P,y}\mathrm{s}^2\psi_x\right)\right)\mathrm{s}\,2\psi_y\dot{\psi}_y\dot{\psi}_z \\ +\left(I_{P,x}+\left(-I_{P,y}+I_{P,z}\right)\mathrm{c}\,2\psi_x\right)\mathrm{c}\,\psi_y\dot{\psi}_x\dot{\psi}_y \\ +\left(I_{P,y}-I_{P,z}\right)\mathrm{s}\,2\psi_x\mathrm{c}^2\psi_y\dot{\psi}_x\dot{\psi}_z \\ +\left(\begin{gathered}0.5\left(-I_{P,y}+I_{P,z}\right)\mathrm{s}\,2\psi_x\,\mathrm{c}\,\psi_y\ddot{\psi}_y \\ +\left(I_{P,z}\mathrm{c}^2\psi_x+I_{P,y}\mathrm{s}^2\psi_x\right)\mathrm{c}^2\psi_y\ddot{\psi}_z\end{gathered}\right) \\ +I_{P,x}\,\mathrm{s}\,\psi_y\left(\ddot{\psi}_x+\mathrm{s}\,\psi_y\ddot{\psi}_z\right)\end{gathered}\right\}$$

The generalized forces are defined as input torques ($\tau_{1a}$ $\tau_{2a}$ $\tau_{3a}$ ) for n = 1; 2; 3 and external force components acting on point $O_{7;z}$ ($F_{extx}$ , $F_{exty}$ and $F_{extz}$ ) for n = 7; 8; 9.
With the formulation of Ln, all terms at the right-hand side of the equation 6.3.1 in the Lagrange dynamics are formulated. For the left-hand side of the equation, constraint equations $\Gamma_k$ correspond to the scalar components of the Equations in 3.1. All these calculations of Dynamics are directly taken from research papers but the calculations of velocity kinematics. Dynamics are not applied in the simulation or any calculations due to the complexity and limited time.

**Features of mechanism**

**Speed-Force trade-off between mechanisms**

Now when we move the robotic leg we generally have to move all the actuators together to obtain a desired kind of motion (like translation in X direction), hence we can apply larger forces on the ground when compared to a serial manipulator with the same actuators and work envelope). Also no actuator has to carry the load of another actuator while moving freely, which also increases efficiency of the movement and contribute to the larger force mentioned in the point above .When we design any kind of mechanism there is always a speed to force trade-off when the actuators cannot be changed, i.e. if we want to increase the speed of the end effecter we have to compensate on max force the end effector can apply. It is very difficult to adjust this trade-off in serial based leg manipulators as we have to change the gearboxes of the motors or try to decrease/increase the weight of each element(keeping size of the robot constant) and even after that, the trade-off is very difficult to perfect. This is not the case for parallel manipulators, as only changing one link length linearly changes the agility to force relation. This point does not bother a engineer much if he/she only wants the robot to walk at slow speed, but when you want the robot to jump or run fast, or do a back flip like the robots at Boston dynamics do this becomes very important.

**Compliance**

Also compliance is very important when building a Quadruped robot. Compliance is a property where the robot's joints move in a spring kind of fashion when a large enough force is applied without actually powering the actuators movement. This may seem unwanted at first as it may decrease the positional accuracy of the robot but this property is very important to balance the robot. Compliant robots automatically align their centre of masses when an unbalanced force is applied, hence making balancing the robot which is a very complex task a little simple. Now compliance can be achieved in the proposed solution by limiting the current to each actuator hence forming a virtual spring. We can also connect a spring as a coupling between the motor and the servo arm which will give desired compliance but may again reduce positional accuracy.

**Errors in Joints**

In serial based manipulators errors in joints may become a huge problem leading to positional error of the foot. These errors can happen due to gear meshing issues, small bending of a component in the mechanism, play in bearing fits, loose threading of bolts and screws, etc. In a serial manipulator error in each joint extenuates and makes a huge difference in the end effecter position, but in case of parallel manipulators, the actuators are mounted in a circular pattern and hence the positional errors effectively cancel each other out leading to very high positional accuracy.

**Motor Sizing**

Motor sizing is very important as it helps us select the most important components of our system, the actuators. Also many times we already have actuators present and in that case we can produce the given calculation in reverse to find the robot parameters.

When sizing a motor we consider 2 most important parameters, shaft angular velocity and maximum output torque. Now if the robot wants to jump it has to lose contact when the joints are at one of their extreme limits in one direction, with the ground and for that to happen the robot should generate contact forces larger than its own weight and the height of jump will depend on both the speed of leg movement as well as force exerted. We assume wrt to our design that the robot has to jump 0.5m in the air in the Z direction against gravity. We now that v=u+a*t and s=u*t+(1/2*a*t^2). Now initial velocity is zero and acceleration due to gravity is -10m/s^2.

We know that mgh=F*s. Where h=height of jump i.e. height between the CG when the robot squats down and the max displacement in air. s is the distance the robot squats down. We add material properties to the design, making most of the design of aluminium, the weight of the robot is nearly 10kgs. We want the robot to jump by 50cm in the air while squatting down for 20cm. So the calculation follows as 10*10*(0.5+0.2)=F*0.2. F=350N. This is the force the robot needs to apply if it has to jump

50cm in the air. Now assuming this force will be equally applied by all actuators, each actuator has to apply 350/12 N ~30N force . Torque required for the motor to pull of this operation=Fon each motor*length of servo arm=30*0.15=4.5Nm.

Now mgh=1/2mv^2. So v^2=0.5gh.3.5. v=1.87m/s. The CG of the robot moves at 1.5m/s. We know the robot has to move in the Z direction by 30cm when it leaves contact from ground. We know that v=u+a*t. We want to find time that is required to move from squat to non-squat position. Acceleration produced by the motors is 350/10=35m/s^2. So 1.87=0+(-10+35)*T. T=0.0748sec. Now we can calculate using inverse kinematics the angles of the joint when displacement in Z is 0.5m and 0.7m respectively. When put in this in the formulas, we find that we need to move the motors by approximately 40deg for squatting by 0.2 m. Then we find motor-speed-rpm=angle-moved/360/T*60=133rpm. Hence we see that low speed motors too can produce such high accelerations required for jumping when using parallel manipulator, hence this is a big advantage for this mechanism.

Now while selecting materials, actuators, power supplies for the robot these factors are considered. This is more useful when working on a real robot rather than a simulation.

**Mounting components for max efficiency**

Running at high speeds and exerting foot forces quickly for explosive dynamic maneuvers requires a leg which must rotate and oscillate rapidly while switching direction throughout every duty cycle. This rapid and cyclic leg swinging requires constant rotational acceleration which is a heavy load on the actuators. Therefore instead of increasing actuator size which consequently increases robot size and total mass the leg design should be such that mass and inertia are minimized while maintaining structural integrity.

Hence we try to reduce the weight of the system by using more Aluminium and less steel. The parallel manipulators have a huge advantage in this regard as the leg does not hold any actuator hence it does not have to support the weight of the actuator leading to lowering the inertia and increasing maneuverability.

Also if most of the mass of the robot is concentrated near the hip i.e. in our case near the motor mountings, and it becomes easier for us to derive the inverse dynamics (as we don’t have to consider the motor’s moment of inertia as it is not moving with the leg as in serial based leg manipulators) and apply the exact amount of force that we want to.

Considering all these parameters the robot is designed in solidworks for maximum efficiency.

**Measuring Normal force on Leg**

Generally all Legged robots have force sensors present on the foot to detect whether the foot is touching the ground or not and if so what normal force does the ground apply on each leg. Hence we can consider the normal force on each leg to be a feedback. With these force values it becomes easier for us to balance the robot during dynamic movements

There is another way of measuring normal force, i.e. by measuring current in each motor. We know that the torque outputted by each actuator is directly propotional to the current being pulled by it at constant voltage. We can then use the jacobian to transform the joint torques to the end effector wrench, and then by using the angular position of the robot we can estimate the normal force on each foot and position of the robot. This force sensing may not be possible in this project as it is easier to check this on a actual model rather than a simulation as most simulation softwares do not support a robot with such a complex design.

**Approach to perform the Work**-

-First the model was designed in Solidworks so that its geometry and mountings became clear.

- Next the inverse kinematics of the model was derived and a MATLAB code was written to get quick results of the IK. We use solidworks to verify the IK.
- Then it was decided what the robot will be used for in this case jumping and running, and motors were sized respectively
- A simulation was attempted in VREP and Gazebo which failed as these softwares wouldn't support parallel structures imported from design softwares
- Simulation was started on MATLAB and path generation using IK is being attempted for single leg as MATLAB couldn't process the whole robot together because of the presence of multiple(12) wrist type joints

**Timeline with milestones-**

October

- Try deriving the IK by using geometry,
- Attempted simulation of robot on V-REP

Nov-Motion study and mathematical modelling of system

- FK
- IK
- Path planning formulation,
- Simulation

Dec-Final Presentation

- Report Making
- Final touches to simulation
- Presentation

Due to problems of design not working on simulation softwares like VREP and Gazebo(as most simulation softwares do not support new parallel manipulators as it is not possible to make a model tree structure of them) I may or may not be able to show the simulation of the model. All simulation softwares don't have a 3dof joint like a ball joint which when imported using a 3D modelling software gets assumed as a fixed joint. To solve this problem the link is connected to the platform using a spherical wrist kind of an arrangement. Also with simulation softwares that support the parallel mechanism structure, the software seems to collapse when a whole Quadruped robot is imported as it is difficult for the software to calculate relative positions of spherical wrist kind of joints and each leg has 3 of such joints and hence the whole robot has 12 of such joints in total which a advanced simulation software like MATLAB is not able to process. Hence it possible I will show the path planning simulation of a single robotic leg using MATLAB. A lot of time was used up in discovering all the shortcomings of the simulation software mentioned above, which were unknown at most forums.

**Running the MATLAB simulation in your own system**

1. Extract all the files in the zip folder.

2. Open matlab and change the path to the path where the contents are extracted.

3. Open the "function_runner.m" matlab code file and run it to create variables in workspace that the simulation will follow. Check if the workspace as variables named ts1 ts2 and ts3

4. Open the Simulink File name "Assem_1_ik.slx" and run the simulation.

5. Here you will see the robot follow the hard coded path in the "function_runner.m" file. If you want to change this path please make necessary changes in the "function_runner.m" file

**Validation Plan:**

I plan to validate all my formulas using a combination of both solidworks and Matlab. I have first written a script in MATLAB that gives joint variables for the end effector position and orientation (Inverse Kinematics.) I then made a design using solidworks, and inserted all the design parameters like link length and positions in MATLAB from the design. Then I enter some end effecter position and orientation in the code and find its corresponding joint variables. Then I go to Solidworks and set the joint variables as I found in MATLAB and confirm whether I get the same end effector position that I entered first in the MATLAB code.

**Conclusion:**

In this project, we firstly try to design a PM module that can function as a leg and follow gaits like the ones followed by generic leg mechanisms. We design the module such that it will be easier for the simulation software to handle and match the dimensions of the most famous Quadraped robot the Spot-mini. The modules are attached to a plate and we try to simulate basic movements on simulation softwares like V-REP and Gazebo. V-REP and Gazebo define the structure of the robot using a tree structure and it is impossible to simulate a PM in a tree style structure, as it is impossible to form closed chain structures in trees. V-REP and Gazebo don't support 3dof joints like ball joints which were a problem too. To go past the 3dof joint issue, I replaced it with a wrist type 3DOF joint with all its axes perpendicular to each other and intersecting at the same point. When we want to manufacture this system we replace this wrist joint with a ball joint. Then I tried to simulate the robot in MATLAB by importing it into Simulink. Even Simulink did not support the ball joint but it could support the design as Simulink defines the robot in a Flowchart kind of a system and it is possible to form closed chain structures in flowcharts.
Then a code is written which calculates the Inverse Kinematics i.e. the 3 joint variables when inputted with the (X,Y,Z) co-ordinates of the feet called the IK_function. If the point is not reachable the code reports a error. Then we try to simulate a trotting gait with the robot and write a script in MATLAB that is called a path planner function. This function calculates the 3 joint variables, by calling the IK_function for a path we hardcode so that the feet follows this trajectory. These joint variables are stored in 3 arrays respectively. Then a from workspace block is used in Simulink to input the 3 joint variable arrays we calculated and use them to simulate and view the trajectory I hardcoded in the path planner function.
Formulations in order to obtain a mathematical model of a 3DOF PM is presented. For inverse kinematic analysis, firstly the constraint equations for moving platform are formulated. Then using the loop equations, from which passive joint variables are eliminated, the active joint parameters are obtained in terms of independent task space parameters. Forward Kinematics are not done in this project as the FK are very complicated in terms of the task space parameters where we obtain a 16$^{th}$ degree polynomial equation we have to solve. We can solve this using numerical analysis but this is beyond the scope of this project. Then, all of the passive joint variables are evaluated and task space parameters are obtained in terms of active and passive joint variables. For the velocity and acceleration level kinematics, derivatives of the loop equations are used and all of the joint and task space velocities and accelerations are obtained.
The singularities of the PM are determined using the Jacobian matrices of the loop equations. 1$^{st}$ and 2$^{nd}$ type of singularities are observed for the PM. The 3$^{rd}$ type singularities are not detected for this PM investigated in this project.
Inverse dynamic analysis for the PM is performed using Lagrange's approach and we obtain a equation for input torques
The calculations for velocity kinematics and inverse dynamics are not used in this project due to complexity and lack of time. For further research we can find the torques required for each joint variable so that it can follow a trajectory.

**Future Scope-**

The Robot that is designed above is mostly for simulation purposes only to simplify geometry and making it easier for the simulation software to handle, but when we go for actual manufacturing of the robot, many design changes such as better design for a leg module for easy connection and removal, Use of materials like Carbon fibre to reduce weight of the structure while maintaining the same or increasing the rigidity of the robot. Also the most difficult and inaccurate part of today's Quadruped

Robots is the dynamics of the system which makes accurate trajectory planning very difficult, A mechanism can be implemented to solve this problem which has a input as a force and its direction and the mechanism exerts the exact amount of force in that particular direction without using complex mathematical equations. I have attached a rough sketch of this idea along with the report. Also Balancing of the robot on 2 legs and while performing dynamic maneuvers is a challenging task which can be undertaken as a future objective. This Robot is a very novel project that can be undertaken as a PHD project in the future. This robot looks like a very good solution to all the problems discussed above for the current Quadruped Robots and a successful implementation of this project might take Quadruped robots into another dimension deriving new problems and a motivation to create a new solution.

**References**-


- https://www.scribd.com/document/24761320/3PRR-Planar-Parallel-Manipulator-Analysis
- https://pdfs.semanticscholar.org/af85/dd6fd66eb9b0121ecd2b687a458cfc5985b4.pdf
- https://www.ri.cmu.edu/pub_files/2016/8/kaloucheThesis.pdf
- https://pdfs.semanticscholar.org/f36c/6407e9a8a2f0a8dbe8ea29a2fbac71e96a58.pdf