MODELING OF KINEMATIC CHAINS

Modeling of a Hexabot Leg

November 10, 2024

Hyejoo Kwon
Godsfavour Ugwu-Gabriel

---

1. **Accoring to the mission dedicated to this limb, what are the desired controlled dof of the foot?**

Ans:

(a) Link and Joint Details

For calculating the DOF in three-dimensional space, we use the modified Grubler's Formula:

$$\text{DOF} = m(N - 1 - J) + \sum_{i=1}^{J} f_i$$

where:

- N = Total number of parts (links), including the base or ground.
- J = Number of connections (joints) between parts.
- m = Degrees of freedom for a single rigid part (3 for flat or 2D systems, and 6 for 3D systems).
- f_i = Degrees of freedom allowed by each joint.

In this hexapod foot setup:

- $N = 4$: We have four links, including the fixed base link.
- $J = 3$: We have 3 revolute joints connecting these links.
- $m = 6$: We have a 3D system
- $f_i = 1$: Each revolute joint provides one rotational degree of freedom.

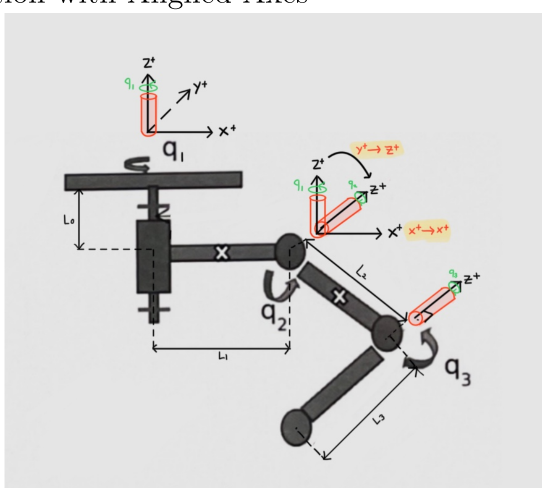
By substituting these values into the formula, we perform the following calculation:

$$\begin{aligned}
\text{DOF} &= 6 \times (N - 1 - J) + \sum_{i=1}^{J} f_i \\
&= 6 \times (4 - 1 - 3) + (1 + 1 + 1) \\
&= 3 \times 0 + 3 \\
&= 0 + 3 = 3
\end{aligned}$$

2. **Q2 : The robot body may be considered as still (same orientation as the Ground frame), so place all the intermediate frames until the foot. Fill in the DH/Khalil formalism table according to the considered frames. Write all the homogeneous transform matrices from a frame to the neighbouring frame and all those from each frame to the body frame (useful for the next question)**

Ans:

(a) Axis Transformation with Aligned Axes



Figure(1)

As seen in the figure 1, sketching the transformation of aligned axes confirms the axis rotations required to define each transformation matrix.

The first transformation is a rotation by q_1 around the z-axis. Since the z-axis remains aligned with the original frame, we don't need to realign the axis.

For the second step, the y-axis is aligned with the z-axis by rotating the x axis by 90-degree. This alignment is only needed at this step to set up the frame for the next rotation by q_2 around the z-axis.

Finally, a rotation around z by q_3 completes the sequence without further changes, preserving alignment.

(b) Translation

The transformation sequentially translates along the x-axis by distances l_1, l_2, and l_3. These translations are determined by the lengths of the links, with l_1 set to 50mm, l_2 to 100mm, and l_3 to 100mm. This approach enables us to make the DH parameter table and homogenous transtormation matrices, ensuring precise limb movement.

(c) DH parameters

The following table summarizes the DH parameters based on the specified translations and rotations for each link:

Link i	a_i (Link Length)	α_i (Link Twist)	d_i (Link Offset)	θ_i (Joint Angle)
1	0	0	0	q_1
2	l_1	90°	0	q_2
3	l_2	0	0	q_3
4	l_3	0	0	0

Using these parameters, the homogeneous transformation matrix is calculated as:

$$T_1^2 = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & 0 \\ \sin(q_1) & \cos(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_1^2 = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & l1 \\ 0 & 0 & -1 & 0 \\ \sin(q_2) & \cos(q_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^3 = \begin{bmatrix} \cos(q_3) & -\sin(q_3) & 0 & l_2 \\ \sin(q_3) & \cos(q_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^4 = \begin{bmatrix} 1 & 0 & 0 & l3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The resulting transformation matrix $T_0^1 \cdot T_1^2 \cdot T_2^3 \cdot T_3^4$ is:

$$T_0^4 = \begin{bmatrix} \cos(q_2+q_3)\cos(q_1) & -\sin(q_2+q_3)\cos(q_1) & \sin(q_1) & \cos(q_1)(l_1+l_3\cos(q_2+q_3)+l_2\cos(q_2)) \\ \cos(q_2+q_3)\sin(q_1) & -\sin(q_2+q_3)\sin(q_1) & -\cos(q_1) & \sin(q_1)(l_1+l_3\cos(q_2+q_3)+l_2\cos(q_2)) \\ \sin(q_2+q_3) & \cos(q_2+q_3) & 0 & l_3\sin(q_2+q_3)+l_2\sin(q_2) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. **Calculate theJacobian matrix, which gives the foot motion relative to the ground frame (or the body frame if it's considered still).**

Ans: The position components from the transformation matrix T_{04} are extracted as follows:

$$P_x = T_{04}(1,4), \quad P_y = T_{04}(2,4), \quad P_z = T_{04}(3,4)$$

The Jacobian matrix is computed by taking the partial derivatives of P_x, P_y, and P_z with respect to the joint angles q_1, q_2, and q_3:

$$J = \begin{bmatrix} \frac{\partial P_x}{\partial q_1} & \frac{\partial P_x}{\partial q_2} & \frac{\partial P_x}{\partial q_3} \\ \frac{\partial P_y}{\partial q_1} & \frac{\partial P_y}{\partial q_2} & \frac{\partial P_y}{\partial q_3} \\ \frac{\partial P_z}{\partial q_1} & \frac{\partial P_z}{\partial q_2} & \frac{\partial P_z}{\partial q_3} \end{bmatrix}$$

The calculated Jacobian matrix J is:

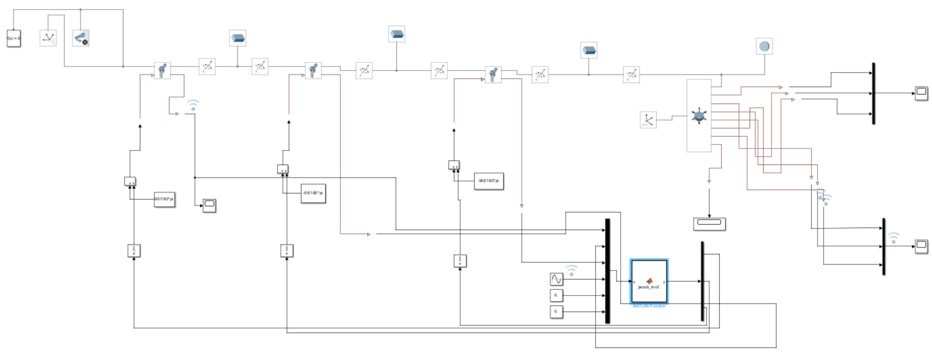
$$J = \begin{bmatrix} -\sin(q_1)\,(l_1+l_3\cos(q_2+q_3)+l_2\cos(q_2)) & -\cos(q_1)\,(l_3\sin(q_2+q_3)+l_2\sin(q_2)) & -l_3\sin(q_2+q_3)\cos(q_1) \\ \cos(q_1)\,(l_1+l_3\cos(q_2+q_3)+l_2\cos(q_2)) & -\sin(q_1)\,(l_3\sin(q_2+q_3)+l_2\sin(q_2)) & -l_3\sin(q_2+q_3)\sin(q_1) \\ 0 & l_3\cos(q_2+q_3)+l_2\cos(q_2) & l_3\cos(q_2+q_3) \end{bmatrix}$$

4. **Q4 : Provided the main robot axis (longitudinal axis to move forwards) is lining with Xground, can we control the leg to generate a strictly forward step (without any other motion)? What would it imply about the robot locomotion? Study the question by modelling the Hexabot leg using the Simcape toolbox of the Matlab solution.**

Ans:

(a) Blueprint of the Hexapod Leg Model

Using the Jacobian matrix derived, we designed a hexapod leg model to achieve precise control over joint movements and simulate realistic leg dynamics. The Jacobian enables calculation of joint velocities to achieve desired end-effector velocities, forming the foundation of the model's control system.



Figure(2): blueprint of the Hexapod Leg

(b) Jacobian Matrix Application in MATLAB Function

The calculated Jacobian matrix helps us find the velocity of each joint, showing the rate of position change needed to create the aimed movement at the end of the leg. BBy integrating these velocities, we calculate each joint's position, which is then used as real-time input in the simulation to control the hexapod leg.

```
function y = jacob_inv2(u)
l1 = 50;
l2 = 100;
l3 = 100;
q1 = u(1);
q2 = u(2);
q3 = u(3);
dx = u(4);
dy = u(5);
dz = u(6);

Jacob3 = [-sin(q1)*(l1 + l3*cos(q2 + q3) + l2*cos(q2)), -cos(q1)*(l3*sin(q2 + q3) + l2*sin(q2)), -l3*sin(q2 + q3)*cos(q1);
           cos(q1)*(l1 + l3*cos(q2 + q3) + l2*cos(q2)), -sin(q1)*(l3*sin(q2 + q3) + l2*sin(q2)), -l3*sin(q2 + q3)*sin(q1);
                                                    0,           l3*cos(q2 + q3) + l2*cos(q2),         l3*cos(q2 + q3)];
y = (inv(Jacob3))*[dx;dy;dz];
```

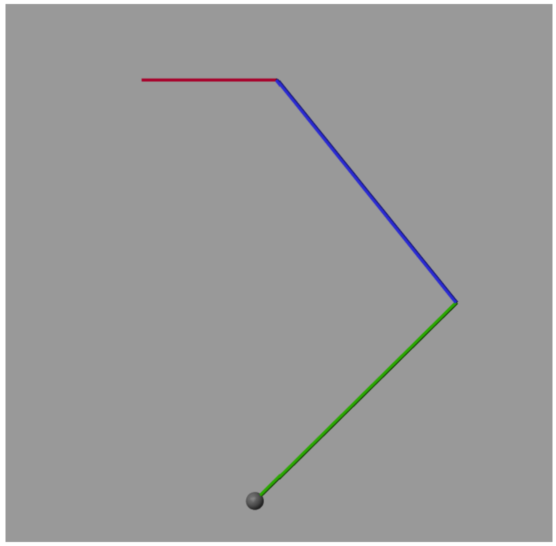
Figure(3) : Jacobian Matrix Function

(c) Input Signal Configuration

In order to understand the hexapod leg's movement for purely forward motion, we adjusted the amplitude of the input signals. Using three different amplitude values 10, 50, and 100, we observed changes in the position and velocity graphs for each case, helping us identify the conditions required for pure forward movement.

(d) Simulation Results

The simulation results show how the hexapod leg moves according to the input commands. We confirmed that the calculated Jacobian function accurately determines the joint velocities needed to follow the intended movement patterns.

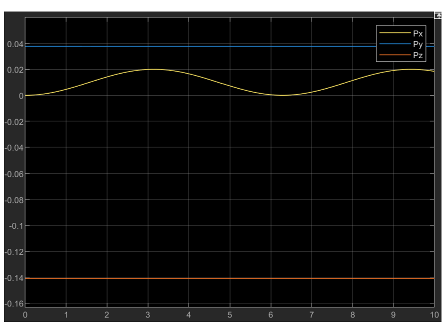


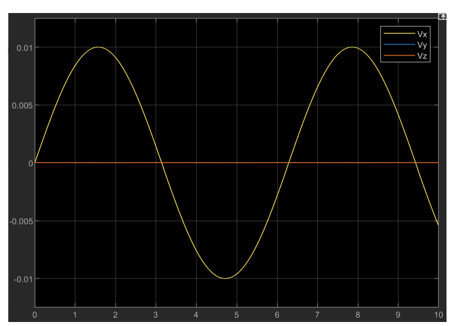
Figure(4): Result of Hexapod Leg

(e) Graphs of Final Velocity and Position Based on Different Amplitude

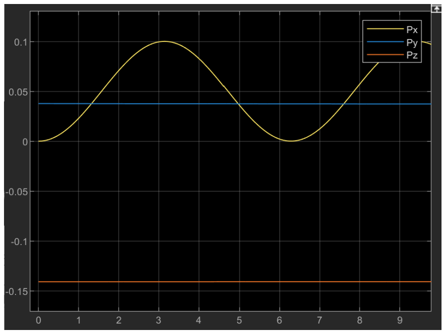
The following velocity (v) and position (p) graphs illustrate the leg's response over time. The results show the model's stability and accuracy in following the trajectory specified by the input signals, analyzed as follows:

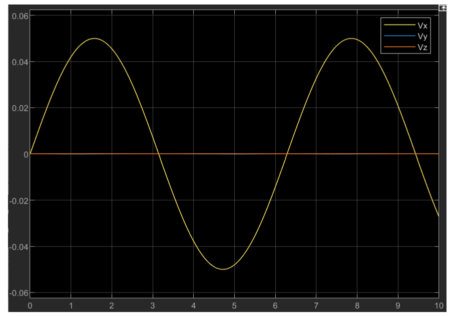
- **Input Parameter: 10**
  At an input amplitude of 10, the robot moves a short distance along the x-axis. The leg movements are well-controlled and stable, with very little vibration, indicating a slow and very steady path.



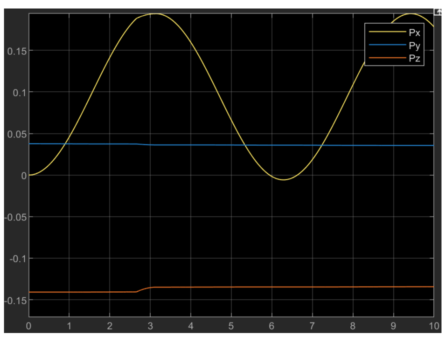


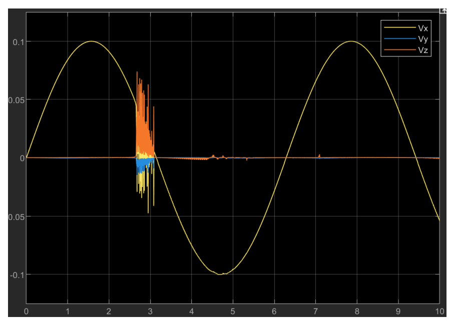
- **Input Parameter: 50**
  With an input amplitude of 50, the robot moves a much greater distance along the x-axis. The leg stays smoothly controlled while covering more ground, showing that the model can handle higher speeds while staying stable.





- **Input Parameter: 100**
  When the input value reaches 100, the robot's leg fully extends, causing singularities and oscillations. This means that high input amplitude can make the system unstable and because of this mechanical limitations, the leg is unlikely to operate with smooth control.





(f) Conclusion 1

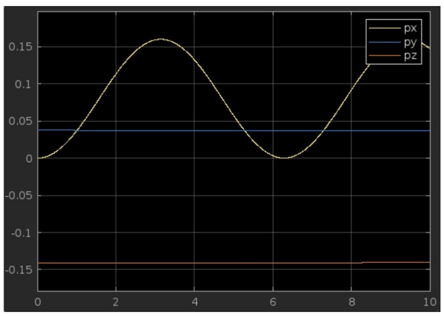
We found that, to achieve purely forward movement, the leg's motion needs to be constrained along the X-axis of the ground frame. This requires precise control of joint velocities and angles to prevent any unintended movement along the Y or Z axes. Using the Jacobian matrix, we calculated the specific joint velocities needed to limit movement to the X-axis alone. Through Simscape simulations, we observed that this forward-only motion works well at low to moderate input values (e.g., 10 or 50). However, when the input values were increased (e.g., to 100), oscillations began to occur, indicating that maintaining purely forward movement becomes difficult due to mechanical instability or resonance effects. For accurate robot locomotion, especially at higher speeds, it's necessary to carefully adjust input signals to prevent unintended deviations, as these could impact the robot's stability and accuracy.

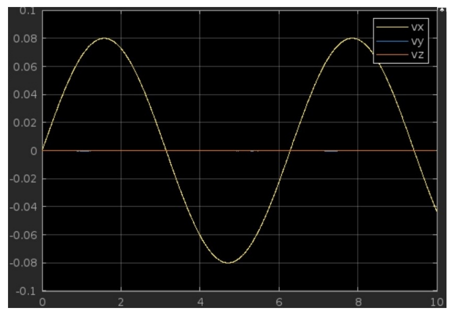
(g) Graphs of Final Velocity and Position Based on Different Frequency

Again, we investigated the response of a hexapod leg model by applying a sinusoidal input signal with an amplitude of 80. To assess how different input frequencies impact the leg's movement, we tested two distinct frequencies: $\omega = 1$

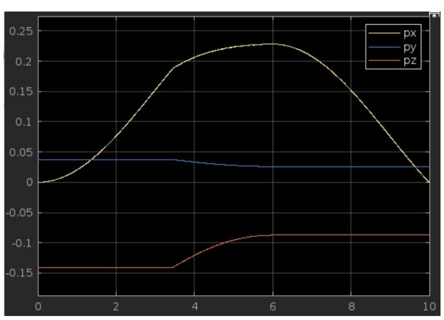
rad/s and $\omega = \frac{\pi}{6}$ rad/s. By analyzing the resulting motion, we aimed to understand the effect of input frequency on achieving controlled, unidirectional movement and identifying any instabilities introduced by singularities.

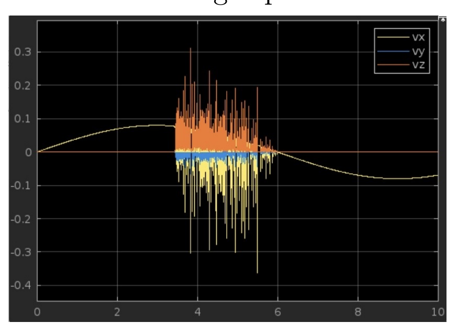
- **Analysis of Motion at 1 rad/s**
  At a frequency of $\omega = 1$ rad/s, the system achieved ideal forward motion, with movement limited entirely to the X-axis. The lack of position or velocity changes along the Y and Z axes confirmed accurate directional control, allowing efficient forward movement without any sideways or vertical drift.





- **Analysis of Motion at $\frac{\pi}{6}$ rad/s**
  At a lower frequency of $\omega = \frac{\pi}{6}$ rad/s, the system showed clear signs of singularity, with noticeable spikes in both position and velocity along the Y and Z axes. These singularities, marked by a nearly zero Jacobian determinant, caused a loss of control over all degrees of freedom. As a result, the leg generated undesirable lateral and vertical motion components, straying significantly from its intended straight path.





(h) Conclusion 2

The results shows that the input frequency plays a crucial role in controlling the hexapod leg's movement. At $\omega \geq 1$ rad/s, the leg achieves precise forward-only steps, maintaining stability and efficiency. However, at less than 1 rad, $\omega = \frac{\pi}{6}$ rad/s, the system encounters singular configurations which causes the leg to deviate from its forward path due to a loss of control over lateral and vertical displacements.

Thus, to generate a strictly forward step (without any other motion, it is important to select input frequencies that prevent singularities, at or above 1 rad/s (for our case study) as frequencies risk inducing singular behavior.