TRAJECTORY PLANNING DELTA PARALLEL ROBOT USING JACOBIAN METHOD

The motion description in cartesian space considers the path generation method. Meaning you should have the path coordinates as a function of time and then trajectory planning in cartesian space will be realized mainly through the Jacobian matrix

$$v = J\,\dot{q}$$

If $\dot{q}$ is velocity of joints and v is the velocity of the EE in cartesian space

the following are the steps to the algorithm of trajectory planning using Jacobian method:

1. Generate the trajectory coordinated as a function of time
2. Calculate the velocity profile (by differentiating the trajectory coordinates)
3. Calculate the inverse kinematic of the multi-point trajectory and calculate the Jacobian matrix for each point
4. Use $v = J\,\dot{\theta}$ for getting velocity profile of the actuator joints

Step 1: Generate the trajectory coordinates as a function of time

in this step a path is formulated as a function of time. As an example, the linear path system in 3D space is used.

So, the path will be:

$$\begin{aligned}(position\ formula)_x &= ??\\ (position\ formula)_y &= constant_1\\ (position\ formula)_z &= constant_2\end{aligned}$$

(And then it is discretized for n points in space-time)

Step 2: Velocity profile calculation

In this step we use the $\frac{d(position\ formula)}{dt} = (position\dot{\ }formula)$ to gain:

$$\begin{aligned}(velocity\ profile\ formula)_x &= ??\\ (velocity\ profile\ formula)_y &= \ 0.0\\ (velocity\ profile\ formula)_z &= \ 0.0\end{aligned}$$

Step 3: inverse kinematics of the multi-point trajectory

The most relevant loop should be picked up for the intended Jacobian analysis. Let $\vec{\theta}$ be the vector made up of actuated joint variables and $\vec{p}$ be the position vector of the moving platform (or end effector) then:

$$\vec{\theta} \equiv \theta_{1i} = \begin{bmatrix} \theta_{11} \\ \theta_{12} \\ \theta_{13} \end{bmatrix}, \vec{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

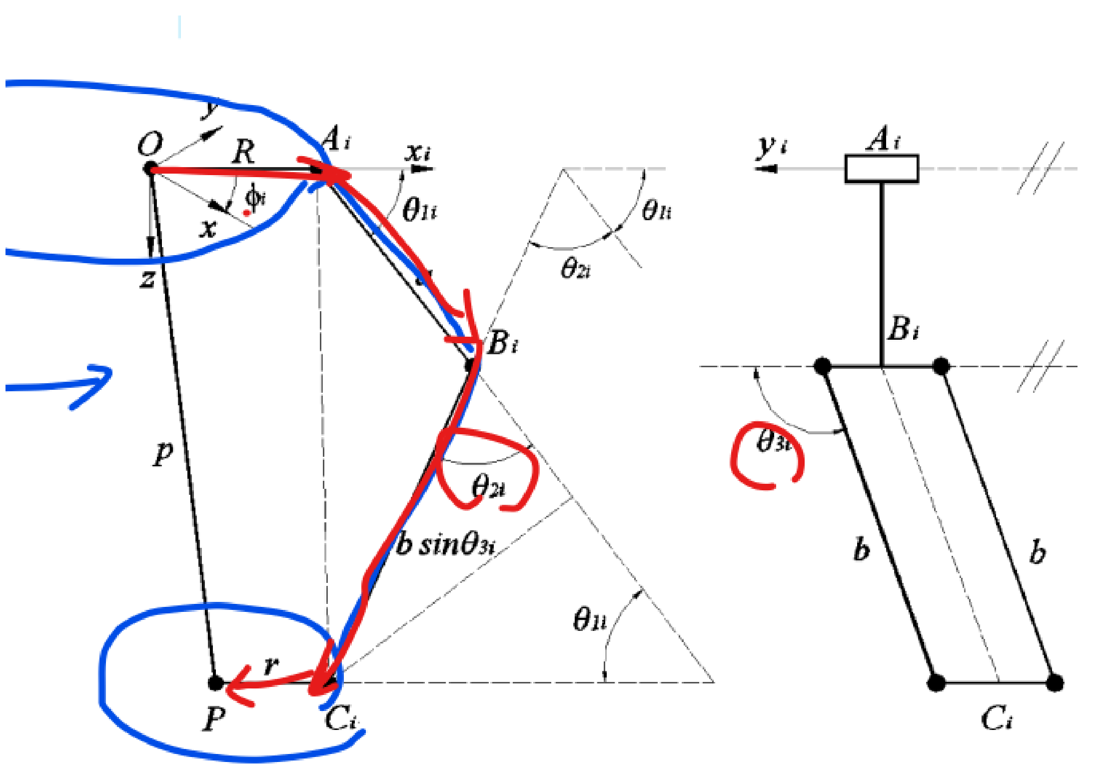

Fig 1-a: kinematics of the delta robot

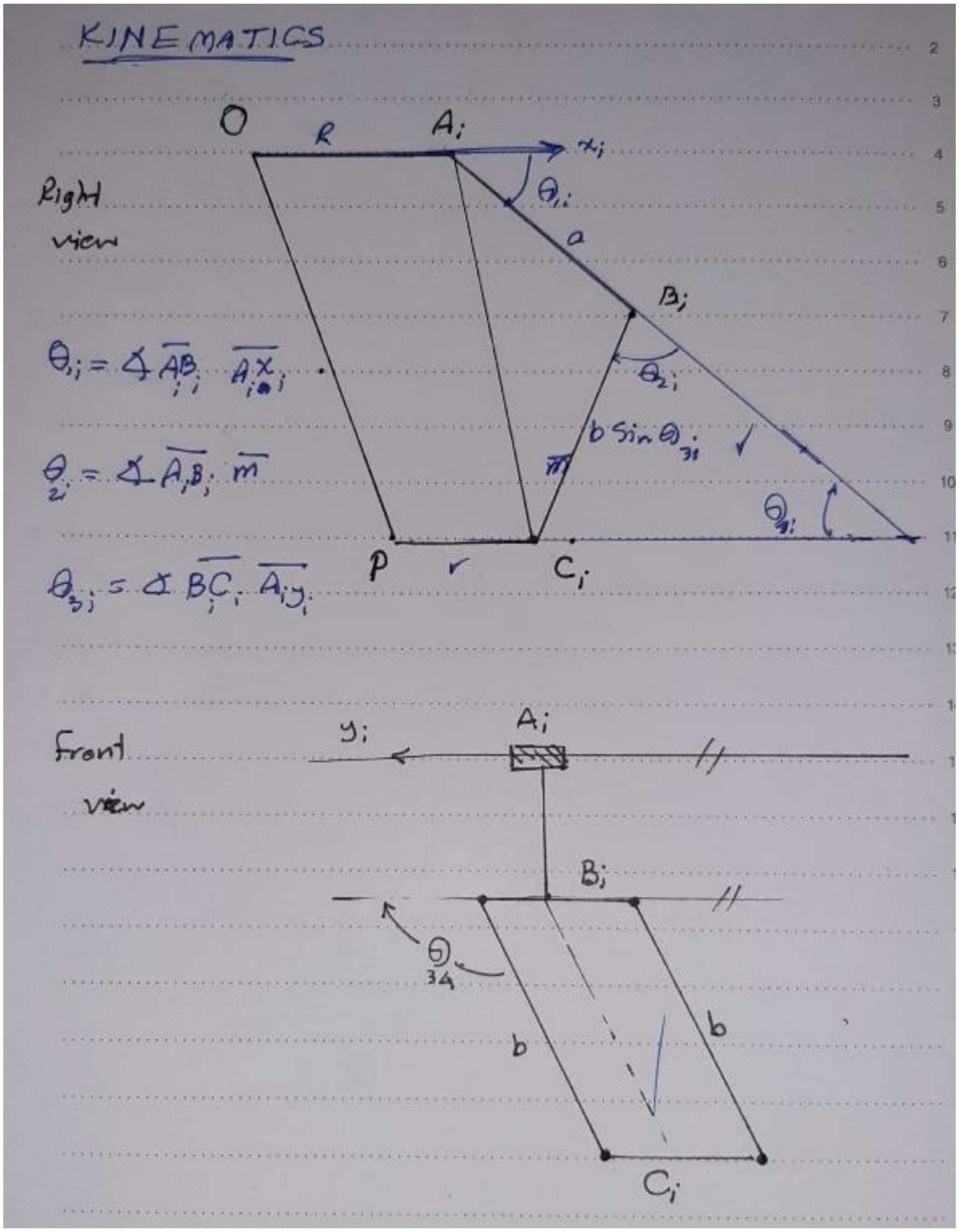

Fig 1-b: kinematics of the delta robot

Jacobian matrix is derived by differentiating the appropriate loop closure equation and rearranging in the following form:

$$J_\theta \dot{\vec{\theta}} = J_P \dot{\vec{p}} = J_P \vec{v}$$

That J_P and J_θ are as followed:

$$J_P = \begin{bmatrix} J_{1x} & J_{1y} & J_{1z} \\ J_{2x} & J_{2y} & J_{2z} \\ J_{3x} & J_{3y} & J_{3z} \end{bmatrix}$$

$$J_\theta = \begin{bmatrix} \sin(\theta_{21})\sin(\theta_{31}) & 0 & 0 \\ 0 & \sin(\theta_{22})\sin(\theta_{32}) & 0 \\ 0 & 0 & \sin(\theta_{23})\sin(\theta_{33}) \end{bmatrix}$$

So how is the above equation, acquired?

Step 3.1: vector summation and θ_{ij} calculation

From Fig 1-a and Fig 1-b you can easily see that:

$$\overrightarrow{OP} + \overrightarrow{PC_\iota} = \overrightarrow{OA_\iota} + \overrightarrow{A_\iota B_\iota} + \overrightarrow{(B_\iota C_\iota)}$$

The equation can be re-written as:

$$\begin{bmatrix} p_x\cos(\Phi_i) - p_y\sin(\Phi_i) \\ p_x\sin(\Phi_i) + p_y\cos(\Phi_i) \\ p_z \end{bmatrix} = \begin{bmatrix} R - r \\ 0 \\ 0 \end{bmatrix} + a\begin{bmatrix} \cos(\theta_{1i}) \\ 0 \\ \sin(\theta_{2i}) \end{bmatrix} + b\begin{bmatrix} \sin(\theta_{3i})\cos(\theta_{2i} + \theta_{1i}) \\ \cos(\theta_{3i}) \\ \sin(\theta_{3i})\sin(\theta_{2i} + \theta_{1i}) \end{bmatrix}$$

If the above equations are solved for $\theta_{1i}, \theta_{2i}, \theta_{3i}$ then:

$$\theta_{3_i} = \arccos\left(\frac{px.sin(\Phi_i) + p_y.cos(\Phi_i)}{b}\right)$$

$$\theta_{1i} = 2 \times \arctan(t)$$

$$\theta_{2i} = \arcsin\left(\frac{p_z - a.sin(\theta_{1i})}{b.sin(\theta_{3i})}\right) - \theta_{1i}$$

Where $t \equiv \frac{(B + (B^2 + A^2 - M^2)^{0.5})}{M + A}$

That

$$A = p_x.\cos(\Phi_i) - p_y.\sin(\Phi_i) - R + r$$

$$B = p_Z$$

$$M = \frac{(A^2 + B^2 + a^2 - (b.\sin(\theta_{3i}))^2)}{2a}$$

Now $\theta_{ij}\ for\ i, j = 1, 2, 3$ are known (as a function of robot geometry and the position of EE)

Step 3.2: Jacobian matrix calculation

Reminding that $J_P\ and\ J_\theta$ are matrices as shown below:

$$J_P = \begin{bmatrix} J_{1x} & J_{1y} & J_{1z} \\ J_{2x} & J_{2y} & J_{2z} \\ J_{3x} & J_{3y} & J_{3z} \end{bmatrix}$$

$$J_\theta = \begin{bmatrix} \sin(\theta_{21})\sin(\theta_{31}) & 0 & 0 \\ 0 & \sin(\theta_{22})\sin(\theta_{32}) & 0 \\ 0 & 0 & \sin(\theta_{23})\sin(\theta_{33}) \end{bmatrix}$$

Now it is possible to calculate J_{ix} with respect to the angles calculated in step 3.1 ($\Phi_i = [0, 120, 240]° \; by \; default$) :

$$J_{ix} = \sin(\theta_{3i})\cos(\theta_{2i} + \theta_{1i})\cos(\Phi_i) + \cos(\theta_{3i})\sin(\Phi_i)$$
$$J_{iy} = -\sin(\theta_{3i})\cos(\theta_{2i} + \theta_{1i})\sin(\Phi_i) + \cos(\theta_{3i})\cos(\Phi_i)$$
$$J_{iz} = \sin(\theta_{3i})\sin(\theta_{2i} + \theta_{1i})$$

Step 4: using Jacobian matrix to relate velocity of the EE to the velocity of the actuator joints

A quick recap of what the algorithm is doing: