DEPARTAMENTO DE
INGENIERÍA
MECÁNICA
UNIVERSIDAD DE SANTIAGO DE CHILE

Validation Kinematic and Dynamic Modeling of a Delta 3 GDL robot through ROS and ADAMS.

Author: Iván Alejandro Fernández Gracia[1]
Teacher Guide: Michael Gabriel Miranda Sandoval[1]

[1]University of Santiago de Chile. School of Engineering.
Department of Mechanical Engineering.

June 17, 2021

Content

1 Motivation

2 Hypothesis and Objectives

3 Theory

4 Development

5 Delta robot specifications

6 Results

7 Conclusions

8 Future lines of research and references

Content

1 Motivation

2 Hypothesis and Objectives

3 Theory

4 Development

5 Delta robot specifications

6 Results

7 Conclusions

8 Future lines of research and references

Motivation

Educational point of view

- Our department is lacking in terms of fostering and creating an environment to **develop** topics in complex robotics.

Theoretical Point of View

- Robot control schemes based **only on position kinematics** (the most abundant) are insufficient for a good interpretation of the results (**accuracy**).

Industrial point of view

- Point of view Chile can be related to robotics, mainly in areas where it has some relevance worldwide, such as fruit exports, where **pick and place** operations are very important.

Content

1 Motivation

2 Hypothesis and Objectives

3 Theory

4 Development

5 Delta robot specifications

6 Results

7 Conclusions

8 Future lines of research and references

Hypothesis

Hipótesis

It is possible to **perform** the **kinematic and dynamic modeling** of a delta robot through free software and **validate** it by means of educational software.

Objectives

General Objective

Design algorithms that control the motion of a delta robot and **validate** it through simulation software.

Specific Objectives

1. **Create** algorithms that compute linear trajectories in Cartesian space with trapezoidal velocity profile.
2. **Create** algorithms that solve the kinematics and dynamics of the robot.
3. **Determine** the robot workspace from imposed constraints.
4. **Simulate** the movement of the mechanical parts of the robot through a visualization tool.
5. **Calculate** the dynamics by means of a mechanical analysis software.
6. **Compare** the results of the dynamics calculated by the algorithms and by the mechanical analysis software.

Content

1 Motivation

2 Hypothesis and Objectives

3 Theory

4 Development

5 Delta robot specifications

6 Results

7 Conclusions

8 Future lines of research and references

Theory: Robot Operating System (ROS)

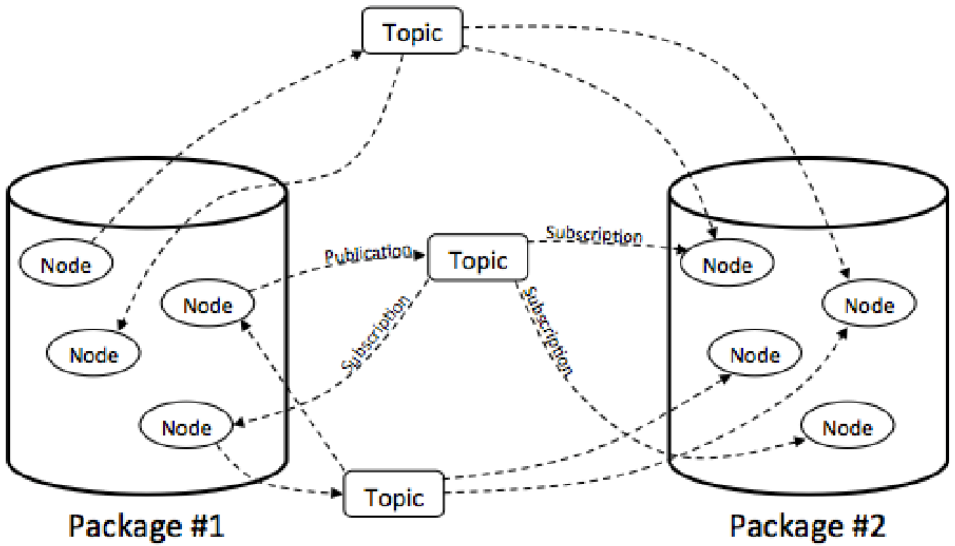
History

The first steps were in the mid-2000s at **Stanford University's robotics lab**. In 2007, **Willow Garage Inc**, a robotics incubator, provided significant resources to extend these concepts much further and create well-tested implementations. From 2013 to the present, ROS is permanently maintained by Google's **Open Source Robotics Foundation (OSRF)** and since 2017 changed its name to **Open Robotics**.

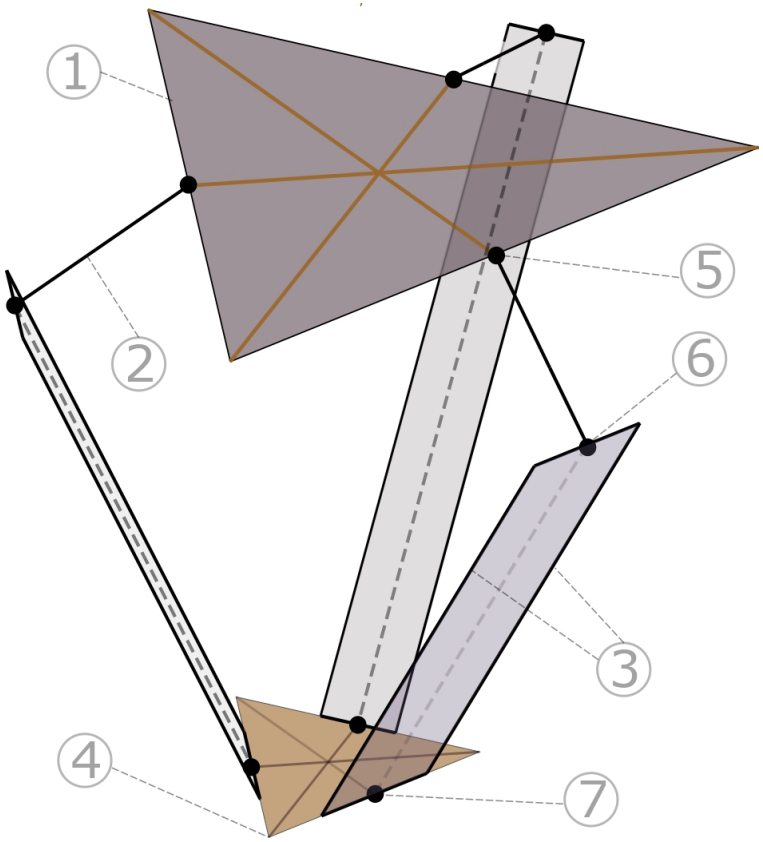
STANFORD UNIVERSITY

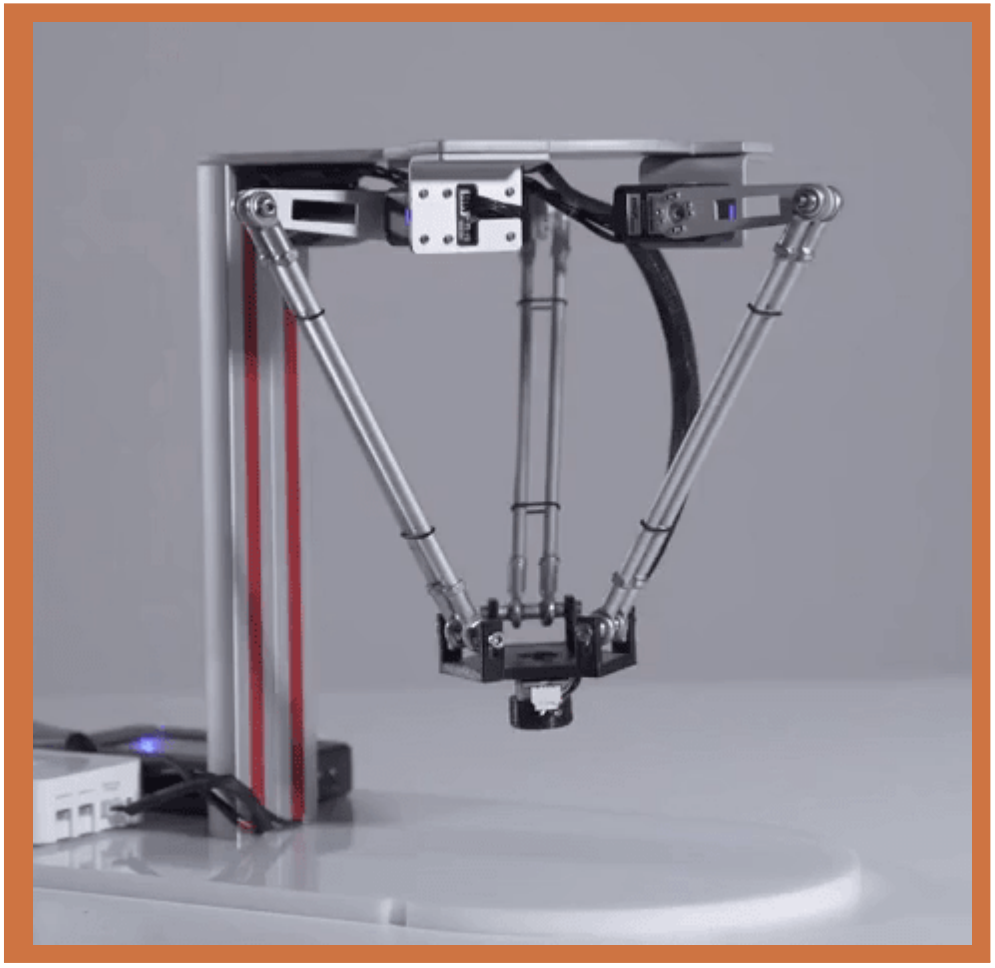
ROS Goals: Peer to peer

Robot Operating System (ROS) is **robotic middleware**, i.e., a collection of frameworks for robot software development. ROS consist of numerous **small computer programs** that connect to each other and **exchange messages** continuously.



Theory: Description delta robot

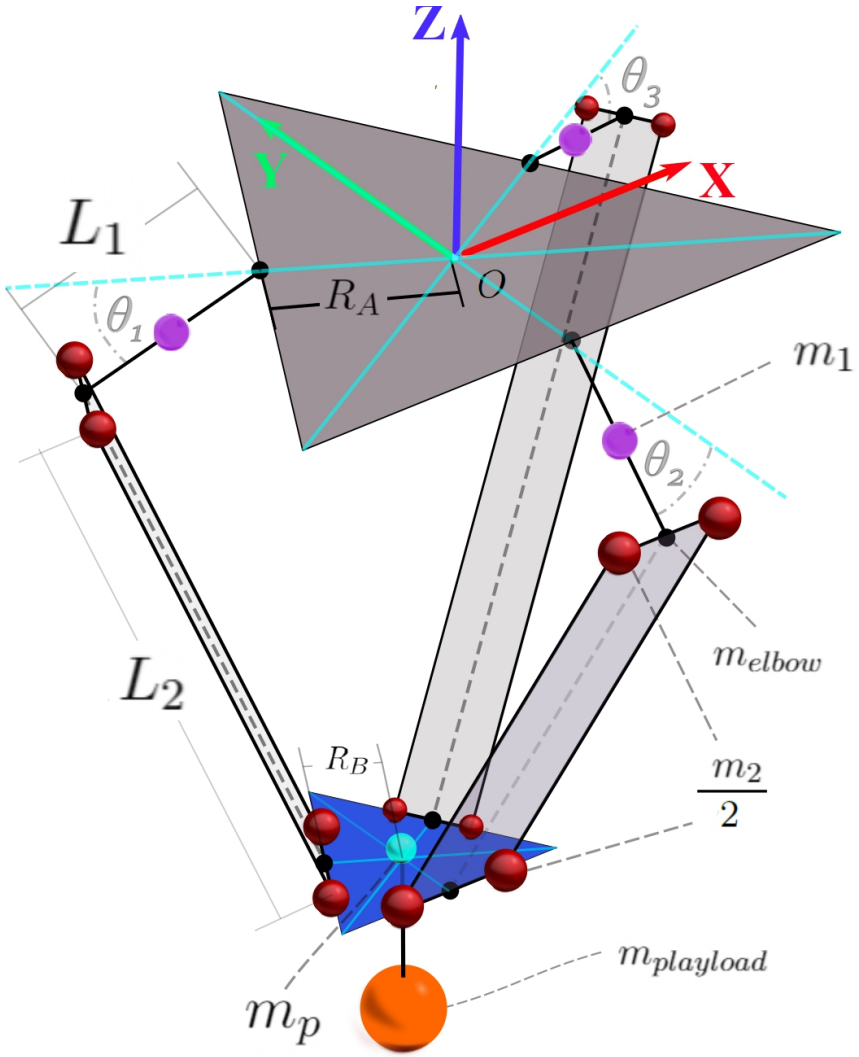




N°	1	2	4	4	5	6	7
Mechanical Parts	Fixed Base	Arm	Forearm	Mobile Base	Actuator o Motor	Spherical Joint	Spherical Joint

Theory: Steps to determine the dynamics

- **Nomenclature and local reference system**
- **Inverse kinematics**
  Direct kinematics
- **Jacobian**
- **Acceleration kinematics**
- **Inverse dynamics**



Global reference frame XYZ and order of arm angles $\theta_{i\in\{1,2,3\}}$.

Theory: Method A (Jacobian)

Close Loop

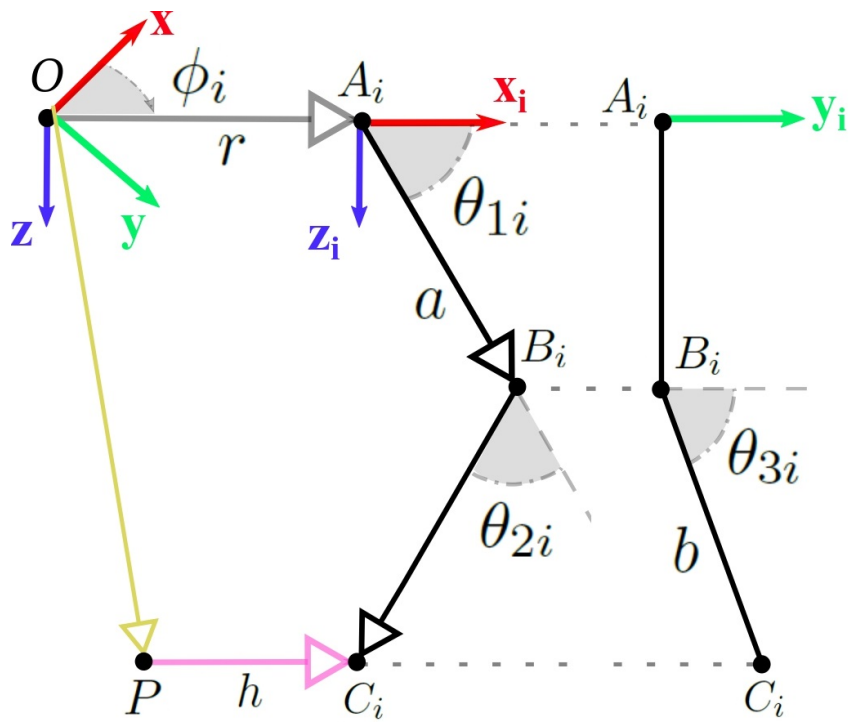
$$\overrightarrow{A_iB_i} + \overrightarrow{B_iC_i} = \overrightarrow{OP} + \overrightarrow{PC_i} - \overrightarrow{OA_i} \; ; \quad i \in \{1,2,3\}$$

By deriving the closed-loop equation and separating out terms v_p and $\dot{\theta}$

$$J_x\, v_p = J_\theta\, \dot{\theta}$$

$$\begin{bmatrix} J_{1x} & J_{1y} & J_{1z} \\ J_{2x} & J_{2y} & J_{2z} \\ J_{3x} & J_{3y} & J_{3z} \end{bmatrix} \begin{bmatrix} v_{px} \\ v_{py} \\ v_{pz} \end{bmatrix} = \begin{bmatrix} J_{1\theta} & 0 & 0 \\ 0 & J_{2\theta} & 0 \\ 0 & 0 & J_{3\theta} \end{bmatrix} \begin{bmatrix} \dot{\theta_{11}} \\ \dot{\theta_{12}} \\ \dot{\theta_{13}} \end{bmatrix}$$

$$\begin{aligned} J_{ix} &= \cos(\theta_{1i}+\theta_{2i})\sin\theta_{3i}\cos\phi_i - \cos\theta_{3i}\sin\phi_i \\ J_{iy} &= \cos(\theta_{1i}+\theta_{2i})\sin\theta_{3i}\sin\phi_i + \cos\theta_{3i}\cos\phi_i \\ J_{iz} &= \sin(\theta_{1i}+\theta_{2i})\sin\theta_{3i} \\ J_{i\theta} &= a\sin\theta_{2i}\sin\theta_{3i} \end{aligned}$$



Jacobian

$$v_p = J\dot{\theta}$$

- $J = J_x^{-1} J_\theta$ is the **Jacobian** of the delta robot
- $v_p = [v_{px}, v_{py}, v_{pz}]^T$ **velocity of point** P of the moving base
- $\dot{\theta} = \left[\dot{\theta_{11}}, \dot{\theta_{12}}, \dot{\theta_{13}}\right]^T$ is the **angular velocity** of the actuators

Theory: Method A (Dynamics)

Theory of Lagrangian equations for systems with holonomous ligatures in explicit form

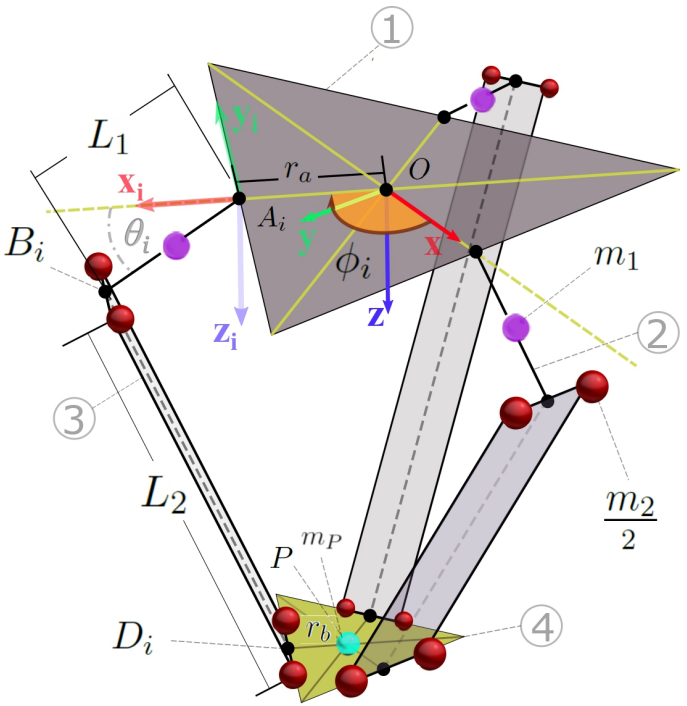
$$\frac{d}{dt}\left(\frac{\delta L}{\delta \dot{q}_j}\right)-\frac{\delta L}{\delta q_j}=\sum_{l=1}^{K^{(h)}}\lambda_l\frac{\delta f_l^{(h)}}{\delta q_j}+Q_j^{(NU)}(q_j)$$

Lagrange Eq. with generalized coordinates $q_j=[X_p,Y_p,Z_p]$

$$(m_p+3m_2)\ddot{X}_p-2\sum_{l=i=1}^{3}\lambda_l(X_P-rcos\phi_i-L_1\cos\theta_i\cos\phi_i)=F_{px}$$

$$(m_P+3m_2)\ddot{Y}_p-2\sum_{l=i=1}^{3}\lambda_l(Y_P-rsin\phi_i-L_1\cos\theta_i\sin\phi_i)=F_{py}$$

$$(m_P+3m_2)\ddot{Z}_p-2\sum_{l=i=1}^{3}\lambda_l(Z_P-L_1\sin\theta_i)-(m_p+3m_2)g=F_{pz}$$



Inverse dynamics: Lagrangian equations with generalized coordinates $q_j=[\theta_1,\theta_2,\theta_3]$

$$\tau_i=\left(\frac{1}{3}m_1+m_2\right)L_1^2\ddot{\theta}_i-\left(\frac{1}{2}m_1+m_2\right)gL_1\cos\theta_i-2\lambda_iL_1[(X_P\cos\phi_i+Y_P\sin\phi_i-r)\sin\theta_i-Z_P\cos\theta_i]$$

Theory: Method B (Jacobian)

Equations of **restriction**

$$\| \overrightarrow{\xi}_{J_i P_i} \|^2 - L_B^2 = \overrightarrow{s_i}^T \cdot \overrightarrow{s_i} - L_B^2 = 0 \quad ; \; i \in \{1,2,3\}$$

Where:

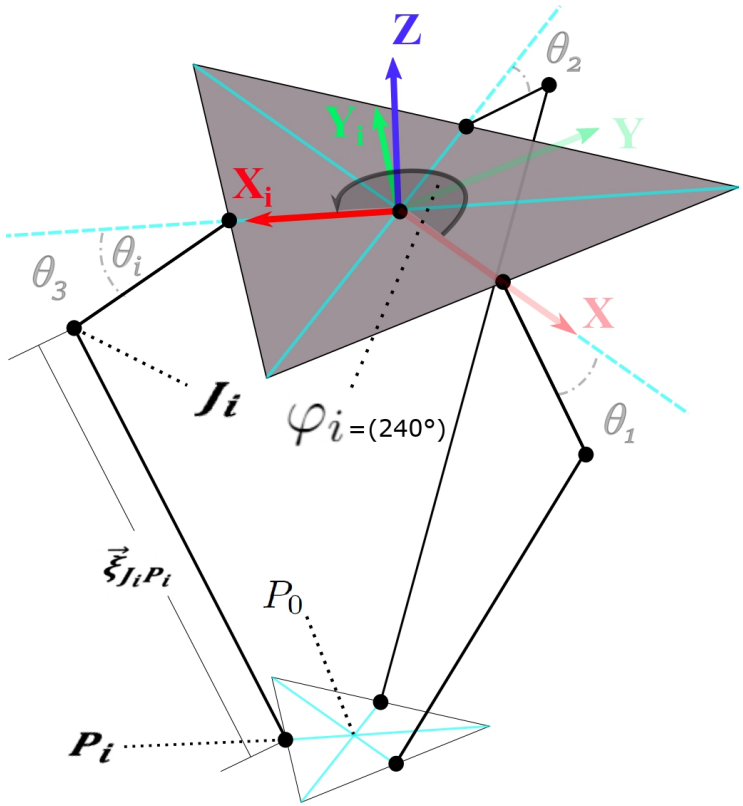
$$\overrightarrow{s_i} = \overrightarrow{P_0} - \left(\overrightarrow{F_i} + \overrightarrow{\xi_{F_i J_i}} \right) = \begin{bmatrix} P_{0x} \\ P_{0y} \\ P_{0z} \end{bmatrix} - R_i^R \left(\begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} L_A \cos(\theta_i) \\ 0 \\ -L_A \sin(\theta_i) \end{bmatrix} \right)$$

By deriving the constraint equation and separating the terms $\overrightarrow{P_0}$ and $\overrightarrow{\theta}$

$$\overrightarrow{\dot{P}_0} = -J_1 \, J_2 \, \overrightarrow{\dot{\theta}}$$

$$\begin{bmatrix} \dot{P}_{0x} \\ \dot{P}_{0y} \\ \dot{P}_{0z} \end{bmatrix} = - \begin{bmatrix} \overrightarrow{s_1}^T \\ \overrightarrow{s_2}^T \\ \overrightarrow{s_3}^T \end{bmatrix}^{-1} \begin{bmatrix} \overrightarrow{s_1}^T \cdot \overrightarrow{b_1} & 0 & 0 \\ 0 & \overrightarrow{s_2}^T \cdot \overrightarrow{b_2} & 0 \\ 0 & 0 & \overrightarrow{s_3}^T \cdot \overrightarrow{b_3} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

Where: $\overrightarrow{b_i} = R_i{}^R [L_A \sin(\theta_i), 0, L_A \cos(\theta_i)]$



Jacobian

$$\overrightarrow{\dot{P}_0} = J \, \overrightarrow{\dot{\theta}}$$

- $J = -J_1 \, J_2$ is the delta robot's **jacobian**
- $\overrightarrow{\dot{P}_0} = \left[\dot{P}_{0x}, \dot{P}_{0y}, \dot{P}_{0z} \right]^T$ is the **velocity of the point** P_0 of the mobile base and $\overrightarrow{\dot{\theta}} = \left[\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3 \right]^T$ is the **angular velocity** of the actuators.

Theory: Method B (Dynamics)

Virtual work: D'Alembert's Principle

Static case: The virtual work δW produced by an **external force** F_i acting on a body producing a virtual **linear displacement** δr_i:

$$\delta W = \sum_{i=1}^{N} F_i * \delta r_i = 0$$

Dynamic case: The above equation can be extended by adding **inertial forces** and **rotation**. Adding the virtual work δW produced by an **external torque** τ acting on the same body produces a **virtual angular displacement** $\delta\theta$:

$$\delta W = \sum_{i=1}^{N} \left[(F_i - m_i a_i) * \delta r_i + \left(\tau - I\ddot{\theta} \right) * \delta\theta \right] = 0$$

Inverse dynamics: torque of actuators

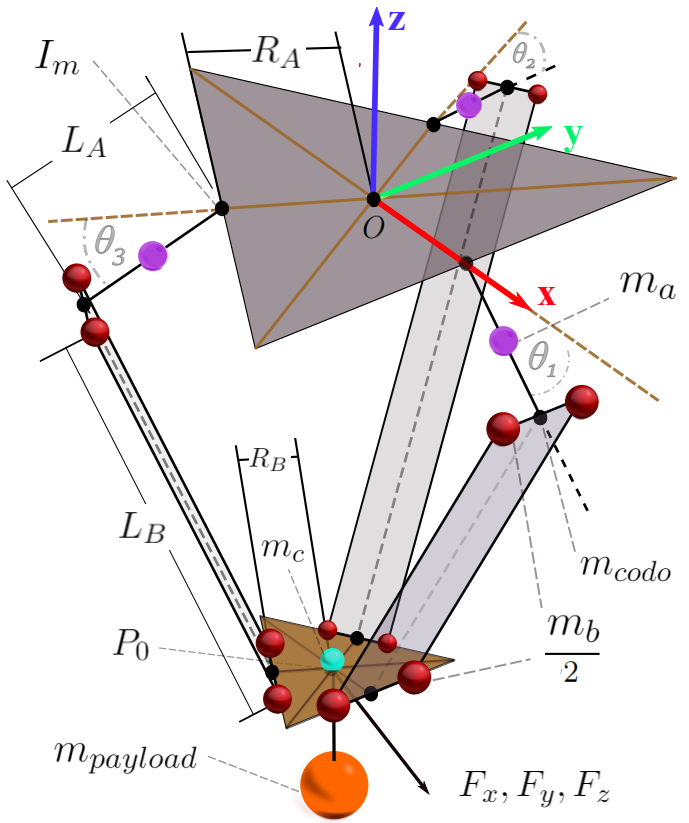
$$\vec{\tau} = I_b \vec{\ddot{\theta}} + J^T m_{nt} \vec{\ddot{P}}_0 - J^T \vec{F_g} - \vec{\tau_{Gb}}$$

$$\vec{\tau} = M(\theta) \vec{\ddot{\theta}} + C\left(\theta, \dot{\theta}\right) \vec{\dot{\theta}} + \vec{G}(\theta)$$

Mass matrix: $M(\theta) = I_b + J^T m_{nt} J$

Coriolis coefficient: $C\left(\theta, \dot{\theta}\right) = J^T m_{nt} \dot{J}$

Gravity terms: $\vec{G}(\theta) = -J^T \vec{F_g} - \vec{\tau_{Gb}}$



$$I_b = [I_{b1}, 0, 0; 0, I_{b2}, 0; 0, 0, I_{b3}]$$

$$I_{bi} = I_m + L_A^2 \left(\frac{m_a}{3} + m_{codo} + 2r m_b \right)$$

$$m_{nt} = m_c + m_{payload} + 3 * 2 * (1 - r) m_b$$

$$\vec{F_g} = m_{nt} [0\,0 - g]^T$$

Theory: MSC ADAMS

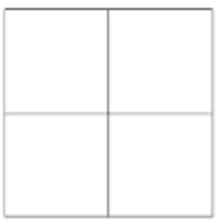
DEPARTAMENTO DE
INGENIERÍA
MECÁNICA
UNIVERSIDAD DE SANTIAGO DE CHILE

Why ADAMS?

State-of-the-art robot simulation software for kinematic and dynamic modeling.

Dynamic simulation and multi-body analysis (MBD)

Displacement = Deformation + Displacement of rigid body



(A) FEM

(B) MBD

Theory: ADAMS/Solver

Generalized Coordinates

ADAMS works with a Cartesian coordinate system, for each part it defines as generalized coordinates

Position + Orientation

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \wedge \varepsilon = \begin{bmatrix} \psi \\ \phi \\ \theta \end{bmatrix} \Rightarrow q_i = \begin{bmatrix} p_i \\ \varepsilon_i \end{bmatrix}$$

Equations of Motion

Generalized Lagrange

$$\frac{d}{dt}\left[\left(\frac{\delta K}{\delta \dot{q}}\right)^T\right] - \left(\frac{\delta K}{\delta q}\right)^T + {\Phi_q}^T \lambda = Q$$

Kinetic Energy

$$K = \frac{1}{2} u^T M u + \frac{1}{2} \overline{\omega}^T \overline{J} \overline{\omega}$$

Theory: Trajectory

Geometric path

A geometric path in **straight line** in **cartesian space** starts with a configuration X_{begin} and up to a final configuration X_{end}. The time scale of the linear trajectory is $s(t)$.

$$X : s \rightarrow \Re^3$$

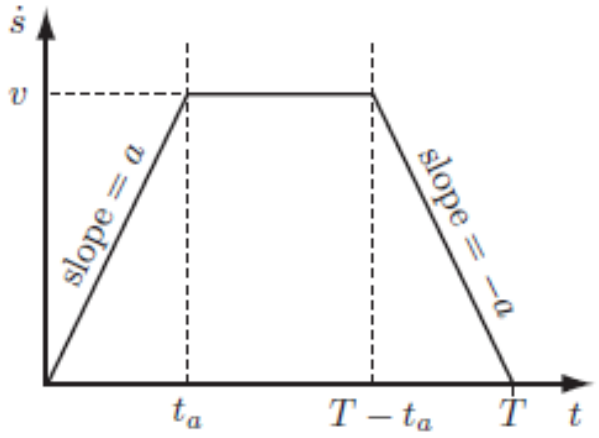
$$X : [0; 1] \rightarrow \Re^3$$

$$X(s) = X_{begin} + s(X_{end} - X_{begin})$$

Time Scale

Trapezoidal velocity profile **(LSPB)** consist of 3 phases:

- Constant **Acceleration** $\ddot{s} = a$ de tiempo t_a
- Constant **Velocity** $\dot{s} = v$ de tiempo $t = T - 2t_a$
- Constant **Deceleration** $\ddot{s} = -a$ de tiempo t_a



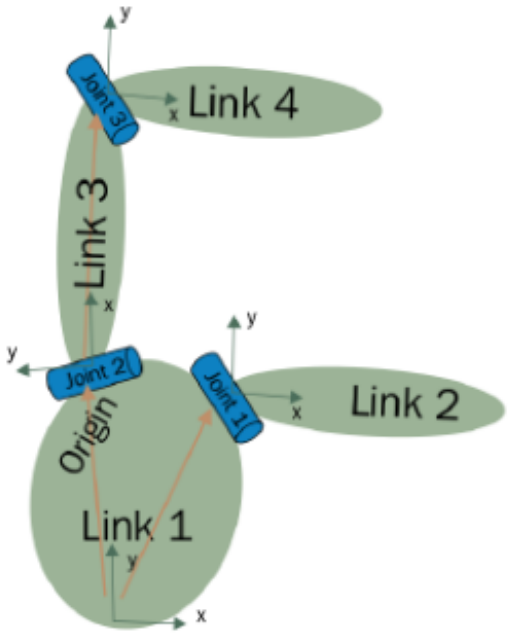
Theory: Visualization

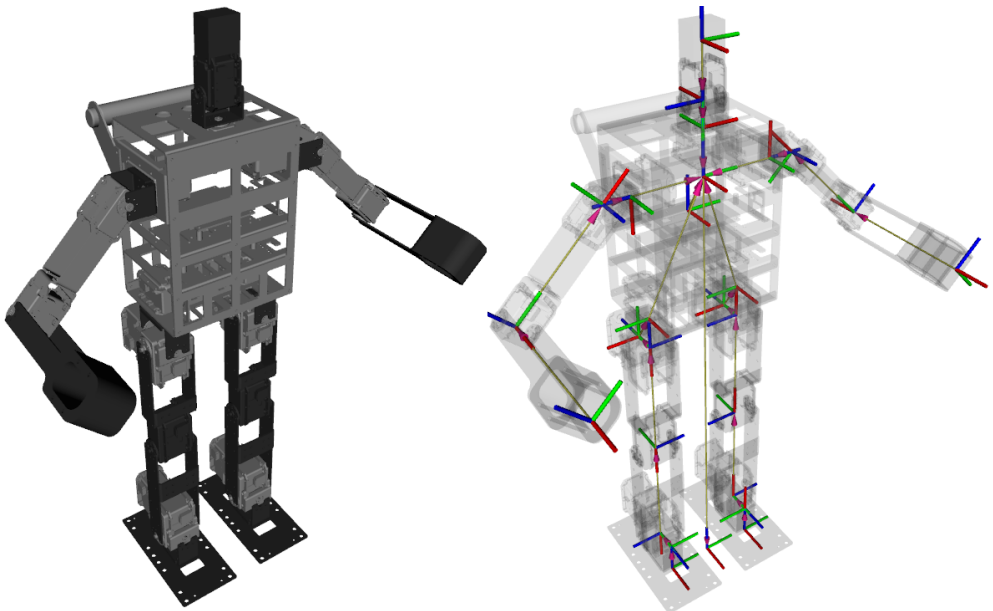
Unified Robot Description Format (URDF)

In ROS it is possible to visualize a **3D model** of a robot by using URDF files. The description of the model basically consists of joining two sets: the set of **links** and the set of **joints**.

Transform Frames (tf)

TF is designed to provide a standard way to track **coordinate frames** and transform data within the entire system over time. The tf package can **track and maintain the relationship** between multiple coordinate frames.

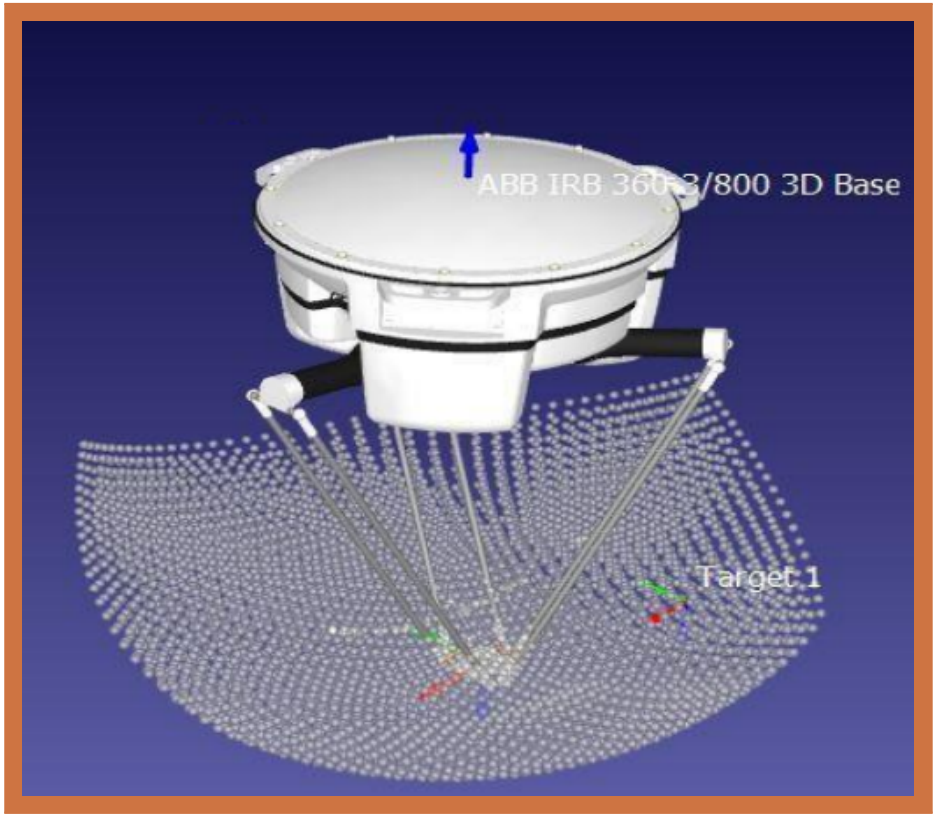




Theory: Workspace

Definition

The **discretization method**, which is based on numerical methods, consists of discretizing the space in three dimensions, solving the **inverse kinematics** for each point and verifying the **constraints** that limit the workspace.



Robot Workspace
ABB irb 360-3/800

Content

1 Motivation

2 Hypothesis and Objectives

3 Theory

4 Development

5 Delta robot specifications

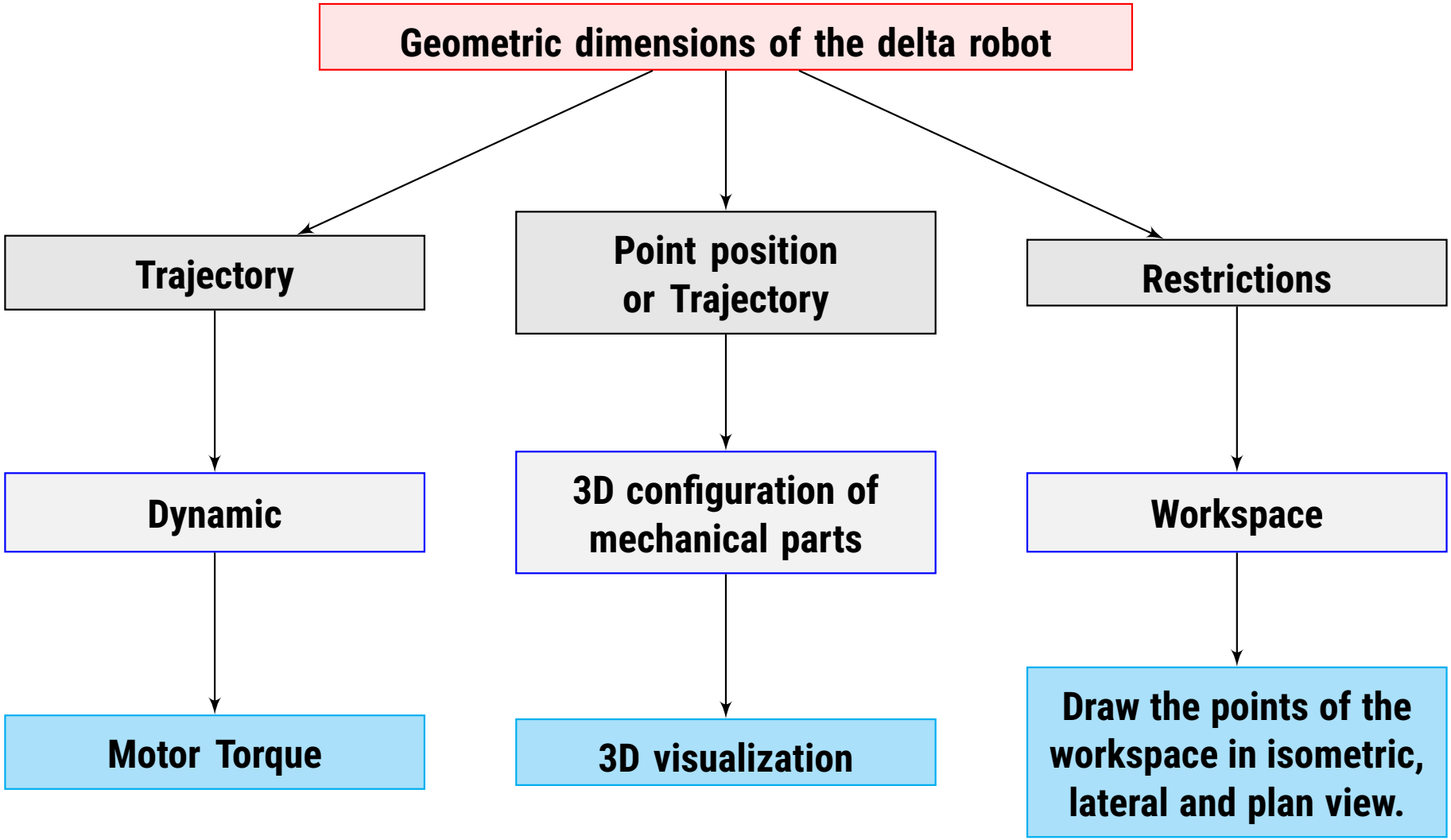
6 Results

7 Conclusions

8 Future lines of research and references

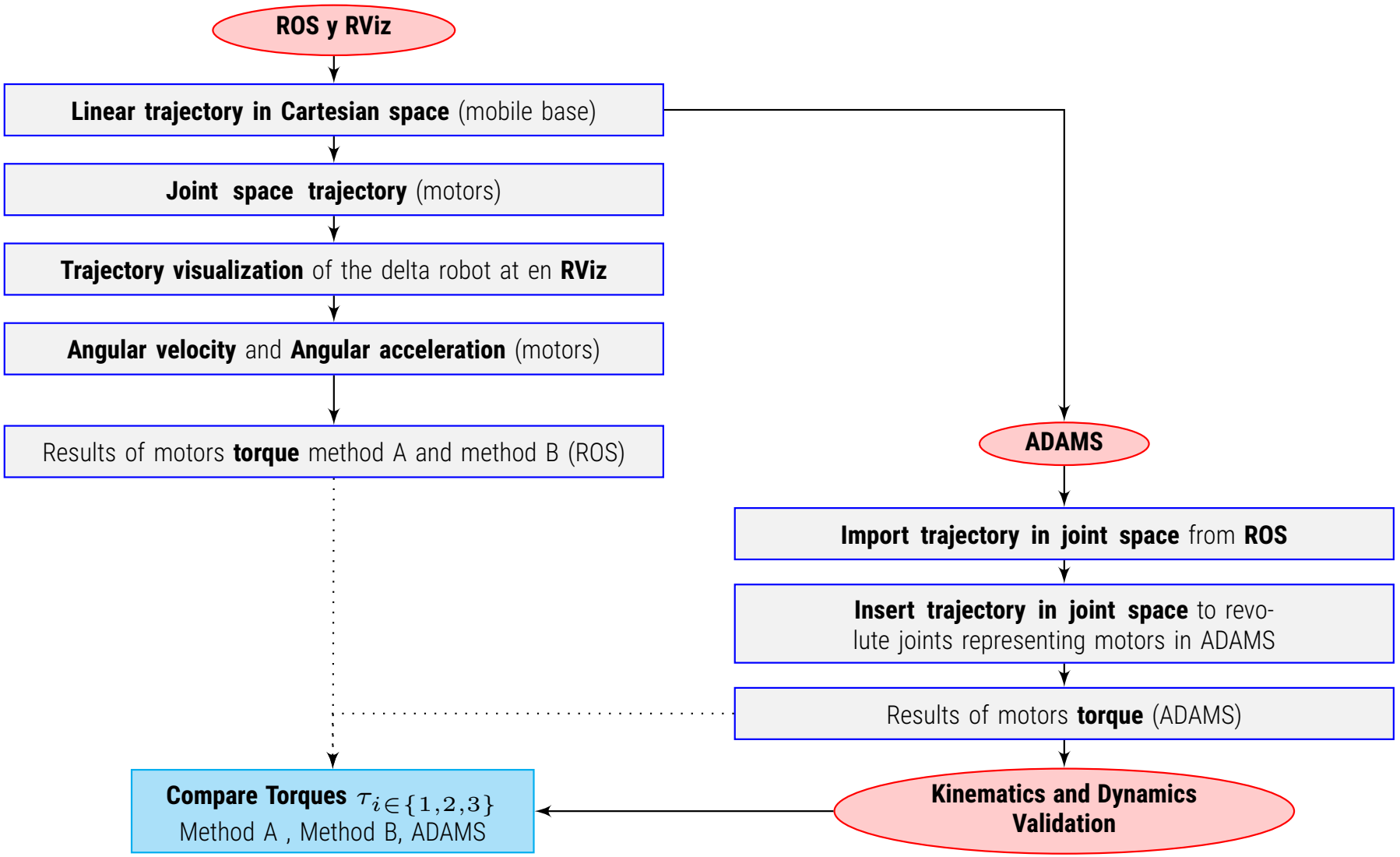
Development: Summary of tasks to be performed

DEPARTAMENTO DE
INGENIERÍA
MECÁNICA
UNIVERSIDAD DE SANTIAGO DE CHILE



Development: Dynamic validation workflow

DEPARTAMENTO DE INGENIERÍA MECÁNICA
UNIVERSIDAD DE SANTIAGO DE CHILE



Development: Simulated trajectories

The trajectories that are created to test the kinematics and dynamics algorithms are:

Trajectory	$P_i(x,y,z)[mm]$	$P_f(x,y,z)[mm]$	$v_{max}[\frac{mm}{s}]$	$a_{max}[\frac{mm}{s^2}]$
1	(-300,0,-450)	(300,150,-750)	2000	40000
2	(-300,-150,-450)	(300,150,-750)	2000	40000
3	**(300,-150,-450)**	**(-300,150,-750)**	**2000**	**40000**
4	(400,150,-450)	(-400,150,-450)	2000	40000
5	(-300,0,-450)	(300,150,-750)	200	10000
6	(-300,-150,-450)	(300,150,-750)	200	10000
7	**(300,-150,-450)**	**(-300,150,-750)**	**200**	**10000**
8	(400,150,-450)	(-400,150,-450)	200	10000

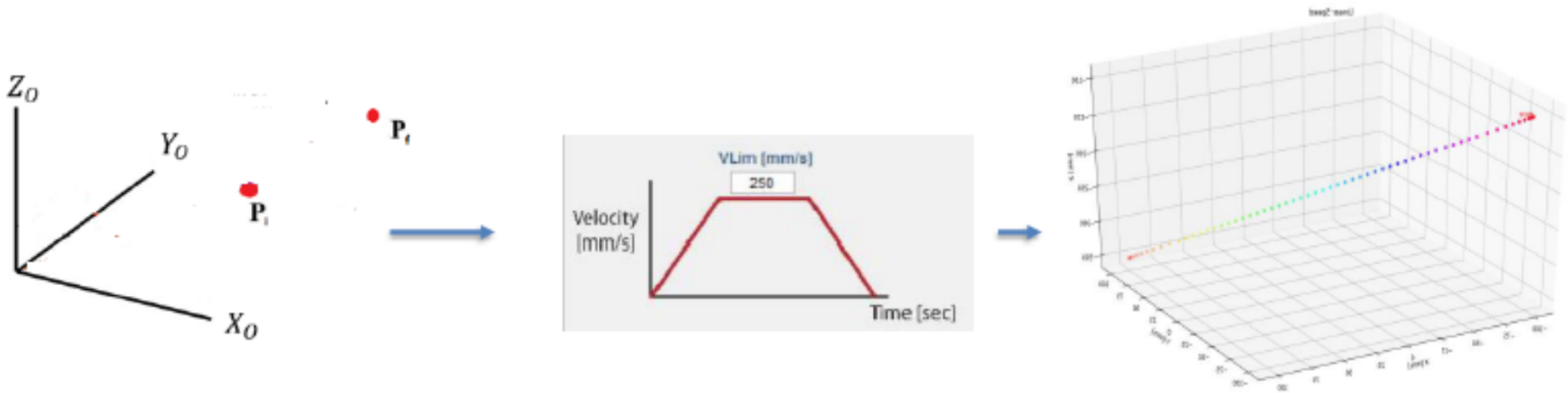
Development: Trajectory

$$P_i = [X_i Y_i Z_i]$$
$$P_f = [X_f Y_f Z_f]$$
$$v_{max}$$
$$a_{max}$$

→ **Trajectory** →

Points of the trajectory in **cartesian space** XYZ and **joint space** $\theta_1\theta_2\theta_3$ whit **time scale** $s(t)$

Trajectory function inputs and outputs



Graphical example of the flow chart for trajectory function.

Development: Method A

DEPARTAMENTO DE INGENIERÍA MECÁNICA
UNIVERSIDAD DE SANTIAGO DE CHILE

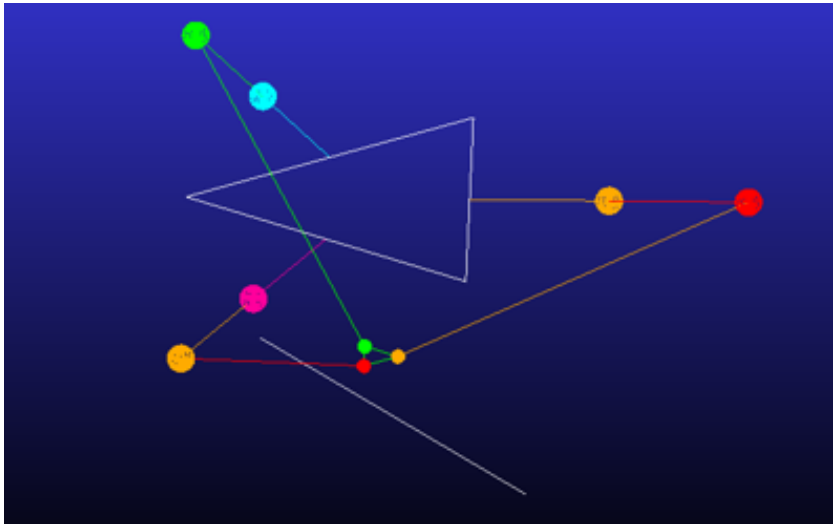
$\theta_1, \theta_2, \theta_3$	**Direct Kinematics**	$E_0(x_0, y_0, z_0)$
$E_0(x_0, y_0, z_0)$	**Inverse Kinematics**	$\theta_1, \theta_2, \theta_3$
$P_x, P_y, P_z, \theta_{11}, \theta_{12}, \theta_{13}$	**Jacobian**	$J, J_\theta, J_x,$ $\theta_{31}, \theta_{32}, \theta_{33},$ $\theta_{21}, \theta_{22}, \theta_{23}$
$P_x, P_y, P_z,$ $\theta_{11}, \theta_{12}, \theta_{13},$ v_{px}, v_{py}, v_{pz}	**Acceleration Kinematics**	$J_\theta, J_x, \dot{J}_\theta, \dot{J}_x,$ $\theta_{31}, \theta_{32}, \theta_{33},$ $\theta_{21}, \theta_{22}, \theta_{23}$
$\ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3,$ $\theta_1, \theta_2, \theta_3,$ $X_p, Y_p, Z_p,$ $\ddot{X}_p, \ddot{Y}_p, \ddot{Z}_p,$ $F_{px}, F_{py}, F_{pz},$ $m_{playload}$	**Inverse Dynamics**	τ_1, τ_2, τ_3

Development: Method B

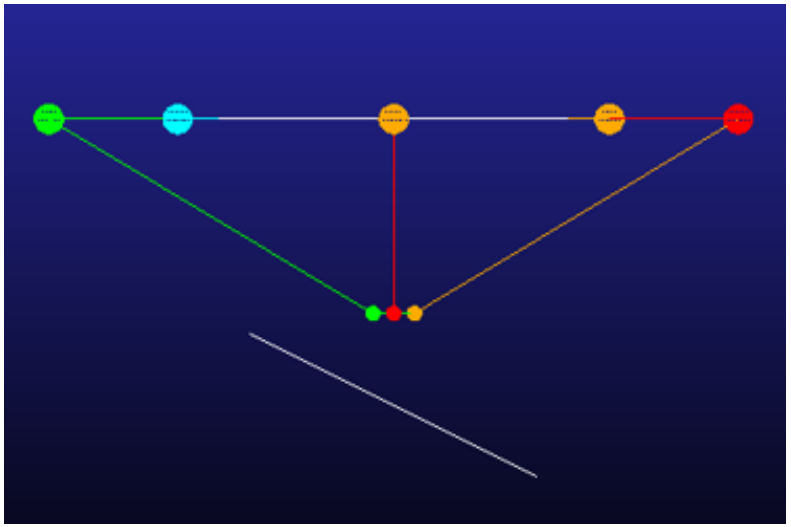
DEPARTAMENTO DE INGENIERÍA MECÁNICA
UNIVERSIDAD DE SANTIAGO DE CHILE

$\theta_1, \theta_2, \theta_3$	**Direct Kinematics**	$P_0(P_{0x}, P_{0y}, P_{0z})$
$P_0(P_{0x}, P_{0y}, P_{0z})$	**Inverse Kinematics**	$\theta_1, \theta_2, \theta_3$
P_{0x}, P_{0y}, P_{0z}, $\theta_1, \theta_2, \theta_3$	**Jacobian**	$-J_1, J_2, J$
P_{0x}, P_{0y}, P_{0z}, $\theta_1, \theta_2, \theta_3$, $\dot{P}_{0x}, \dot{P}_{0y}, \dot{P}_{0z}$	**Acceleration Kinematics**	$-J_1, J_2, J$, $\overrightarrow{\dot{s}_i}^T, K, \overrightarrow{\dot{\theta}}, \overrightarrow{s_i}^T$
$\ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3$, $\theta_1, \theta_2, \theta_3$, F_{px}, F_{py}, F_{pz}, $J, m_{playload}, \dot{J}$, $\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3$	**Inverse Dynamics**	τ_1, τ_2, τ_3

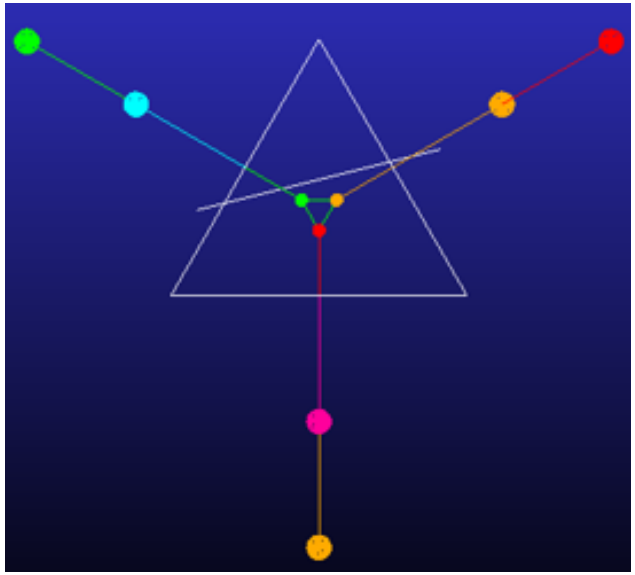
Development: ADAMS (Robot delta model)



(a) Isometric View



(b) Front View

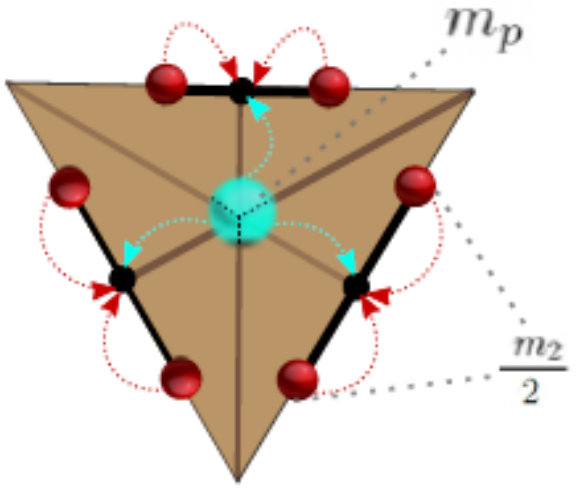


(c) Top View

Development: ADAMS (Simplifications)

Mass Simplification

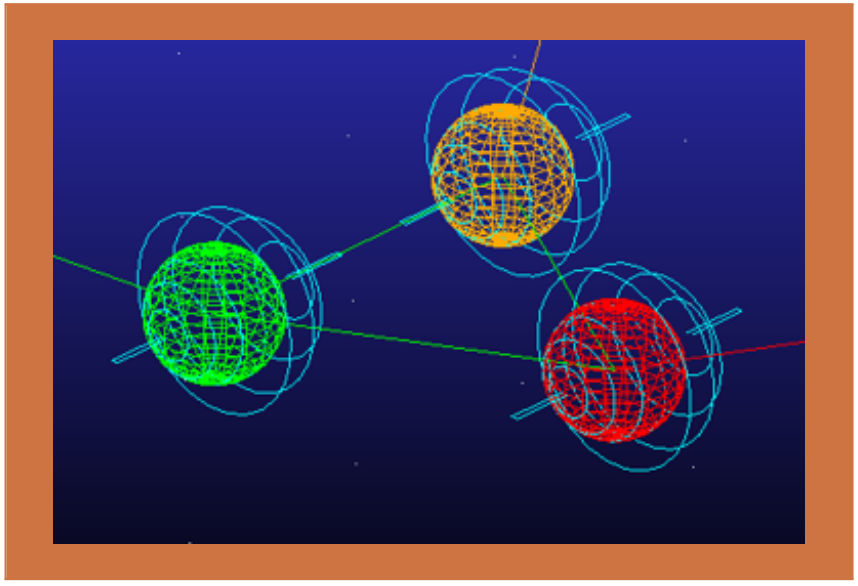
The mass of the effector (green dot) is divided into three portions and added to the three masses of the robot forearms (black dots).



$$M_{total,junta} = m_2 + \frac{1}{3}m_p$$

Joint - fixed base simplification

Difficulty in connecting the three open kinematic chains of the robot arms into closed chains.



Development: Visualization in VRiz

DEPARTAMENTO DE INGENIERÍA MECÁNICA
UNIVERSIDAD DE SANTIAGO DE CHILE

Trajectory	$\theta_{i\in\{1,2,3\}}$
Trajectory	XYZ
Time Scale	$s(t)$

Calculation of joint values

Angle	θ_{2i}
Angle	θ_{3i}
URDF joint names	string[ ]

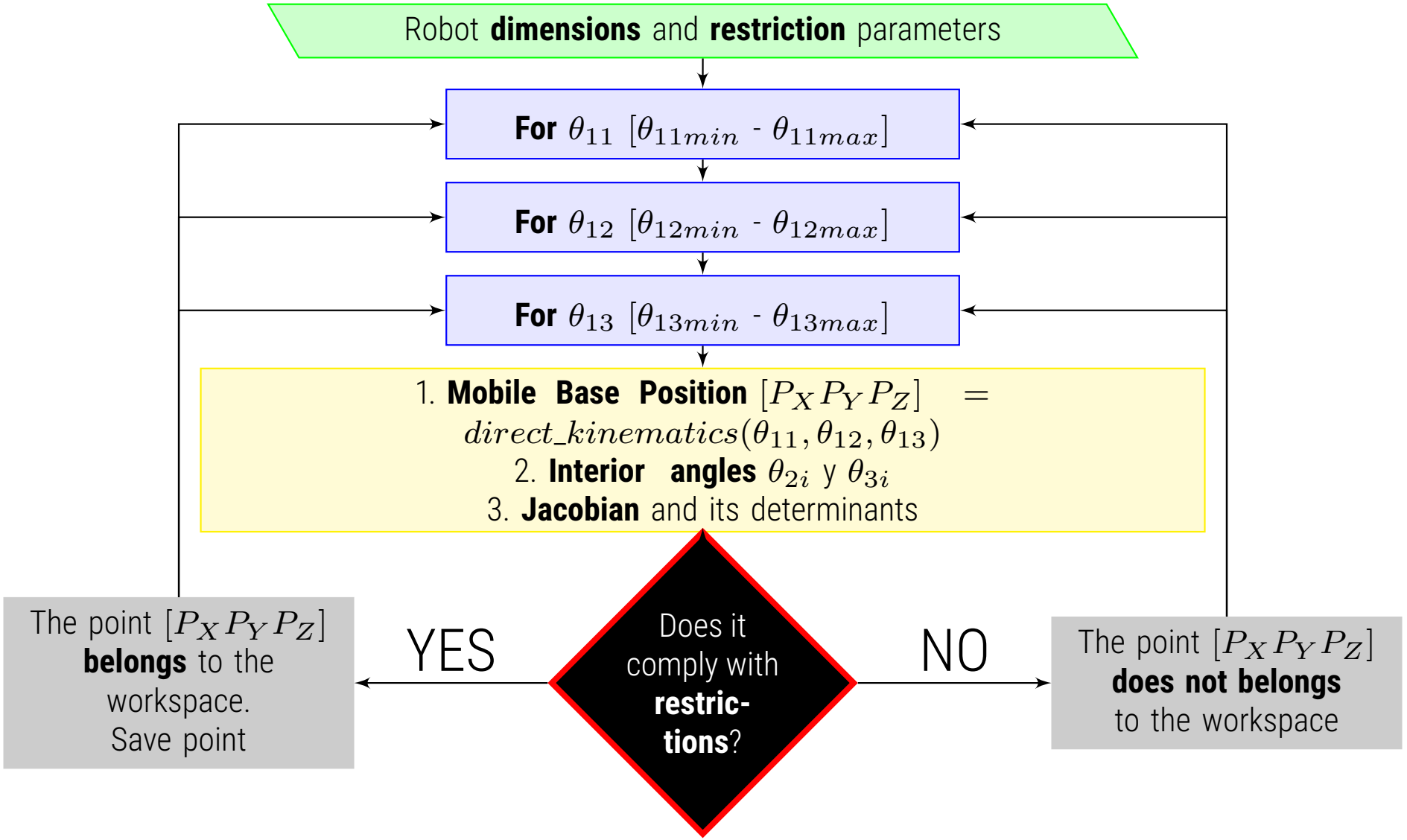
TF (Transform Frames)
Calculate transformation matrices of joint frames according to their current value and the URDF model of the delta robot.

Visualization of the URDF model of the delta robot + TF in RViz

Nodes to obtain the visualization of the delta robot

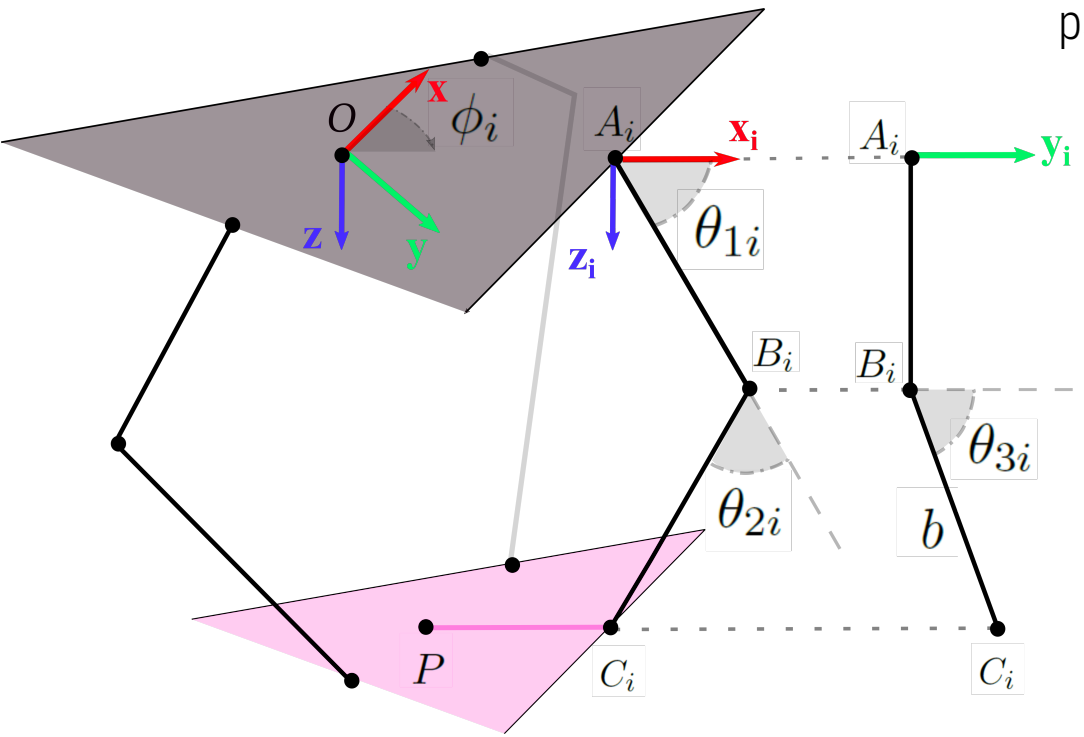
Development: Workspace

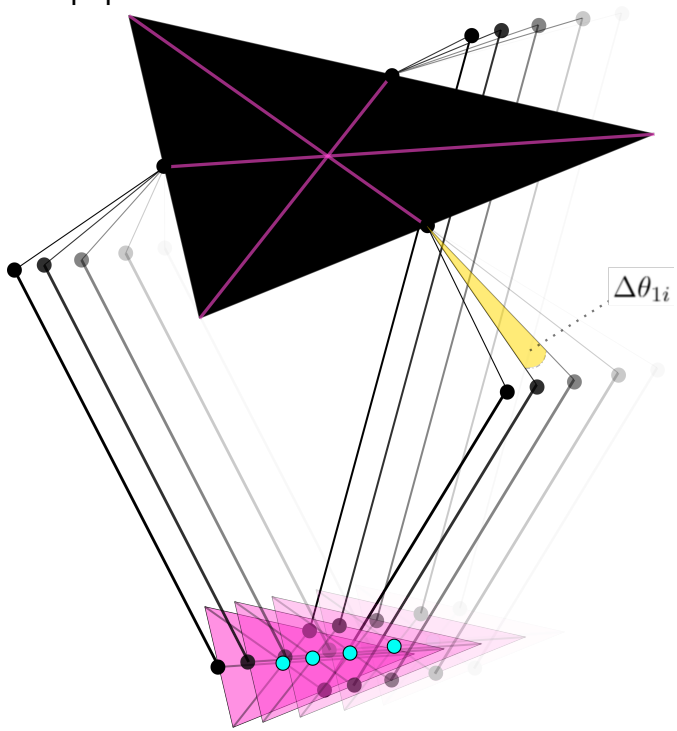
DEPARTAMENTO DE INGENIERÍA MECÁNICA
UNIVERSIDAD DE SANTIAGO DE CHILE



Development: Workspace restrictions

- **Limits** imposed on **actuator angles** $[\theta_{1i,min}\text{-}\theta_{1i,max}]$ for each actuator $i \in \{1, 2, 3\}$.
- Resolution based on actuator step size, i.e. **discretization** $\Delta\theta_{1i}$ of the range imposed by the limits of the previous point for each actuator $i \in \{1, 2, 3\}$.
- **Internal angles** restrictions θ_{2i} y θ_{3i} based on joint restrictions .
- Singularities that are determined by **the determinant of the Jacobian** $J = J_x^{-1} J_\theta$ when it is close to 0.
- **Limits** X,Y y Z. They are usually geometric volumes such as cylinders or parallelepipeds.





Development: Workspace

Restriction	Explanation	Min	Max
θ_{1i}	Angle arm	$-90°$	$90°$
$\Delta\theta_{1i}$	Discretization of ranges θ_{1i}	$5°$	
θ_{2i}	Interior angle	$5°$	$175°$
θ_{3i}	Interior angle	$45°$	$135°$
J_x	Jacobian mobile base	$6 * 10^{-1}$	
J_θ	Jacobian actuators	$4 * 10^{-3}$	
X	X workspace limit	$-400[mm]$	$+400[mm]$
Y	Y workspace limit	$-400[mm]$	$+400[mm]$
Z	Z workspace limit	$-300[mm]$	$-750[mm]$

Content

1 Motivation

2 Hypothesis and Objectives

3 Theory

4 Development

5 Delta robot specifications

6 Results

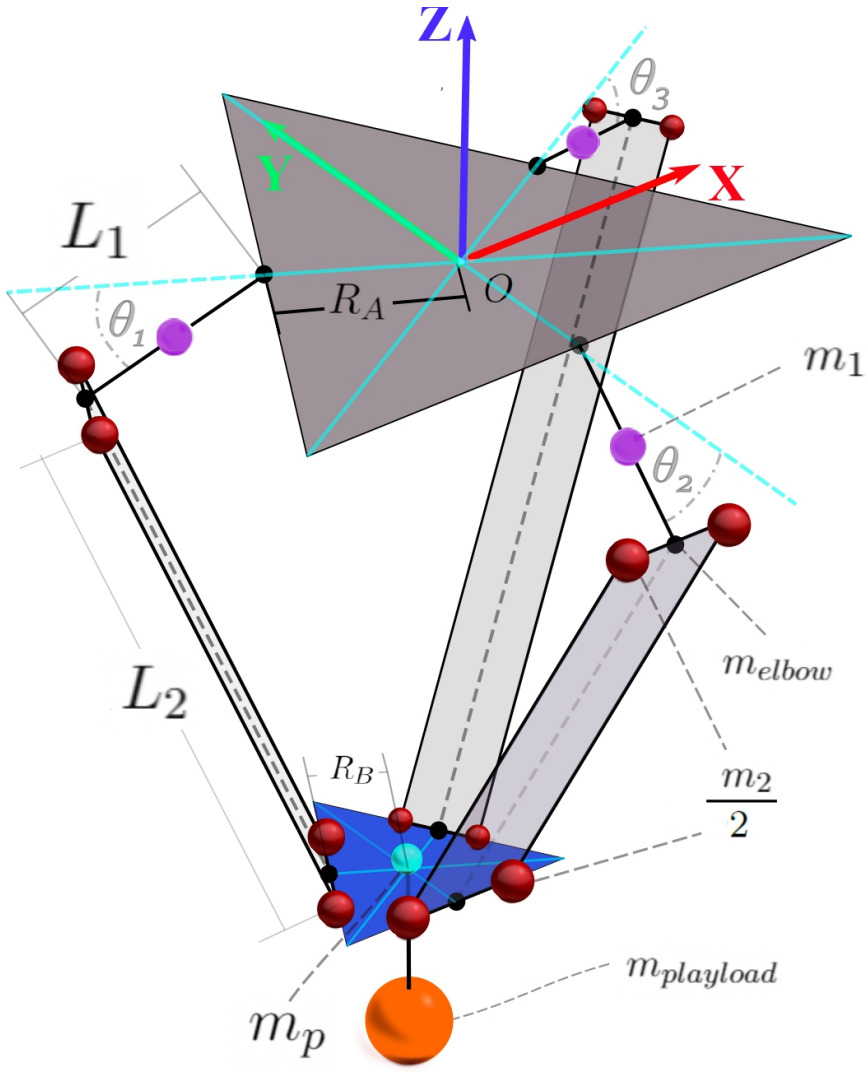
7 Conclusions

8 Future lines of research and references

Delta robot specifications

Parameter	Value
L_1	620 [mm]
L_2	880 [mm]
R_A	210 [mm]
R_B	50 [mm]
m_1	2.213 [kg]
m_2	0.6575 [kg]
m_p	0.510 [kg]
$m_{playload}$	0 [kg]
m_{elbow}	0 [kg]
I_m	0 [kg*m^2]
r_{mass}	0.5
g	9.81 [m/s^2]

Delta robot parameters



Graphical representation of delta robot

Content

1 Motivation

2 Hypothesis and Objectives

3 Theory

4 Development

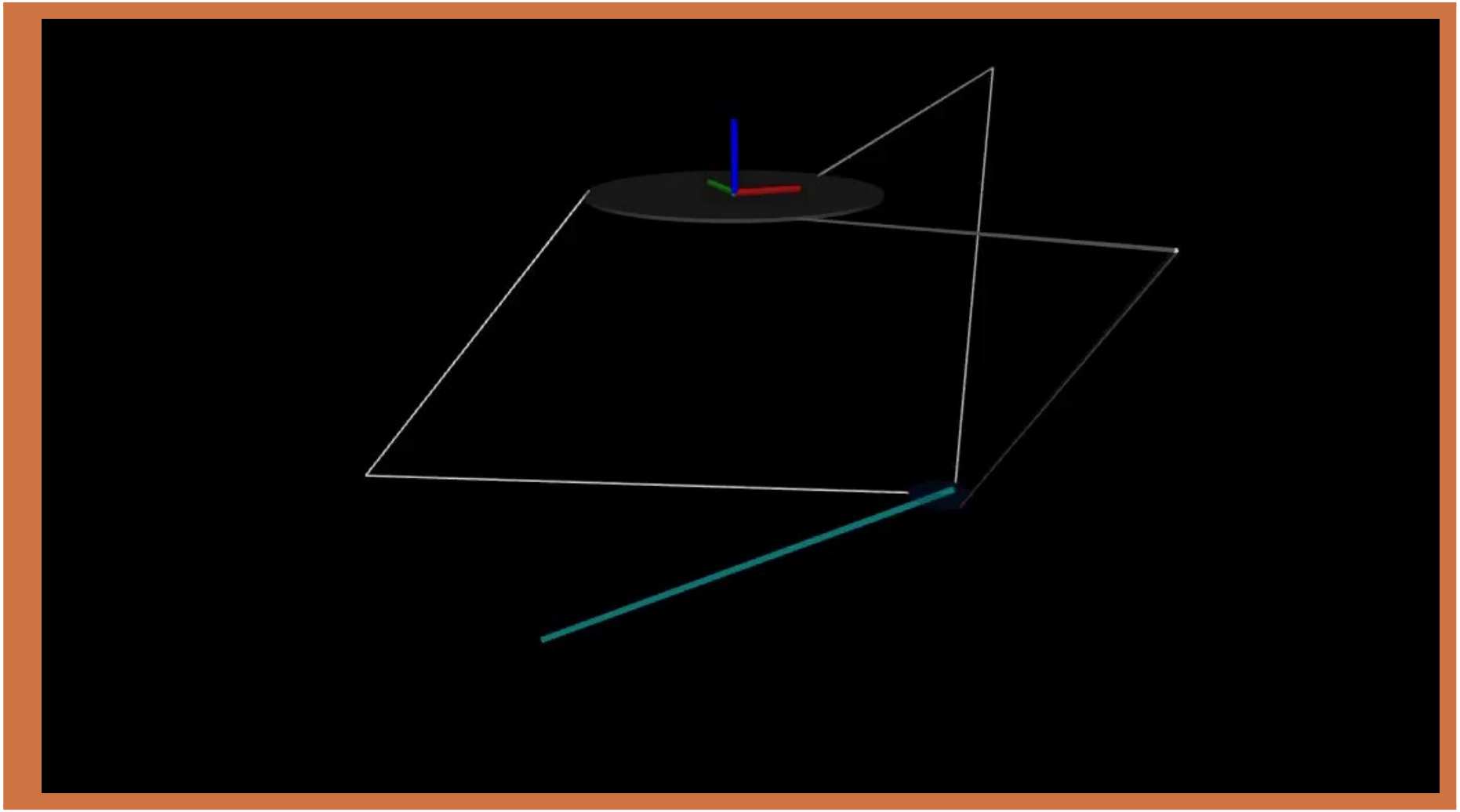
5 Delta robot specifications

6 Results

7 Conclusions

8 Future lines of research and references

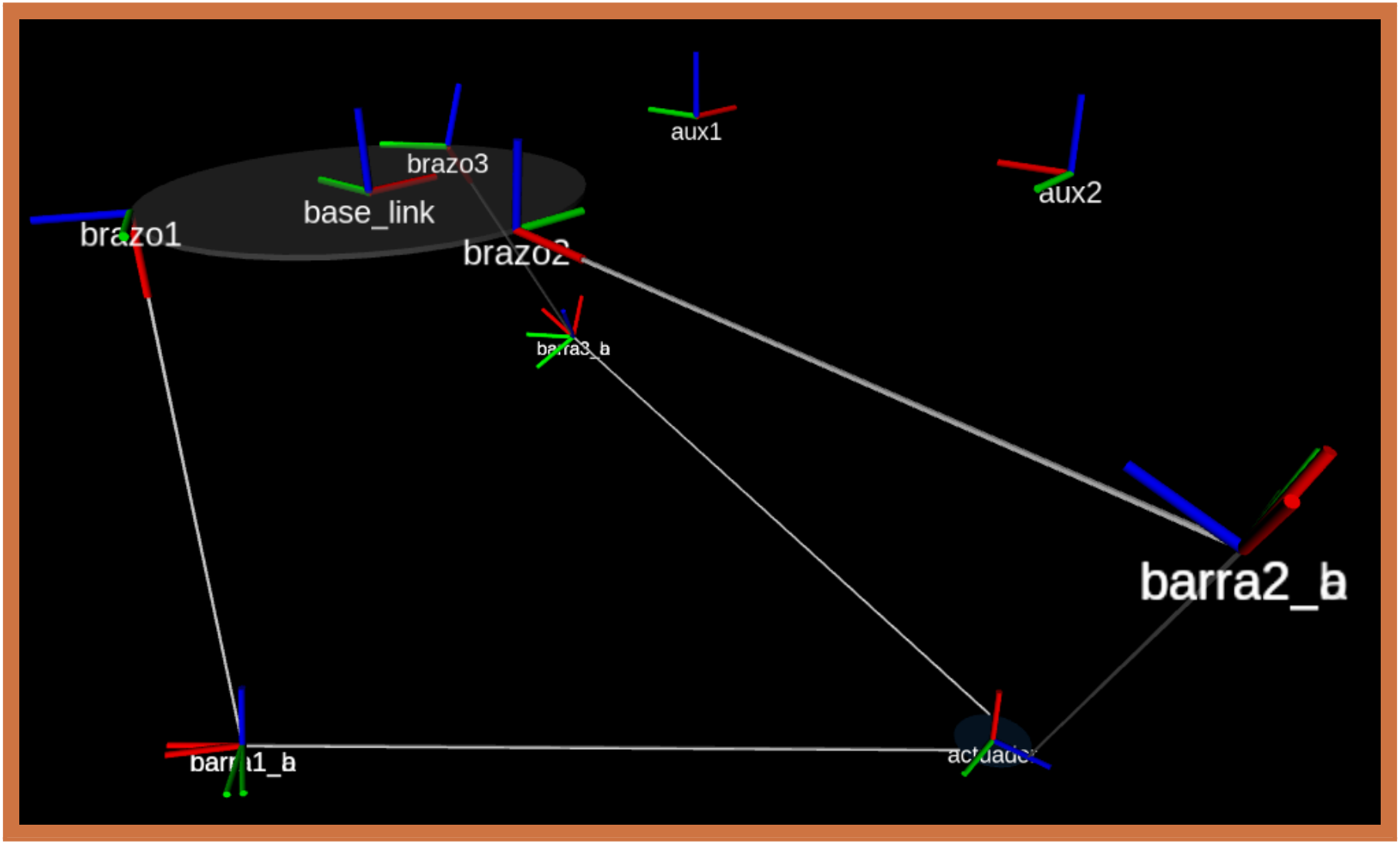
Results: Visualization in RViz



Simulation of **trajectory 3** in **RViz**

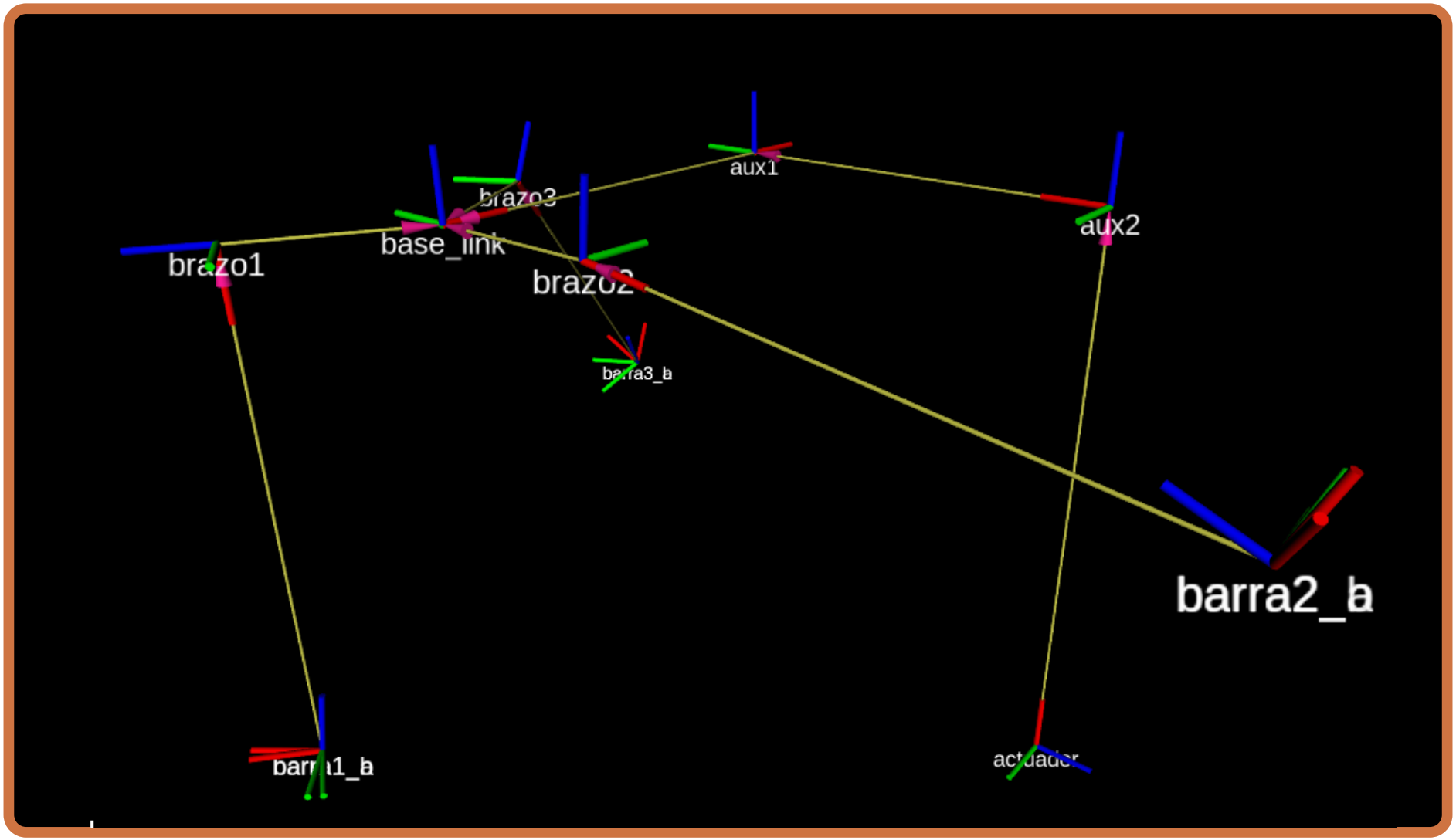
Results: Visualization in RViz

DEPARTAMENTO DE
INGENIERÍA
MECÁNICA
UNIVERSIDAD DE SANTIAGO DE CHILE



3D visualization of links and **frames** of joints of the robot delta in **RViz**.

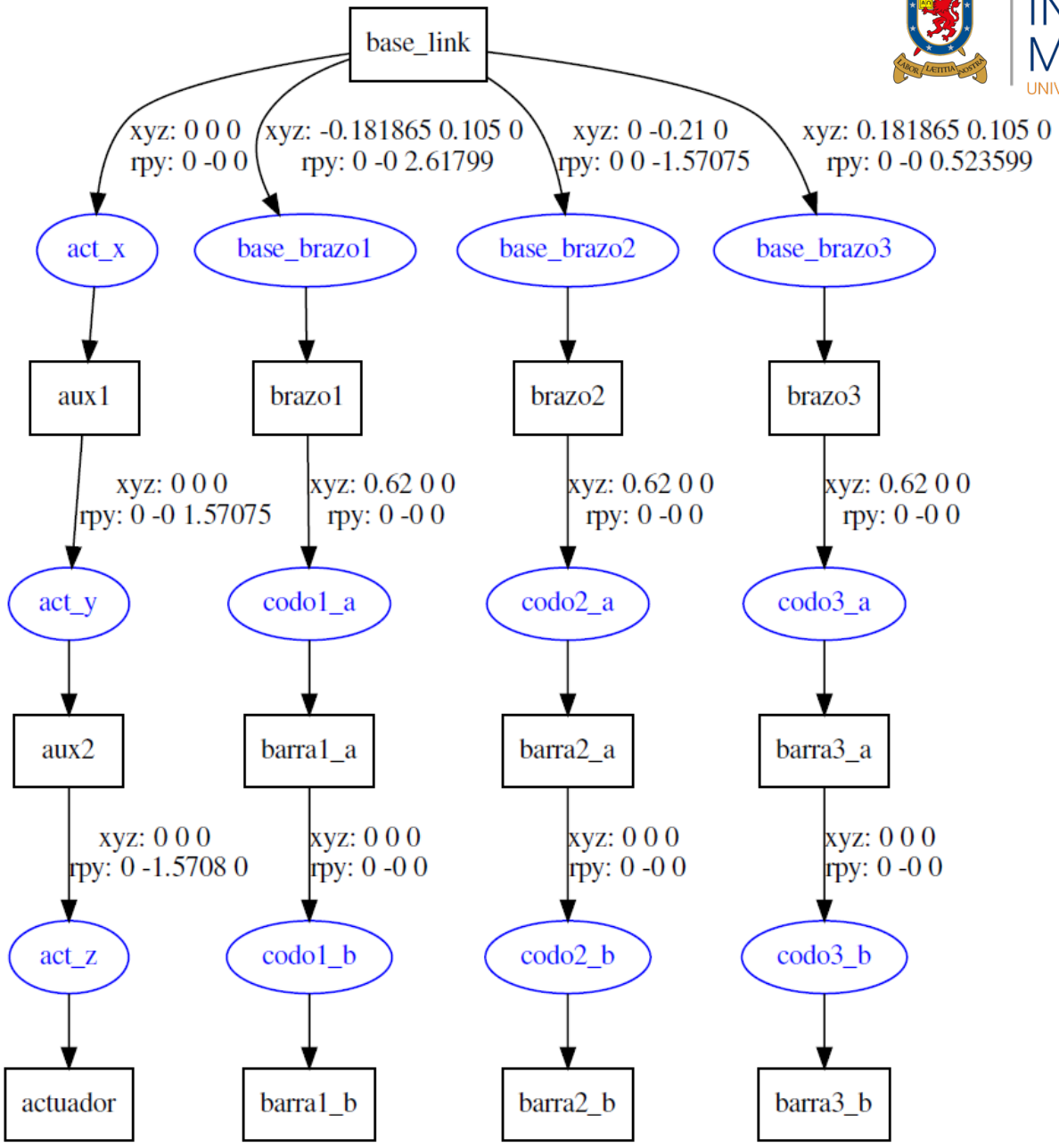
Results: Visualization in RViz



Connection between links and **frames** joints of the delta robot in **RViz**.

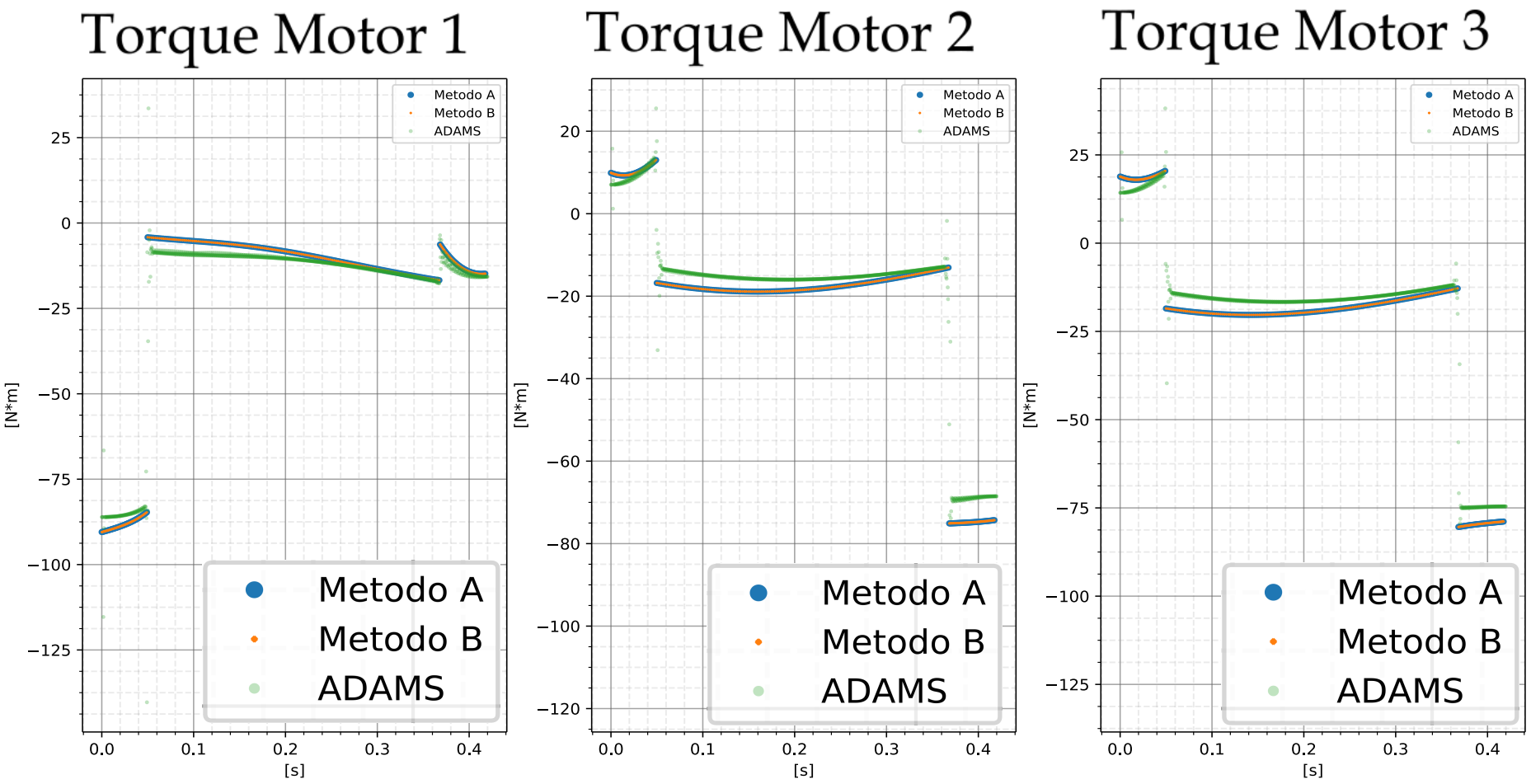
Results:

Graphical representation of the parent-child relationship of the delta robot model **URDF** in RViz.



Results: Inverse Dynamics

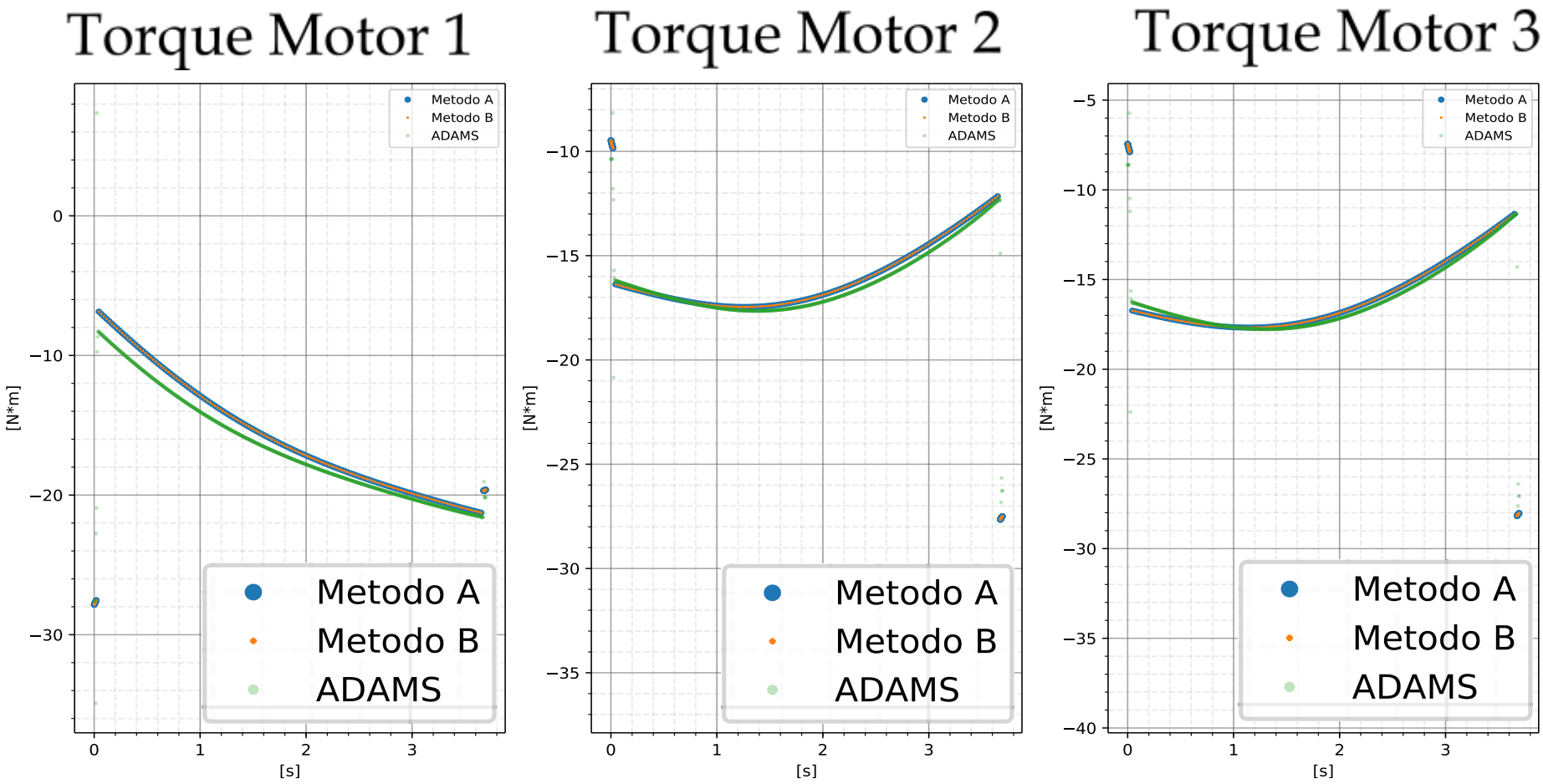
DEPARTAMENTO DE
INGENIERÍA
MECÁNICA
UNIVERSIDAD DE SANTIAGO DE CHILE



Torque of actuators for **trajectory 3 (Speed)**

Results: Inverse Dynamics

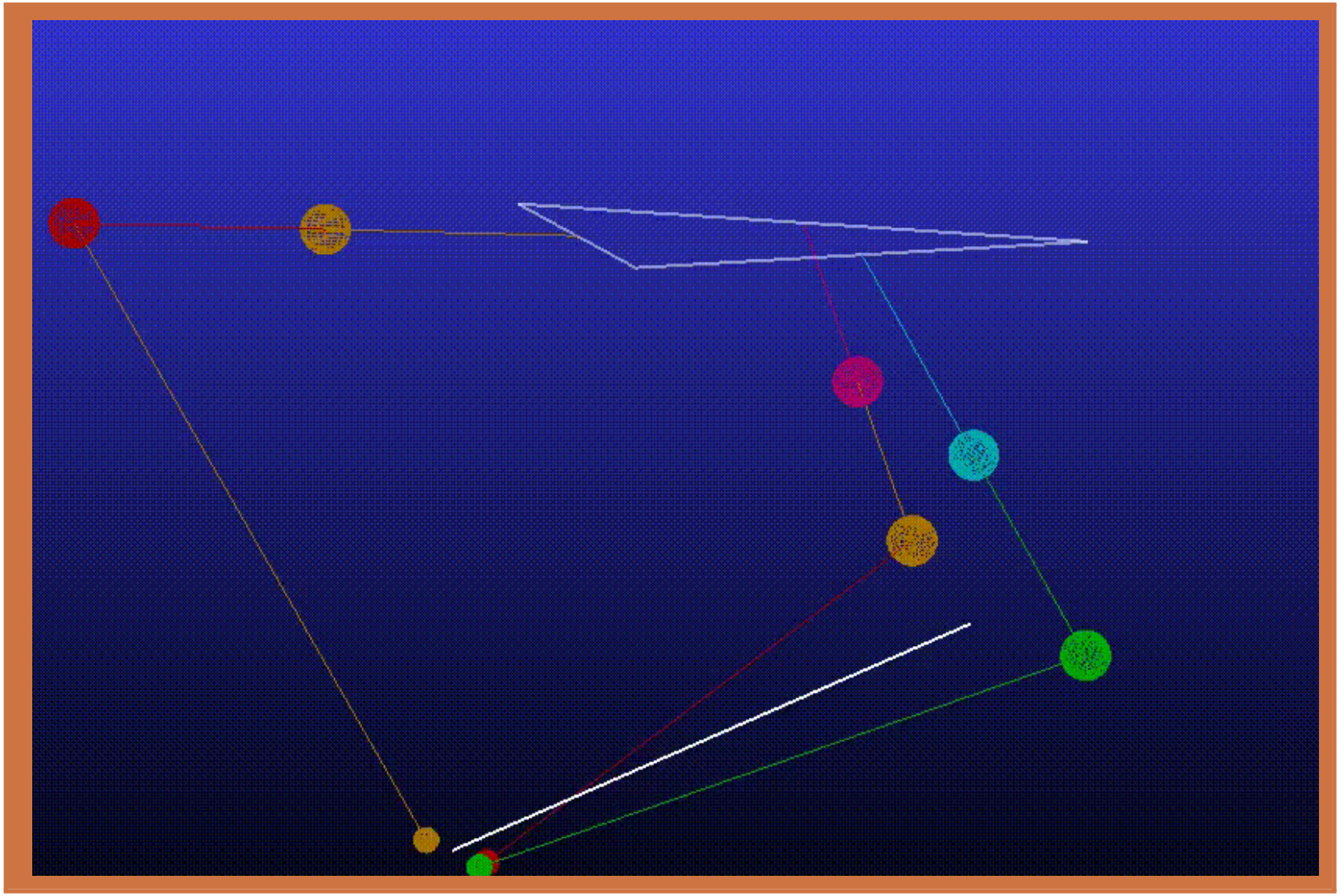
DEPARTAMENTO DE
INGENIERÍA
MECÁNICA
UNIVERSIDAD DE SANTIAGO DE CHILE



Torque of actuators for **trajectory 7 (Low)**

Results: Visualization and Dynamics ADAMS

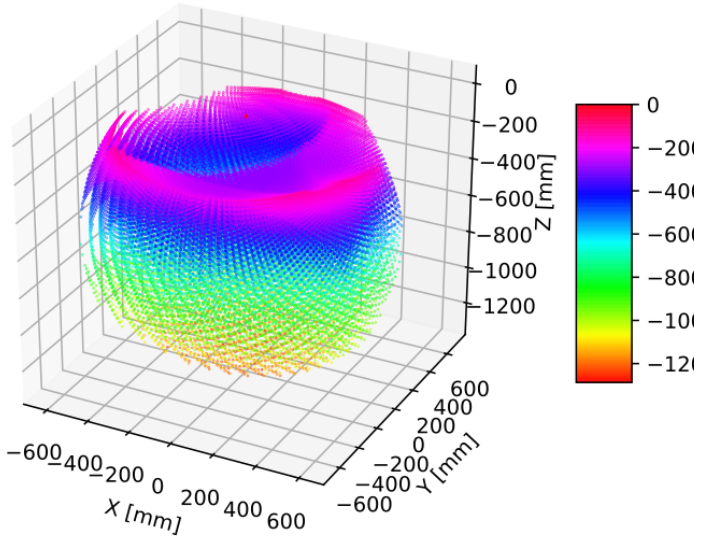
DEPARTAMENTO DE
INGENIERÍA
MECÁNICA
UNIVERSIDAD DE SANTIAGO DE CHILE



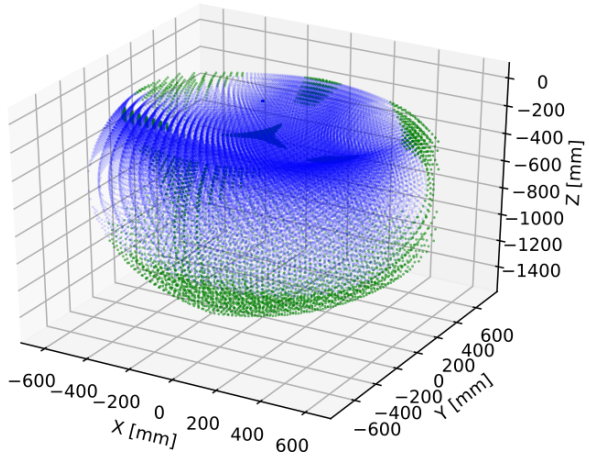
Simulation of **trajectory 1** in **ADAMS**

Results: Workspace

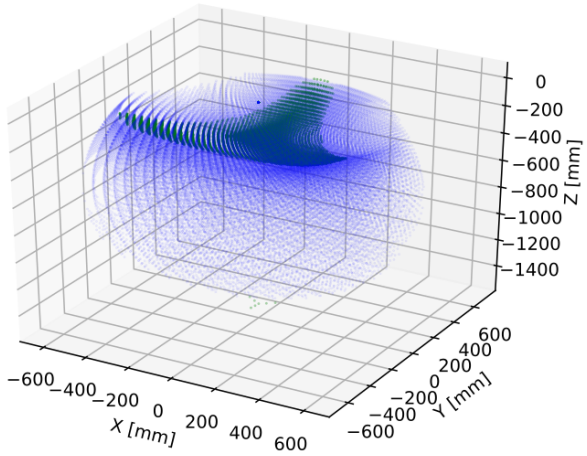
DEPARTAMENTO DE
INGENIERÍA
MECÁNICA
UNIVERSIDAD DE SANTIAGO DE CHILE



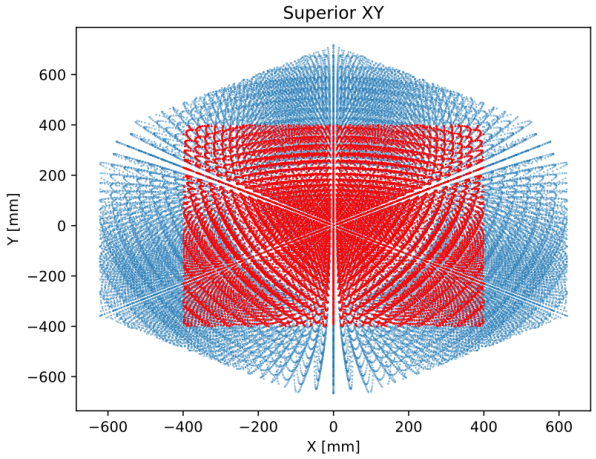
(a) Reachable points with **angular restrictions** **Color scale** represents height in z-axis



(b) Restricted points $J_x \approx 0$ **(green)** and angular restrictions **(blue)**



(c) Restricted points $J_\theta \approx 0$ **(green)** and angular restrictions **(blue)**



(d) Plane view XZ of angular restrictions, $J \approx 0$ and effective workspace **(red)**

Content

1 Motivation

2 Hypothesis and Objectives

3 Theory

4 Development

5 Delta robot specifications

6 Results

7 Conclusions

8 Future lines of research and references

Conclusions

1. **Kinematic and dynamic validation:** The results of the dynamics of methods A and B are almost identical by negligible decimals (10^{-18}). Comparing the results of both methods vs. ADAMS software it can be said that the curves are similar, both in value and shape. The existing small errors are mainly due to the fact that the configuration in ADAMS is similar, but not the same as in ROS, in that the mass simplification (effector mass division) and the joint - fixed base simplification exist only in ADAMS.
2. **Workspace:** The objective of determining the working space of the delta robot, identifying singularities and critical points to prevent damage to the mechanical parts and/or actuators is accomplished. Among the particularities of each constraint it is observed that the **spherical joint** θ_3 mainly modifies the shape of the workspace, while the **geometry** of the robot changes the size (limits). On the other hand, points that produce singularities when the determinant of the **Jacobian** is close to zero are discarded, avoiding infinite forces on the actuators and unreachable points.
3. **ROS Visualization (RViz):** The three-dimensional model on which the geometry of the delta robot has been implemented is created. The 3D visualization helps to check both the **trajectory algorithms**, which must generate linear geometric paths, and the **inverse kinematics** algorithms, which must make the arms follow the effector along the trajectory.

Conclusions

4 **Robot Operating System (ROS):** The ros environment was presented as a functional middleware for developing complex robotics software, where multiple data and tasks must be solved asynchronously. The tasks were segmented into three nodes: **3D visualization** of the mechanical parts, **calculation of the actuator dynamics** from linear trajectories of the mobile base and **workspace** of the delta robot. These nodes are possible to split into more nodes, further segmenting the tasks following the single functionality principle. This allows a better reuse of the developed codes, avoiding as they say reinventing the wheel.[ea]ch time a robotic program is developed.

5 **Code recycling:** All developments were carried out step by step and in such a way that it is possible to vary the values that characterize the robot: geometric dimensioning, masses and inertias and workspace restrictions. This makes it possible to determine the kinematics and dynamics of any other delta robot, without having to re-code the algorithms.

Content

1 Motivation

2 Hypothesis and Objectives

3 Theory

4 Development

5 Delta robot specifications

6 Results

7 Conclusions

8 Future lines of research and references

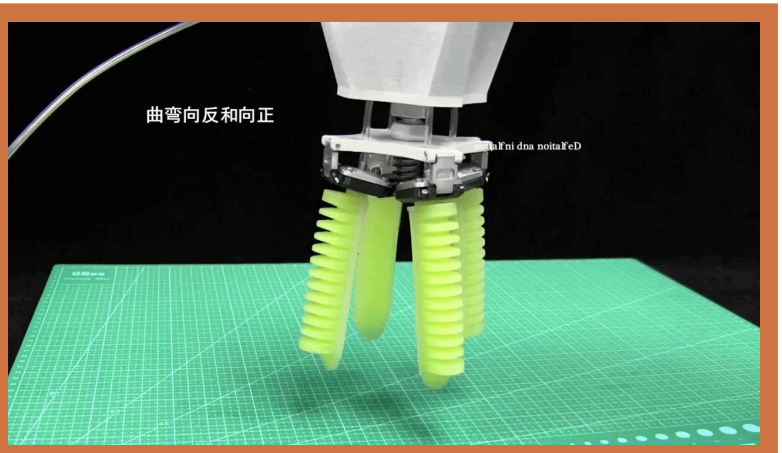
Future lines of research

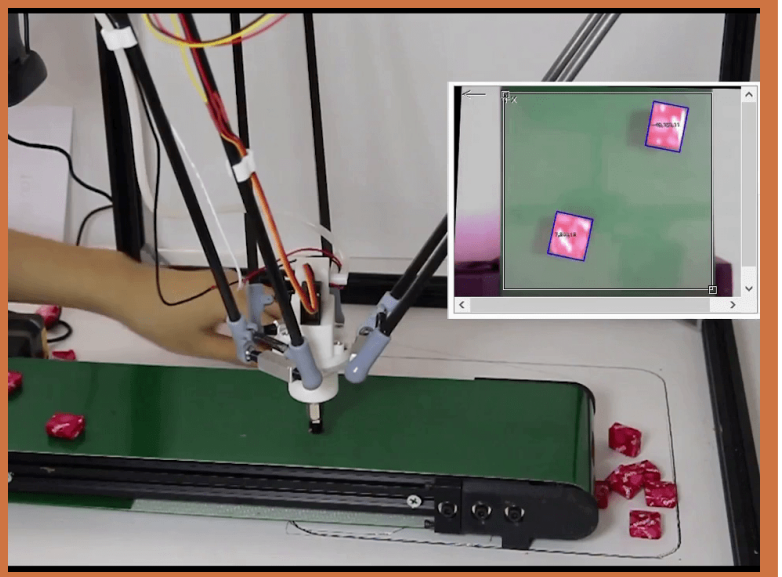
Future lines of research

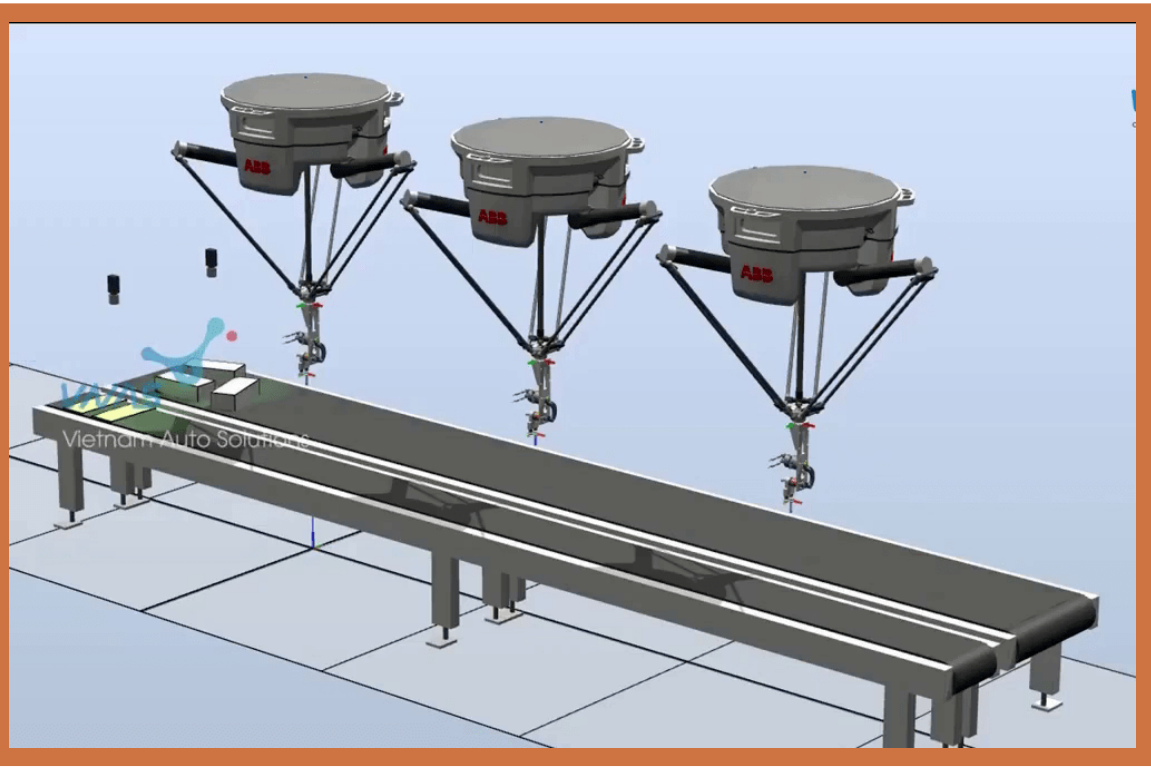
- **Dimensional optimization::** Based on lowest energy consumption
- **Trajectory optimization:** Minimum energy and/or minimum trayectory time
- **Pick and place trajectory:** Pythagoras-Hodograph quintic curves (PH)
- **Calibration with neural networks:** Visual positioning
- **Gripper:** Handling of fragile objects
- **Computer vision:** Detection of objects on conveyor belts
- **Motors:** Stepper motors and driver control
- **Torque curve:** Maximum and minimum torques vs. angular velocity of motors

Future lines of research

DEPARTAMENTO DE
INGENIERÍA
MECÁNICA
UNIVERSIDAD DE SANTIAGO DE CHILE







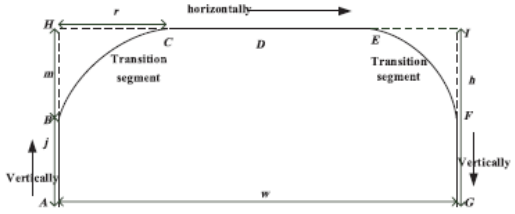


FIGURE 3. Pick-and-place operation path.

$$J = \int_{t_0}^{t_f} (\kappa + (1-\kappa)u^2)dt$$

References

References

- **Repository** GitHub:
  `https://github.com/IvanFernandezGracia/delta_robot_tesis`