



Validación de modelacion cinemática y dinámica de un robot delta 3 GDL a través de ROS y ADAMS

Autor: Iván Alejandro Fernández Gracia¹

Autor: Rodrigo Eduardo Soto Castro¹

Profesor Guia: Michael Gabriel Miranda Sandoval¹

¹Universidad de Santiago de Chile. Facultad de Ingeniería.

Departamento de Ingeniería Mecánica

12 de abril de 2021



- 1 Motivación
- 2 Hipótesis y Objetivos
- 3 Teoría
- 4 Desarrollo
- 5 Especificaciones robot delta
- 6 Resultados
- 7 Conclusiones

- 1 Motivación
- 2 Hipótesis y Objetivos
- 3 Teoría
- 4 Desarrollo
- 5 Especificaciones robot delta
- 6 Resultados
- 7 Conclusiones

Punto de vista educacional

- **Desconexión** entre la industria y la academia latente en nuestro departamento de ingeniería mecánica de la USACH.

Punto de vista teórico

- Modelar sistemas robóticos complejos produce problemas en la **sincronía** de los datos y **altos costos** en tiempo de **desarrollo** e **implementación**.
- Los esquemas de control de robots basados solo en la **cinemática de posición** son **insuficientes** para una excelente precisión.

Punto de vista industrial

- Uso en los últimos años de robots paralelos en operaciones de '**pick and place**' en diversas áreas en la industria, tales como la **agronomía y manufactura**. Estas áreas son una de las más desarrolladas en Chile según el PIB real.



1 Motivación

2 Hipótesis y Objetivos

3 Teoría

4 Desarrollo

5 Especificaciones robot delta

6 Resultados

7 Conclusiones



Hipótesis

Es posible **realizar** el **modelamiento cinemático y dinámico** de un robot delta a través de softwares libres y **validarlo** por medio de un software educativo.

Objetivos



Objetivo general

Diseñar algoritmos que controlen el movimiento de un robot delta y **validarlo** a través de software de simulación.

Objetivos específicos

- 1 **Crear** algoritmos que calculen trayectorias lineales en el espacio cartesiano con perfil de velocidad trapezoidal.
- 2 **Crear** algoritmos que resuelva la cinemática y dinámica del robot.
- 3 **Determinar** el espacio de trabajo del robot a partir de restricciones impuestas.
- 4 **Simular** el movimiento de las piezas mecánicas del robot a través de una herramienta de visualización.
- 5 **Calcular** la dinámica por medio de un software de análisis mecánico.
- 6 **Comparar** los resultados de la dinámica calculados por los algoritmos y por el software de análisis mecánico.

Contenido



1 Motivación

2 Hipótesis y Objetivos

3 Teoría

4 Desarrollo

5 Especificaciones robot delta

6 Resultados

7 Conclusiones

Teoría: Robot Operating System (ROS)



Historia

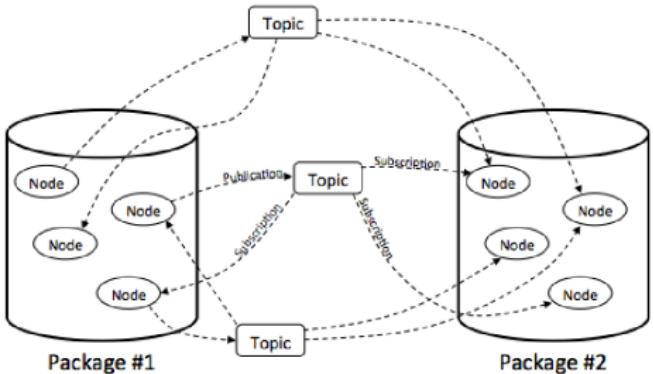
Los primeros pasos fueron a mediados de la década de 2000 en el laboratorio de robótica de la **universidad de Stanford**. En 2007, **Willow Garage Inc.**, una incubadora de robótica, proporcionó importantes recursos para extender estos conceptos mucho más y crear implementaciones bien probadas. Desde el 2013 hasta el presente, ROS es mantenido permanentemente por **Open Source Robotics Foundation (OSRF)** de Google y desde el 2017 cambio su nombre a **Open Robotics**.



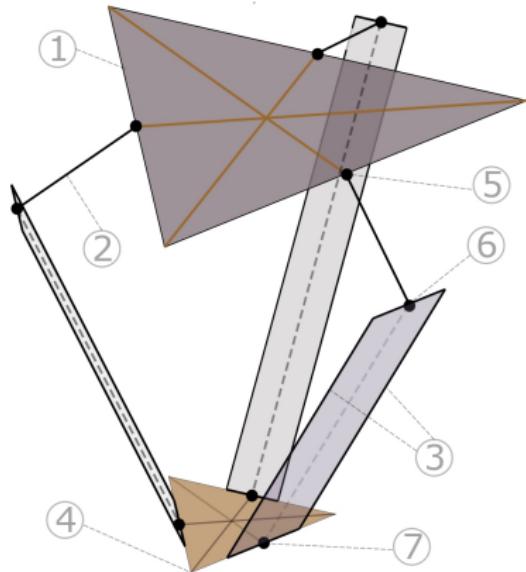
STANFORD
UNIVERSITY

Objetivos de ROS: Peer to peer

Robot Operating System (ROS) es un **middle-ware robótico**, es decir, una colección de frameworks para el desarrollo de software de robots. ROS consisten en **numerosos programas pequeños** de computadora que se conectan entre sí e **intercambian mensajes** continuamente.

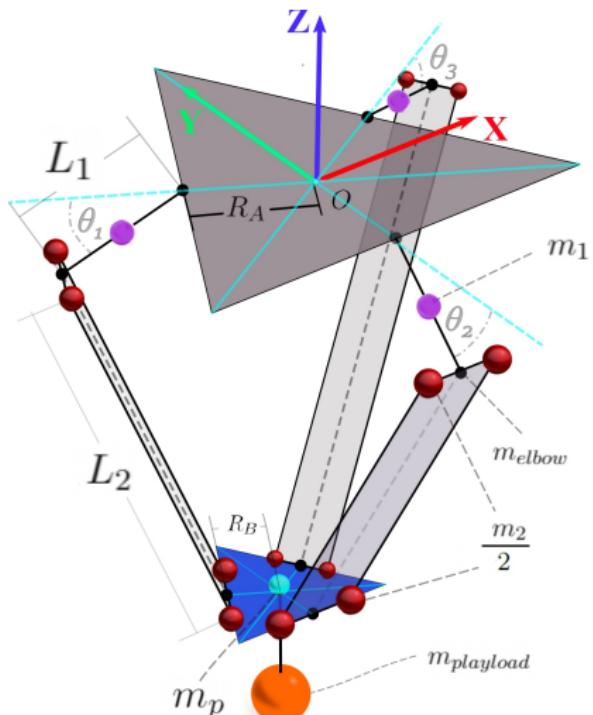
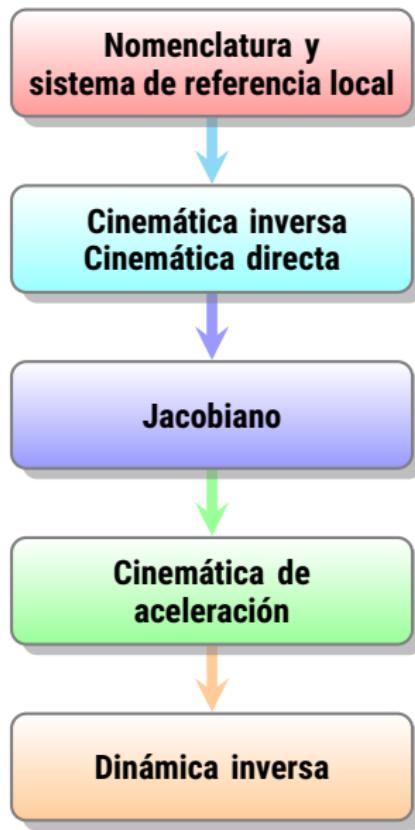


Teoría: Descripción robot delta



Nº	1	2	4	4	5	6	7
Piezas Mecánicas	Base fija	Brazo	Antebrazo	Base Móvil	Actuador o junta	Junta esférica	Junta esférica

Teoría: Pasos para determinar la dinámica



Sistema de referencia global XYZ y orden de ángulos de los brazos $\theta_{i \in \{1,2,3\}}$.

Teoría: Método A (Jacobiano)



Bucle cerrado

$$\overrightarrow{A_i B_i} + \overrightarrow{B_i C_i} = \overrightarrow{OP} + \overrightarrow{PC_i} - \overrightarrow{OA_i} ; \quad i \in \{1, 2, 3\}$$

Al derivar la ecuación del bucle cerrado y separar términos v_p y $\dot{\theta}$

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

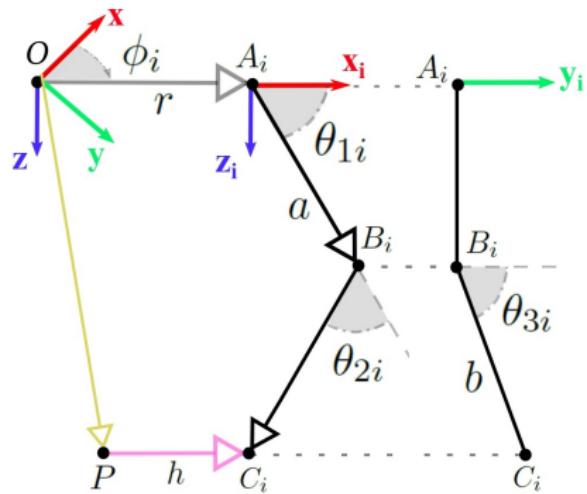
$$\begin{bmatrix} J_{1x} & J_{1y} & J_{1z} \\ J_{2x} & J_{2y} & J_{2z} \\ J_{3x} & J_{3y} & J_{3z} \end{bmatrix} \begin{bmatrix} v_{px} \\ v_{py} \\ v_{pz} \end{bmatrix} = \begin{bmatrix} J_{1\theta} & 0 & 0 \\ 0 & J_{2\theta} & 0 \\ 0 & 0 & J_{3\theta} \end{bmatrix} \begin{bmatrix} \dot{\theta}_{11}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

$$J_{1x} = \cos(\theta_{1i} + \theta_{2i}) \sin \theta_{3i} \cos \phi_i - \cos \theta_{3i} \sin \phi_i$$

$$J_{1y} = \cos(\theta_{1i} + \theta_{2i}) \sin \theta_{3i} \sin \phi_i + \cos \theta_{3i} \cos \phi_i$$

$$J_{1z} = \sin(\theta_{1i} + \theta_{2i}) \sin \theta_{3i}$$

$$J_{i\theta} = a \sin \theta_{2i} \sin \theta_{3i}$$



Jacobiano

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

- $J = J_x^{-1} J_\theta$ es el **jacobiano** del robot delta
- $v_p = [v_{px}, v_{py}, v_{pz}]^T$ es la **velocidad del punto** P de la base móvil
- $\dot{\theta} = [\dot{\theta}_{11}, \dot{\theta}_{12}, \dot{\theta}_{13}]^T$ es la **velocidad angular** de los actuadores

Teoría: Método A (Dinámica)

Teoría ecuaciones de Lagrange para sistemas con ligaduras holónomas en forma explícita

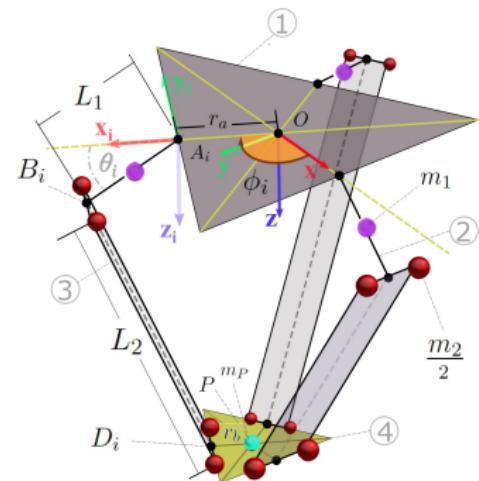
$$\frac{d}{dt} \left(\frac{\delta L}{\delta \dot{q}_j} \right) - \frac{\delta L}{\delta q_j} = \sum_{l=1}^{K^{(h)}} \lambda_l \frac{\delta f_l^{(h)}}{\delta q_j} + Q_j^{(NU)}(q_j)$$

Ec. de Lagrange con coordenadas generalizadas $q_j = [X_p, Y_p, Z_p]$

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

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

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



Dinámica inversa: Ecuaciones de Lagrange con las coordenadas generalizadas $q_j = [\theta_1, \theta_2, \theta_3]$

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



Teoría: Método B (Jacobiano)

Ecuaciones de restricción

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

Donde:

$$\vec{s}_i = \vec{P}_0 - (\vec{F}_i + \vec{\xi}_{F_i J_i}) = \begin{bmatrix} P_{0x} \\ P_{0y} \\ P_{0z} \end{bmatrix} - R_i^R \left(\begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} L_A \cos(\theta_i) \\ 0 \\ -L_A \sin(\theta_i) \end{bmatrix} \right)$$

Al derivar la ecuación de restricción y separando los términos \vec{P}_0 y $\vec{\theta}$

$$\vec{P}_0 = -J_1 J_2 \vec{\theta}$$

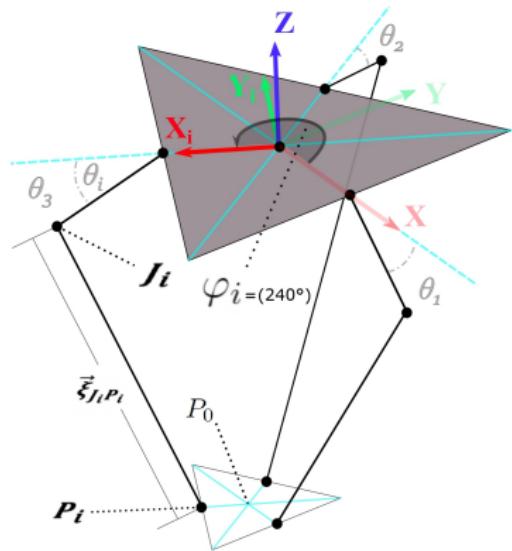
$$\begin{bmatrix} \dot{P}_{0x} \\ \dot{P}_{0y} \\ \dot{P}_{0z} \end{bmatrix} = - \begin{bmatrix} \vec{s}_1^T \\ \vec{s}_2^T \\ \vec{s}_3^T \end{bmatrix}^{-1} \begin{bmatrix} \vec{s}_1^T \cdot \vec{b}_1 & 0 & 0 \\ 0 & \vec{s}_2^T \cdot \vec{b}_2 & 0 \\ 0 & 0 & \vec{s}_3^T \cdot \vec{b}_3 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

Donde: $\vec{b}_i = R_i^R [L_A \sin(\theta_i), 0, L_A \cos(\theta_i)]$

Jacobiano

$$\vec{P}_0 = J \vec{\theta}$$

- $J = -J_1 J_2$ es el **jacobiano** del robot delta
- $\vec{P}_0 = [\dot{P}_{0x}, \dot{P}_{0y}, \dot{P}_{0z}]^T$ es la **velocidad del punto** P_0 de la base móvil y $\vec{\theta} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]^T$ es la **velocidad angular** de los actuadores



Teoría: Método B (Dinámica)

Trabajo virtual: Principio de D'Alembert

Caso estático: El trabajo virtual δW producido por una **fuerza externa** F_i que actúa sobre un cuerpo produciendo un **desplazamiento lineal virtual** δr_i :

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

Caso dinámico: La ecuación anterior se puede extender agregando **fuerzas iniciales** y rotación. Agregando el trabajo virtual δW producido por un **torque externo** τ que actúa sobre el mismo cuerpo produciendo un **desplazamiento angular virtual** $\delta\theta$:

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

Dinámica inversa: torque de actuadores

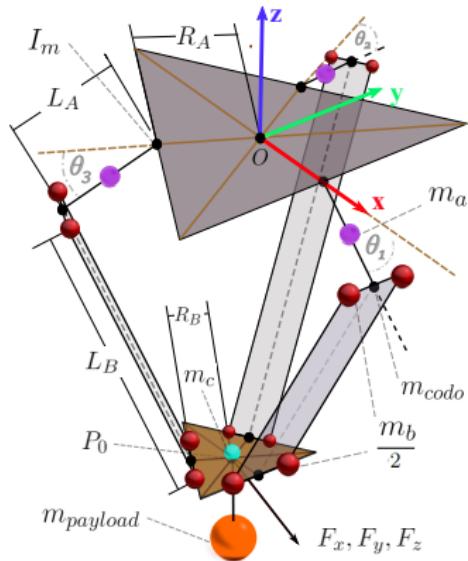
$$\vec{\tau} = I_b \vec{\dot{\theta}} + J^T m_{nt} \vec{P}_0 - J^T \vec{F}_g - \vec{\tau}_{Gb}$$

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

$$\text{Matriz de masa: } M(\theta) = I_b + J^T m_{nt} J$$

$$\text{Coeficiente de Coriolis: } C(\theta, \dot{\theta}) = J^T m_{nt} J$$

$$\text{Términos de gravedad: } \vec{G}(\theta) = -J^T \vec{F}_g - \vec{\tau}_{Gb}$$



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

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

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

$$\vec{F}_g = m_{nt} [0 \ 0 \ -g]^T$$



¿Por qué ADAMS?

Software a la vanguardia en simulación de robots para modelación cinemática y dinámica.

Simulación dinámica y análisis multi-cuerpo (MBD)

Desplazamiento = Deformación + Desplazamiento de cuerpo rígido

(A) FEM

(B) MBD



Coordenadas generalizadas

ADAMS trabaja con un sistema cartesiano de coordenadas, para cada parte define como coordenadas generalizadas

Posición + Orientación

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

Ecuaciones de Movimiento

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

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

Teoría: Trayectoria



Trayectorias en línea recta

Una trayectoria en **línea recta** en el **espacio cartesiano** comienza con una configuración X_{inicio} y hasta una configuración final X_{fin} . La escala temporal de la trayectoria lineal es $s(t)$.

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

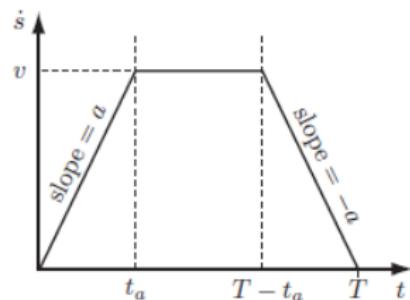
$$X : [0; 1] \rightarrow \Re^3$$

$$X(s) = X_{\text{inicio}} + s(X_{\text{fin}} - X_{\text{inicio}})$$

Escala de tiempo: Perfil de velocidad trapezoidal (LSPB)

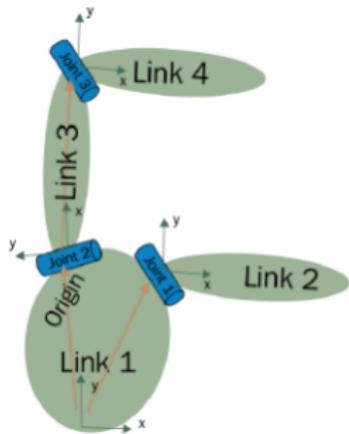
Constan de 3 fases:

- **Aceleración constante** $\ddot{s} = a$ de tiempo t_a
- **Velocidad constante** $\dot{s} = v$ de tiempo $t = T - 2t_a$
- **Desaceleración constante** $\ddot{s} = -a$ de tiempo t_a



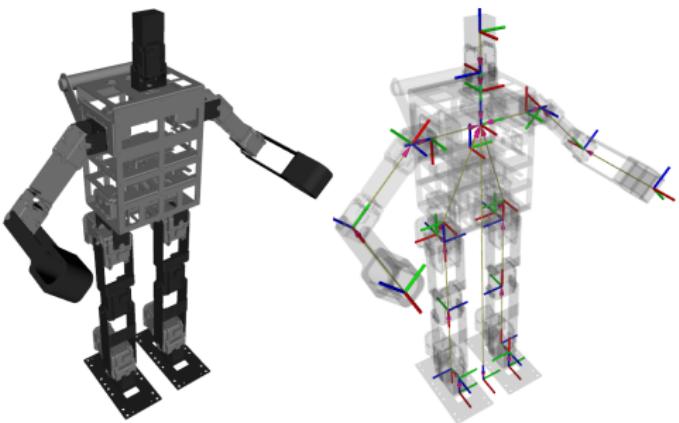
Unified Robot Description Format (URDF)

En ROS es posible visualizar un **modelo 3D** de un robot mediante el uso de archivos URDF. La descripción del modelo consiste básicamente en unir dos conjuntos: el conjunto de **enlaces (link)** y el conjunto de **uniones (joint)**.



Transform Frames (tf)

Tf está diseñada para proporcionar una forma estándar de realizar un seguimiento de los **marcos de coordenadas** y transformar los datos dentro de todo el sistema a lo largo del tiempo. El paquete tf puede **rastrear y mantener la relación** entre múltiples marcos de coordenadas.

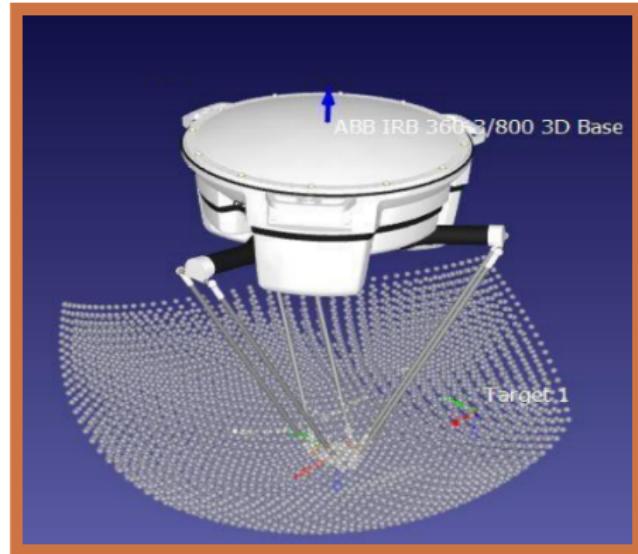


Teoría: Espacio de trabajo



Definición

El **método de discretización**, que se basa en métodos numéricos, consiste en discretizar el espacio en tres dimensiones, resolviendo la **cinemática inversa** para cada punto y verificando las **restricciones** que limitan dicho espacio de trabajo

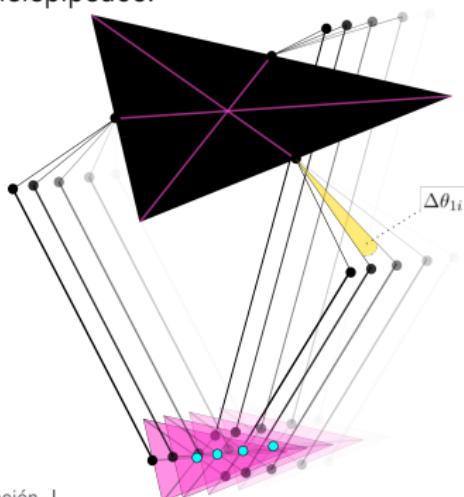
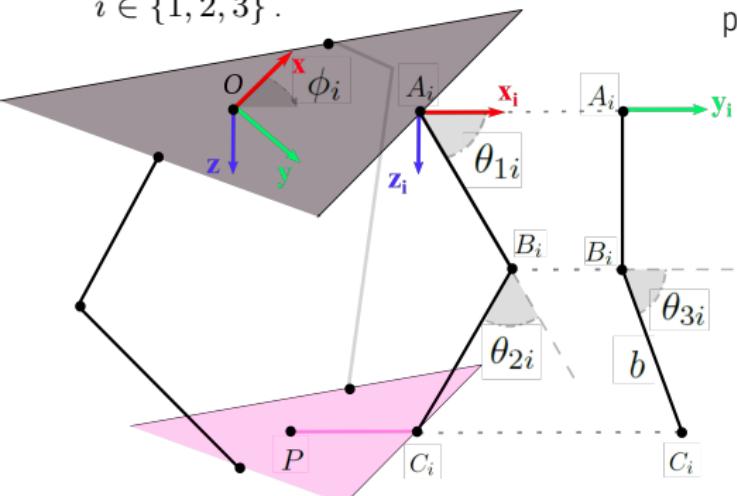


Espacio de trabajo robot
ABB irb 360-3/800

Teoría: Restricciones espacio de trabajo



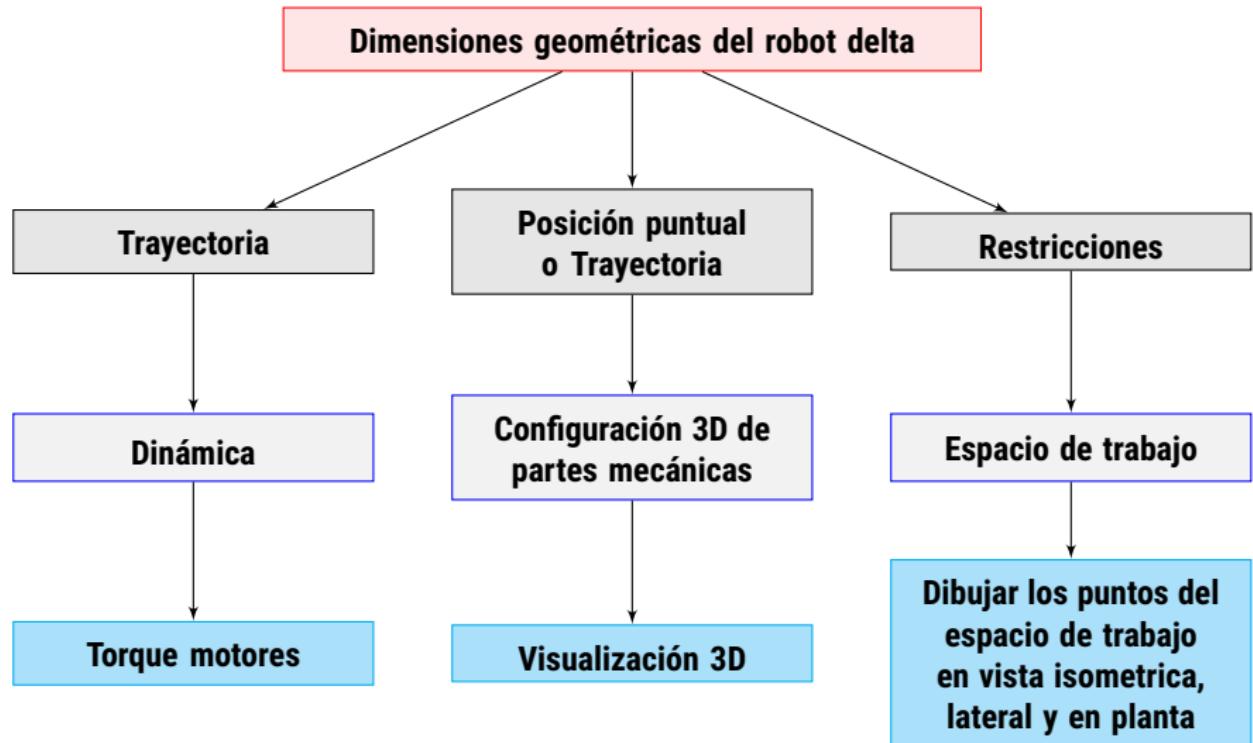
- **Límites** impuesto en los **ángulos de los actuadores** $[\theta_{1i,min} - \theta_{1i,max}]$ para cada actuador $i \in \{1, 2, 3\}$.
- Resolución basada en tamaño del paso de los actuadores, es decir, la **discretización** $\Delta\theta_{1i}$ del rango impuesto por los límites del punto anterior para cada actuador $i \in \{1, 2, 3\}$.
- Restricciones de **ángulos internos** θ_{2i} y θ_{3i} en base a restricción de las juntas .
- Singularidades que se determinan mediante el **determinante del jacobiano** $J = J_x^{-1}J_\theta$ cuando este es cercano a 0.
- **Límites** X, Y y Z . Generalmente son volúmenes geométricos como cilindros o paralelepípedos.



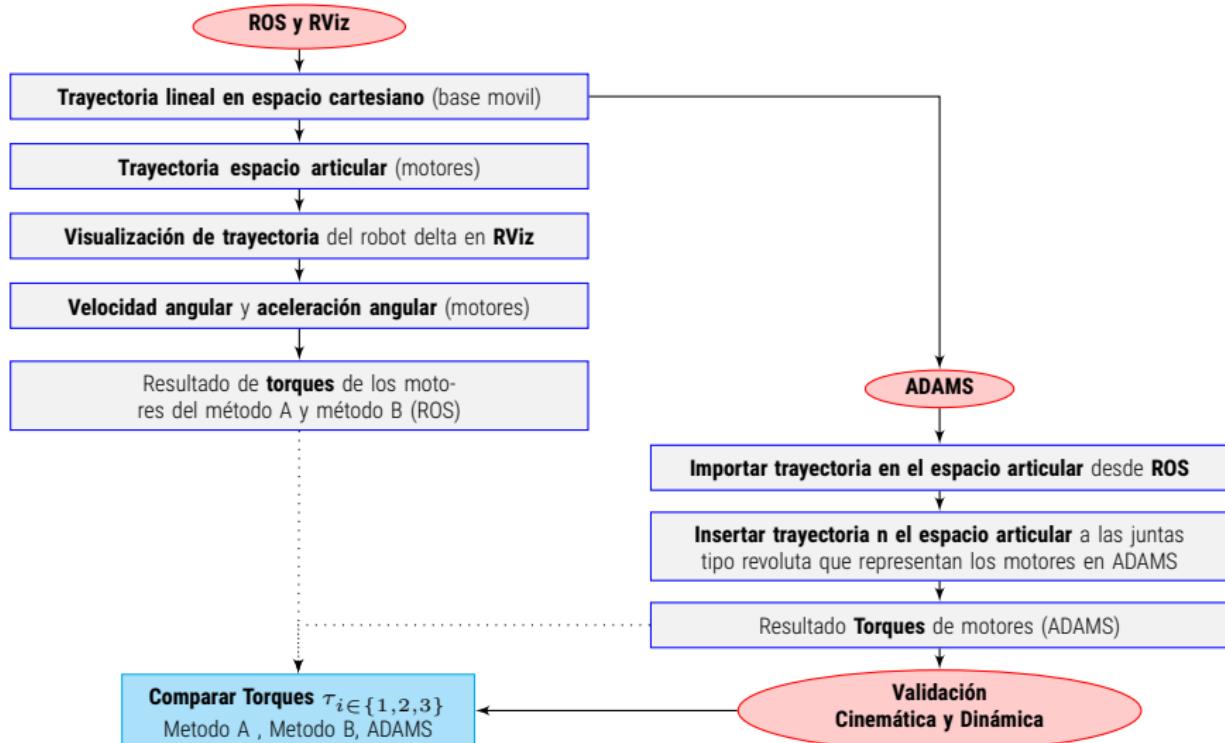


- 1 Motivación
- 2 Hipótesis y Objetivos
- 3 Teoría
- 4 Desarrollo
- 5 Especificaciones robot delta
- 6 Resultados
- 7 Conclusiones

Desarrollo: Resumen de tareas a desarrollar



Desarrollo: Flujo de trabajo validación dinámica



Desarrollo: Trayectorias simuladas



Las trayectorias que se crean para comprobar los algoritmos de cinemática y dinámica son:

Trayectoria	$P_i(x, y, z)[mm]$	$P_f(x, y, z)[mm]$	$v_{max}[\frac{mm}{s}]$	$a_{max}[\frac{mm}{s^2}]$
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

Desarrollo: Trayectoria

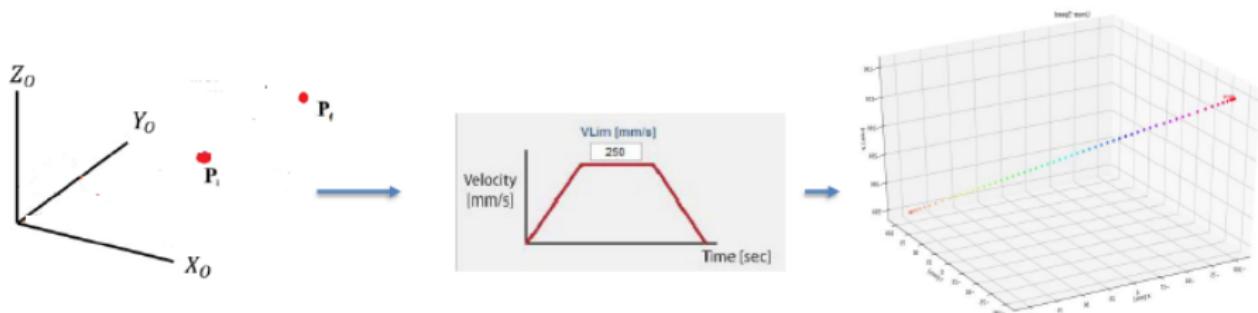


$$\begin{aligned}P_i &= [X_i Y_i Z_i] \\P_f &= [X_f Y_f Z_f] \\v_{max} & \\a_{max} &\end{aligned}$$

Trayectoria

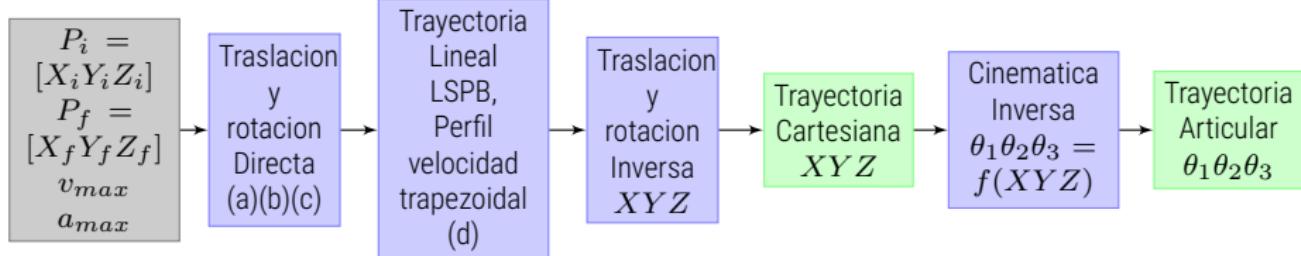
Puntos de la trayectoria en el
espacio cartesiano XYZ y
espacio articular $\theta_1 \theta_2 \theta_3$
con **escala temporal** $s(t)$

Entradas y salidas de la función de trayectoria

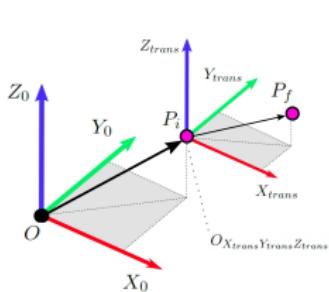


Ejemplo gráfico del diagrama de flujo para función

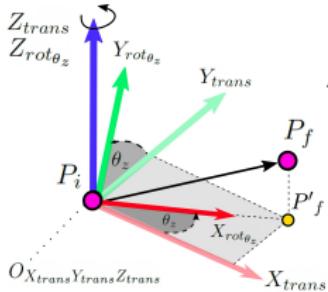
Desarrollo: Trayectoria



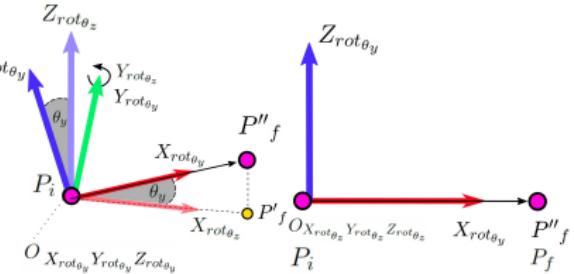
Subfunciones de función trayectoria



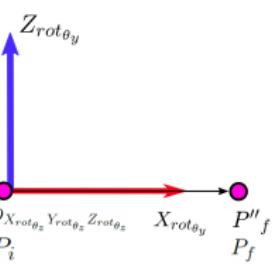
(a) Traslación



(b) Rotación θ_z

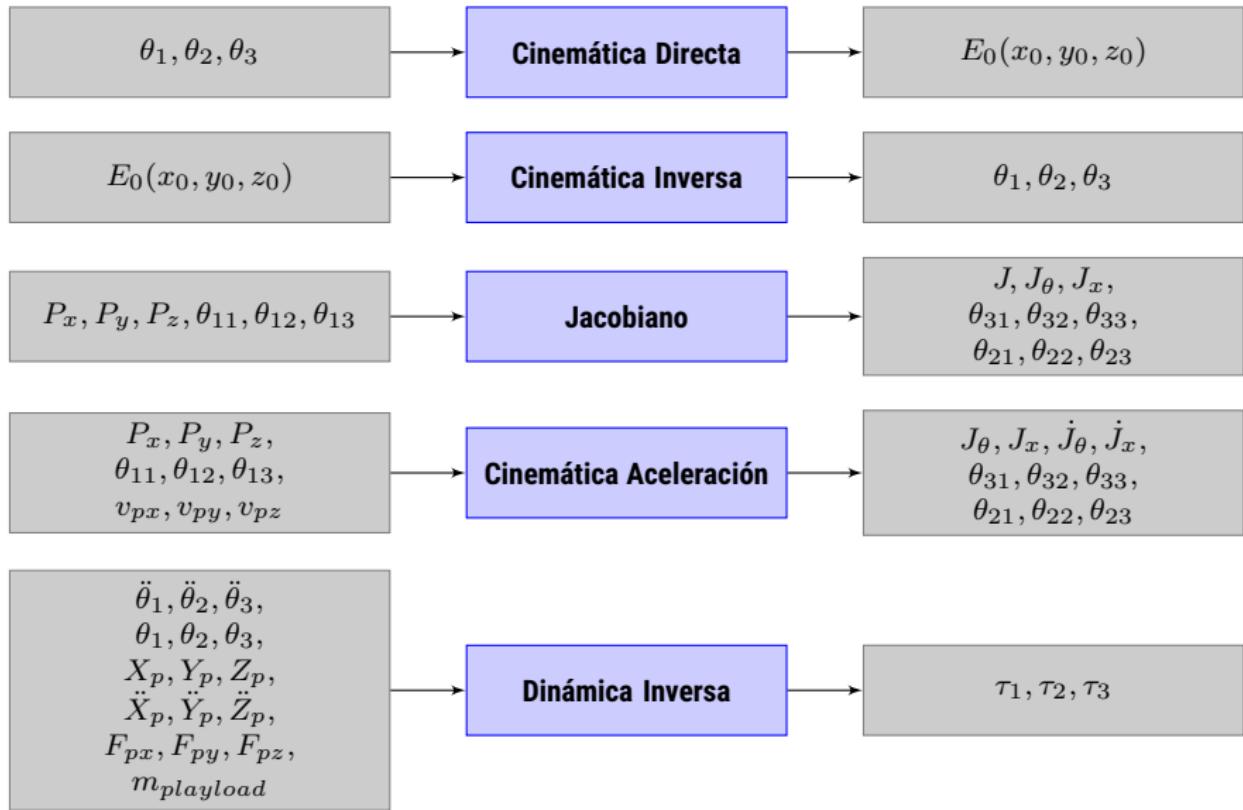


(c) Rotación θ_y

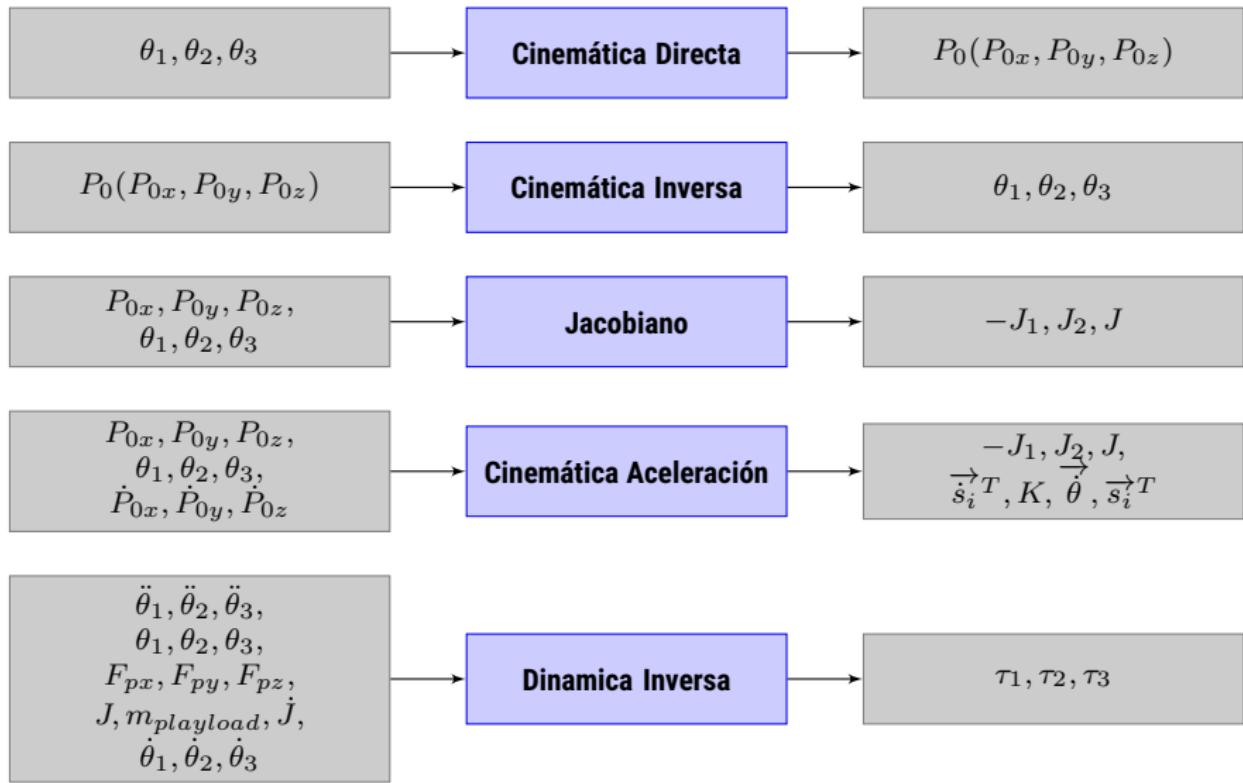


(d) Se aplica LSPB

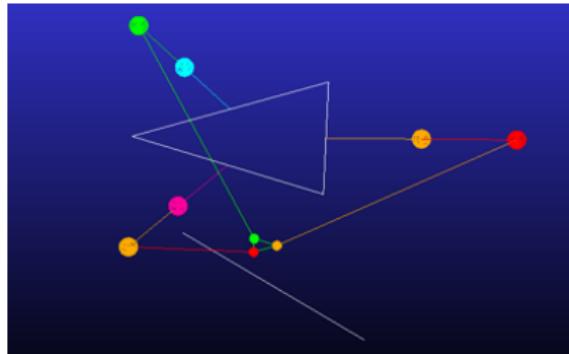
Desarrollo: Método A



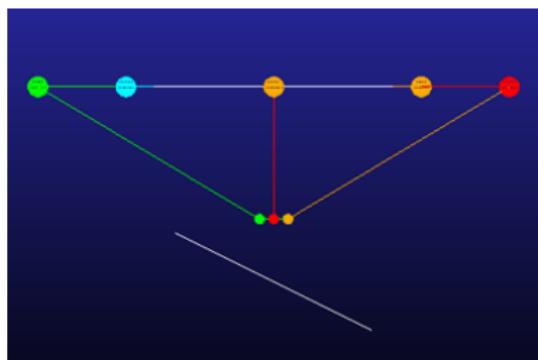
Desarrollo: Método B



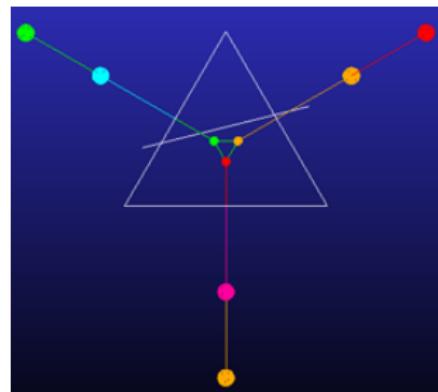
Desarrollo: ADAMS (Modelo)



(a) Vista Isométrica



(b) Vista Frontal



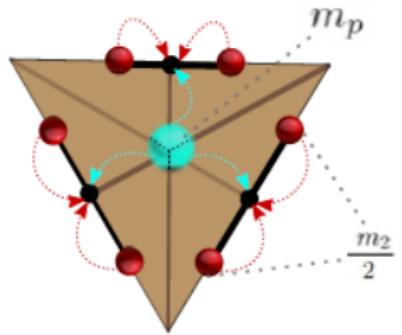
(c) Vista Superior

Desarrollo: ADAMS (Simplificaciones)



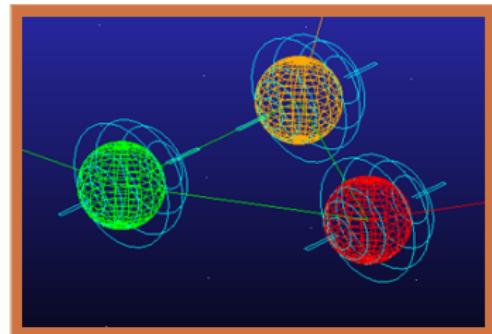
Simplificación de Masas

La masa del efecto (punto verde) se divide en tres porciones y se le suma a las tres masas de los antebrazos del robot (puntos negros).



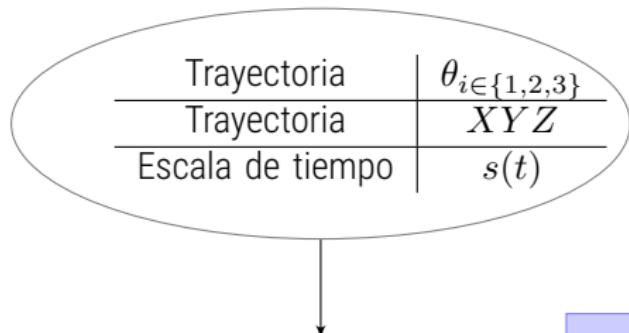
Simplificación de junta - base fija

Dificultad para unir las tres cadenas cinemáticas abiertas de los brazos del robot en cadenas cerradas.



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

Desarrollo: Visualización en RViz



Calculo de valores de juntas (joint)

Ángulo	θ_{2i}
Ángulo	θ_{3i}
Nombres de juntas URDF	string[]

TF (Transform Frames)

Calcula matrices de transformación de frames de joint según su valor actual y el modelo URDF del robot delta

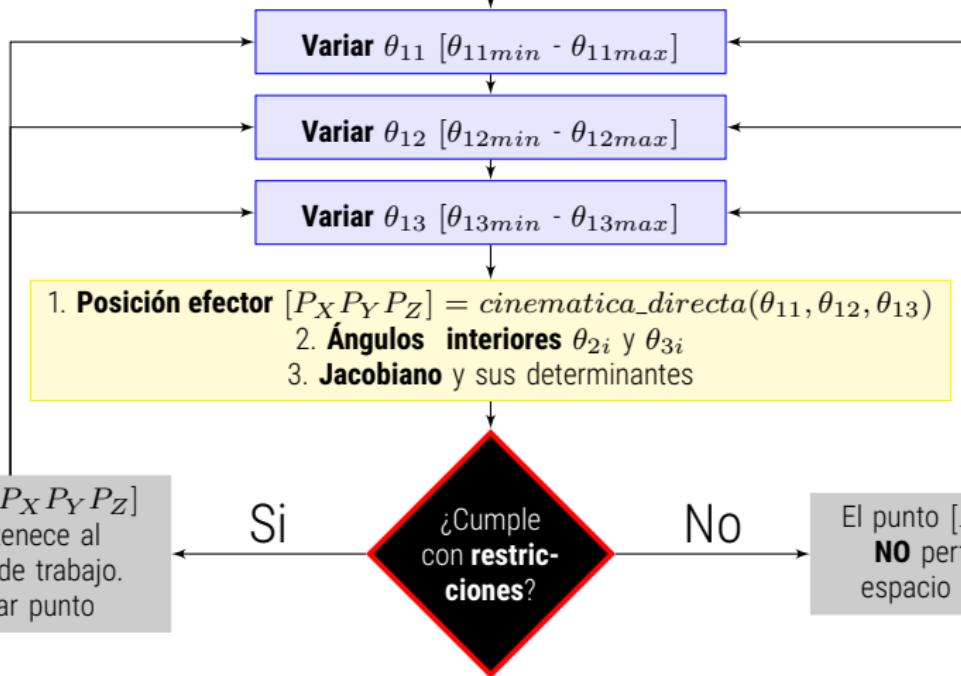
Vizualizacion del modelo URDF del robot delta + TF en RViz

Nodos para obtener la visualización del robot delta

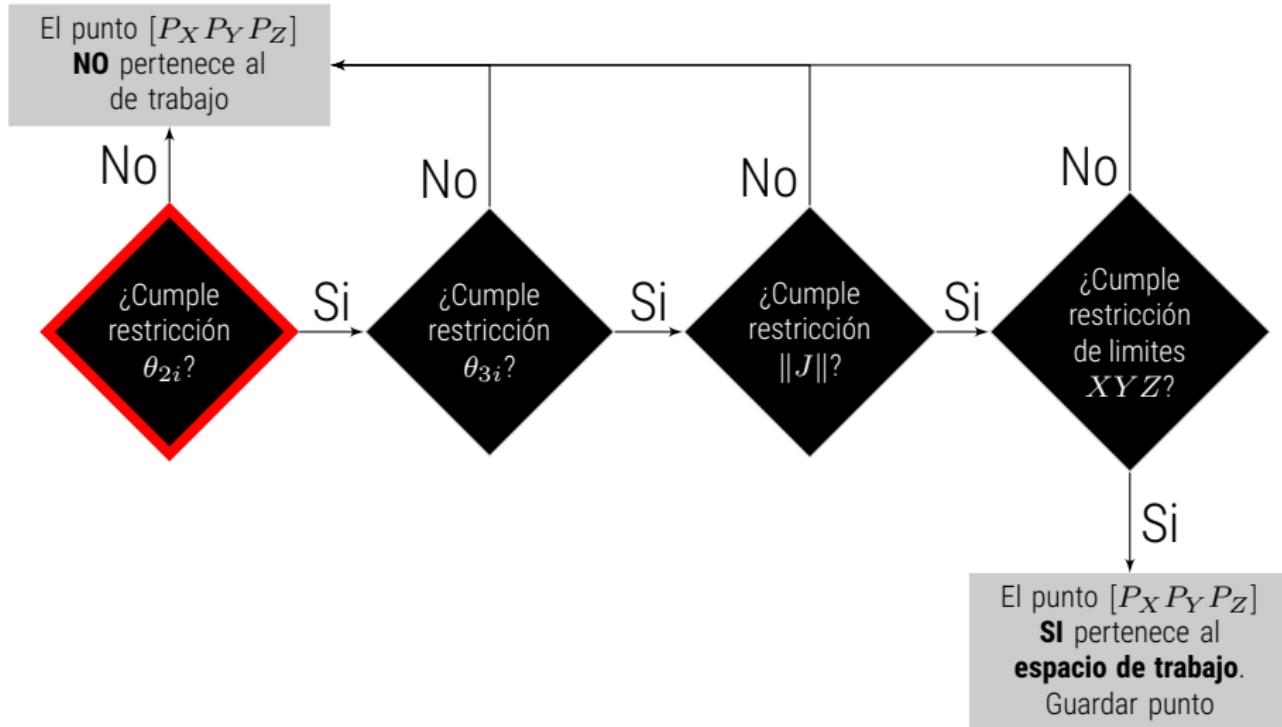
Desarrollo: Espacio de trabajo



Dimensiones del robot y parámetros de restricciones



Desarrollo: Espacio de trabajo



Desarrollo: Espacio de trabajo



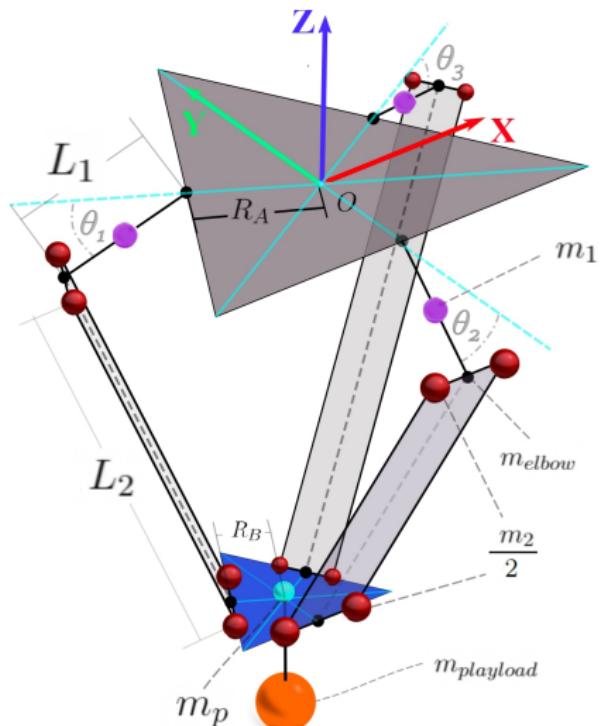
Restricción	Explicación	Min	Max
θ_{1i}	Ángulo brazo	-90°	90°
$\Delta\theta_{1i}$	Discretización de rangos θ_{1i}	5°	
θ_{2i}	Ángulo interior	5°	175°
θ_{3i}	Ángulo interior	45°	135°
J_x	Jacobiano base móvil	$6 * 10^{-1}$	
J_θ	Jacobiano actuadores	$4 * 10^{-3}$	
X	Límite X de espacio de trabajo	-400[mm]	+400[mm]
Y	Límite Y de espacio de trabajo	-400[mm]	+400[mm]
Z	Límite Z de espacio de trabajo	-300[mm]	-750[mm]

- 1 Motivación
- 2 Hipótesis y Objetivos
- 3 Teoría
- 4 Desarrollo
- 5 Especificaciones robot delta
- 6 Resultados
- 7 Conclusiones

Especificaciones robot delta

Parámetro	Valor
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]
$m_{payload}$	0 [kg]
m_{elbow}	0 [kg]
I_m	0 [$\text{kg} \cdot \text{m}^2$]
r_{mass}	0.5
g	9.81 [m/s ²]

Parámetros robot delta



Representación gráfica robot delta



- 1 Motivación
- 2 Hipótesis y Objetivos
- 3 Teoría
- 4 Desarrollo
- 5 Especificaciones robot delta
- 6 Resultados
- 7 Conclusiones

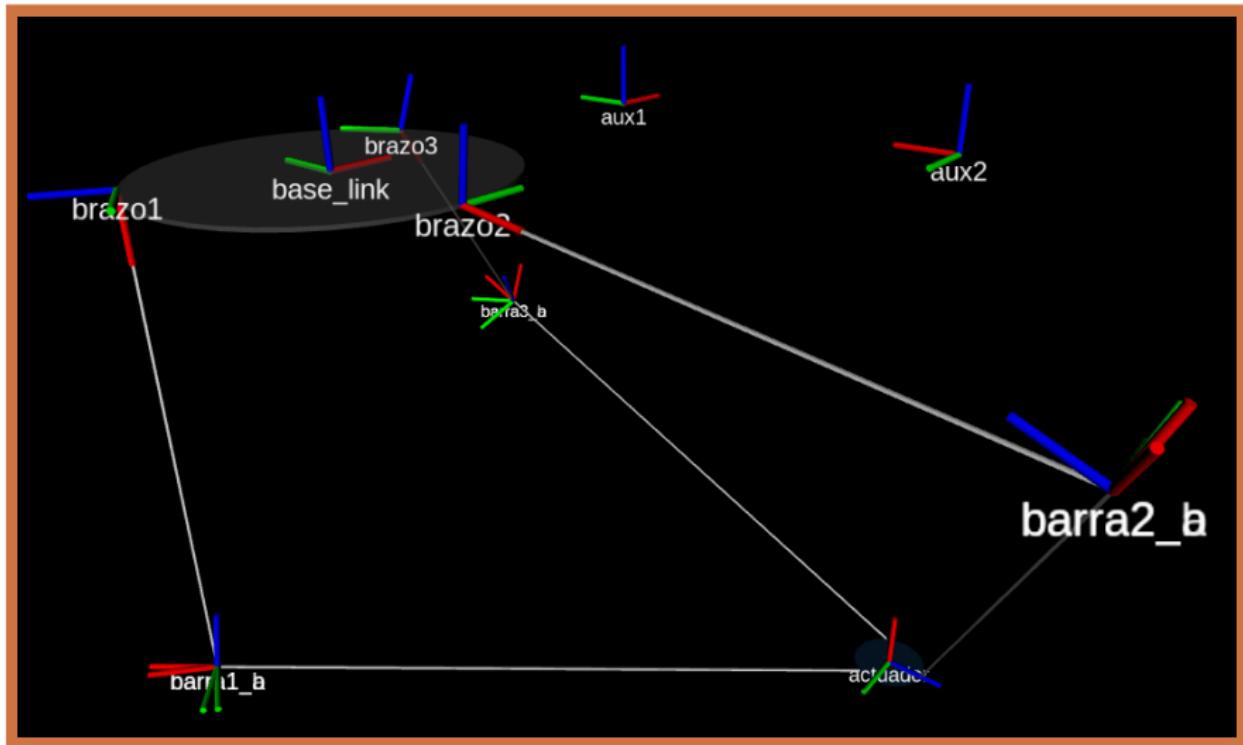
Resultados: Visualización en RViz



DEPARTAMENTO DE
INGENIERÍA
MECÁNICA
UNIVERSIDAD DE SANTIAGO DE CHILE

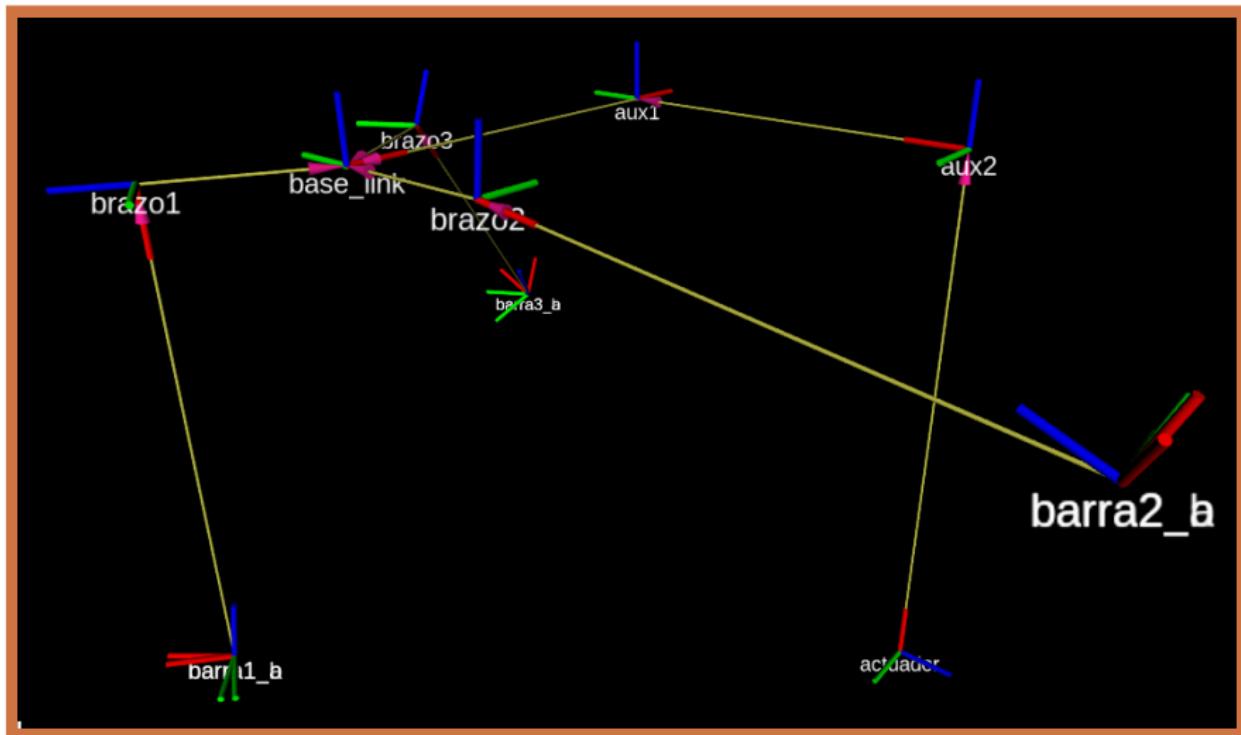
Simulación de **trayectoria 3** en **RViz**

Resultados: Visualización RViz



Visualización 3D de links y **frames** de joints del robot delta en **RViz**

Resultados: Visualización RViz

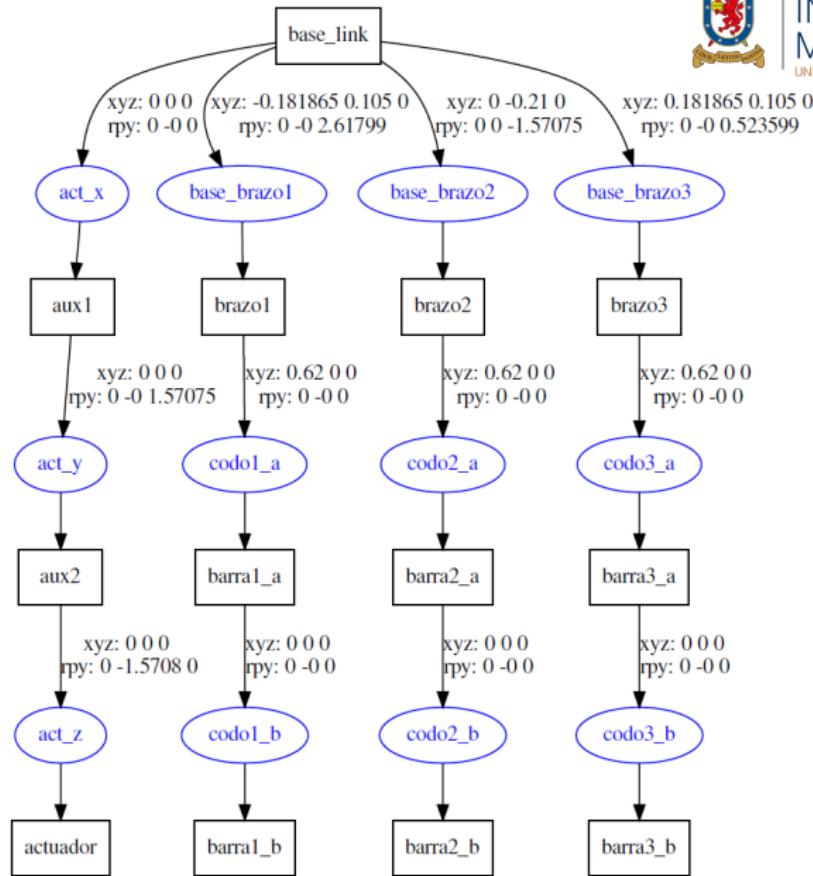


Conexión entre los links y **frames** joints del robot delta en **RViz**

Resultados:



Representación gráfica de la relación padre-hijo del modelo URDF del robot delta en RViz



Resultados: Visualización ADAMS

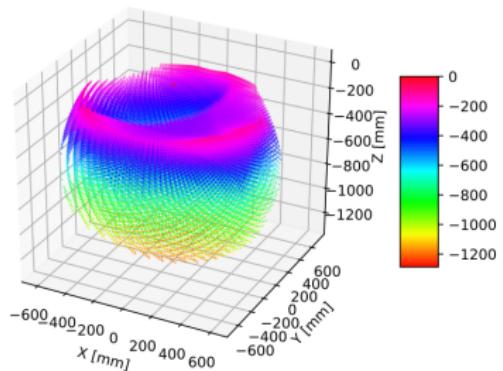


DEPARTAMENTO DE
INGENIERÍA
MECÁNICA
UNIVERSIDAD DE SANTIAGO DE CHILE

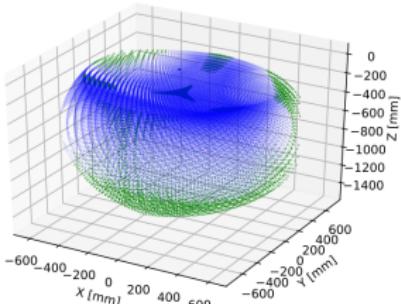


Simulación de **trayectoria 1** en **ADAMS**

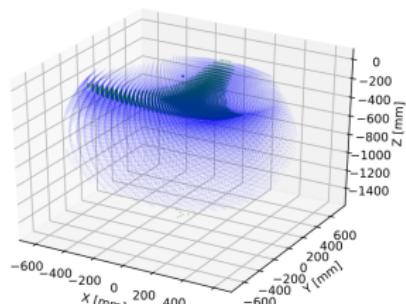
Resultados: Espacio de trabajo



(a) Puntos alcanzables con restricciones angulares (**Escala de colores** representa altura en eje z)

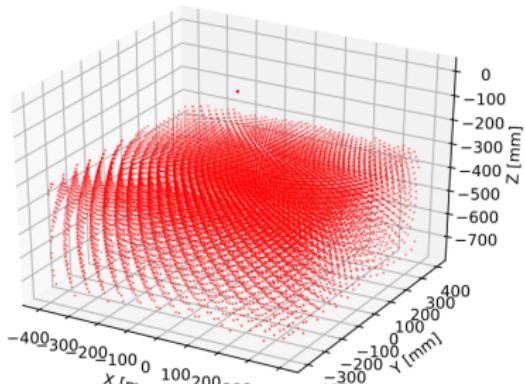


(b) Puntos con restricción $J_x \approx 0$ (verde)
y restricciones angulares (azul)

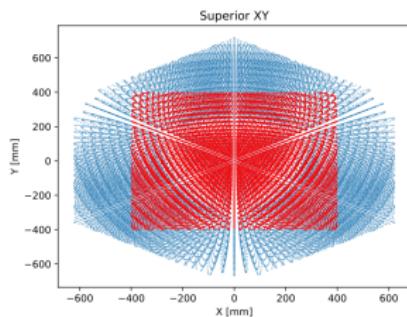


(c) Puntos con restricción $J_\theta \approx 0$ (verde)
y restricciones angulares (azul)

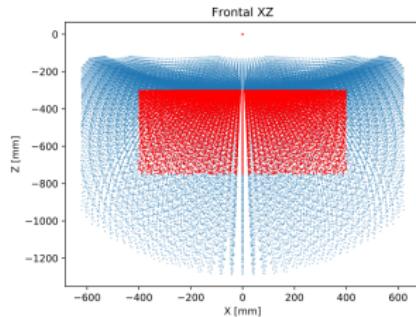
Resultados: Espacio de trabajo



(a) Espacio de trabajo



(b) Vista plano $X\,Y$ de restricciones angulares y espacio de trabajo



(c) Vista plano $X\,Z$ de restricciones angulares y espacio de trabajo



- 1 Motivación
- 2 Hipótesis y Objetivos
- 3 Teoría
- 4 Desarrollo
- 5 Especificaciones robot delta
- 6 Resultados
- 7 Conclusiones



Conclusiones

- 1 Validación cinemática y dinámica:** Los resultados de la dinámica de los métodos A y B son casi idénticos por decimales insignificantes (10^{-18}). Al comparar los resultados de ambos métodos vs el software ADAMS se puede decir que las curvas son similares, tanto en su valor como en su forma. Los pequeños errores existentes se deben principalmente a que la configuración en ADAMS es similar, pero no igual a la de ROS, en cuanto que la simplificación de masas (división de masa del efecto) y la de junta - base fija, existen solo en ADAMS.
- 2 Espacio de trabajo:** Se cumple el objetivo de determinar el espacio de trabajo del robot delta, identificando singularidades y puntos críticos para prevenir daños en las partes mecánicas y/o en los actuadores. Entre las particularidades de cada restricción se observa que la **rótula** θ_3 modifica principalmente la forma del espacio de trabajo, mientras que la **geometría** del robot cambia el tamaño (límites). Por otro lado se descartan los puntos que producen singularidades cuando el determinante del **jacobiano** es cercano a cero, evitando fuerzas infinitas en los actuadores y puntos no alcanzables.
- 3 ROS Visualization (RViz):** Se consigue crear el modelo tridimensional sobre el cual se ha implementado la geometría del robot delta. La visualización 3d ayuda a comprobar tanto los **algoritmos de trayectoria**, que deben generar caminos geométricos lineales, como los algoritmos de **cinemática inversa**, que deben lograr que los brazos sigan al efecto en dicha trayectoria.

- 4 **Robot Operating System (ROS):** Se presentó el ambiente de ros como un middleware funcional para desarrollar software de robótica compleja, donde múltiples datos y tareas deban ser resueltas asincrónicamente. Las tareas se segmentaron en tres nodos: **visualización 3D** de las partes mecánicas, **cálculo de la dinámica** de los actuadores a partir de trayectorias lineales de la base móvil y **espacio de trabajo** del robot delta. Estos nodos son posibles de separar en más nodos, segmentando aún más las tareas siguiendo el principio de funcionalidad única. Esto permite reutilizar de mejor forma los códigos desarrollados, evitando como se dice "reinventar la rueda" cada vez que se desarrolla un programa robótico.
- 5 **Reutilización de código:** Todos los desarrollos se realizaron paso a paso y de tal forma que sea posible variar los valores que caracterizan el robot: dimensionamiento geométrico, masas e inercias y restricciones del espacio de trabajo. Esto permite determinar la cinemática y dinámica de cualquier otro robot delta, sin tener que realizar nuevamente los códigos de los algoritmos.

Futuras líneas de investigación y referencias



Futuras líneas de investigación

- **Optimización dimensional:** Basado en el menor consumo de energía
- **Optimización de trayectoria:** Mínima energía y/o tiempo mínimo de trayectoria
- **Trayectoria pick-and-place:** Curvas quínticas de Pitágoras-Hodógrafa (PH)
- **Calibración con redes neuronales:** Posicionamiento visual
- **Pinzas:** Manipulación de objetos frágiles
- **Visión por computadora:** Detección de objetos en cinta transportadora
- **Actuadores:** Motores paso a paso y control de drivers
- **Curva de torque:** Torques máximos y mínimos vs la velocidad angular de motores

Referencias

- **Repositorio** en GitHub del proyecto de tesis:
https://github.com/IvanFernandezGracia/delta_robot_tesis
https://github.com/asimov-dev/delta_robot_tesis