Universidad de Ingeniería Y Tecnología

Departamento de Ingeniería Mecatrónica

Fundamentos de Robótica

Proyecto final

Integrantes

Barraza Ramírez, Juan Luis

Flores Chagua, Del Piero

Palma Rodríguez, Diego Alonso V.

Solórzano Pinzás, Renzo Lucciano

Profesor: Ph.D Oscar E. Ramos

Lima - Perú

2020-2

1. Introducción

Los robots manipuladores seriales son herramientas bastante útiles para realizar tareas como la manipulación de objetos. Este tipo de robots son más conocidos en el ámbito industrial debido a su eficiencia para realizar tareas repetitivas. Sin embargo, estos pueden ser utilizadas para otras aplicaciones como son los vehículos de exploración espacial o *rovers* [1]. La implementación de un brazo robótico serial permite brindar mayor autonomía al robot puesto que le permite realizar acciones como la recolección de objetos y facilitar la exploración en otros medios. Por esta razón, el presente proyecto pretende diseñar, visualizar y simular un robot de 6 grados de libertad de tal manera que sea capaz de ser adaptado a un robot de exploración. Es decir, este robot debe ser ligero y sencillo de implementar a diferencia de otros robots dirigidos a la industria. Asimismo, en este tipo de robots se busca realizar el control tanto en el espacio articular como operacional para poder formar trayectorias del efector final para ciertas tareas del robot.

1.1. Brazo robótico

El brazo robótico de estudio es de tipo 6R, es decir, posee 6 articulaciones de revolución y por ende, cuenta con 6 grados de libertad. En la Figura 1 se muestra dónde están ubicadas las articulaciones de dicho brazo. Cabe mencionar que la imagen del robot fue tomada en base al robot manipulador IRB6600 de ABB, pues este también es un robot 6R. Es decir, el plano fue tomado de las especificaciones de dicho robot, con fines estéticos de este reporte. Sin embargo, las medidas del robot son diferentes y serán especificadas en la sección de cinemática directa, para el cálculo de los parámetros de Denavit-Hartenberg. Cabe mencionar que para visualizar el diseño 3D del robot, se tomó como base el robot *mirobot* de la empresa Wlkata. Asimismo, el repositorio del proyecto puede encontrarse en [2]

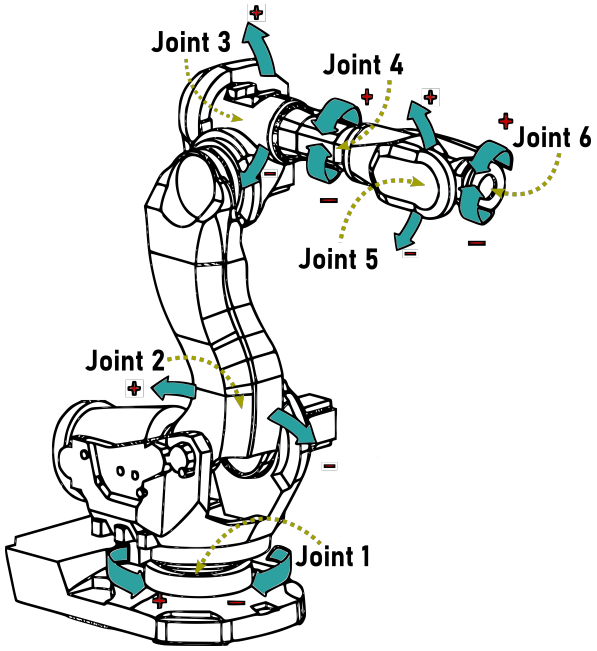


Figura 1: Ubicación de las articulaciones del robot

1.2. Componentes del Robot

Esta sección se centrará en mencionar todos los componentes necesarios para la implementación de un prototipo del robot especificado. En la Tabla 1 se observa los componentes divididos en tipos. En estos se encuentra la estructura mecánica, la cual se describiría como la parte superficial del robot que podría ser realizada con impresión 3D; los actuadores, en donde se puede elegir entre usar motores paso a paso o servomotores para las articulaciones del robot; sensores, los cuales leen información del estado interno del robot (propioceptivos) y del entorno del mismo (exteroceptivos); y también se define los dispositivos usados para el control en bajo y alto nivel. Todos estos componentes serían útiles al momento de realizar los métodos de control y procesos de cinemática directa e inversa que se describirán posteriormente en este informe.

Tipo de componente		**Componente del robot**
Estructura mecánica		- Eslabones (hecho con plástico ABS) - Articulaciones de revolución - Base (hecho con plástico ABS) - Efector final (garra mecánica)
Fuente de poder		Baterías lipo o Fuentes de 24 v
Potencia		Puentes H (pololu) y controlador de motores Sabertooth
Actuadores		Motores paso a paso o Servomotores
Transmisión		Harmonic Drives
Sensores	Propioceptivos	- Enconders absolutos - Giroscopio - IMU
	Exteroceptivos	- Sensor de Fuerza/Torque - Cámara
Sistemas de control	Bajo nivel	Microcontrolador (Arduino)
	Alto nivel	Single Board Computer (Raspberry Pi)

Tabla 1: Componentes del brazo robot

2. Modelo del robot

En esta sección se presentará el modelo del robot en el formato URDF. Para obtener dicho modelo primero se elaboró el archivo `robotfr.xacro` del robot. La estructura de este archivo se divide como sigue:

- En primer lugar, se incluyen los archivos .xacro que contienen los materiales y colores de los eslabones, tanto para Gazebo como Rviz. Estos archivos se denominaron `colores.gazebo` y `materiales.xacro`.

- Luego, se elabora la estructura cinemática del robot, la cual se puede visualizar en la figura 2. En resumen, el robot presenta un eslabón base (base_link), el cual actua como unión entre el entorno y el robot. Además, presenta 6 eslabones (*links*) y 6 articulaciones de tipo revolución (*joints*). Es importante resaltar que dentro cada *link* se especificaron las propiedades inerciales, de visualización y de colisión. Asimismo, se utilizó el comando `<mesh>` tanto para la visualización y simulación del robot. Los archivos 3D que se utilizaron fueron obtenidos del repositorio *open source* mostrado en [3].

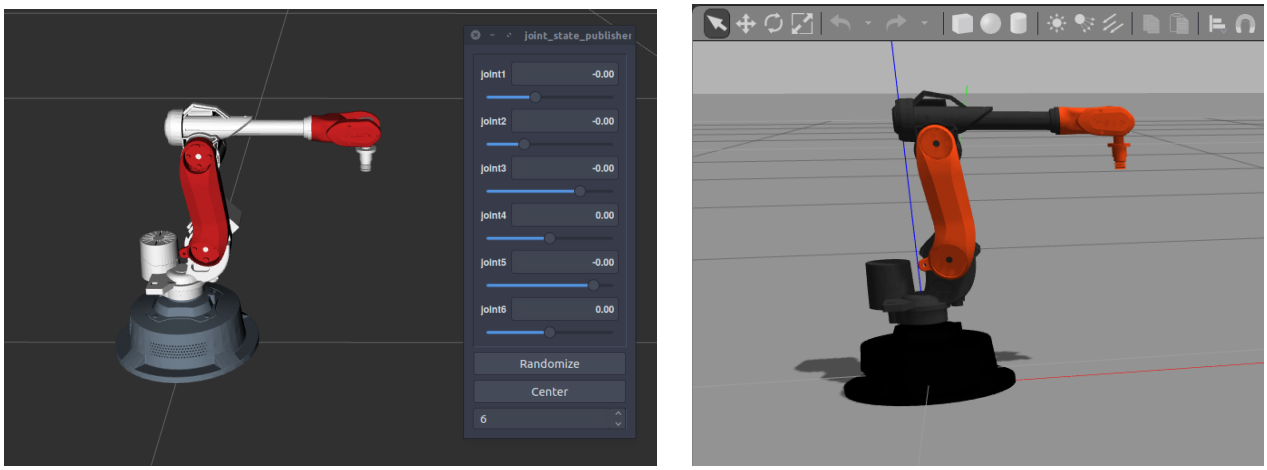
- Además de ello, se agrega el comando de `<transmission>`, en el cual se especifica el tipo de transmisión que generará el actuador, el nombre de la articulación a la cual corresponde el actuador, el tipo de interfaz de hardware que se utiliza en el robot y la reducción mecánica. En este caso se utilizó por defecto la interfaz de tipo *EffortJointInterface* y una reducción mecánica de 1. Para ello, se definió un `macro`, dado que es necesario especificar el tipo de transmisión para cada articulación.
- Finalmente, se especifica el plugin de control (`gazebo_ros_control`), el cual asigna las interfaces de hardware y el controlador mediante el archivo `libgazebo_ros_control.so`. Esto es necesario para realizar la simulación en Gazebo.

Para convertir el archivo *xacro* a un archivo URDF `(robot.urdf)` se utilizó el siguiente comando:

```
$ rosrun xacro xacro --inorder robotfr.xacro > robotfr.urdf
```

Luego de realizar el URDF del robot es necesario crear los archivos con la extensión `.launch`, los cuales nos permitiran visualizar el robot en Rviz y realizar la simulación en Gazebo. El archivo que corresponde a Rviz se denomina `display.launch` y el que corresponde a Gazebo es el archivo `control_gazebo.launch`.

En la Figura 2a se puede observar al robot en el entorno de Rviz y en la Figura 2b en el entorno de Gazebo.



(a) Modelo del robot en Rviz

(b) Modelo del robot en Gazebo

Figura 2: Modelo del robot en los entornos de RViz y Gazebo

3. Cinemática directa

Para obtener la cinemática directa se utilizará la convención estándar de Denavit Hartenberg. Por ello, primero se definen los sistemas de referencia de acuerdo a la convención. Estos se muestran en la Figura 3.

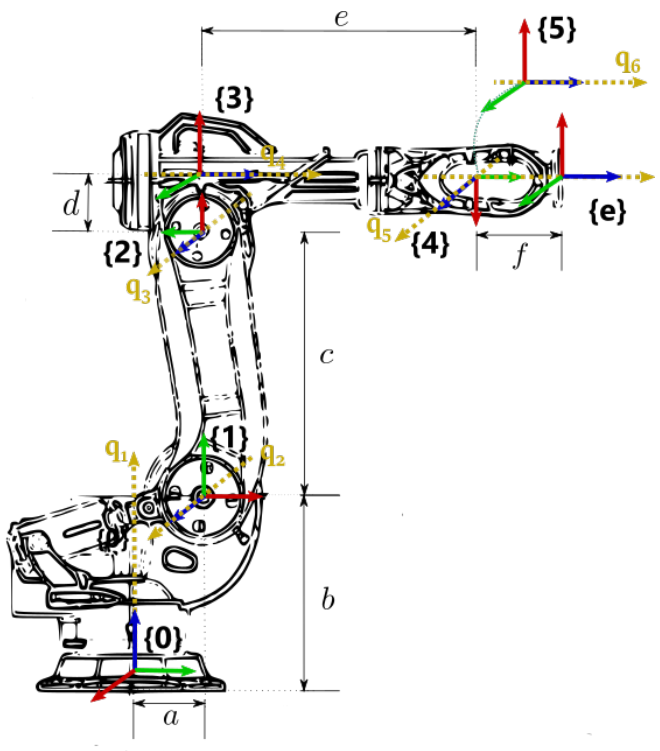


Figura 3: Medidas del brazo robótico

Donde se tienen las siguientes medidas:

Variable	valor (mm)
a	32
b	127
c	108
d	20
e	169
f	70

Tabla 2: Medidas principales del robot.

A partir de esto, se determinan los parámetros de DH como se muestran en la siguiente tabla.

Articulación i	d	θ	a	α
1	b	$\pi + q_1$	-a	$\frac{\pi}{2}$
2	0	$\frac{\pi}{2} + q_2$	c	0
3	0	$\pi + q_3$	-d	$\frac{\pi}{2}$
4	e	$\pi + q_4$	0	$\frac{\pi}{2}$
5	0	$\frac{\pi}{2} + q_5$	0	$\frac{\pi}{2}$
6	-f	q_6	0	π

Tabla 3: Parámetros de Denavit-Hartenberg.

Teniendo los parámetros de DH se utilizará la transformación homogénea que relaciona cada sistema con respecto al sistema anterior. Este es de la forma:

$$ {}^{i-1}T_i = \begin{bmatrix} cos(\theta_i) & -cos(\alpha_i)sin(\theta_i) & sin(\alpha_i)sin(\theta_i) & a_i cos(\theta_i) \\ sin(\theta_i) & cos(\alpha_i)cos(\theta_i) & -sin(\alpha_i)cos(\theta_i) & a_i sin(\theta_i) \\ 0 & sin(\alpha_i) & cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1} $$

Entonces, con los parámetros obtenidos se obtienen las transformaciones homogéneas de cada sistema con respecto al anterior.

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

$$ {}^{1}T_2 = \begin{bmatrix} -sin(q_2) & -cos(q_2) & 0 & -c \cdot sin(q_2) \\ cos(q_2) & -sin(q_2) & 0 & c \cdot cos(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3} $$

$$ {}^{2}T_3 = \begin{bmatrix} -cos(q_3) & 0 & -sin(q_3) & d \cdot cos(q_3) \\ -sin(q_3) & 0 & cos(q_3) & d \cdot sin(q_3) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4} $$

$$ {}^{3}T_4 = \begin{bmatrix} -cos(q_4) & 0 & -sin(q_4) & 0 \\ -sin(q_4) & 0 & cos(q_4) & 0 \\ 0 & 1 & 0 & e \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5} $$

$$ {}^{4}T_5 = \begin{bmatrix} -sin(q_5) & 0 & cos(q_5) & 0 \\ cos(q_5) & 0 & sin(q_5) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6} $$

$$
{}^{5}T_{6} = \begin{bmatrix} cos(q_6) & sin(q_6) & 0 & 0 \\ sin(q_6) & -cos(q_6) & 0 & 0 \\ 0 & 0 & -1 & -f \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}
$$

Luego, para obtener la orientación y posición del efector final con respecto a la base se realizan multiplicaciones de estas transformaciones de forma sucesiva, así como se muestra a continuación.

$$
{}^{0}T_{6} = ({}^{0}T_{1})({}^{1}T_{2})({}^{2}T_{3})({}^{3}T_{4})({}^{4}T_{5})({}^{5}T_{6})
$$

De esta forma, se obtiene una matriz bastante extensa. Por ello, se obtendrá la matriz del efector final con respecto a la base para una cierta configuración de las articulaciones. A modo de comprobar la configuración inicial planteada, se tomarán valores cero.

$$
\mathbf{q} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^{T} \tag{8}
$$

Para esta configuración, la matriz de transformación ${}^{0}T_{6}$ queda de la forma:

$$
{}^{0}T_{6} = \begin{bmatrix} 1 & 0 & 0 & a+e \\ 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & b+c+d-f \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}
$$

Si se observa la figura 3, se puede comprobar que la orientación y posición coinciden con lo obtenido en la matriz. De la misma forma, se ha comprobado este resultado implementando la configuración y representándola de forma visual con la librería para robótica serial proporcionada en Python.

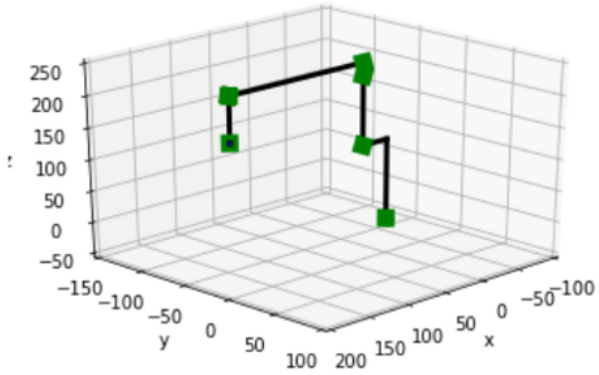


Figura 4: Visualización de la configuración para valores articulares cero (ejes en mm)

Con esta visualización, se identifica la obtención de la configuración inicial del robot mostrado en la figura 3 tanto en posición y orientación, confirmando la correcta obtención de los parámetros de Denavit Hartenberg. De forma arbitraria, se plantea otra configuración de las articulaciones para observar el comportamiento del robot. Estos valores articulares se presentan en el siguiente vector:

$$\mathbf{q} = \begin{bmatrix} -1{,}15 & 1{,}03 & -2{,}5 & 3{,}5 & -3{,}31 & 2{,}4 \end{bmatrix}^T \tag{10}$$

Con el cual se obtiene la siguiente matriz de transformación del efector final con respecto a la base en función de las variables.

$$^0T_6 = \begin{bmatrix} -0{,}36 & -0{,}63 & 0{,}68 & 0{,}41a + 0{,}35c - 0{,}41d + 0{,}041e - 0{,}68f \\ -0{,}63 & -0{,}38 & -0{,}68 & -0{,}912a - 0{,}78c + 0{,}91d - 0{,}092e - 0{,}68f \\ 0{,}69 & -0{,}68 & -0{,}26 & b + 0{,}51c + 0{,}1d + e - 0{,}26f \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

De esta forma, se obtiene la siguiente configuración del robot.

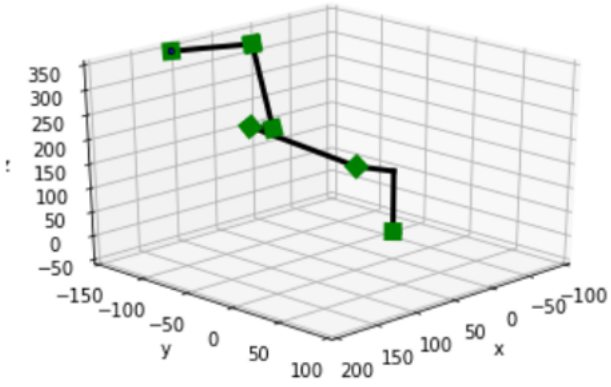


Figura 5: Visualización de nueva configuración arbitraria (ejes en mm)

Posteriormente, se pasa procede a realizar la visualización de la cinemática directa en el programa RViz para comprobar los resultados obtenidos. En primer lugar, se tiene la configuración inicial del robot.

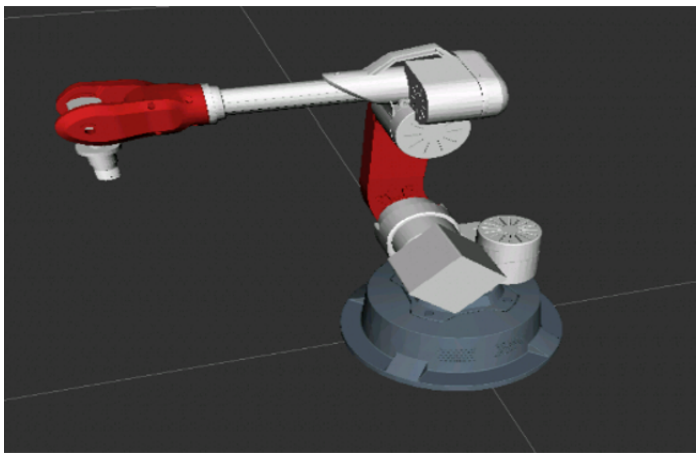


Figura 6: Configuración inicial en RViz

Posteriormente, se muestra la configuración final del robot en la figura 7

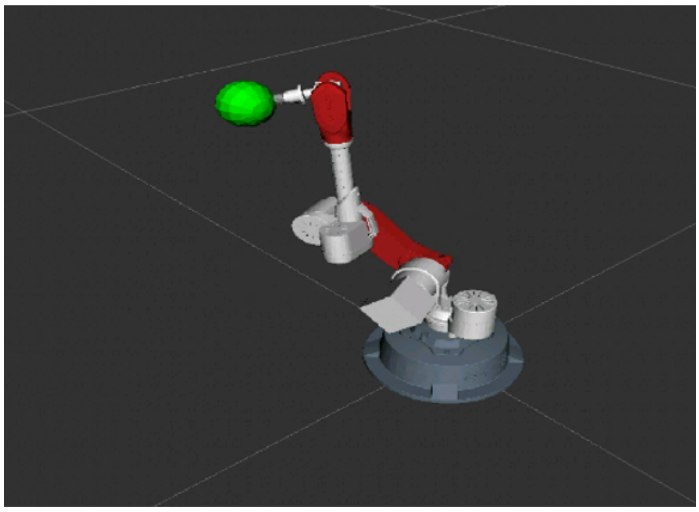


Figura 7: Configuración final en RViz

Como se observa, la configuración de las figuras 6 y 7 son similares a las figuras 4 y 5 respectivamente. De forma numérica se puede comprobar al obtener las mismas transformaciones homogenéneas del sistema del efector final con respecto a la base en ambos casos, lo cual comprueba los cálculos correctos de los parámetros DH.

4. Cinemática inversa

Para el análisis de la cinématica inversa del brazo robótico se utilizará un método numérico, debido a que el robot cuenta con 6 grados de libertad. Para ello, se ha optado por realizar el método de Newton, dada su rápida convergencia en pocas iteraciones. Para poder explicar este método, se realiza un diagrama de flujo que muestra el procedimiento de este (Figura 8).

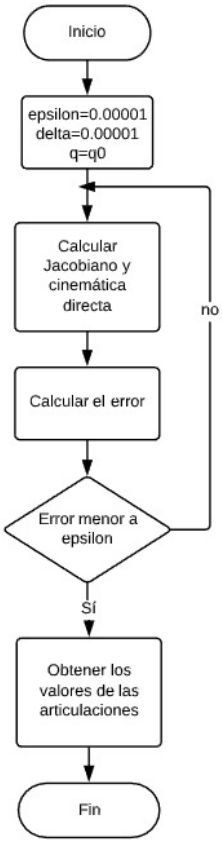


Figura 8: Diagrama de flujo-Método de Newton

Como se puede observar, este método iterativo consiste en llevar la posición actual del efector final a una posición deseada minimizando el error. De esta forma, se obtienen los valores de las articulaciones. Así, una vez teniendo claro la metodología que se utilizará para calcular el valor de las articulaciones deseadas, se necesita calcular el jacobiano, dado que el valor de las articulaciones depende de la inversa de este multiplicado por el error. Esto puede ser verificado en el programa realizado en MATLAB.

Para poder realizar este método, se toma la cinemática directa realizada anteriormente y se utilizaron los valores de la Tabla 2, dado que ahí se tienen las longitudes de cada link. La matriz de transformación homogénea del efector final respecto a la base cuando los valores articulares son cero es la siguiente:

$$^0T_6 = \begin{bmatrix} 1 & 0 & 0 & 201 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 185 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

Es decir,cuando todos los valores articulares son cero, la posición deseada e igual a:

$$x_{des} = \begin{bmatrix} 201 \\ 0 \\ 185 \end{bmatrix} \tag{13}$$

Sin embargo, para poder utilizar una configuración diferente, se tomará una posición deseada distinta. Nuestra posición deseada será igual a:

$$x_{des} = \begin{bmatrix} -0{,}2 \\ 0{,}15 \\ 0{,}28 \end{bmatrix} \tag{14}$$

Una vez dado la posición deseada del efector final, se debe dar como parámetro un vector con las posiciones iniciales de los 6 valores articulares, puesto que la diferencia entre los valores articulares finales y los iniciales es la que se tratará de minimizar. Para probar lo mencionado, se especifican los siguientes valores iniciales:

$$q_0 = \begin{bmatrix} 0{,}0 \\ -1{,}0 \\ 1{,}7 \\ -2{,}2 \\ -1{,}6 \\ 0{,}0 \end{bmatrix} \tag{15}$$

Luego de haber obtenido los valores de la posición deseada y una configuración inicial para las articulaciones, se procederá a realizar la metodología mostrada en el diagrama de flujo presentado anteriormente. De esta manera, la salida obtenida es igual a la configuración articular deseada; es decir, ya se tendría la cinemática inversa. Los valores obtenidos bajo la configuración inicial dada es igual a:

$$q_f = \begin{bmatrix} -22{,}365 \\ 51{,}427 \\ -45{,}911 \\ -1{,}319 \\ -43{,}788 \\ 0 \end{bmatrix} \tag{16}$$

Sin embargo, con el fin de comprobar si lo mencionado se ha realizado de manera correcta, se realizará la cinemática directa bajo esta configuración para ver si es que se obtiene como posición del efector final, el valor de la posición deseada. Para esto, se aplicará el mismo método explicado en la sección de cinématica directa. De esta manera se obtuvo la siguiente matriz de transformación

homogénea:

$$ {}^{0}T_{6} = \begin{bmatrix} -0{,}62 & 0{,}717 & 0{,}319 & -0{,}200 \\ 0{,}445 & -0{,}014 & 0{,}869 & 0{,}150 \\ 0{,}646 & 0{,}697 & -0{,}310 & 0{,}28 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17} $$

A partir de (17), se puede corroborar viendo la última columna que la posición deseada se mantuvo, es decir se obtuvo los valores propuestos inicialmente.
Así como en la cinemática directa, se puede comprobar los resultados en RViz. Donde se tiene de igual forma la configuración y posición inicial del efector.

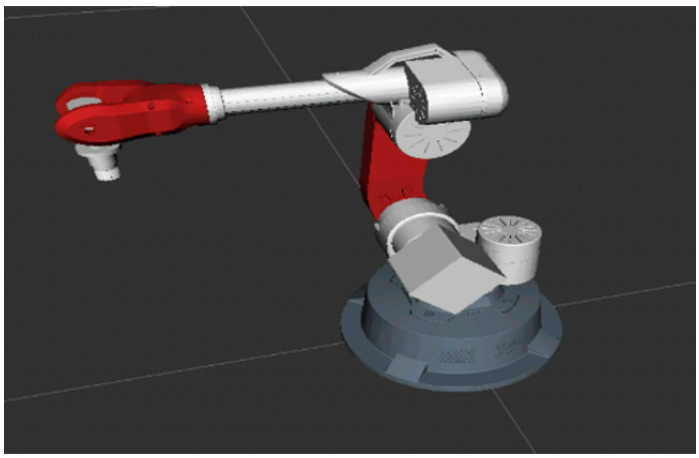


Figura 9: Posición inicial

Posteriormente la configuración obtenida por cinemática inversa para llegar a la posición cartesiana deseada.

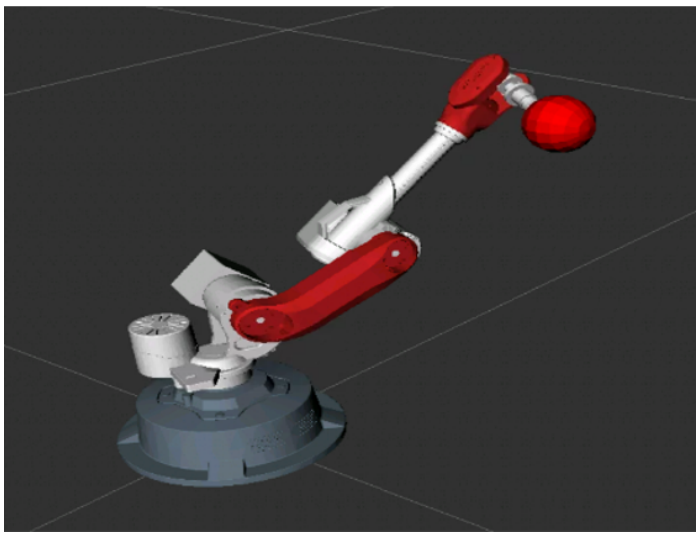


Figura 10: Posición deseada del efector final

5. Control cinemático

El control cinemático es un método de control en el que se utiliza la cinemática diferencial del robot para determinar la trayectoria que debe tomar el efector final y reducir el error entre la posición deseada con la real. Este método trabaja en el espacio operaciona, por lo que solo se enfoca en llegar a la posición y orientación deseada del efector final y no se trata de controlar las articulaciones individuales del brazo robot.

Para realizar este control se hace uso de la cinemática directa, la cual ya se realizó previamente; se debe calcular el jacobiano analítico y también se debe realizar el algoritmo de control cinemático. En esta sección se realizará el control cinemático solo de la posición.

El cálculo del jacobiano, también llamado jacobiano de la tarea, se puede calcular con métodos numéricos. Este se obtiene derivando la posición con respecto a las articulación del robot, ver (18).

$$J_A = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \frac{\partial x}{\partial q_3} & \frac{\partial x}{\partial q_4} & \frac{\partial x}{\partial q_5} & \frac{\partial x}{\partial q_6} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \frac{\partial y}{\partial q_3} & \frac{\partial y}{\partial q_4} & \frac{\partial y}{\partial q_5} & \frac{\partial y}{\partial q_6} \\ \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \frac{\partial z}{\partial q_3} & \frac{\partial z}{\partial q_4} & \frac{\partial z}{\partial q_5} & \frac{\partial z}{\partial q_6} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial q_1} & \frac{\partial \mathbf{x}}{\partial q_2} & \frac{\partial \mathbf{x}}{\partial q_3} & \frac{\partial \mathbf{x}}{\partial q_4} & \frac{\partial \mathbf{x}}{\partial q_5} & \frac{\partial \mathbf{x}}{\partial q_6} \end{bmatrix} \tag{18}$$

Donde $\mathbf{x}$ es la posición en el espacio cartesiano: $\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$

Computacionalmente estas derivadas parciales pueden calcularse con diferencias finitas.

$$\frac{\partial x}{\partial q_i} \approx \frac{\mathbf{x}(\mathbf{q} + \partial q_i) - \mathbf{x}(\mathbf{q})}{\partial q_i} \tag{19}$$

Luego de haber calcular el jacobiano se realiza un algoritmo iterativo en el que se actualiza el valor articular del robot con respecto al error de posición. Lo primero que se hace es calcular este error de posición:

$$\mathbf{e} = \mathbf{x} - \mathbf{x}_d \tag{20}$$

Para realizar un control de posición de primer orden y que el error converja a cero se utiliza la siguiente ley de control cinemática:

$$\dot{\mathbf{e}}^* = -k\mathbf{e} \tag{21}$$

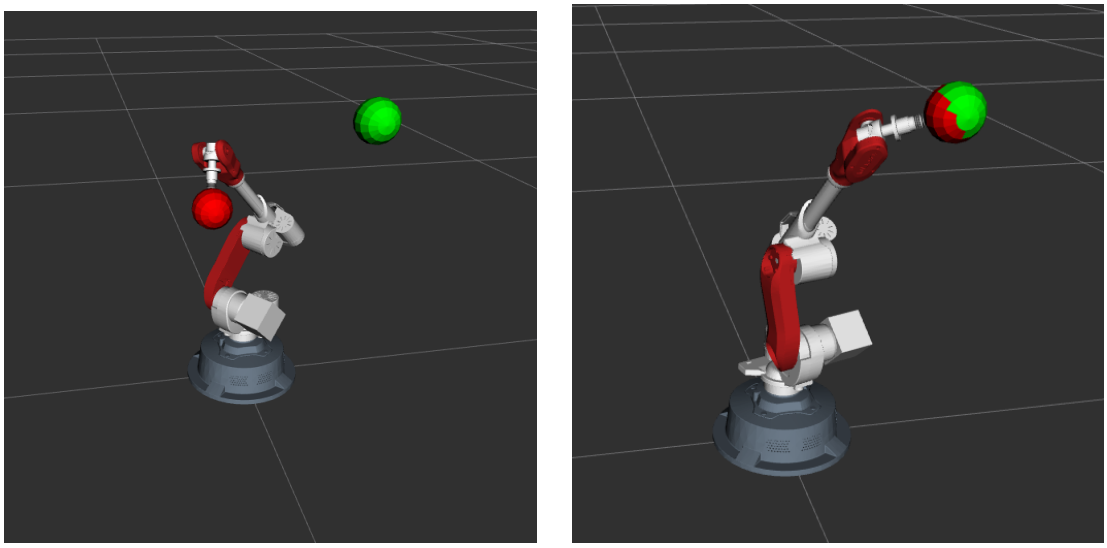
Donde k es un escalar conocido como ganancia cinemática y debe tomar un valor adecuado para tener el mejor comportamiento de la trayectoria del robot. Con la derivada del error usando (21) y la inversa del jacobiano analítico se puede calcular las velocidades articulares con cinemática diferencial.

$$\dot{\mathbf{q}} = J^{\#}\dot{\mathbf{e}} \tag{22}$$

Finalmente se integra las velocidades articulares usando la integración de Euler en la cual se actualiza el valor anterior de $\mathbf{q}$:

$$\mathbf{q}_k = \mathbf{q}_{k-1} + \Delta t \dot{\mathbf{q}}_k \tag{23}$$

Donde Δt es el periodo de control que tiene el algoritmo del robot.
Este algoritmo se implementó en un script de python para realizar la visualización del control cinemático en RViz. En este script se definió una configuración inicial del robot, la cual fue la configuración con las articulaciones en 0, y una posición deseada en el plano cartesiano. Asimismo se crea un loop para actualizar constantemente la configuración articular del robot, con el algoritmo descrito anteriormente, hasta llegar cerca a la posición deseada. Esto se logra comparando la norma del error con un valor de parada, llamado comunmente *epsilon* y se determinó con un valor de 0.001. Como ejemplo se definió la posición deseada como $\mathbf{x}$=[-0.1, 0.15, 0.4] y se ejecutó el programa. Algunas imágenes de la visualización se presentan a continuación:



(a) Inicio del robot con control cinemático (b) Fin del robot con control cinemático

Figura 11: Control Cinemático del robot

Para comprobar que el efector final está llegando a la posición deseada se ha creado la gráficas de la posición real vs. la posición deseada en cada eje cartesiano con respecto al tiempo (Fig. 12). Asimismo para comprobar que la ley de control cinemática converje a cero, como se determinó en la ecuación (21), en la Fig. 13 se muestran las gráficas del error en cada eje cartesiano.

En la Fig. 12 se puede ver como la posición en cada eje cartesiano (color azul) converge hacia el valor indicado para el ejemplo (línea verde). Según las gráficas también se puede determinar que el efector final llega a la posición deseada en un tiempo de 2 segundos aproximadamente y que la respuesta del control presenta un comportamiento sobreamortiguado. De la Fig. 13 se puede observar que el error (línea azul) verdaderamente tiende a cero (el cual fue señalado con una línea verde) con el tiempo debido a la ley de control definida en el algoritmo.

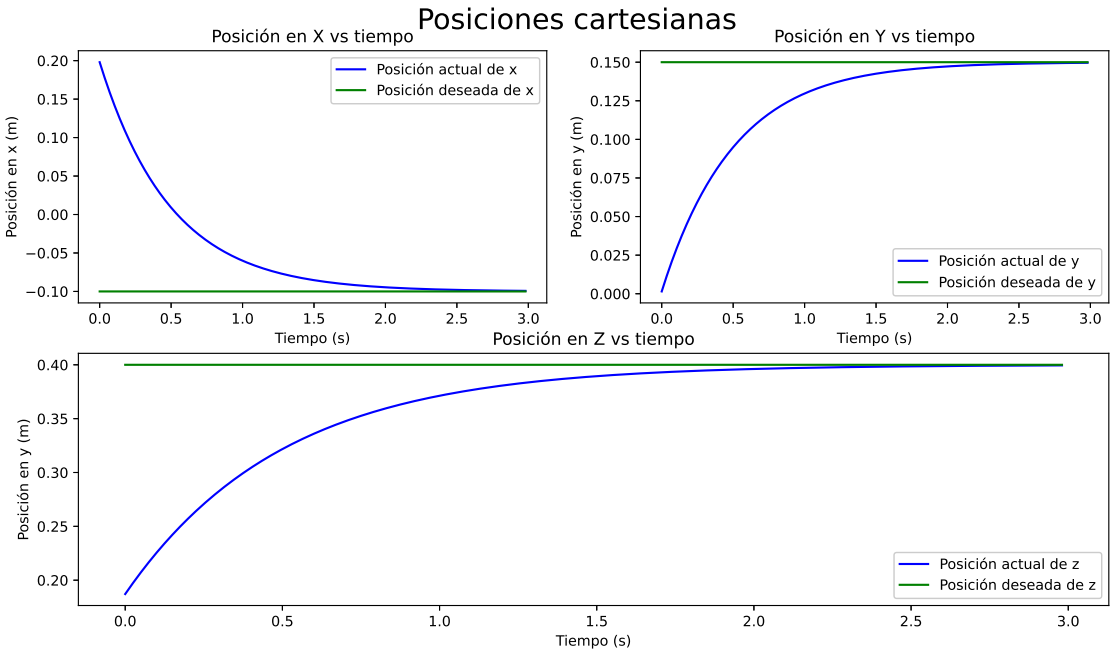


Figura 12: Gráficas de las posiciones cartesianas con respecto al tiempo con control cinemático

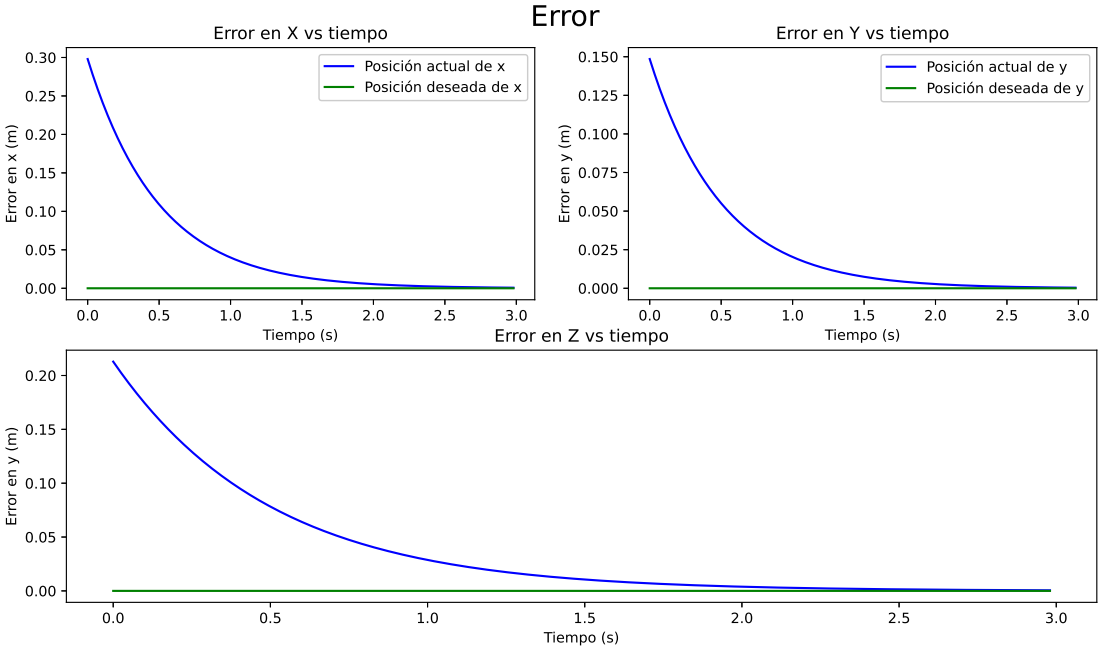


Figura 13: Gráficas del error de las posiciones cartesianas con respecto al tiempo con control cinemático

6. Control Dinámico

El control dinámico, a diferencia del control cinemático, requiere del modelo dinámico del robot, esto debido a que la ley de control está compuesta por las matrices que representan al robot. Ejemplo de esto podemos ver en el control por dinámica inversa, donde la ley de control u está compuesta por la matriz de Coriolis, el vector de gravedad y la matriz de inercia. Por este motivo, con el fin de encontrar la ley de control, se hizo uso del paquete RBDL, puesto que nos facilitaba el cálculo de estas matrices, ya que en caso contrario hubiera sido necesario realizarlo de manera analítica. Al utilizar dicho paquete es necesario calcular cada uno de estas matrices en un orden determinado. Este orden comienza con el cálculo del vector de gravedad, luego la matriz de Coriolis y finalmente la matriz de inercia. Al obtener cada término de manera independiente, se procede a colocar la ley de control de la siguiente manera:

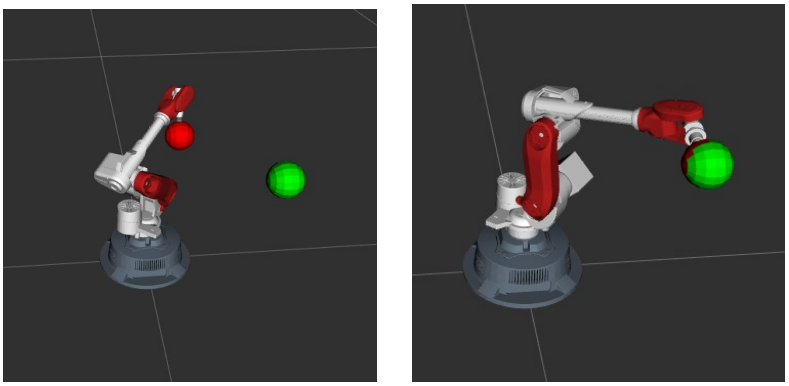
$$u = C\dot{q} + g + M(\ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_p - q))$$

La idea al realizar el control por dinámica inversa es compensar todos los términos del modelo dinámico a través de la ley de control. Para esto debemos realizar algunos despejes matemáticos, con el fin de obtener una ecuación que dependa únicamente del error y de las constantes de un controlador PD. Esta ecuación se representa de la siguiente manera:

$$\ddot{e} + K_d\dot{e} + K_p e = 0$$

Así, una vez obtenida la ecuación anterior, sabemos que el error debe tender a 0. Por lo que se debe comenzar a sintonizar los parámetros del controlador PD, para poder obtener un comportamiento deseado, es decir, un tiempo de establecimiento pequeño y un comportamiento críticamente amortiguado. Para ello, se escogen los valores de tal manera que $K_d = 2\sqrt{K_p}$.

A partir de lo anterior se implementa el algoritmo para obtener la ley de control y con ello realizar el control dinámico del robot. En la Figura 14 se observan las gráficas del robot al inicio y al final del movimiento. En la imagen de la izquierda, figura 14a, se observa al robot en su posición inicial, donde la posición del efector final se representa con un marcador rojo. En la imagen de la derecha, figura 14b, se puede observar que el efector final llegó a la posición deseada, puesto que ambas bolitas se encuentra superpuestas.



(a) Inicio del robot con control dinámico

(b) Fin del robot con control dinámico

Figura 14: Control Dinámico del robot

Esto se logró utilizando las siguientes matrices de K_p y K_d:

$$\mathbf{K_p} = diag(\begin{bmatrix}3 & 3 & 3 & 3 & 3 & 3\end{bmatrix}) \text{ y } \mathbf{K_d} = diag(\begin{bmatrix}3{,}46 & 3{,}46 & 3{,}46 & 3{,}46 & 3{,}46 & 3{,}46\end{bmatrix})$$

Además de ello, se realizaron las gráficas de las trayectorias realizadas por cada articulación, con el fin de verificar que estas cumplen con el comportamiento deseado. En la Figura 15 se muestra el comportamiento de las articulaciones q_0 y q_1. Como se observa, no existe un porcentaje de sobreimpulso y se llega al valor deseado en ambas articulaciones. Asimismo, el tiempo de establecimiento de ambas trayectorias es rápida, puesto que se llega a la referencia en un tiempo menor a 5 segundos.

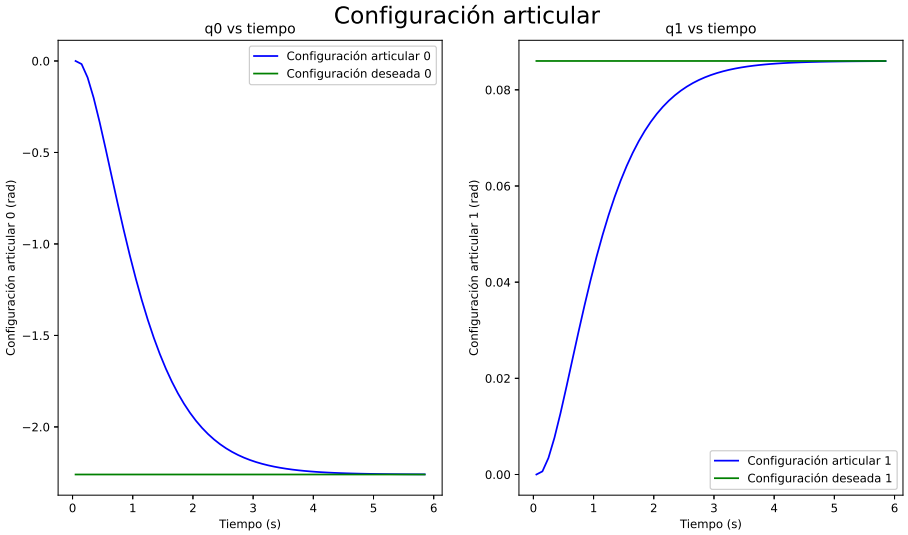


Figura 15: Configuración articular de q0 y q1 con control dinámico

En la Figura 16 se muestra el comportamiento de las articulaciones q_2 y q_3. Se puede observar que el comportamiento se mantiene, cumpliendo con los requerimientos solicitados.

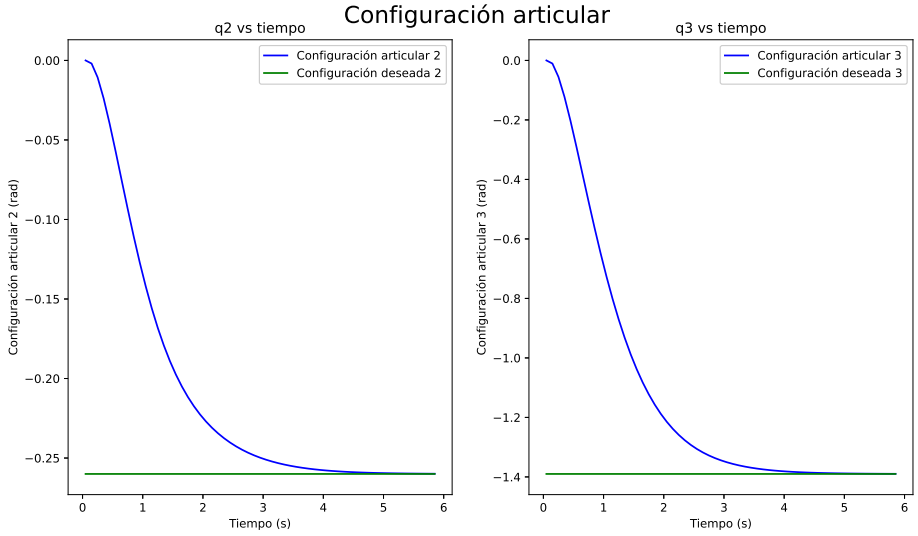


Figura 16: Configuración articular de q2 y q3 con control dinámico

Finalmente, en la figura 17 se observa que la articulación q_4 presenta un comportamiento similar al de las articulaciones anteriores. Sin embargo, la articulación q_5 si cuenta con un ligero sobreimpulso puesto que sobrepasa la referencia para luego recién estabilizarse.

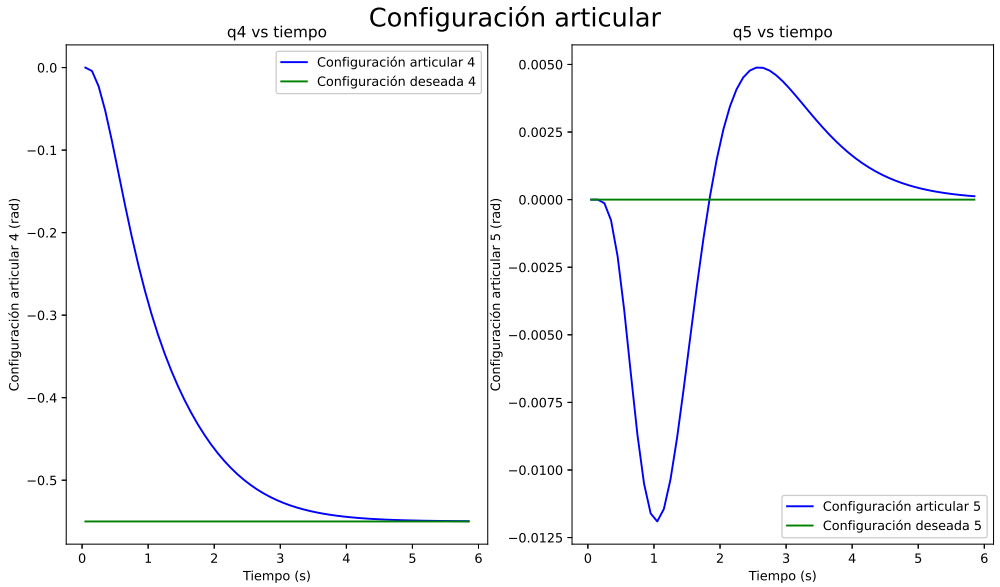


Figura 17: Configuración articular de q4 y q5 con control dinámico

En cuanto al espacio operacional, se puede analizar las posiciones cartesianas del efector final durante la trayectoria dada. En la figura 18 se observa que el efector final llega a la posición deseada en sus 3 ejes en un tiempo aproximado de 5 segundos.

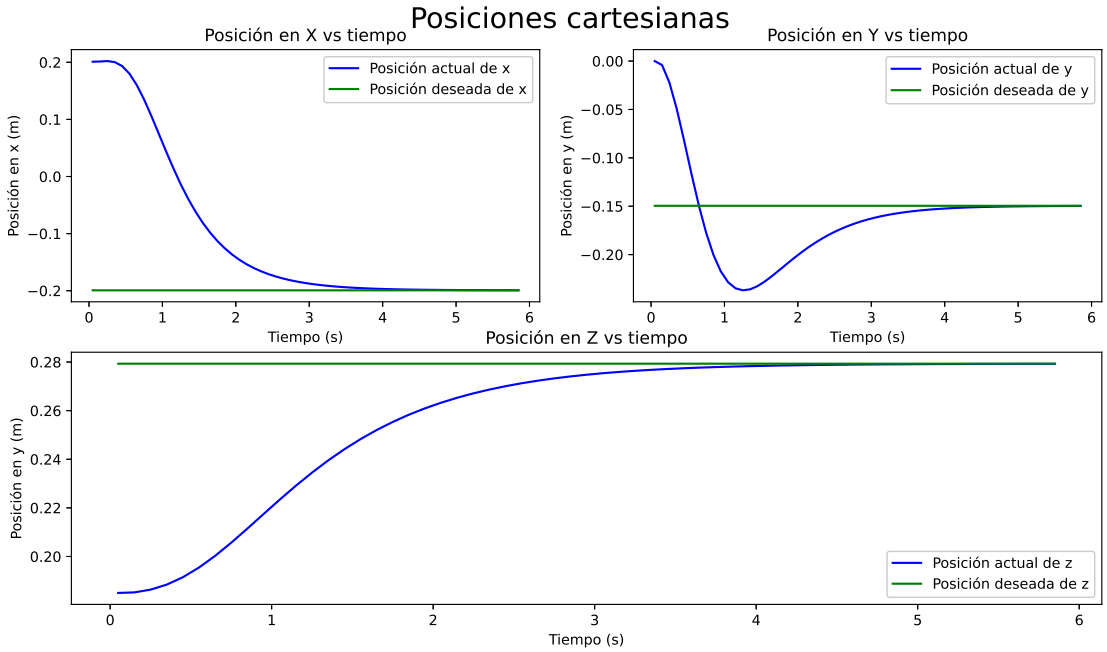


Figura 18: Gráficas de las posiciones cartesianas con respecto al tiempo con control dinámico

Así como se ha analizado el control dinámico del robot manipulador por dinámica inversa, existen otros métodos con los cuales también se puede realizar el control dinámico de un robot. Con el fin de comparar dichos métodos, se realizó el control PD + Compensación de gravedad. En este método, la ley de control u se representa de la siguiente manera:

$$u = g + K_p e - K_d \dot{q}$$

En este método, a diferencia del anterior, no se debe compensar cada término del modelo del robot, sino solamente el vector de gravedad. Además, hay que tener en cuenta que el comportamiento de cada articulación también varía respecto al control por dinámica inversa. Para visualizar este comportamiento, se graficó el comportamiento de cada articulación. En la figura 19, se puede ver que las articulaciones tienen un comportamiento subamortiguado, además de tener un tiempo de establecimiento mucho mayor al del control por dinámica inversa.

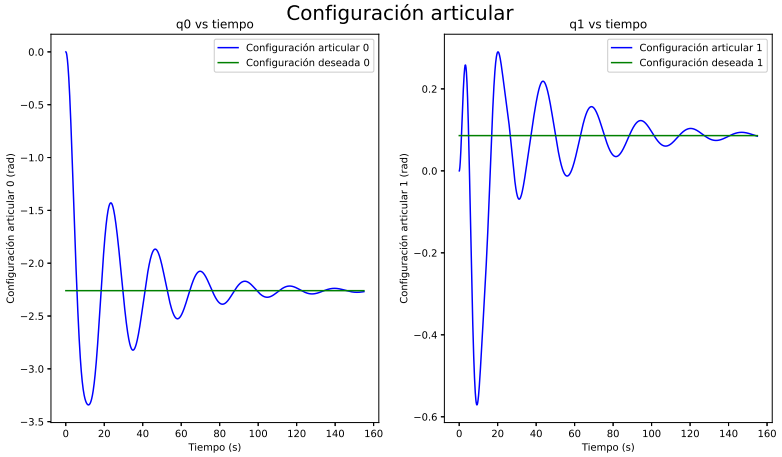


Figura 19: Configuración articular de q0 y q1 con compensación de gravedad

En la figura 20 y 21, se puede ver que las articulaciones q_2, $q_3, q_4 y q_5$ también tienen un comportamiento subamortiguado, además de tener un tiempo de establecimiento mucho mayor al del control por dinámica inversa.

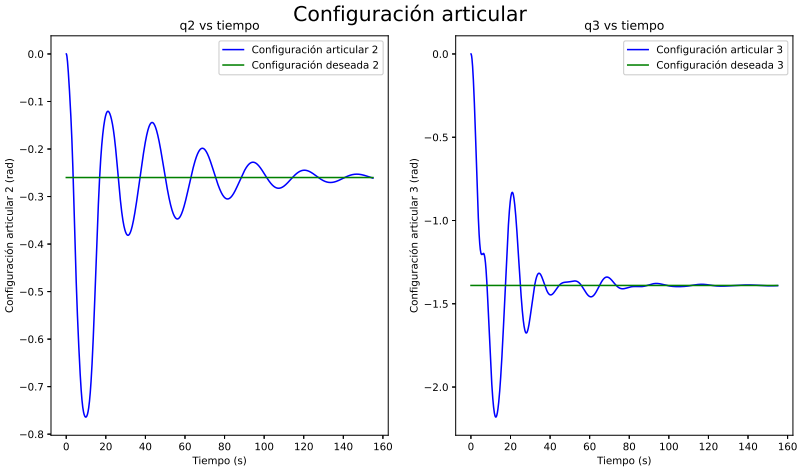


Figura 20: Configuración articular de q2 y q3 con compensación de gravedad

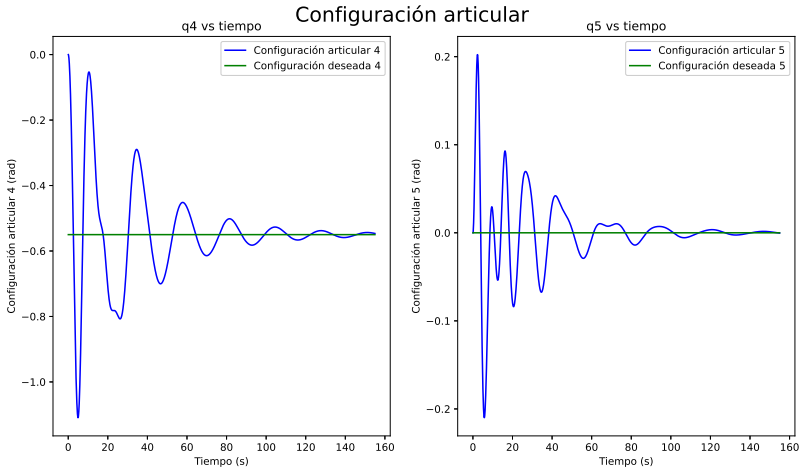


Figura 21: Configuración articular de q4 y q5 con compensación de gravedad

En cuanto al espacio operacional, se puede analizar las posiciones cartesianas del efector final durante la trayectoria dada. Como se puede ver en la figura 22, el efector final llega a la posición deseada en sus 3 ejes. Sin embargo, estas siguen teniendo un mayor tiempo de establecimiento y un comportamiento subamortiguado, por lo que genera muchos sobreimpulsos durante la trayectoria.

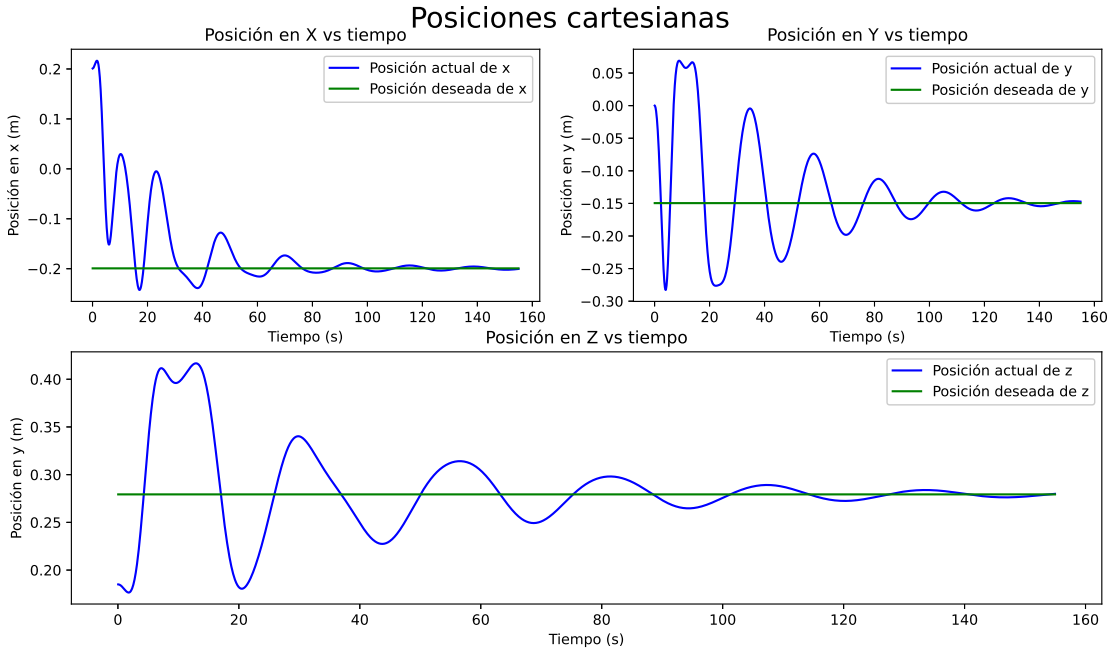


Figura 22: Gráficas de las posiciones cartesianas con respecto al tiempo con compensación de gravedad

Después de haber realizado el control dinámico por ambos métodos, se puede concluir que para el robot manipulador conviene utilizar el control por dinámica inversa, puesto que tiene un menor tiempo de establecimiento considerable. Además que no genera sobreimpulsos durante la trayectoria del efector final por llegar a la posición deseada.

7. Conclusiones

- Con respecto al modelo del robot, se decidió trabajar con archivos de tipo `xacro`, debido a que estos nos permiten incluir archivos similares y de este modo tener un código más ordenado. Además de ello, se pueden generar `macros` que nos brindan mayor flexibilidad y reducen las líneas de código. Asimismo, luego de configurar nuestro archivo `xacro` es posible convertirlo a formato URDF mediante el *script* `xacro.py`. De este modo, se asegura que el archivo URDF se realizó de manera correcta y ya no es necesario alguna modificación.

- La cinemática inversa permite obtener la configuración de las articulaciones del robot a partir de la posición del efector final. Cabe resaltar que existen más de una opción de valores articulares que cumplan con la posición deseada. Los valores articulares obtenidos dependen del valor inicial de estos valores ya que pueden converger en distintos ángulos.

- En cuanto al control cinemático se desarrolló un algoritmo de control basado en cinemática diferencial, el cual se enfocaba en la reducción del error entre la posición deseada y la posición actual del efector final. El algoritmo de control desarrollado actualizaba los valores articulares del robot en cada iteración con el cálculo del Jacobiano de posición y del error de posición. Se pudo ver en las gráficas que en cada coordenada cartesiana la posición actual llegaba a la posición deseada con un comportamiento sobreamortiguado y con un tiempo de establecimiento relativamente corto.

- Sobre el control dinámico, se realizaron dos métodos distintos con el fin de visualizar las respuestas temporales de cada articulación. Se puede concluir que en el caso de este robot manipulador, el método de control por dinámica inversa es más eficiente debido al menor tiempo de establecimiento que tiene con respecto al método de control por compensación de gravedad. Asimismo, en cuanto a su comportamiento temporal, se pudo ver que el método de control por dinámica inversa obtuvo un comportamiento críticamente amortiguado, mientras que el otro método tenía un comportamiento subamortiguado. Este tipo de comportamiento se caracteriza por contar con sobreimpulsos a lo largo del tiempo generando ciertas vibraciones en los links del robot a lo largo de su movimiento.

8. Recomendaciones

- Para obtener una mejor resolución del diseño 3D se recomienda utilizar archivos *collada* (extensión *.dae*). Este tipo de archivos permiten tener diseños 3D con mayor calidad y además se le puede agregar colores y texturas más realistas.

- En el control cinemático se utilizó una ganancia cinemática k para tener una ley de control que reduzca el error de posición. Se podría probar diferentes valores de esta ganancia con el objetivo de tener una respuesta más rápida o más lenta dependiendo de los requerimientos que posea el robot para una tarea específica.

- En el control dinámico se utilizaron dos métodos para comparar el comportamiento temporal de cada articulación. Se recomendaría utilizar más métodos con el objetivo de analizar más respuestas temporales y ver cuál se adapta mejor a los requerimientos que posea el robot para una tarea específica.

- Es importante definir tanto los límites articulares cómo los límites de posición del efector final. Esto debido a los límites físicos de los componentes del robot. Si no se realiza una correcta limitación de los valores articulares, los motores que componen estos pueden verse dañados físicamente, por lo que es recomendable respetar los parámetros de seguridad y simular el movimiento del robot en un software como Gazebo antes de implementarlo en el robot real para evitar daños irreparables.

Referencias

[1] R. Siegwart, *Introduction to autonomous mobile robots*. Cambridge: Cambridge University Press, 2004.

[2] Visit https://github.com/DPR00/ProjectFR

[3] Visit https://github.com/wlkata/Mirobot-STL