

Robot manipulador con plataforma omnidireccional*

*Proyecto Final para el curso de Fundamentos de Robótica

Cabeza José

Dep. Ingeniería Mecatrónica
UTECH

jose.cabeza@utec.edu.pe

Carita Manuel

Dep. Ingeniería Mecatrónica
UTECH

manuel.carita@utec.edu.pe

Custodio Jairo

Dep. Ingeniería Mecatrónica
UTECH

jairo.custodio@utec.edu.pe

Rosario Quispe

Dep. Ingeniería Mecatrónica
UTECH

rosario.quispe@utec.edu.pe

Abstract—En este paper se presenta a un robot móvil con llantas omnidireccionales que posee anclado a su chasis un brazo robótico de 6 grados de libertad que le permite una aplicación más consistente en un medio físico. El brazo robótico que se utilizó para este proyecto es el KUKA KR4 Agilus. En resumen, se seleccionaron los componentes que utilizarían para una implementación real, luego se realizó un modelamiento del robot en URDF tanto del chasis como del brazo. Asimismo, se modeló la cinemática directa e inversa del robot. Con ello, se realizó el control cinemático tanto del brazo como de la plataforma omnidireccional. Por último, se modeló la dinámica del brazo robótico y se aplicó un control por dinámica inversa en el espacio articular y en el operacional. La visualización del robot con sus controles aplicados se realizó en RViz mediante el uso de ROS.

Index Terms—Robot Omnidireccional, Brazo robótico, ROS, Control, Python

I. INTRODUCCIÓN

Los robots manipuladores móviles están sujetos a sus respectivas restricciones cinemáticas y los grados de libertad del brazo manipulador que se encuentra encima de la plataforma. Estos sistemas combinan las ventajas de las plataformas móviles y su respectivo brazo robótico, puesto que tiene un mayor espacio de trabajo y este mismo ofrece mayor funcionalidad de operación (Avilés, 2018, p. 133).

Las aplicaciones de este tipo de robot están en el ámbito de seguridad militar, operaciones espaciales, construcciones, servicio personal y empresarial. En el ámbito militar y de seguridad, estos sistemas robóticos realizan tareas de desminado y manejo de explosivos. Por otro lado, también realizan tareas de manejo de materiales peligrosos radiactivos, los cuales sería muy riesgoso si una persona lo realiza. En este proyecto, el propósito del robot manipulador con plataforma omnidireccional es principalmente ser parte de la automatización de almacenaje en diversas empresas.

II. COMPONENTES

Los componentes para el robot móvil omnidireccional se basa en los datasheet del robot móvil MPO-500 y del brazo manipulador en el modelo Kuka KR 4 AGILUS de 6 grados de libertad. A continuación se explica los componentes pertenecientes a cada robot.

A. Para el brazo manipulador:

- Servomotores: Los servomotores que se utilizarán son los tipo brushless debido a que no generan fricción y por ende producen menos calentamiento y brindan un mejor rendimiento.
- Encoder: Para el sistema de sensado, cabe resaltar que el motor tipo brushless elegido viene con encoder incluido.
- Unidad de control KR C5 micro: La unidad de control KR C5 ofrece el máximo rendimiento, conectividad y flexibilidad en el espacio más reducido, puesto que se encuentra incorporado en el robot. Además, este tipo de Unidad de control tiene ventajas por ser pequeña, flexible e inteligente.

B. Para robot omnidireccional:

- Motor paso a paso: Este tipo de motor se eligió por su versatilidad y mayor precisión de posicionamiento.
- Encoder: El motor paso a paso viene incorporado con un encoder de tipo incremental óptico, la cual es recomendado para este tipo de motores por los incrementos utilizados.
- Raspberry PI 4: Para el sistema de control se utilizará esta placa de desarrollo por su facilidad de interfaz que cuenta con ROS.
- Driver Puente H: Para accionar los steppers y controlar el sentido de giro se eligió un driver puente h gracias a su compatibilidad con tarjetas de desarrollo como Raspberry.
- Ruedas Mecanum Wheels: Se necesitarán 4 ruedas mecanum wheels, este tipo de ruedas son ruedas con rodillos unidos a su circunferencia. Los rodillos se colocan en diagonal o en un ángulo de 45 grados con respecto al eje de rotación de la rueda para que la rueda ejerza fuerza en dirección diagonal cuando se mueve hacia adelante o hacia atrás.
- Pack de baterías de Tipo AGM sealed lead acid 24 V/50 Ah: Este tipo de baterías es un acumulador ideal para los sistemas que necesitan potencias elevadas. Además, tienen un sistema de carga más rápida y su fácil sistema de cambio.

III. MODELO DEL ROBOT

Para el modelo del robot, se usó los archivos CAD brindados por KUKA (Kuka KR4 agilus) y Neobotix (MPO-500), los cuales fueron ensamblados mediante el programa CAD Autodesk Fusion 360

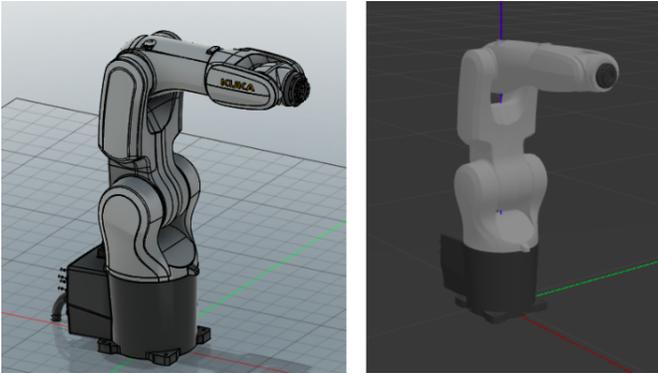


Fig. 1. Modelo del brazo manipulador en Fusion 360 y Gazebo

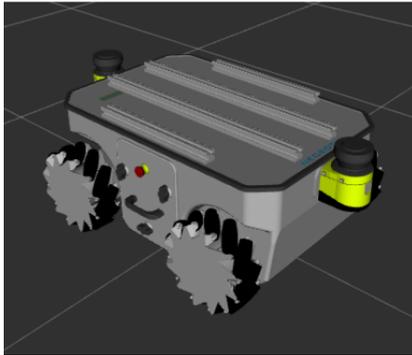


Fig. 2. Modelo de la base móvil omnidireccional

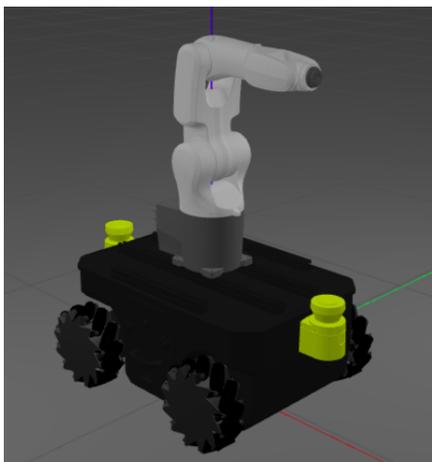


Fig. 3. Modelo del robot manipulador con base móvil

Para la simulación de movimiento de la base móvil del robot, se usó el plugin de gazebo `ros_planar_move` el cual es

usado por la mayoría de usuarios para la cinemática de robots móviles holonómicos

IV. CINEMATICA DIRECTA E INVERSA

La cinemática directa de un robot relaciona la posición y la orientación del efector final en función de las posiciones de las articulaciones. Por otro lado, la cinemática inversa relaciona a las posiciones articulares en función de la posición y orientación del efector final. Para el desarrollo de este proyecto se utilizó el método convencional Denavit-Hartenberg para la cinemática directa y el método numérico de Newton para la cinemática inversa. Esta última hace uso de jacobiano para poder aplicarse el método recursivo de Newton.

A continuación, se expone en detalle, la obtención de la cinemática directa e inversa para el robot KUKA KR4 Agilus.

A. Método de Denavit-Hartenberg

El método de Denavit-Hartenberg ubica de manera convencional y pautada los sistemas de referencia en función a las posiciones y orientaciones de las articulaciones uno con respecto al anterior. Para ello, se hizo uso de la hoja técnica donde se muestra el espacio de trabajo del robot Kuka KR4 Agilus y las distancias correspondientes entre las articulaciones tal como se especifica en la figura 4.

Gráfica del campo de trabajo

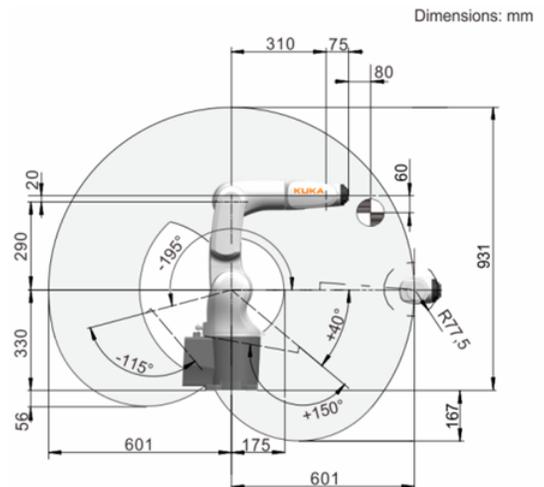


Fig. 4. Espacio de trabajo

A partir del espacio de trabajo se determinó las ubicaciones de los sistemas de referencias con sus respectivas orientaciones y se gráfico convenientemente tal como se muestra en la figura 5, para poder obtener los parámetros del método.

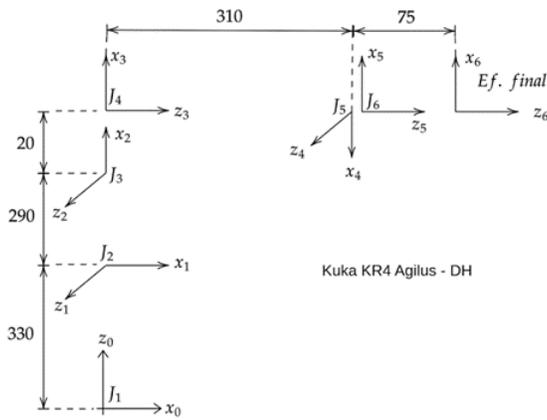


Fig. 5. Método Denavit-Hartenberg

Los parámetros obtenidos a partir de la figura anterior se muestran en la siguiente tabla (Figura 6).

i	d	θ	a	α
1	330	q_1	0	90°
2	0	$-q_2+90^\circ$	290	0°
3	0	$-q_3$	20	90°
4	310	$-q_4+180^\circ$	0	90°
5	0	$-q_5+180^\circ$	0	90°
6	75	$-q_6$	0	0°

Fig. 6. Parámetros del método D-H

B. Cinemática Directa en RViz

Con la parámetros calculados, se puede hacer uso de los mismos para convertirlo en una transformación homogénea del sistema de referencia actual respecto al anterior 1. Haciendo una multiplicación de transformaciones se puede obtener la correspondiente la transformación homogénea que relaciona al sistema de referencia inicial con la del efector final, esto es la cinemática directa.

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

Para comprobar que el cálculo de los parámetros se realizaron de manera correcta, se procede a realizar la visualización del modelo en Rviz. La primera configuración que se probó fue cuando todas las articulaciones eran cero.

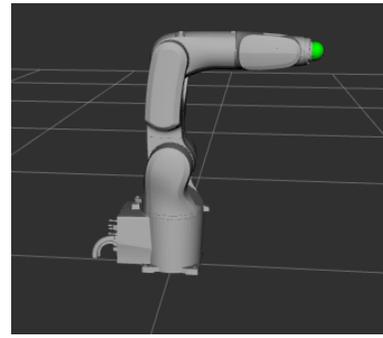


Fig. 7. Configuración inicial

Como se muestra en la figura 7, el robot está ubicado de forma similar a lo que se muestra en la hoja técnica, y la pelotita verde se ubica justo al final del efector final. Esto indica que la cinemática se realizó de manera correcta, porque la pelotita está asociada a la transformación homogénea que muestra en la figura 8.

$$\begin{bmatrix} 0. & 0. & 1. & 0.385 \\ 0. & -1. & 0. & 0. \\ 1. & 0. & 0. & 0.64 \\ 0. & 0. & 0. & 1. \end{bmatrix}$$

Fig. 8.

También se probó con otra configuración articular aleatoria para poder confirmar el correcto modelamiento de la cinemática directa. Se utilizó los valores articulares de $q = [0.9, 0.8, 0.5, -1.3, -0.4, 0.4]$. En RViz se obtuvo la siguiente imagen.

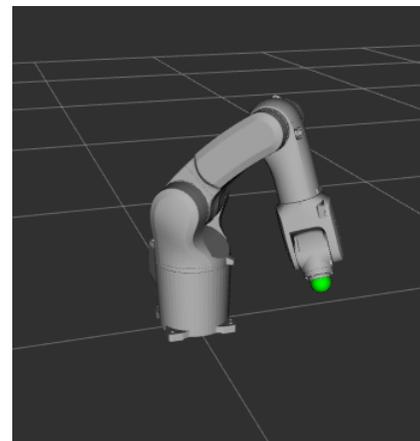


Fig. 9. Configuración articular 2

Como se puede visualizar la pelotita verde se ubica al final del efector final del robot y su transformación homogénea con respecto al sistema de referencia base se muestra en la 10. Por lo tanto, se confirma que se implementó de manera correcta la cinemática directa.

```

[[ 0.86  -0.036  0.509  0.231]
 [-0.064 -0.997  0.038  0.246]
 [ 0.507 -0.066 -0.86   0.174]
 [ 0.     0.     0.     1.   ]]

```

Fig. 10. Transformación homogénea para configuración 2

Por último, se unieron las dos partes del robot, la plataforma omnidireccional con el brazo manipulador de 6 grados de libertad y se probó la cinemática directa, esto funcionó correctamente. Por ello, la correcta visualización de la pelotita verde se muestra a continuación en la figura VI-B.



Fig. 11. Robot omnidireccional con brazo manipulador

C. Método de Newton

Para hallar la cinemática inversa, existen diferentes métodos. El utilizado en este proyecto fue un método numérico llamado de Newton. Este es una manera recursiva de hallar el valor correspondiente a las articulaciones teniendo como dato al jacobiano, una posición articular inicial y una posición del efector final deseada. Se utiliza la siguiente expresión para hallar las posiciones articulares hasta que el error sea ínfimo.

$$q_{k+1} = J^{-1}(q_k)(x_d - f(q_k))$$

D. Cinemática inversa en Rviz

Se probó con dos valores deseados. En el primer caso el valor deseado de la posición del efecto final es $x_d = [0.024, 0.225, 0.649]$ y las posiciones articulares iniciales son $q_0 = [0.0, -1.8, 1.7, -2.2, -1.6, 0.0]$. El algoritmo converge y esto se muestra en la siguiente gráfica, pues el error tiende hacia cero.

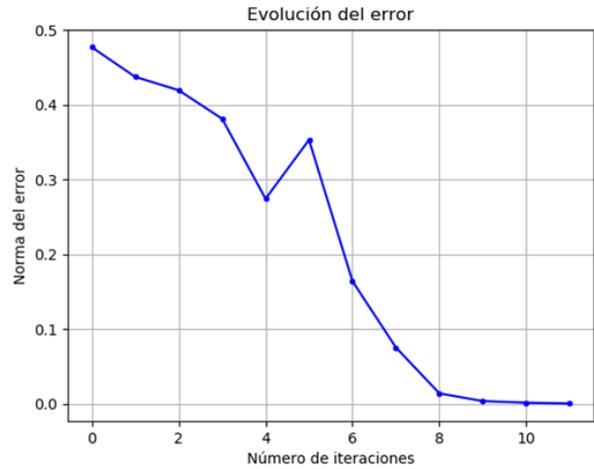


Fig. 12. Evolución de la norma del error

También se puede comprobar mediante Rviz, donde se ve como la pelotita roja que refleja posición actual del efector fina ha llegado a la posición deseada que se ve representada en la pelotita verde.

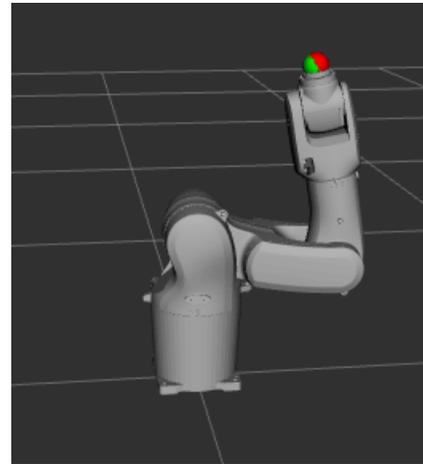


Fig. 13. Visualización en Rviz

Asimismo, se realizó con la posición deseada $x_d = [-0.03, 0.125, 0.749]$ y con valores articulares iniciales $q_0 = [0.0, -1.8, 1.7, -2.2, -1.6, 0.0]$. La evolución del error se muestra en la gráfica 14 y su comprobación por medio de Rviz en la figura VI-C.

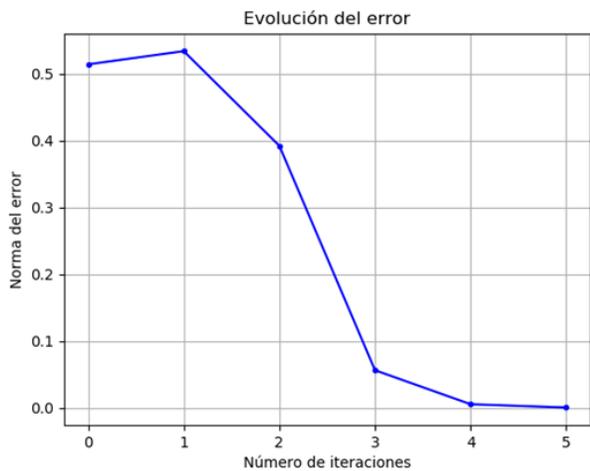


Fig. 14. Evolución del error

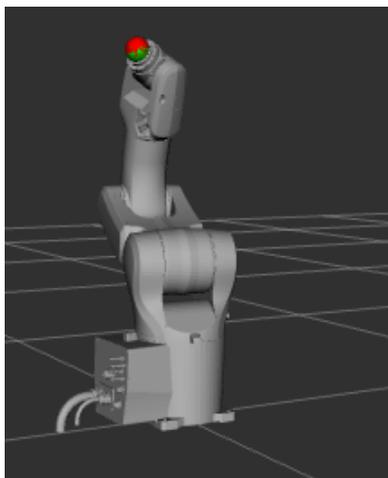


Fig. 15. Visualización en Rviz

De esta manera, podemos llegar a una conclusión parcial de que la cinemática directa e inversa para el robot Kuka KR4 Agilus se realizó de manera correcta

V. CONTROL CINEMÁTICO

A. brazo robotico

Para el caso del brazo se tomo en cuenta la ley de control presente en la siguiente ecuacion:

$$\dot{e}^* = -ke$$

Y tomando en cuenta la conversion de la posicion mediante la integracion de euler:

$$q_k = q_{k-1} + \Delta t \dot{q}_k$$

Y que la derivada de la posicion articular tiene la forma:

$$\dot{q} = J^\# \dot{e}$$

Donde $J^\#$ es el pseudo inversa de moore-Pensore que tiene la forma de :

$$J^\# = (J^T J)^{-1} J^T$$

De esta manera, implementando en python lo previamente mencionado se obtuvieron los resultados de la figura 17 y 18. Cabe resaltar que el punto que se quiere alcanzar es $x_d = [0.4, 0.4, 0.4]$

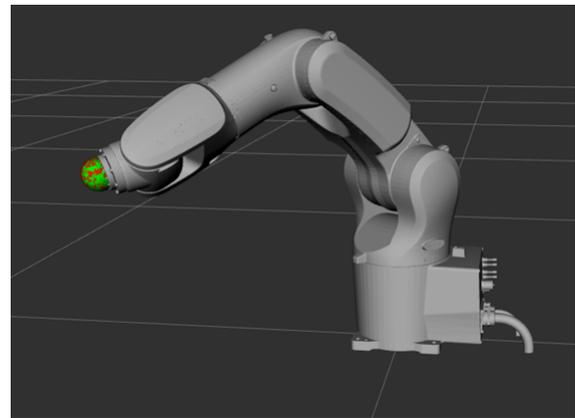


Fig. 16. Vizualizacion del control cinematico de la posicion

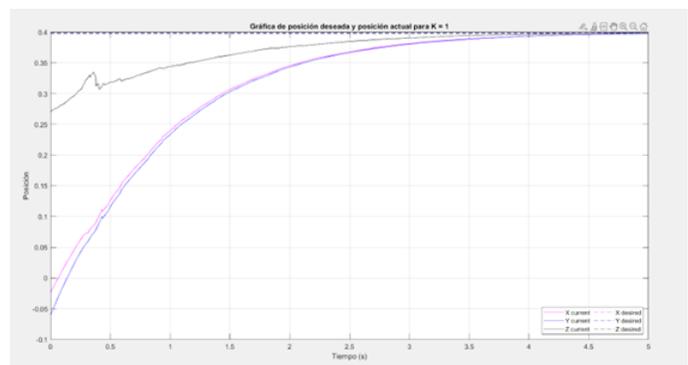


Fig. 17. Posicion del efector final

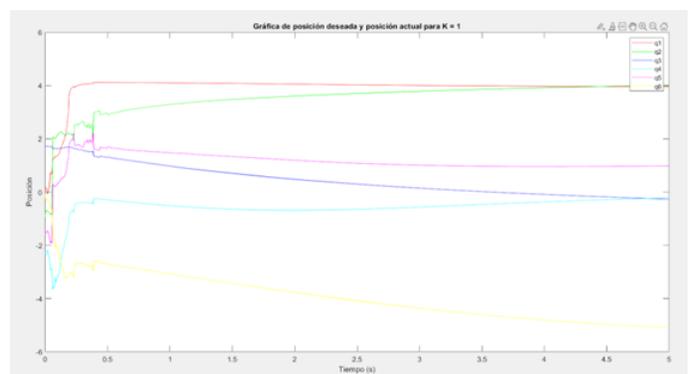


Fig. 18. Evolucion de las posiciones articulares

Asimismo, para las singularidades se utilizara la matriz Jacobiana presente en la siguiente figura 19 dados sus

elementos simbolicos.

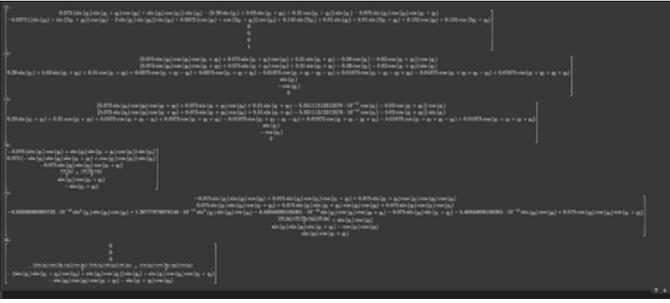


Fig. 19. Matriz jacobiana

Cabe resaltar que el procedimiento para hallar la matriz jacobiana se encuentra en la referencias.

De esta manera, teniendo en cuenta eliminar los elementos más pequeños y utilizando matlab en lugar de python se tendra la determinante.

$$\det(J) = (29\sin(q4)965\cos(q1))/2 + 965 * \cos(q1) - \arctan(124/957) + 2(q2))/2 - (29965(1/2)\cos(\arctan(31/2) + (q1) + (q2)))/2 + (29 965(1/2)\cos(\arctan(31/2) - (q1) + (q2)))/2)/1000000 \quad (2)$$

Tomando en cuenta que para evitar singularidades esta matriz debe ser diferente de 0 algunas de las configuraciones que haría tendría singularidad sería la siguiente.

$$q4! = 0 - \pi \quad (3)$$

Teniendo en cuenta que la ecuacion 2 y 3 se deberían cumplir a la vez

$$q1! = 0 - \pi/2$$

$$q2! = 0 - -\pi/2$$

B. Robot omnidireccional

Para el caso de la base móvil omnidireccional, se utilizó un plugin en gazebo *planar_move_plugin* para su simulación y obtener los datos de odometría del robot, para mediante el controlador del plugin, enviar datos de entrada para la velocidad lineal y angular del robot. Asimismo, estos datos son publicados en los tópicos `\odom` (para la posición del robot respecto al sistema inercial) y `\cmd_vel` para la velocidad lineal y angular del mismo. De este modo, se presenta el diagrama de tópicos y nodos obtenido mediante el comando *rqt_graph*:

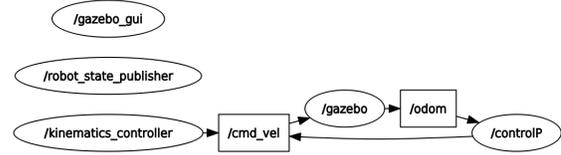


Fig. 20. Diagrama de los nodos y tópicos de rqt_graph

En la imagen 20, se observa que el nodo *controlP* está suscrito al tópico *odom*, y publica al tópico *cmd_vel*, la cual el nodo de *gazebo* está suscrita a esta. Es mediante el nodo de *controlP* el cual se implementa el control proporcional de posición en el plano cartesiano. Para ello, se usa la siguiente ley de control:

$$\dot{x} = K_p(x_d - x)$$

donde \dot{x} es la derivada temporal de la posición y orientación. De manera que, expresado de forma matricial, obtenemos:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} K_p & 0 & 0 \\ 0 & K_p & 0 \\ 0 & 0 & K_p \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix}$$

La relación entre las velocidades en el sistema de referencia inercial (\dot{x}), y las velocidades lineal (v) y angular w en el sistema del robot pueden ser representadas como:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = S \begin{bmatrix} v_x \\ v_y \\ w \end{bmatrix}$$

$$S = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

La matriz S permite relacionar las velocidades respecto al sistema inercial de referencia respecto a el sistema del robot móvil. Asimismo, usando las expresiones mostradas es posible expresar las velocidades lineales y angulares en función del error:

$$\begin{bmatrix} v \\ w \end{bmatrix} = S^\# K_p(x_d - x)$$

Al implementar el control en el robot móvil omnidireccional en gazebo, se obtiene la siguiente trayectoria y se observa que para una referencia de posición de (3,2), el robot llega a su posición y orientación deseada:

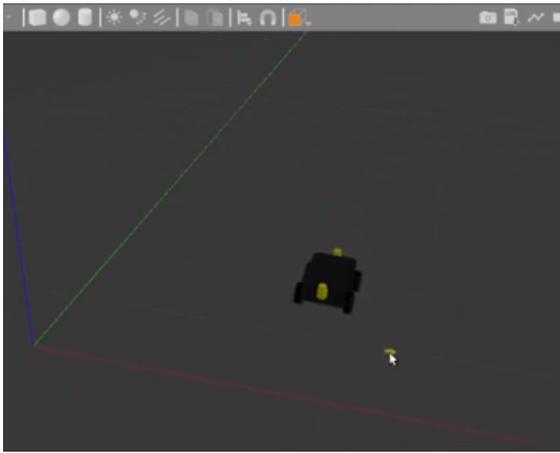


Fig. 21. Control de posición cartesiana de plataforma omnidireccional en gazebo

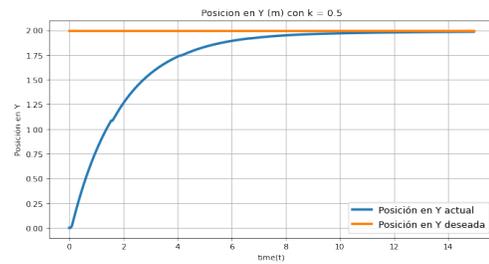


Fig. 24. Posición en y

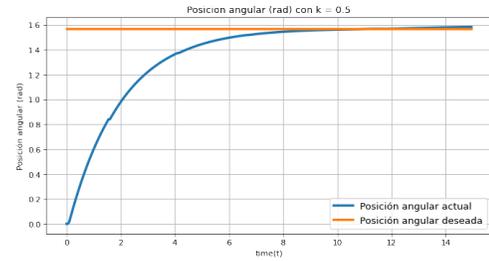


Fig. 25. Posición en z

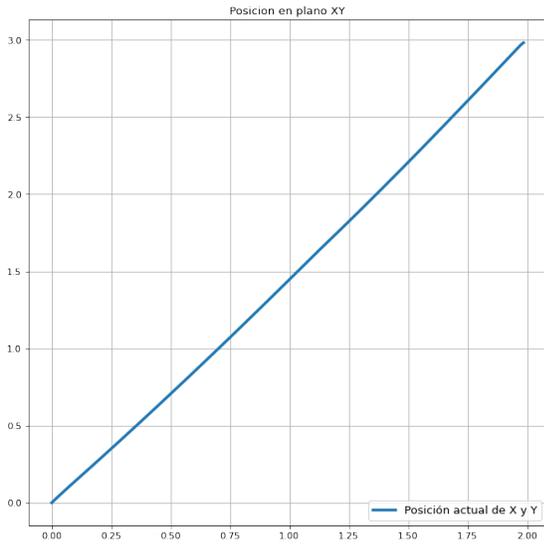


Fig. 22. Trayectoria en x e y de la plataforma omnidireccional

C. Control Cinemático de Brazo Robótico y Base Móvil

Integrando ambos controles presentados previamente, se puede hacer un control por cinemática inversa de un punto en el espacio fuera del workspace original del brazo manipulador para que junto a una base móvil, este pueda llegar a acercarse al punto deseado hasta estar dentro de su espacio de trabajo operacional. Para la implementación de este, se probó con una posición de $(-1,-1, 0.45)$.

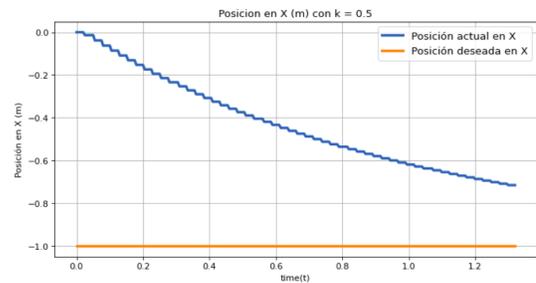


Fig. 26. Control en x de plataforma omnidireccional

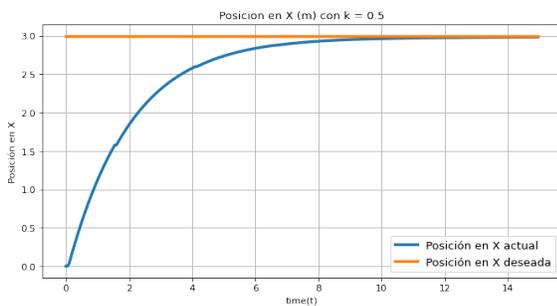


Fig. 23. Posición en x

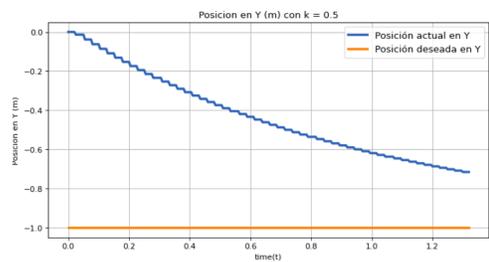


Fig. 27. Posición en y de plataforma omnidireccional

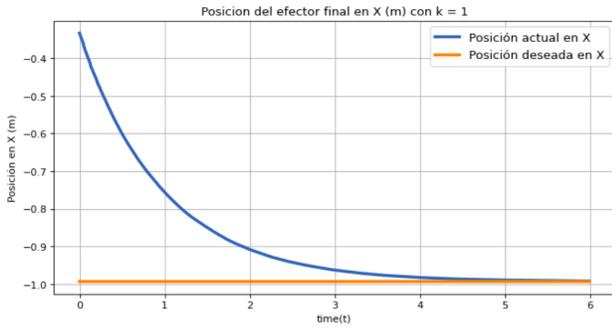


Fig. 28. Control en x del brazo manipulador

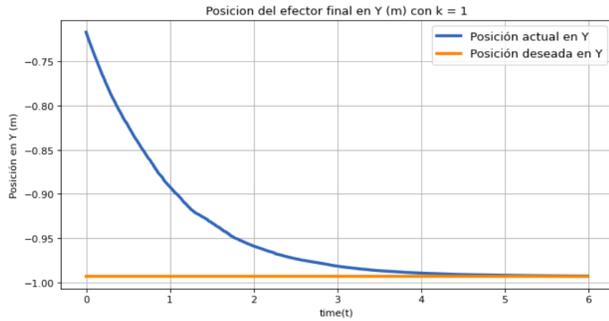


Fig. 29. Control en y del brazo manipulador

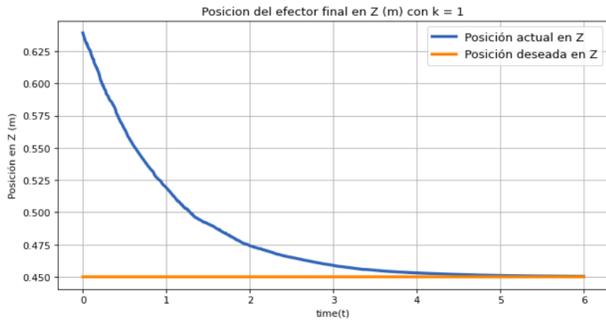


Fig. 30. Control en z del brazo manipulador

Se puede observar en las gráficas obtenidas que la base móvil no llega a la referencia el cual es en la coordenada (-1,-1), puesto que cuando la norma del error de posición llega a ser menor a 0.65 (límite de espacio operacional del robot manipulador), la base móvil se detiene para que el brazo manipulador empiece con el control cinemático. Es por ello que también se observa que las coordenadas del efector final en las gráficas de su posición en X, Y y Z, no empiezan en la posición de su configuración inicial, sino en la posición en la cual la base móvil llegó. Asimismo, llegan satisfactoriamente al punto deseado.

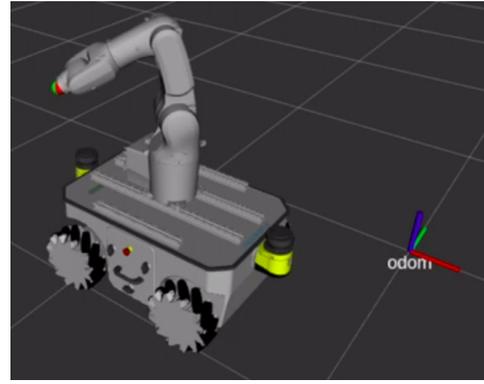


Fig. 31. Visualización de la cinemática inversa del brazo robótico con una posición fuera de su espacio de trabajo en Rviz

VI. DINÁMICA Y CONTROL DINÁMICO

La dinámica de un robot manipulador se modela mediante la siguiente ecuación. Esta está compuesta de 3 componentes: el primero correspondiente a la matriz de inercia, el segundo relacionado con la matriz de coriolis y centrífuga y el tercer elemento el vector de gravedad.

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = u$$

A. Modelo dinámico con RBDL

Para hallar los términos de la expresión anterior existen diferentes formas, para este caso se hizo uso de la librería “Rigid Body Dynamics Library” que en función de q, \dot{q}, \ddot{q} se obtiene el vector de torque y fuerzas u .

Dada una posición articular q , su derivada \dot{q} y su segunda derivada \ddot{q} como se muestra a continuación. Se despeja el vector de gravedad utilizando la función de la librería RBDL.

$$\ddot{q} = \dot{q} = 0$$

$$M(q)\ddot{q} + c(\dot{q}, q)\dot{q} + g(p) = \text{InverseDynamics}(q, \dot{q}, \ddot{q})$$

$$M(q) * 0 + c(\dot{q}, q) * 0 + g(p) = \text{InverseDynamics}(q, 0, 0)$$

$$g(p) = \text{InverseDynamics}(q, 0, 0) = \tau$$

Luego se asume que la aceleración articular es cero y como ya se conoce el vector de gravedad, se determina el vector de fuerzas centrífugas y coriolis.

$$\ddot{q} = 0$$

$$M(q)\ddot{q} + c(\dot{q}, q)\dot{q} + g(p) = \text{InverseDynamics}(q, \dot{q}, \ddot{q})$$

$$M(q) * 0 + c(\dot{q}, q)\dot{q} + g(p) = \text{InverseDynamics}(q, \dot{q}, 0)$$

$$c(\dot{q}, q)\dot{q} = \text{InverseDynamics}(q, \dot{q}, 0) - g(p)$$

Por último, procede a hallar los términos de la matriz de inercia realizando iteraciones.

$$\ddot{q} = e_i, \quad \dot{q} = 0$$

$$M(q)\ddot{q} + c(\dot{q}, q)\dot{q} + g(p) = \text{InverseDynamics}(q, \dot{q}, \ddot{q})$$

$$M(q) * e_i + c(\dot{q}, q) * 0 + g(p) = \text{InverseDynamics}(q, 0, e_i)$$

$$m_i(q) = \text{InverseDynamics}(q, 0, e_i) - g(p)$$

De esta manera, dada las posiciones articular con sus derivas para un momento especifica se puede determinar las fuerzas y torques resultantes del mecanismo.

B. Control por dinámico inversa en el espacio articular

El control dinámico en el espacio articular necesita del modelamiento dinámico realizado anteriormente para establecer el control sobre cada una de las articulaciones. Como se trata de un control en el espacio articular, la posición del efector final deseado se produce en consecuencia de controlar cada articulación a una referencia. La ley de control que se utilizó se muestra a continuación.

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

Se aplicó esta ley control para el robot KUKA KR4 Agilus y los resultados de las posiciones articulares se muestran en la figura VI-C. La constante K_p utilizado fue de 0.3 para cada articulación y K_d se halló con $K_d = 2\sqrt{K_p}$ para tener un comportamiento sobreamortiguado.

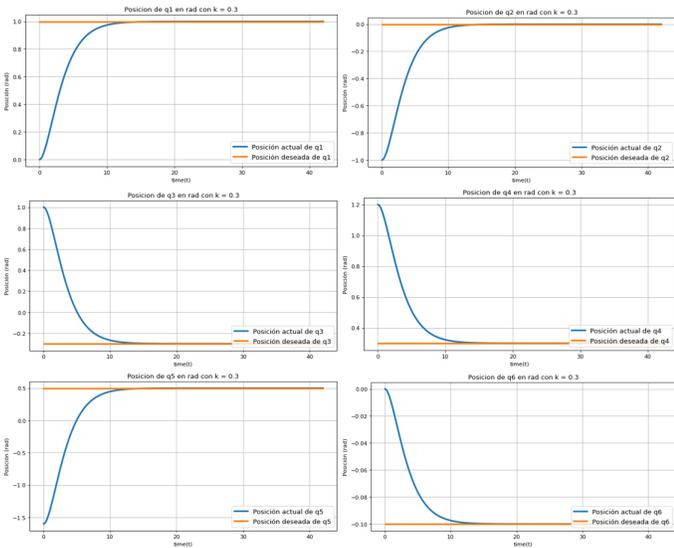


Fig. 32. Control de las articulaciones

Como se puede observar cada una de las articulaciones se alinea hacia su valor deseado. De esto podemos concluir que el control se realizó de manera adecuada.

Ahora también se evalúa el control para la posición del efector final. Un resumen de esto se muestra en la figura 33.

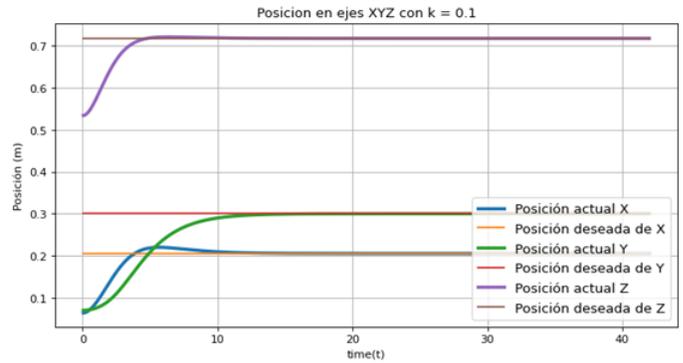


Fig. 33. Posición del efector final

Como se puede notar, el control de la posición del efector final se cumple en cada uno de los ejes x, y e z. Es por ello que se puede concluir que el control se realizó de manera correcta. El control podría ser más rápido es decir un menor tiempo de establecimiento si se aumenta el K_p , pero también podría incrementar el sobre impulso.

C. Control por dinámica inversa en el espacio operacional

Este control también necesita del modelamiento de la dinámica del robot para poder establecer una ley de control. Esta se muestra a continuación.

$$u = M(q)J_A^{-1}(\ddot{x}_d - \dot{J}_A\dot{q} + K_d\dot{e} + K_p e) + C(q, \dot{q})\dot{q} + g(q)$$

Como se puede notar depende del error y de la derivada del error por lo que se tiene que almacenar los errores pasados para obtener su derivada.

Los resultados del control aplicado en Rviz se muestra a continuación.

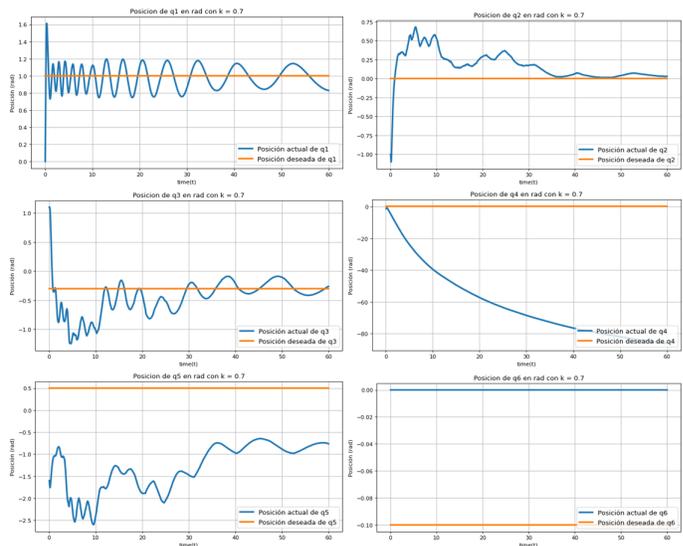


Fig. 34. Control de las articulaciones

Se puede notar que las posiciones de las articulaciones no llegan a posición deseada, esto se debe a que es un control en el espacio operacional y no en el articular, por lo que importa que el efecto final llegue a su posición deseada. También se puede notar que algunas articulaciones se quedan oscilando.

Como se puede notar en la figura 35 el efector final si llega a su posición ideal en cada uno de los ejes x, y, z. Sin embargo lo realiza de manera oscilatoria, esto se debe a que solo se realiza un control de la posición más no de la orientación.

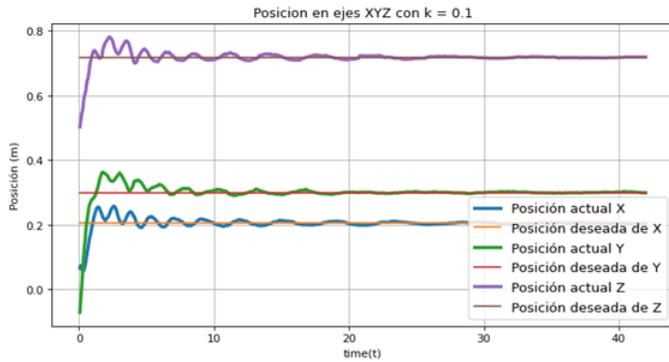


Fig. 35. Posición del efector final

Esto se puede solucionar realizando también el control de la orientación o indicando en el algoritmo de control que una vez que la norma del error sea menor a cierto valor deje de ejecutar el control y las articulaciones se queden en el estado en el que se encuentran.

VII. CONCLUSIONES Y RECOMENDACIONES

- Primero, de acuerdo a lo analizado en el control cinemático de posición, es posible colegir que los robots móviles holonómicos presentan un mejor control a comparación de los no holonómicos, debido a que poseen libertad de movimiento en el eje X e Y del sistema de referencia del robot móvil.
- Segundo, en cuanto al control dinámico, se puede notar que en el espacio articular tanto las posiciones articulares como la posición del efector final llegan a su posición deseada, mientras en el control por dinámica inversa en el espacio operacional solo llega el efector final a su posición deseada mientras articulaciones se quedan oscilando. Esto se debe a que solo se hizo un control de posición y no de orientación.
- Tercero, se puede concluir que una plataforma omnidireccional integrado a un brazo robótico permite que el robot realice tareas más específicas y con mayor precisión.
- Cuarto, cabe mencionar que un robot con ruedas mecanum son muy útiles, pero, tienen limitaciones con respecto a los terrenos o superficies que se puede recorrer, por ello, en una implementación real se recomienda probarlo en superficies regulares.
- Por último, como recomendación, se sugiere realizar una generación de trayectorias basado en el entorno para que

la plataforma omnidireccional pueda realizar acciones más “inteligentes”.

VIII. ANEXOS

El código utilizado se subió a un repositorio en GitHub. El link de acceso es el siguiente: https://github.com/manul30/fdr_manipulator_movil_robot.

En este se puede notar 6 carpetas. En la primera denominada “cinematica” se encuentra el modelamiento tanto de la cinemática directa e inversa. En la segunda carpeta, “control_cinematico” se realizó el modelo cinemático diferencial y el control cinemático del robot. En la tercera carpeta, “control_dinamico” se ubica los dos controles aplicados al brazo robótico mencionadas en la sección VI. Por último, las últimas tres carpetas contienen información relacionada al modelamiento tanto del brazo manipulador como del chasis omnidireccional en URDF y Gazebo. Asimismo, el desarrollo de la matriz jacobiana se encuentra en el siguiente enlace: https://colab.research.google.com/drive/1Dsg7hCRUDb_NAdd-jc3W0AYWZi4_VIot?usp=sharing

REFERENCES

- [1] Áviles, O., Mauledoux, M., y Rubiano, O. (2018). Electronic Architecture for a Mobile Manipulator. VOL. 14 NO. 02. <https://doi.org/10.3991/ijoe.v14i02.7672>
- [2] Neobotix MPO-500. Data sheet of the MPO-500. <https://www.neobotix-robots.com/products/mobile-robots/mobile-robot-mpo-500>
- [3] Kuka KR 4 AGILUS. Data sheet of KR 4 AGILUS. <https://www.kuka.com/es-es/productos-servicios/sistemas-de-robot/robot-industrial/kr-4-agilus>