

CONTROL Y SISTEMAS

“CONTROL CINEMÁTICO ROBOT MÓVIL OMNIDIRECCIONAL”



NOMBRE: GABRIEL EZEQUIEL GODOY

PROFESOR: Dr. Ing. RODRIGO GONZALEZ

AÑO: 2018



RESUMEN

Para el presente informe se desarrolla el modelado y simulación de un robot móvil omnidireccional. Esto se lleva a cabo mediante la utilización de la herramienta informática Matlab, más concretamente los subprogramas Simulink y Simscape.

En primer lugar se describen las características y las configuraciones que pueden adquirir este tipo de robot móvil. Luego se expone la cinemática correspondiente al tipo de configuración seleccionada para este proyecto. Dicho modelo cinemático se implementa en simulink para su comprobación y luego se aplica un tipo de control PID para poder controlar su posición.

Una vez concretado lo anterior se desarrolla una manera de visualización de un modelo CAD exportado desde Solidworks para poder implementarlo en Simscape y de esta manera observar los movimientos realizados por este dispositivo.

Los robots móviles han tenido un gran desarrollo en los últimos años en diferentes entornos, desde los industriales hasta los académicos. La construcción de un prototipo de plataforma móvil actualmente es totalmente viable gracias al abaratamiento de los costos de diferentes piezas incluyendo las mecánicas (ruedas, plataforma y soportes), hardware (placas controladoras, drivers, sensores, motores, etc) y software (programas de diseño y simulación). Por esto es que hoy en día hay un campo de estudio importante alrededor de estos dispositivos debido a la gran cantidad de aplicaciones que se le encuentran para distintos ambientes, como industrial, transporte y exploración.

Dentro de la gama de robots móviles se encuentran los denominados omnidireccionales. Un ejemplo de un dispositivo de esta clase sumamente desarrollado, es el caso del robot móvil “OmniMove” fabricado por la empresa Kuka, que consiste en una plataforma de transporte de cargas pesadas. También ha insertado en el mercado una plataforma de trabajo levadiza, “Kuka Triple Lift” y un sistema robótico complejo “KuKa Quantec”, constituido por un brazo robotico sobre una plataforma móvil omnidireccional.



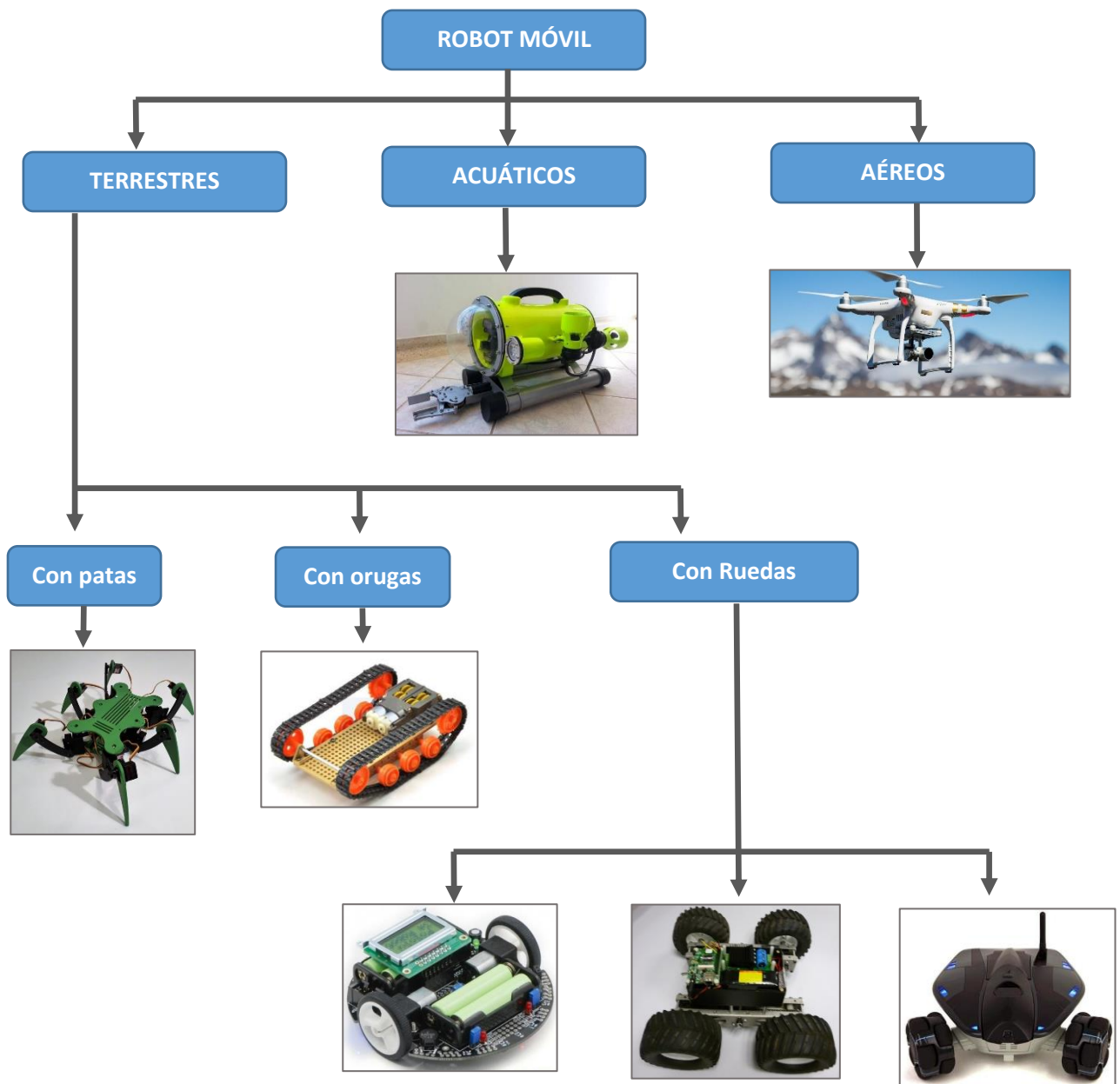
Fig. 1: Diferentes modelos de plataformas omnidireccionales de la empresa KUKA

ROBOT MÓVIL.

Un robot móvil puede clasificarse de diversas maneras, una es según el ambiente de trabajo donde se pueden clasificar en terrestres, acuáticos o aéreos. Dentro de los nombrados como terrestres se pueden agrupar según su tipo de locomoción en robot con patas, con ruedas o con orugas.

Para los robots que tienen ruedas dependiendo de los tipos de ruedas y de la disposición que adquieran las mismas existen diferentes tipos y según su locomoción existen:

- Diferencial
- Síncrona
- Triciclo
- Ackerman
- Omnidireccionales



Dentro de esta gama de robots móviles con locomoción a través de ruedas existen los denominados robots móviles omnidireccionales. Se nombran de esta manera debido a la capacidad que tienen de poder desplazarse en el plano en cualquier dirección y orientación.

Esta capacidad la brinda el tipo de rueda que utilizan y su disposición en la plataforma. Actualmente existen 2 tipos de ruedas no convencionales que pueden ser utilizadas en estos tipos de robot, las ruedas con rodillos y las ruedas suecas o “mecanum”. La diferencia entre ambas es la posición que adquieren los rodillos. Las mismas se muestran en la siguiente figura:

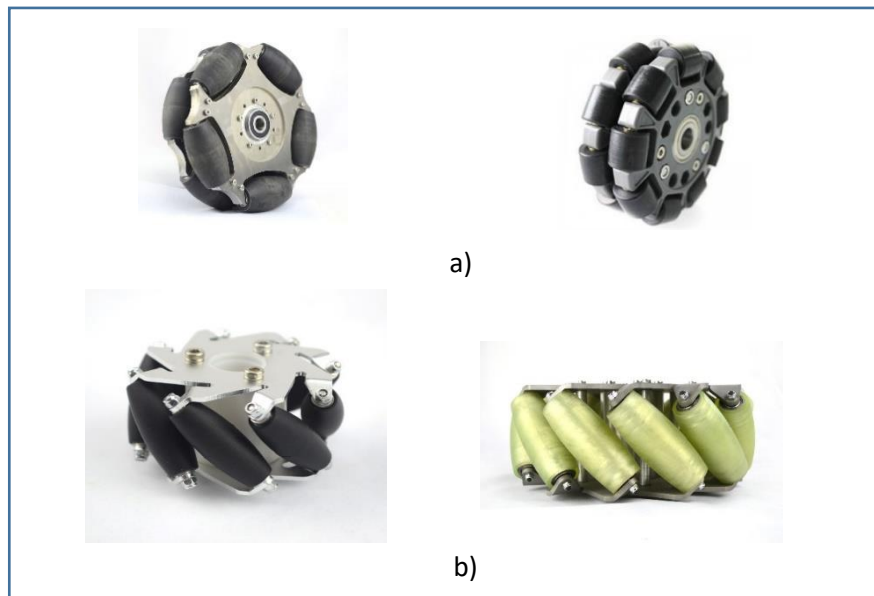


Fig. 2: a) ruedas con rodillos. b) ruedas mecanum

Para conseguir movimientos en todas las direcciones es necesario ubicar las ruedas de diferentes maneras, pueden ser de 3 o 4 ruedas.

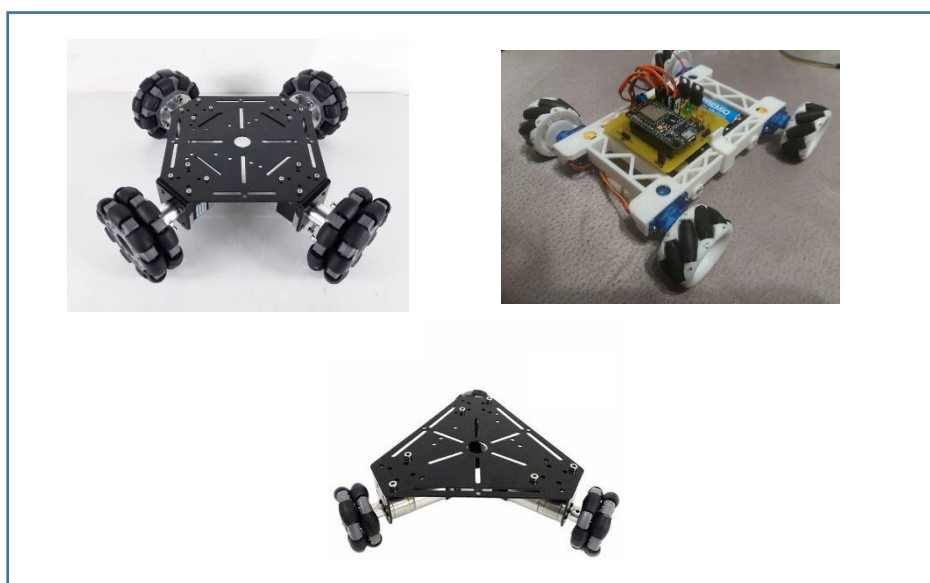


Fig 3: Distintas configuraciones con ruedas no convencionales.

Para el presente proyecto se selecciona una plataforma con 4 ruedas omnidireccionales del tipo mecanum, cuyos rodillos se encuentran a $+45^\circ$ y -45° . Dichas ruedas tienen que ir ubicadas de manera que al combinarse el giro de cada una pueda obtenerse una resultante de velocidad deseada del móvil. En la siguiente figura se muestra la disposición que se adopta:

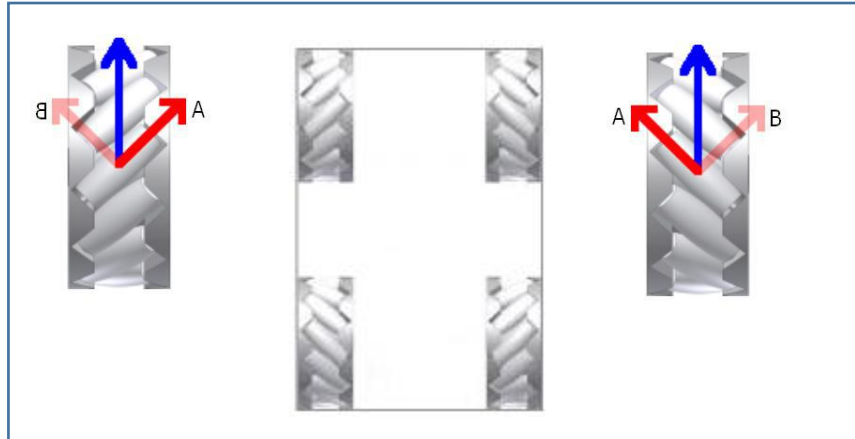


Fig.4 : Posicion de las ruedas en plataforma

Dado este tipo de configuración y al poder mover cada una de las ruedas de manera independiente es posible lograr que la plataforma pueda desplazarse en distintas direcciones. La flecha de color verde indicaría el sentido de la resultante que representaría hacia donde se dirige el móvil:

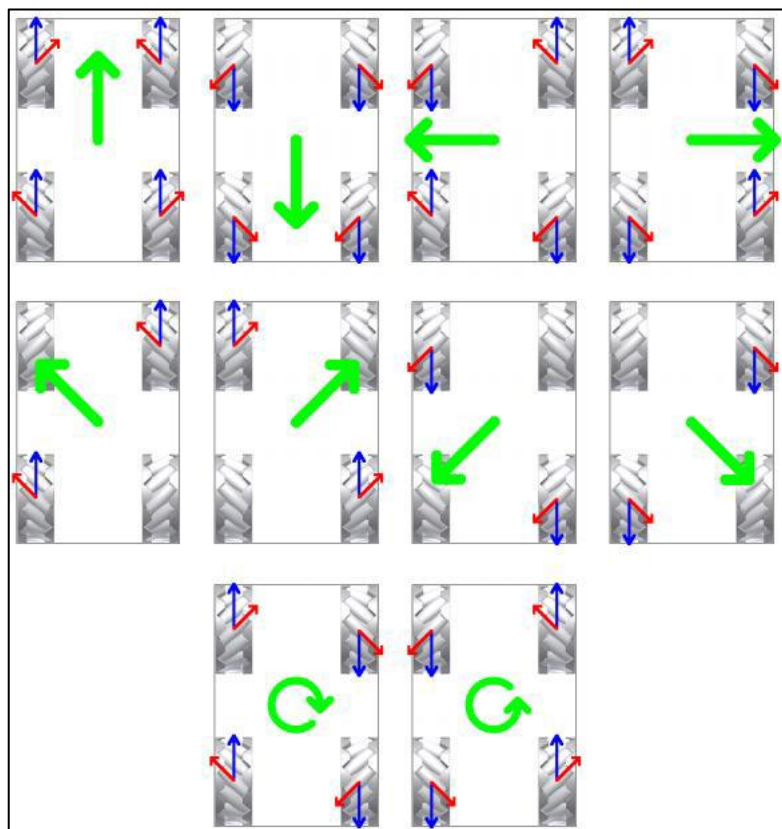


Fig. 5: Movimiento de la plataforma, según su la velocidad de las ruedas.

En esta sección se describe el modelo cinemático del robot omnidireccional con ruedas mecanum. La cinemática de un robot móvil describe la posición y orientación del mismo en función de las variables del sistema. Además se realizan las siguientes consideraciones:

- El robot se mueve en una superficie plana
- Los ejes de guiado son perpendiculares al suelo
- No hay deslizamiento
- Robot como sólido rígido, no se deforma
- Las trayectorias se pueden aproximar como arcos de circunferencia entre dos periodos de muestreo consecutivos

Las ruedas giran con una velocidad angular $\dot{\varphi}_i$ con $i=1,2,3,4$. El radio de la rueda es R , este parámetro incluye tanto el radio de la llanta con el radio de los rodillos. Como se indica en la siguiente figura el ángulo de rotación de la plataforma alrededor del punto característico S . El ancho de la plataforma se toma como $2S_y$, y la distancia entre el punto S y el punto medio del eje delantero y trasero es S_x

Todo el siguiente desarrollo de las siguientes ecuaciones se realiza con respecto a un sistema de referencia ubicado en el centro de masa de la plataforma x_p, y_p, z_p .

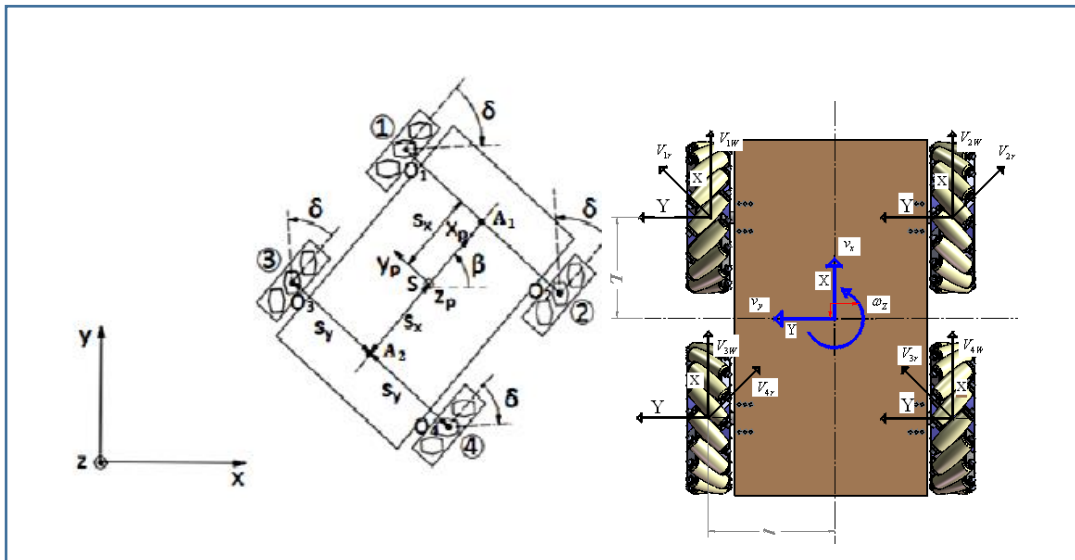


Fig.6 : Sistema de coordenadas en la plataforma

Definiendo como

V_{ir} : velocidad de los rodillos

V_{iw} : velocidad tangencial de la rueda mecanum (siempre en dirección de X)

La velocidad lineal en las coordenadas x_p e y_p (o como indica el gráfico de la derecha en $X Y$) de cada rueda es:

$$V_{ixp} = V_{iw} + \left(\frac{V_{ir}}{\sqrt{2}} \right) \quad ; \quad V_{iyp} = \left(\frac{V_{ir}}{\sqrt{2}} \right)$$



$$V_{ixp} = v_{sx_p} - \dot{\beta} S_y$$

$$V_{iyp} = v_{sy_p} + \dot{\beta} S_x$$

Desarrollando las ecuaciones anteriores y considerando que $V_{iw} = \dot{\phi}_i R$, se obtienen las siguientes relaciones:

$$\dot{\phi}_1 = \frac{1}{R} [v_{sx_p} - v_{sy_p} - \dot{\beta}(S_x + S_y)] \quad (1)$$

$$\dot{\phi}_2 = \frac{1}{R} [v_{sx_p} + v_{sy_p} + \dot{\beta}(S_x + S_y)] \quad (2)$$

$$\dot{\phi}_3 = \frac{1}{R} [v_{sx_p} + v_{sy_p} - \dot{\beta}(S_x + S_y)] \quad (3)$$

$$\dot{\phi}_4 = \frac{1}{R} [v_{sx_p} - v_{sy_p} + \dot{\beta}(S_x + S_y)] \quad (4)$$

Las ecuaciones anteriores pueden ser representadas de manera matricial de la siguiente manera:

$$\dot{\phi} = J V_{sp}$$

Donde

$$\dot{\phi} = \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix} ; \quad J = \begin{bmatrix} \frac{1}{R} & -\frac{1}{R} & -\frac{(S_x+S_y)}{R} \\ \frac{1}{R} & \frac{1}{R} & \frac{(S_x+S_y)}{R} \\ \frac{1}{R} & \frac{1}{R} & -\frac{(S_x+S_y)}{R} \\ \frac{1}{R} & -\frac{1}{R} & \frac{(S_x+S_y)}{R} \end{bmatrix} ; \quad V_{sp} = \begin{bmatrix} v_{sx_p} \\ v_{sy_p} \\ \dot{\beta} \end{bmatrix}$$

De esta manera queda representado el modelo cinemático inverso del robot omnidireccional. Entonces las ecuaciones (1) (2) (3) (4) permiten obtener las velocidades angulares de cada una de las ruedas en función de las velocidades lineales de la plataforma v_{sx_p} ; v_{sy_p} y de la velocidad angular de la misma $\dot{\beta}$.

Para resolver un problema cinemático directo, debe realizarse una operación de inversión de la matriz J. La matriz J es una matriz rectangular, por lo tanto, para realizar su inversión se puede usar el Teorema de Moore-Penrose sobre inversión de matrices rectangulares. Una vez que se aplica el teorema mencionado, se obtiene la siguiente relación

$$J_{od} = (J^T J)^{-1} J^T$$

En donde queda por resultado final la siguiente matriz:

$$J_{od} = \frac{1}{4} \begin{bmatrix} R & R & R & R \\ -R & R & R & -R \\ -R & R & -R & R \\ (S_x + S_y) & (S_x + S_y) & (S_x + S_y) & (S_x + S_y) \end{bmatrix}$$

Entonces la solución del problema cinemático directo es el siguiente

$$V_{sp\dot{\phi}} = J_{od} \dot{\phi}$$

Donde las relaciones entre la velocidad lineal de la plataforma y las velocidades angulares de las ruedas:

$$v_{sx_p} = \frac{R}{4} [\dot{\phi}_1 + \dot{\phi}_2 + \dot{\phi}_3 + \dot{\phi}_4] \quad (5)$$

$$v_{sy_p} = \frac{R}{4} [-\dot{\phi}_1 + \dot{\phi}_2 + \dot{\phi}_3 - \dot{\phi}_4] \quad (6)$$

$$\dot{\beta} = \frac{R}{4(S_x + S_y)} [-\dot{\phi}_1 + \dot{\phi}_2 - \dot{\phi}_3 + \dot{\phi}_4] \quad (7)$$

De esta manera queda representada la cinemática directa del robot móvil omnidireccional, ya que las últimas ecuaciones obtenidas permiten encontrar las velocidades lineales en x_p e y_p y la angular del robot $\dot{\beta}$ a partir de los valores de las velocidades angulares de las ruedas. Desde la obtención de estas variables es posible determinar tanto la velocidad resultante como la dirección de movimiento del móvil:

$$Vr = \sqrt{v_{sx_p}^2 + v_{sy_p}^2}$$

$$\beta = \tan^{-1} \left(\frac{v_{sy_p}}{v_{sx_p}} \right)$$

Implementación en simulink del modelo cinemático

A partir de la cinemática determinada en el apartado anterior se realiza la implementación las ecuaciones en el software Matlab/Simulink. Este programa permite cargar diferentes modelos matemáticos a través de una conexión bloques con parámetros matemáticos. Se elige este tipo de programa debido a un entorno amigable que permite de una manera interactiva poder probar diferentes ecuaciones.

La siguiente figura muestra lo anteriormente mencionado con un modelo que al ingresar un valor de velocidad angular para cada rueda, se obtiene como dato de salida la velocidad lineal y angular de la plataforma. Para esto se necesitó del aporte del diseño del modelo de la plataforma móvil como la definición del radio de la rueda ($R=0.03$) y los parámetros $s_x=0.067$ y $s_y=0.0565$.

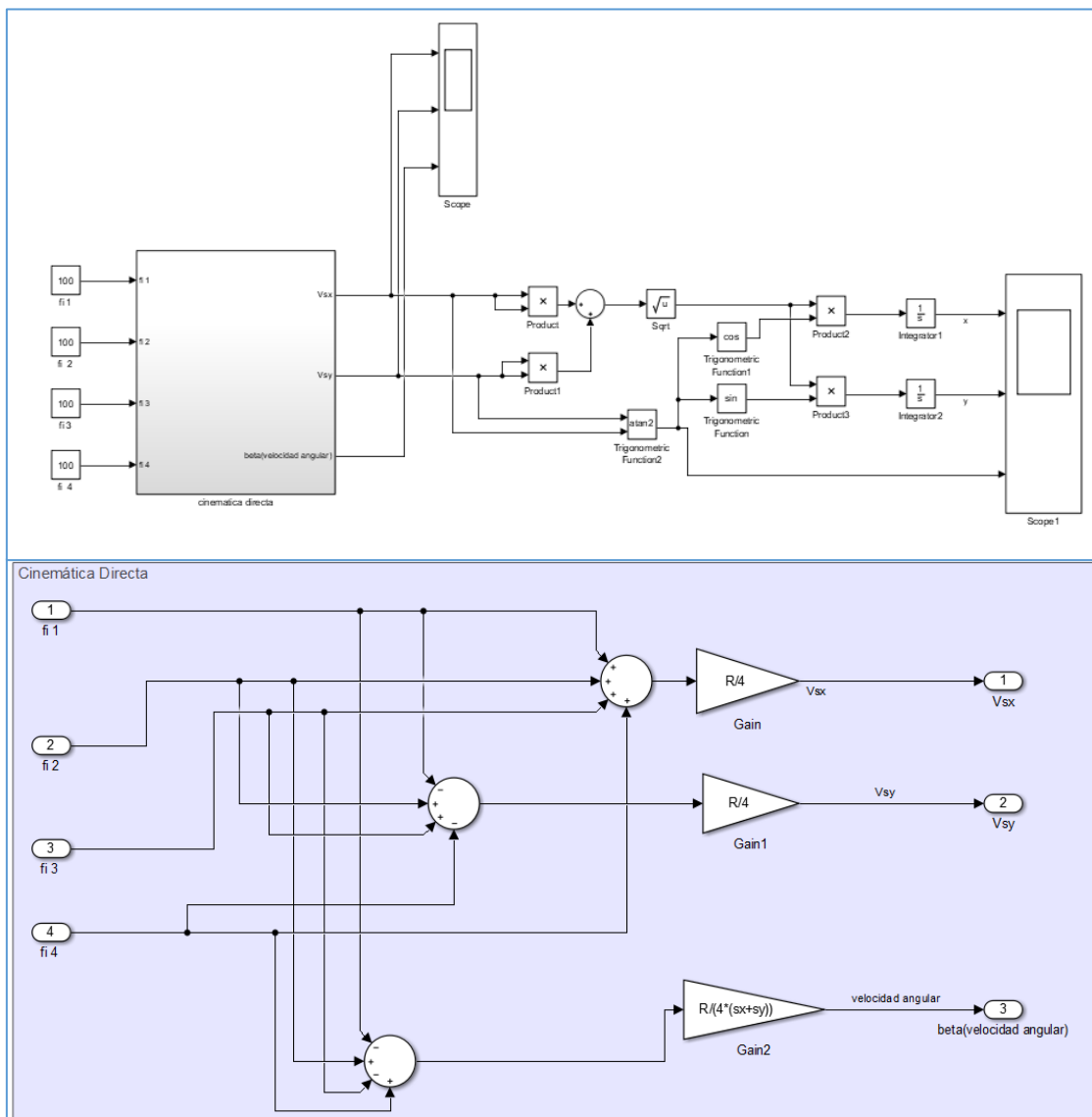


Fig.7 : Modelo cinemático implementado en simulink.

Utilizando tanto el modelo cinemático directo como el inverso de la plataforma se procede a simular el comportamiento de la misma. A demás se agrega un modelo matemático que representa el motor de corriente continua que llevaría cada una de las ruedas los cuales se controlan de manera independiente. En la figura 7 se muestra el esquema del motor, considerando el subsistema mecánico eléctrico.

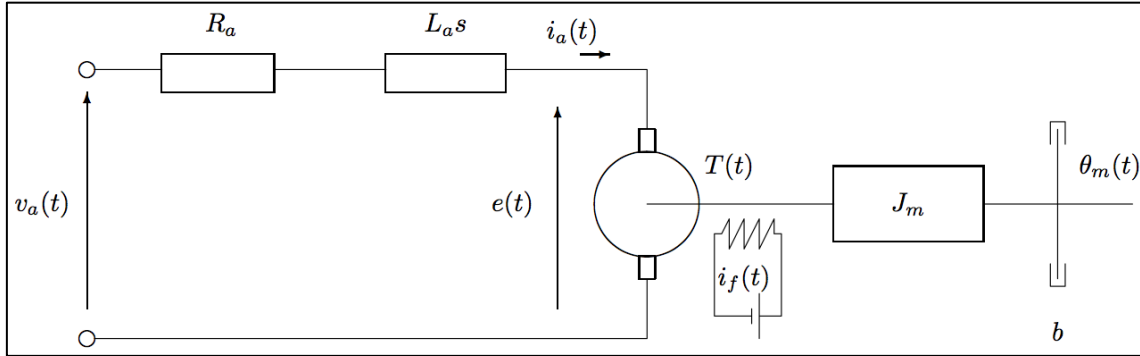


Fig.8 : Esquema de un motor de corriente continua.

Gracias a la función de transferencia se representa la relación que existe entre la señal de salida y la de entrada.

$$\frac{\theta_m(s)}{V_a(s)} = \frac{K_t}{s[(J_m s + b)(L_a s + R_a) + K_t K_e]}$$

Donde se tiene a :

θ_m : posición angular del eje del motor (señal de salida)

V_a : tensión de alimentación motor (señal de entrada)

K_t : constante de proporcionalidad entre torque y la corriente del devanado de inducido

K_e : constante de proporcionalidad la fuerza electromotriz y la velocidad de giro

J_m : inercia del motor

b : constante de fricción del motor

L_a : inductancia de la armadura

R_a : resistencia eléctrica de la armadura

Y sabiendo que

$$s \theta_m(s) = W_m(s)$$

Se puede obtener la función de transferencia que relaciona la velocidad del eje del motor con la tensión de armadura aplicada en los bornes:

$$\frac{W_m(s)}{V_a(s)} = \frac{K_t}{J_m L_a s^2 + (J_m R_a + b L_a) s + (b R_a + K_t K_e)}$$

Este modelo de motor DC se aplica en simulink con una reducción de 19:1 con el agregado de un controlador del tipo PID para el control de su velocidad. Para ajustar los parámetros de dicho controlador se utiliza la herramienta de control disponible en Matlab (PID Tuner) :

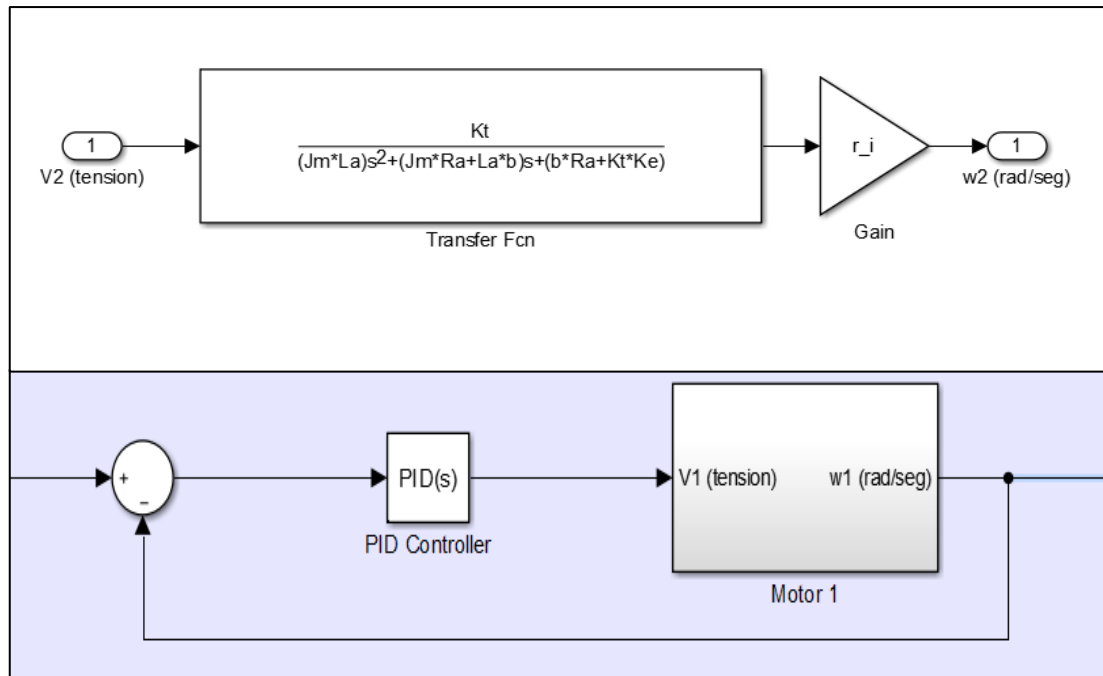


Fig.9 :implementacion modelo de motor dc con control PID de velocidad.

MODELO DE CONTROL CINEMATICO.

En esta sección se explica el desarrollo del control de posición del robot móvil. Esto se logra implementando tanto el modelo cinemático directo como el modelo inverso, en la siguiente figura se muestra un esquema ilustrativo del lazo de control a utilizar:

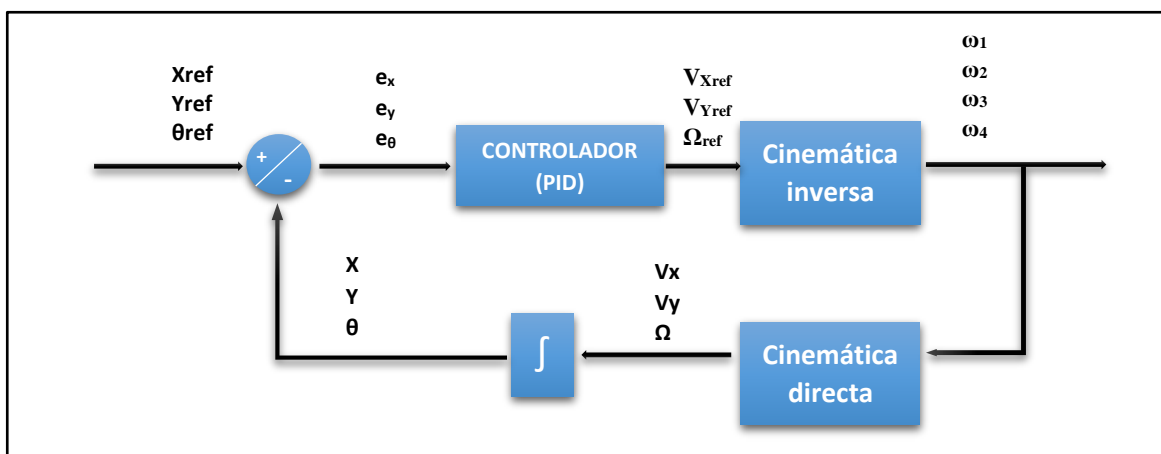


Fig.10 : Lazo de control cinemático robot móvil omnidireccional.

Explicación del lazo de control:

Se indican las consignas de referencias x_{ref} , y_{ref} , θ_{ref} que se comparan con las coordenadas de posición reales para obtener el error de entre las mismas:

$$e_x = X_{ref} - X$$

$$e_y = Y_{ref} - Y$$

$$e_\theta = \theta_{ref} - \theta$$

Este error ingresa en el controlador, para luego calcular el valor de las velocidades de referencia vx_{ref} , vy_{ref} , Ω_{ref} que ingresan al bloque de la Cinemática Inversa para obtener las velocidades de cada rueda $\omega_1, \omega_2, \omega_3, \omega_4$. Una vez obtenidas estas velocidades, se realimentan e ingresan al bloque de la cinemática directa y poder obtener las velocidades del robot actuales vx, vy, Ω . Luego se integran para encontrar las posiciones actuales X, Y, θ . Todo este lazo de control se implementa posteriormente en simulink para realizar las pruebas correspondientes.

Para el controlador se seleccionó un controlador del tipo PID con ganancias que se ajustan con la herramienta de control que posee Matlab, *PID Tunner*. Este programa permite modificar los parámetros del controlador según las características deseadas de respuestas, por ejemplo una velocidad de respuesta más o menos rápida y una respuesta más o menos robusta. En la siguiente imagen se observa la venta de opciones. Se aclara que para cada consigna lleva su propio controlador.

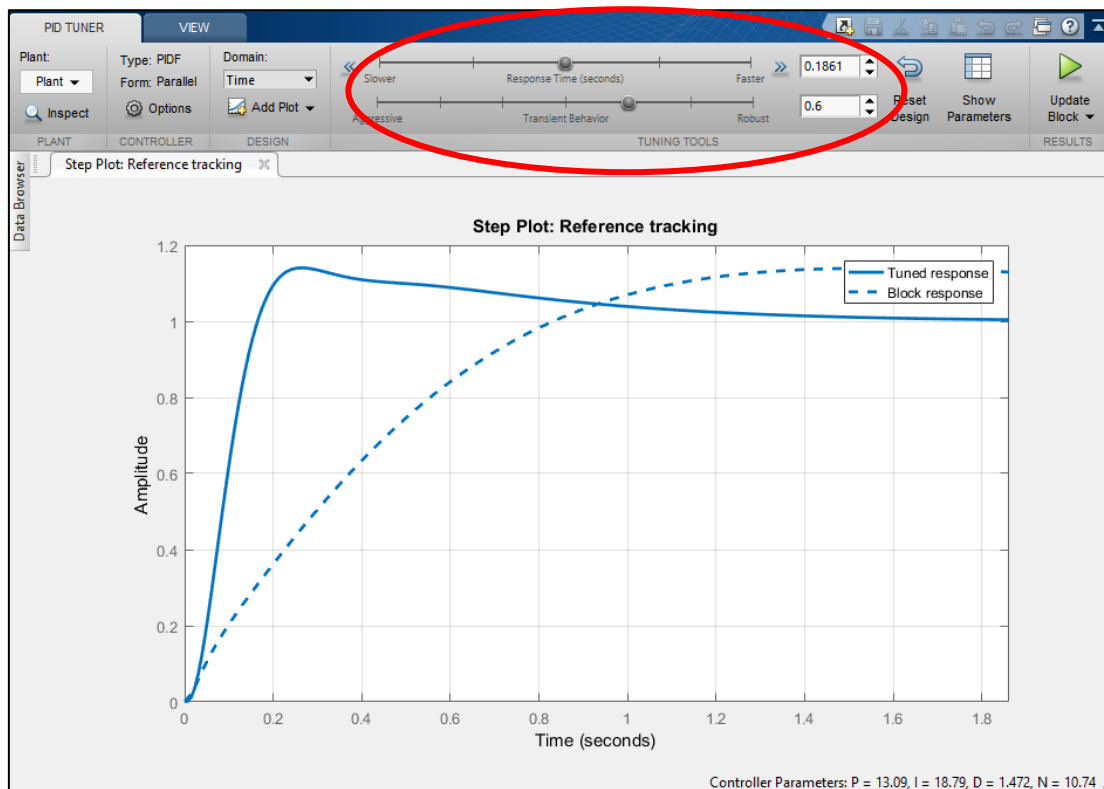


Fig.11 : PID Tunner para ajustar a una respuesta deseada.

Se realiza una prueba generando 6 puntos (x,y, θ), indicando posición y orientación. Esto se crea en un código previamente en Matlab indicando la cantidad de puntos, las respectivas posiciones y orientaciones además de la velocidad máxima deseada. En la siguiente captura de pantalla se muestra una sección de dicho código.

```

29 - cant_puntos=6;
30 - x_i=0;
31 - y_i=0;
32 - x=[x_i 1 -1 -1 1 0];
33 - y=[y_i 1 1 -1 -1 0];
34
35 - tita_i=0;
36 - tita=[tita_i 0 0 0 0 0];
37
38 %%CALCULO DISTANCIA
39 - vel_max=0.15 ;%m/s velocidad maxima del movil
40 - w_max=0.1;

```

Fig.12 : Carga de puntos deseados en Matlab.

Una vez ejecutado el código en Matlab se procede a simular en la plataforma Simulink. En la figura 12 se observan los parámetros que se ajustaron para los PID de los motores y de la plataforma completa. Luego en la figura 13 se observan resultados para las posiciones en x e y, como su orientación.

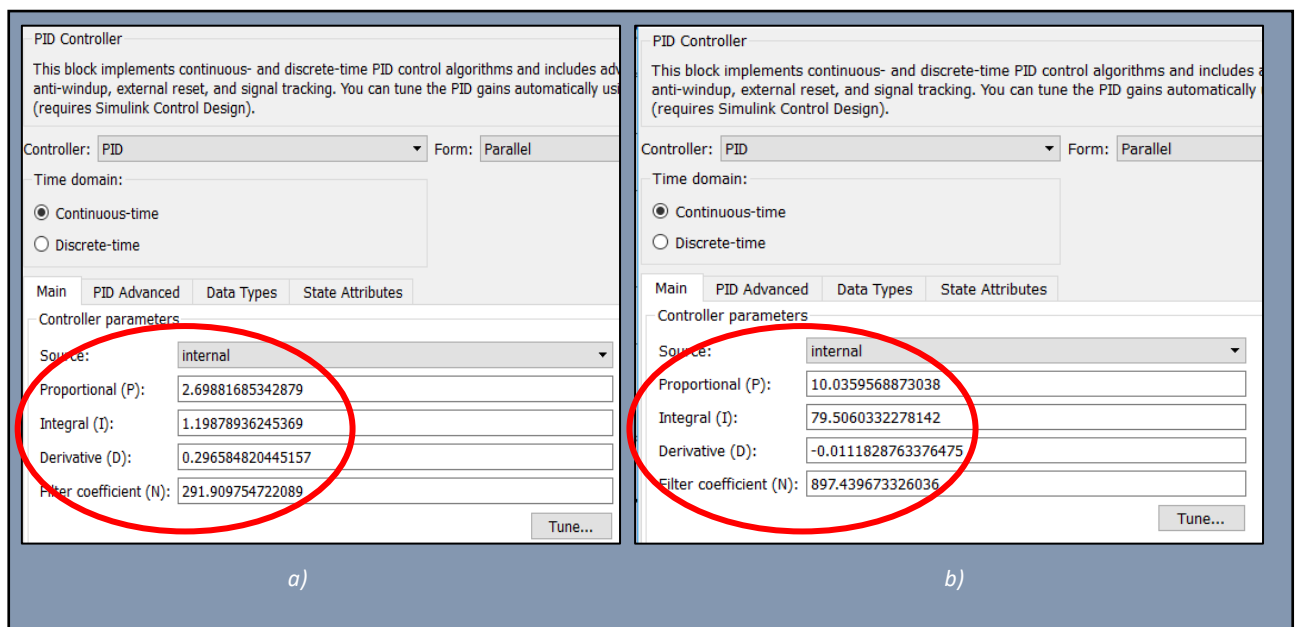


Fig.13 : a) PID posiciones b) PID motores

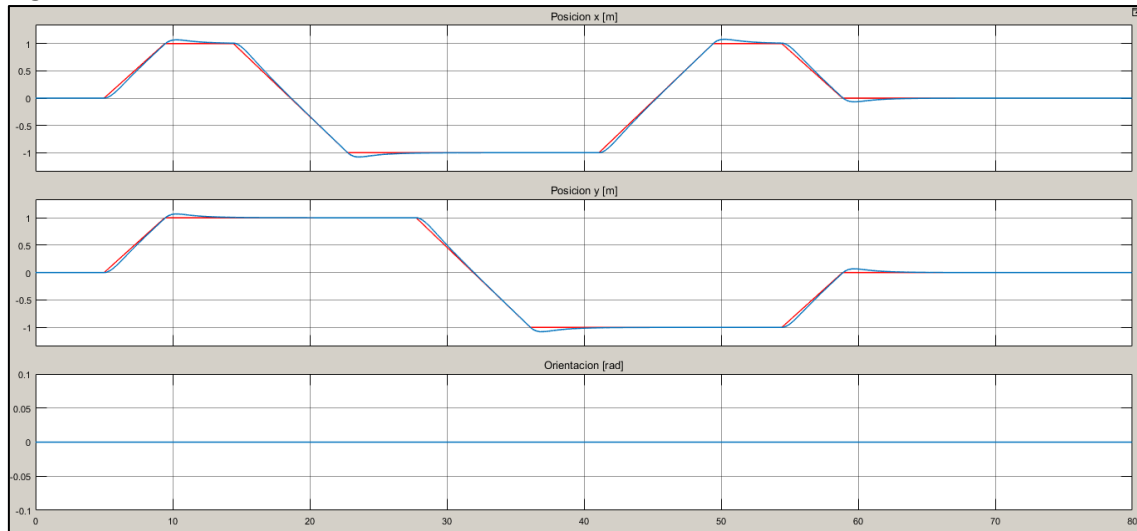


Fig.14: Seguimiento de la posicion del robot móvil.

Los gráficos anteriores muestran la consigna de posición en color rojo y la posición real en color azul. Una vez ajustados los parámetros del. Con el tiempo de simulación adecuado se puede observar como el robot móvil es capaz de seguir y posicionarse en las coordenadas de referencia. Es posible apreciar un error que al realizar un acercamiento se puede ver que es de aproximadamente 0.05 m en la posición y un error menor en la orientación (del orden de 10^{-18}) (figura 14).

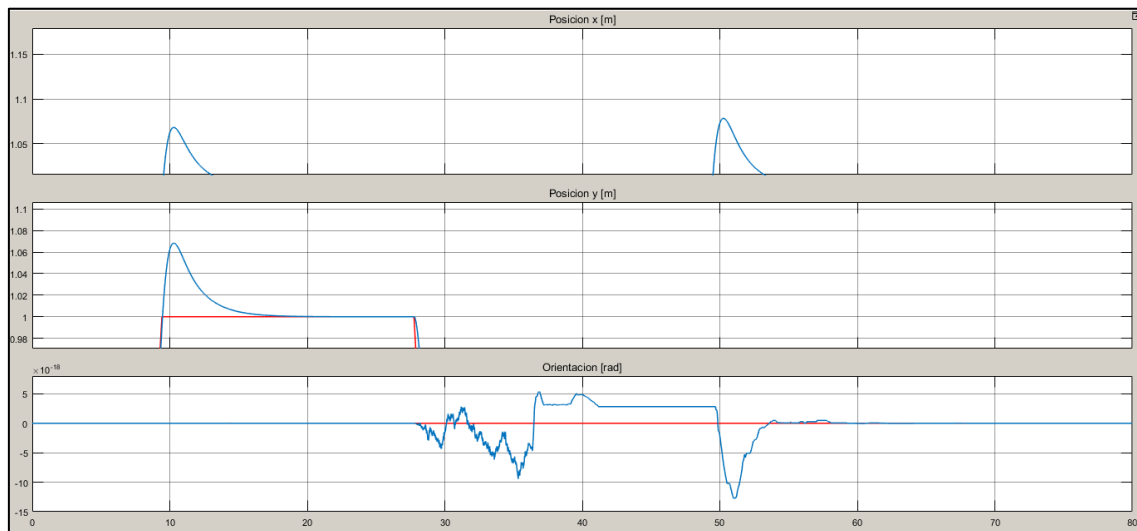


Fig.15: Error en la seguimiento de la consigna.

Tomando los datos desde simulink y exportándolos al workspace de Matlab se pueden graficar las posiciones en las que se ubica el robot en el plano, apreciando los errores de posición observados anteriormente:

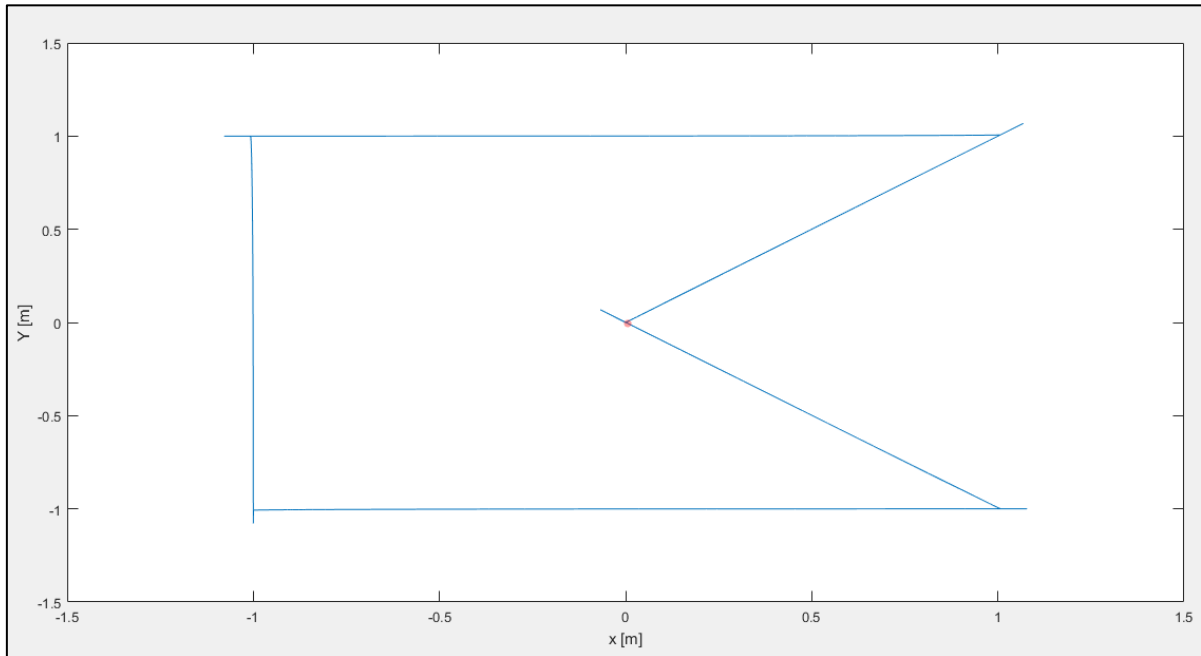


Fig.16: Recorrido del robot móvil.

A la vez se observan las velocidades que van adquiriendo los cuatro motores a lo largo del recorrido, teniendo la velocidad de consigna en rojo y la velocidad real en color azul (figura 16). A partir de esto se observa una velocidad en los motores siendo un máximo de 18 rad/seg (1600 rpm) aproximadamente:

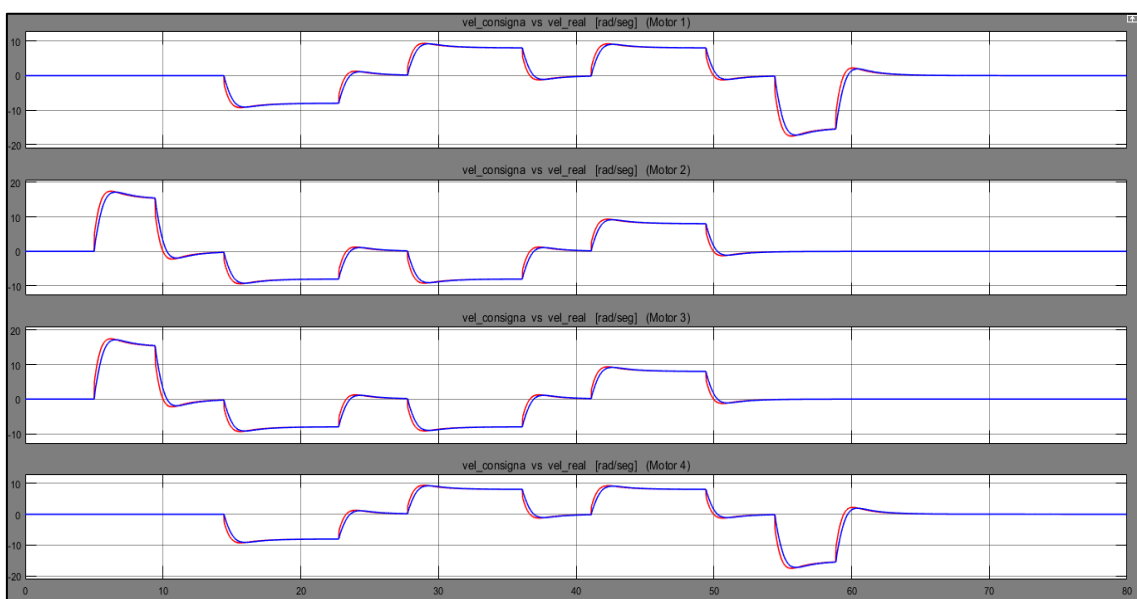


Fig.17: Velocidades de los motores.

El error de posición puede minimizarse cambiando los parámetros del controlador, es decir una velocidad de respuesta más veloz. Pero esto provocaría que las velocidades en las ruedas aumenten de manera considerable, en otras palabras se exigiría a los motores que eleven las revoluciones. Otra manera es aumentar la velocidad máxima cargada para que el móvil llegue a los puntos en menor tiempo, pero ocasionaría el mismo inconveniente con respecto a la velocidad de los motores.

Además de lo anterior se adiciona la simulación de un ruido en la lectura del sensor que mide la velocidad del motor (encoder, mide posición de eje y se luego deriva con respecto al tiempo en la realidad). La varianza que se coloca es de $1/360$ rad/s. Este error al ser un valor bajo no altera al comportamiento total del sistema. Solo se observa una muy baja variación en la orientación del robot pero este valor continua siendo bajo.

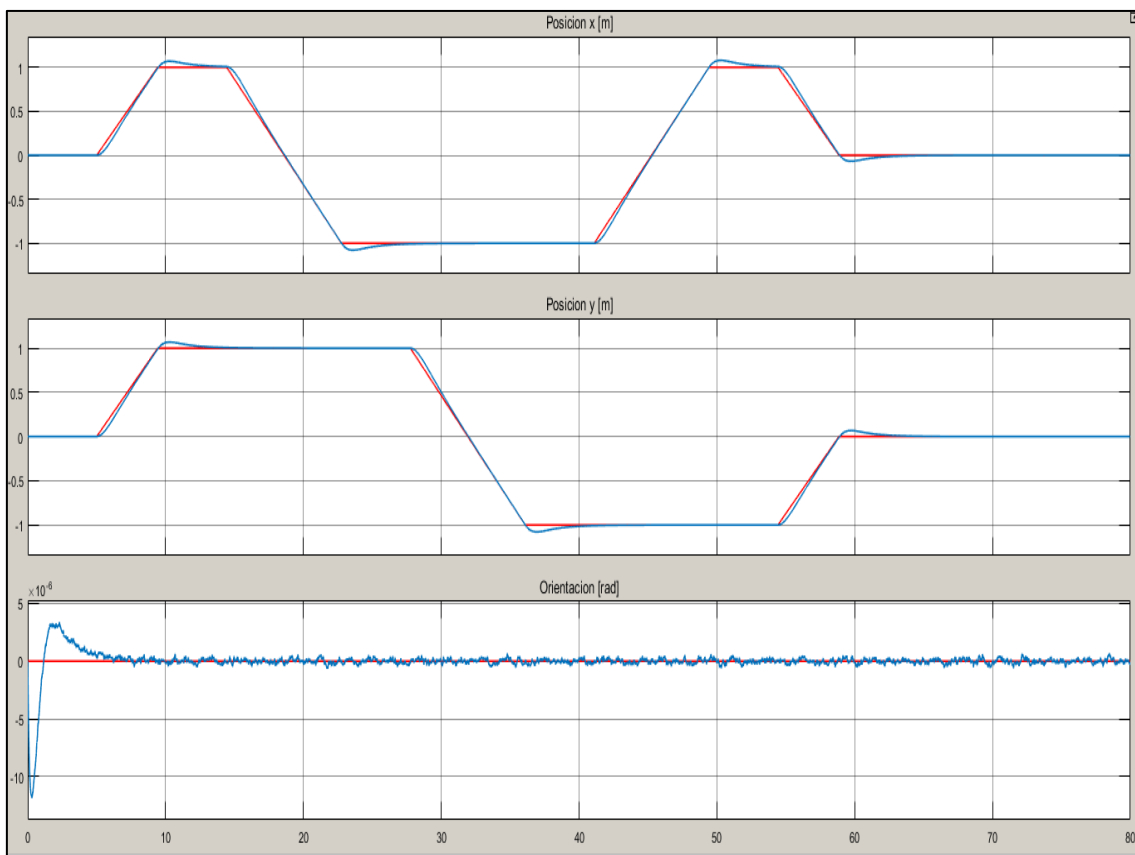


Fig.18: Comportamiento del sistema con la simulación de ruido en los motores.

Visualización con Simscape

Dentro del entorno de Simulink se encuentra Simscape que permite crear rápidamente modelos de sistemas físicos. Con esta herramienta se construyen modelos de componentes físicos basados en conexiones físicas que se integran directamente con diagramas de bloques y otros paradigmas de modelado. Es posible modelar motores eléctricos, rectificadores tipo puente, actuadores hidráulicos y sistemas de refrigeración mediante el ensamblaje de componentes fundamentales en un esquema.

Simscape ayuda a desarrollar sistemas de control y a probar el rendimiento a nivel de sistema. En este caso se utilizó para poder visualizar el movimiento del robot omnidireccional y poder acercarse al comportamiento que tendría en la realidad.

En una primera instancia se crea el modelo de la plataforma en el software de diseño CAD en 3D. El modelo de las ruedas es descargado de internet de la página web “grabcad”, en la cual se comparten diferentes diseños mecánicos en 3D.

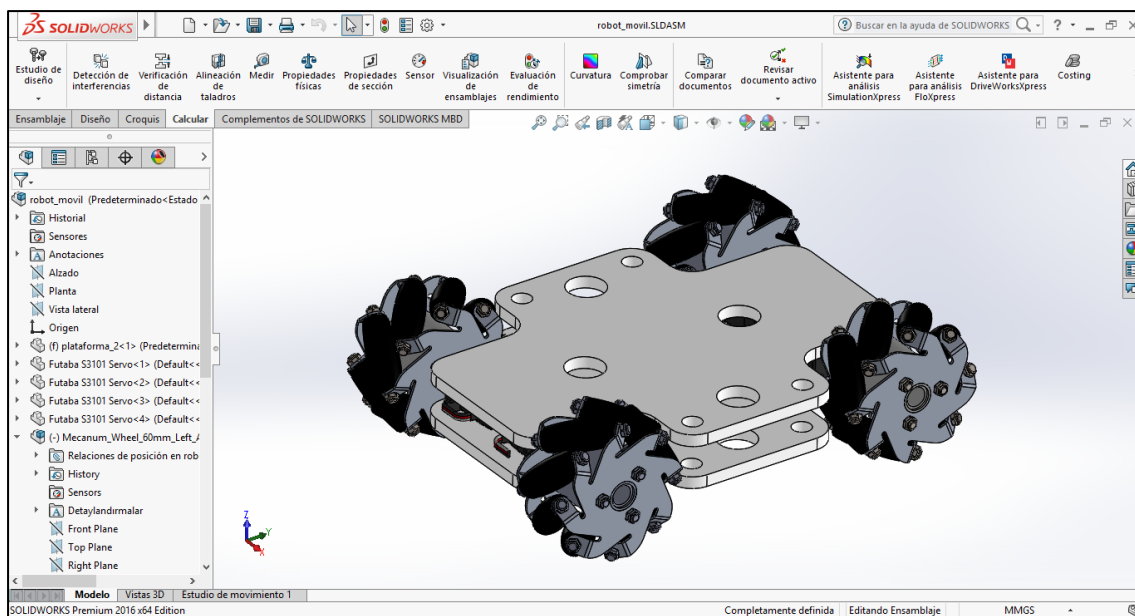


Fig.19: Diseño del modelo en Solidworks.

Además de permitir el diseño en 3D el software Solidworks posee diferentes complementos para posteriores análisis. Uno de ellos es el complemento “*Simscape Multibody Link*”, el cual permite exportar el modelo creado a un archivo de extensión “.xml”. Este tipo de archivo se ejecuta en Matlab y es visualizado como un esquema de bloques en Simulink/Simscape.

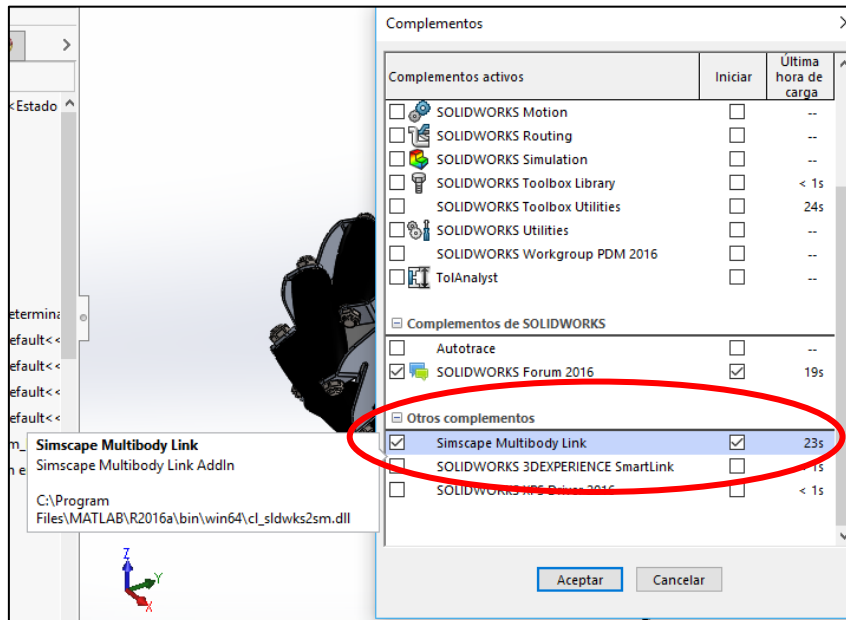


Fig.20: Complemento que permite exportar el modelo creado en Solidworks a Simscape.

En Simscape el modelo importado solo representa el modelo creado en solidworks representando solo la parte solida diseñada, es decir los ejes de referencia de cada pieza solo se encuentran con transformaciones de traslación y rotación rígidas. Para poder dotar al modelo de movimiento dentro de este nuevo entorno se agregan dos componentes importantes que ayudan a la representación física del robot, un bloque de junta de revolución y otro de junta planar, que permitirán el movimiento relativo de las ruedas con la plataforma y el movimiento de la plataforma con respecto a un sistema de referencia respectivamente.

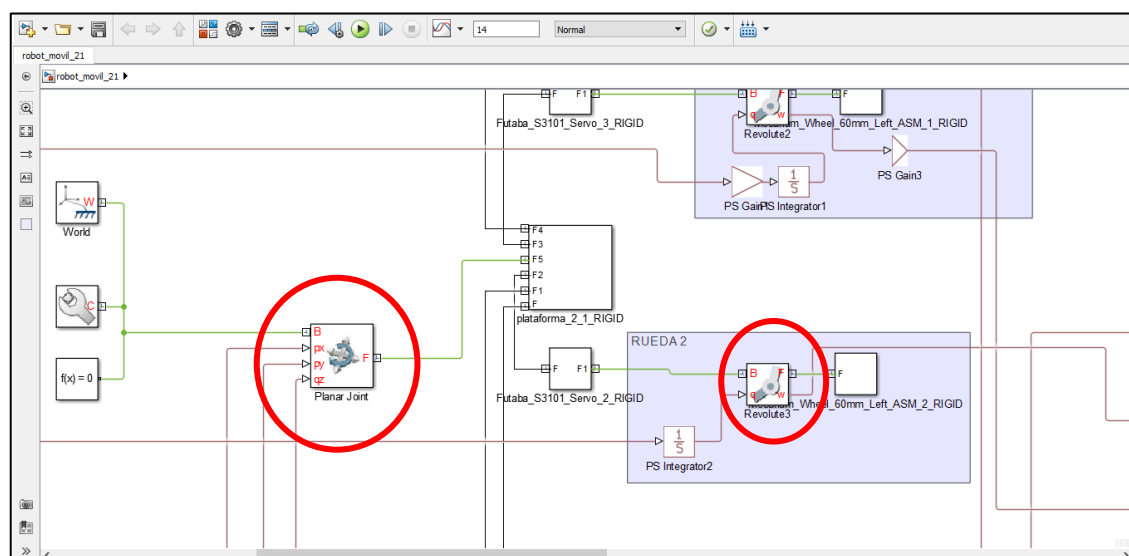


Fig.21: Modelo en simscape con bloques agregados que representan las diferentes juntas.

Con este modelo cargado en Simscape, a través del entorno Simulink es posible conectar a este nuevo modelo importado al modelo cinemático que se planteó anteriormente. De esta manera se visualiza el comportamiento del robot omnidireccional dentro del entorno de simulación que posee el software MATLAB.

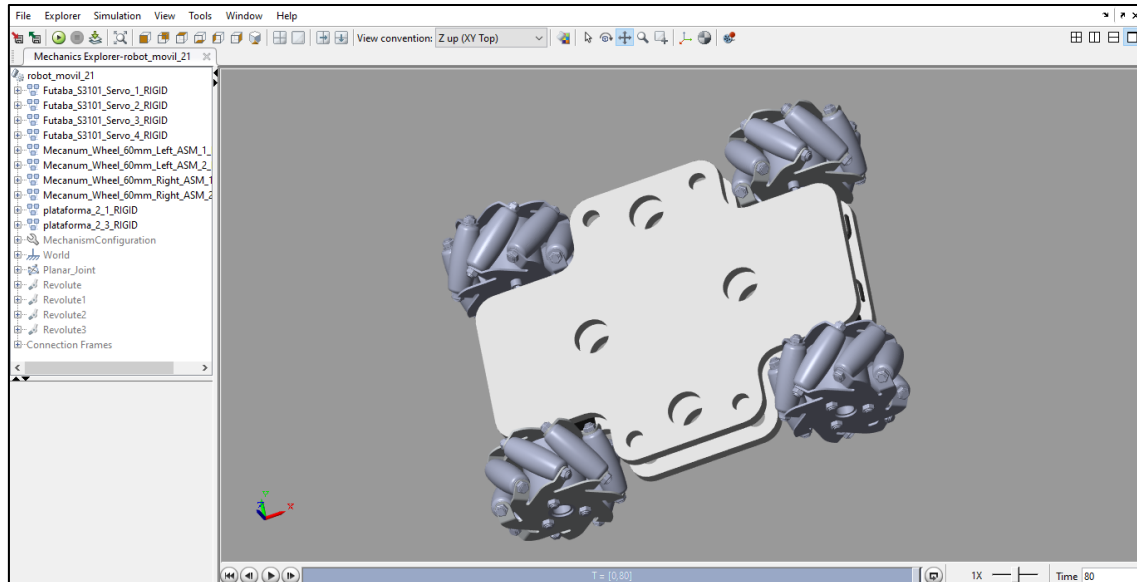


Fig.22: Visualización del comportamiento del robot móvil al ejecutar la simulación en Simscape.



Conclusiones y mejoras

Los resultados obtenidos en el presente proyecto fueron satisfactorios. Se manipularon programas como Matlab y Simulink lo que permitió mejorar habilidades a la hora de modelar un sistema físico e implementar un modelo matemático para representar el comportamiento en una simulación. También se logró adquirir conocimientos tanto de Solidworks como de Simscape, logrando una interacción de los mismos para exportar el modelo de un programa a otro.

Con el análisis de los resultados, que se fueron obteniendo en el transcurso del trabajo, fue posible detectar diferentes situaciones y compararlas con la realidad, por ejemplo en una primera simulación con parámetros iniciales en los controladores se detectaron valores muy elevados en las velocidades de los motores significando que en la realidad deban ser motores de gran tamaño para alcanzar dichas revoluciones, lo que no sería muy práctico.

Se deja planteado para un futuro la mejora de la implementación de otros tipos de control, comparando las respuestas con los implementados. Además se propone el agregado de un autómata de control que permita enviar consignas de posición (o velocidad) al sistema a medida que el error tienda a un valor cercano a 0, lo que ahorra la ejecución de código en Matlab para poder generar dichas consignas. Como última mejora, se podría implementar un programa dedicado a la generación de trayectorias y lograr que el robot se desplace de un lugar a otro realizando cualquier tipo de recorrido.



Referencias:

- Apuntes catedra Control y Sistemas
- OGATA, KASTUHICO, "INGENIERÍA DE CONTROL MODERNA", QUINTA EDICIÓN.
- OLLERO, "ROBÓTICA: MANIPULADORES Y ROBOTS MÓVILES".
- Modelado motor dc, documentación de ayuda en internet:
http://isa.uniovi.es/ISAwiki/index.php/Modelado_de_un_motor_CC
- MATHWORKS. Documentación de ayuda en internet:
Simscape:
 - <https://www.mathworks.com/videos/physical-modeling-with-simscape-1492698267787.html> (modelado sistema físico)
 - <https://la.mathworks.com/videos/simulation-of-collision-avoidance-for-mobile-robots-97765.html?requestedDomain> (simulación en Simscape)
 - <https://la.mathworks.com/help/phymod/smlink/ug/export-robot-assembly-from-solidworks-software.html> (exportación de modelo 3D)