

Universidad Nacional de Córdoba
Facultad de Ciencias Exactas, Físicas y Naturales
Córdoba, Argentina

Trabajo Integrador de Sistemas de Control 1

ROBOT DE DOS RUEDAS AUTOBALENCEADO



Autor: Riba, Franco

Docentes: Ing. Claudio Agüero – Ing. Reinado Reyes

Contacto: franco.riba@mi.unc.edu.ar

Año de cursado de la asignatura: 2021

ÍNDICE

1. Introducción.....	1
2. Definición del Problema	1
2.1 Principio de Funcionamiento	1
2.2 Conceptos Teóricos de Sistemas de Control	2
2.2.1 Bloques del Sistema.....	2
2.2.2 Señales del Sistema y sus Niveles	2
2.2.3 Posibles Perturbaciones	3
2.2.4 No Linealidades Involucradas	3
3. Modelado Matemático de la Planta.....	4
3.1 Consideraciones Previas	5
3.1.1 Nomenclatura Básica Utilizada	5
3.1.2 Parámetros del Robot.....	6
3.1.3 Sistema de Coordenadas Utilizado	6
3.2 Dinámica del Sistema de Rotación.....	7
3.3 Dinámica del Cuerpo Principal	9
3.4 Obtención de las Ecuaciones de Movimiento	10
3.5 Linealización de las Ecuaciones de Movimiento	11
3.6 Dinámica del Actuador	13
3.7 Incorporación de la Dinámica del Actuador a las EDM.....	16
3.8 Función de Transferencia.....	16
3.8.1 Modelado en Espacio de Estados.....	16
3.8.2 Obtención de la FTLA	18
3.8.3 Estudio de Estabilidad Absoluta (LA)	20
4. Especificaciones de Diseño.....	22
4.1 Error en Estado Estable.....	22
4.2 Sobre Pasamiento Máximo	22
5. Diseño del Controlador	23
5.1 Marco Teórico.....	23
5.1.1 Acción de control P	23
5.1.2 Acción de control PI.....	23
5.1.3 Acción de control PD.....	24
5.1.4 Acción de control PID	24
5.2 Diseño	25
6. Simulación en Matlab	27
6.1 Mecánica	27
7. Conclusiones.....	28

8. Mejoras Futuras al Proyecto	29
9. Recursos.....	30
Bibliográficos.....	30
10. Anexo – Implementación	31
10.1 Hardware.....	31
10.1.1 Driver L298N.....	31
10.1.2 Motorreductores	33
10.1.3 Acelerómetro y Giroscopio MPU-6050	35
10.1.4 Alimentación.....	38
10.1.5 Módulo Bluetooth HC-05	38
10.2 Mediciones.....	39
10.2.1 Masa del Sistema de Rotación	39
10.2.2 Masa del Cuerpo Principal	39
10.2.3 Radio de la Rueda	39
10.2.4 Longitud del Centro de la Rueda al Centro de Masa del Chasis	40
10.2.5 Inercia Mecánica del Chasis J_b	42
10.2.6 Inercia Mecánica de la Rodadura J_w	47

Acceso a recursos:

Repositorio en GitHub del proyecto (código Arduino + simulaciones):

<https://github.com/francoriba/sisControl/tree/master>

Videos y fotos de la evolución del proyecto:

<https://drive.google.com/drive/folders/1bMKZYu2w7RBSlc7AJ7NAG8CdF2A5MKAy?usp=sharing>

1. Introducción

El robot de dos ruedas auto balanceado consiste en plataforma robótica con dos ruedas que actúan independientemente y un centro de masa que se encuentra sobre el eje de rotación de las ruedas. Este tipo de robots han tenido un rápido avance en la última década y aparecieron en múltiples áreas de la vida cotidiana de las personas. Los ejemplos más notables que hacen uso de los mismos principios son los vehículos de transporte urbano como pueden ser los llamados Segway o los Hoverboards, sin embargo, se pueden hallar multitud de aplicaciones en distintos campos, como los dispositivos de transporte de equipajes, robots de asistencia y sillas de ruedas. Se trata de un proyecto no muy sencillo que engloba la mayoría de los conceptos aprendidos en la materia Sistemas de Control 1 y que da lugar a poner en práctica nuestras capacidades de diseño y programación.

2. Definición del Problema

2.1 Principio de Funcionamiento

Este sistema basa su funcionamiento en el mismo principio que rige el balanceo del clásico sistema de péndulo invertido con carrito motorizado mostrado en la Fig. 1. En este sistema cuando el péndulo cae hacia adelante el carrito avanza al frente para volver al estado de equilibrio como se muestra en la Fig. 2., por el contrario, cuando el péndulo cae hacia atrás, el carrito retrocede con el mismo objetivo como se indica en la Fig. 3.

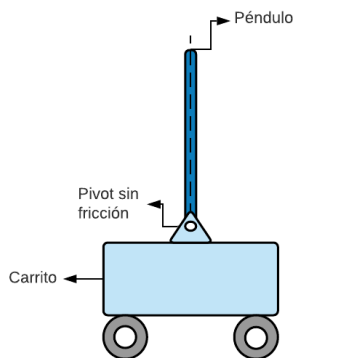


Fig. 3. Péndulo invertido inclinado hacia adelante y acción restauradora

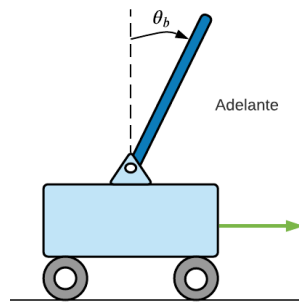


Fig. 2. Péndulo invertido en equilibrio

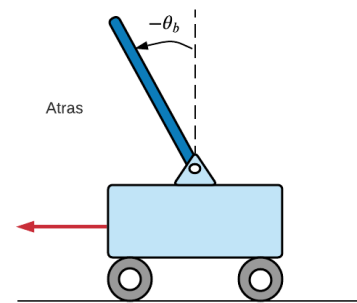


Fig. 1. Péndulo invertido hacia atrás y acción restauradora

Podemos transformar fácilmente este sistema en el de del robot auto balanceado reemplazando el carrito por dos motorreductores con escobillas cuyos ejes de salida pasarán a ser pivotes entre las ruedas y el chasis (péndulo).

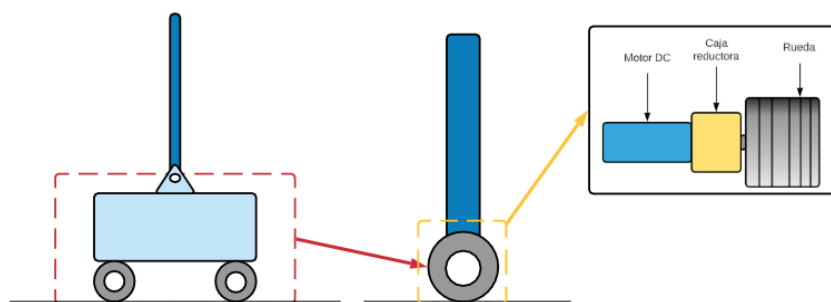


Fig. 4. Conversión del péndulo invertido en el robot auto balanceado

2.2 Conceptos Teóricos de Sistemas de Control

Este apartado está dedicado a relacionar y asociar conceptos teóricos básicos de Sistemas de Control con nuestro sistema de estudio.

2.2.1 Bloques del Sistema

- Detector de error: implementado vía software mediante placa de desarrollo Arduino UNO.
- Controlador: utilizaremos un controlador digital, concretamente la placa de desarrollo Arduino UNO, a nivel de software el controlador contará con un algoritmo de control PID.
- Amplificador: modulo controlador para motores, doble puente H - L298N.
- Actuador o Accionador: motorreductores genéricos de robótica.
- Planta o Proceso: dinámica del robot autobalanceado propiamente dicho, formado por lo que más adelante en este informe llamaremos Cuerpo Principal y Sistema de Rotación.
- Unidad de realimentación o Captador: sensor tipo IMU (Unidad de Medición Inercial), concretamente el módulo MPU 6050 de 6 ejes que integra un acelerómetro (mide aceleración lineal en 3 ejes), un giroscopio (mide velocidad angular en 3 ejes) y un DMP (Procesador de movimiento digital que permite obtener la fusión de los datos del acelerómetro y del giroscopio independientemente del microcontrolador)

En el anexo podrá encontrar el detalle del hardware utilizado donde se explica con mayor detalle el funcionamiento, características y otros aspectos a considerar de los componentes utilizados.

2.2.2 Señales del Sistema y sus Niveles

- Señal de mando o entrada: ángulo de inclinación $\theta_{b_{set}}$ [°] deseado para el robot. Este parámetro se configura vía software en el microcontrolador.
- Señal controlada o salida: rigurosamente se tienen dos señales de salida, la velocidad angular $\dot{\theta}_b$ [$\frac{^\circ}{seg}$] del chasis del robot (que captará el giroscopio) y aceleración lineal a [$\frac{m}{s^2}$] (que captará el acelerómetro). Sin embargo, para simplificar el análisis consideraremos directamente al ángulo de inclinación del chasis del robot θ_b [°] como única señal de salida ya que es esto lo que deseamos controlar.
- Señal de error: diferencia entre el ángulo de inclinación deseado y el ángulo de inclinación calculado por procesador de movimiento digital del MPU6050, esto es $e(t) = \theta_{b_{set}} - \theta_b(t)$ [°].
- Señal de control: tensión $v_a(t)$ [V] que el amplificador aplica a los motorreductores (señal de alta potencia generada por el driver) en función del ancho de pulso de la señal PWM de 8 bits (señal de baja potencia generada por el controlador) comandada por el algoritmo de compensación PID a implementar en el microcontrolador.
- Señal realimentada: ángulo de inclinación $\theta_b(t)$ detectado (en verdad calculada a partir de los valores de velocidad y aceleración sensados) por el sensor.

NOTA: ver diagrama de bloques del sistema al final del anexo del informe.

2.2.3 Posibles Perturbaciones

Las perturbaciones más frecuentes a la que será sometido el sistema son por un lado la propia gravedad y por el otro el dedo humano ya que una vez se tenga el robot funcionando podremos examinar sus limitaciones obstaculizando su funcionamiento en distintos niveles. Otra perturbación de la misma naturaleza podría originarse cuando el robot impacte contra algún objeto ya que no estaremos incorporando al robot un sistema de detección de obstáculos, sin embargo, esto sería posible utilizando sensores ultrasónicos o infrarrojos, aunque quedará pendiente como posible mejora futura del proyecto.



Otras perturbaciones no tan intuitivas que pueden presentarse podrían ser aquellas relacionadas a la presencia de error y demoras en la estimación de la variable de salida debido a la forma de operación de los sensores y la presencia de ruido. La eficiencia de los algoritmos utilizados también es un aspecto para considerar ya que comúnmente la etapa de censado y realimentación requiere filtros Kalman o filtros complementarios para la obtención de $\theta_i(t)$, es por esta razón que para esta tarea se ha optado por utilizar el algoritmo de fusión del DMP integrado al MPU 6050 ya que este utiliza técnicas estadísticas y trabaja con cuaterniones (una extensión de los números reales similar a los números complejos).

2.2.4 No Linealidades Involucradas

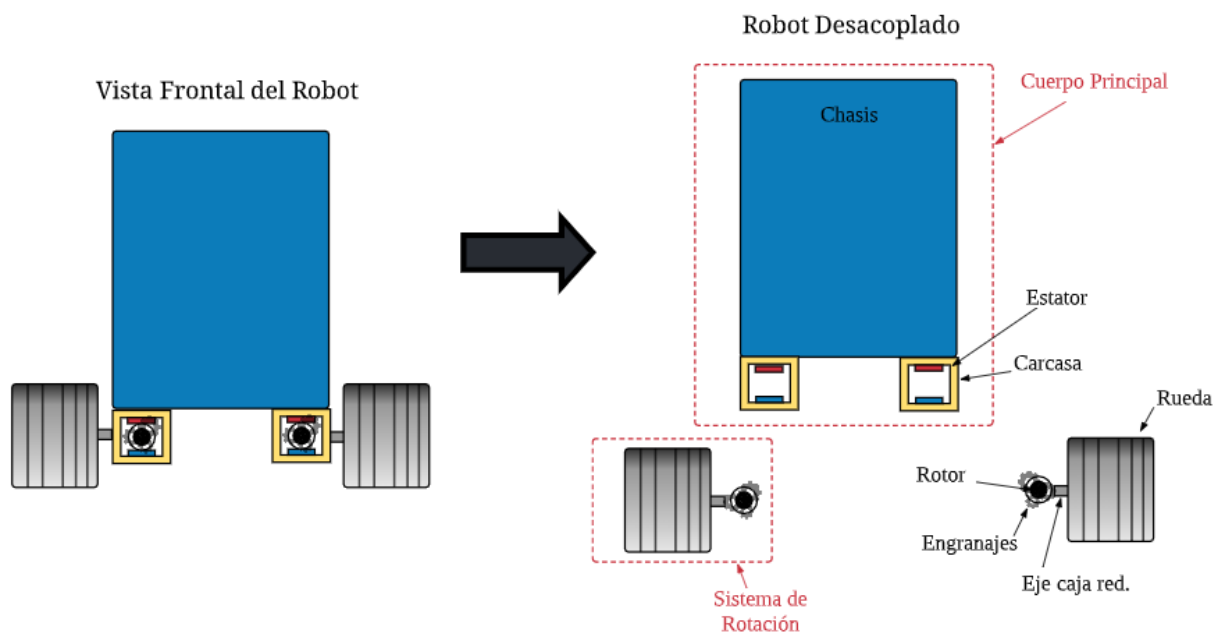
- Fuerzas de fricción no lineales en el motor y la caja reductora.
- Efectos de saturación en el circuito del controlador L298N que puedan producirse al alcanzar la máxima corriente permitida.
- No linealidades propias del modelado matemático del cuerpo principal (desarrollado en sección 3.5)

3. Modelado Matemático de la Planta

El objetivo final de esta sección es la obtención de la función de transferencia de la planta. Debemos aclarar que basándonos en [[https://en.wikipedia.org/wiki/Plant_\(control_theory\)](https://en.wikipedia.org/wiki/Plant_(control_theory))] en este trabajo se considerará que la planta contempla dos elementos, el primero es el “Actuador” representado por el motorreductor y el segundo es el “Proceso” que para nuestro sistema podría considerarse representado por el conjunto chasis-ruedas. Esto puede diferir de lo indicado en ciertas bibliografías que consideran como elementos separados a los actuadores y a la planta, lo cierto es que en nuestro robot los motorreductores se encuentran rígidamente enlazados al chasis y las ruedas, es por ello por lo que resulta conveniente considerar la planta como sistema que combina los elementos motorreductor, chasis y rueda. Dado que el motorreductor funciona como nexo entre el chasis y las ruedas proponemos dividir la planta en dos subsistemas acoplables entre sí, los subsistemas que proponemos son:

- **Cuerpo Principal:** este subsistema se asemeja funcionalmente al péndulo en el sistema de péndulo invertido con carrito. Los elementos que incluye son:
 - ✓ Chasis (plataformas con los componentes y varillas)
 - ✓ Estator
 - ✓ Rodamientos/bujes/rulemanes/bolilleros del eje del motor
 - ✓ Carcasa de motorreductor
- **Sistema de Rotación:** este subsistema se asemeja al carrito en el sistema de péndulo invertido con carrito. Los elementos que incluye el Sistema de Rotación son:
 - ✓ Rueda
 - ✓ Rotor del motor eléctrico
 - ✓ Engranajes de la caja reductora
 - ✓ Eje de salida de la caja reductora

La figura representa la división de partes planteada para el robot:

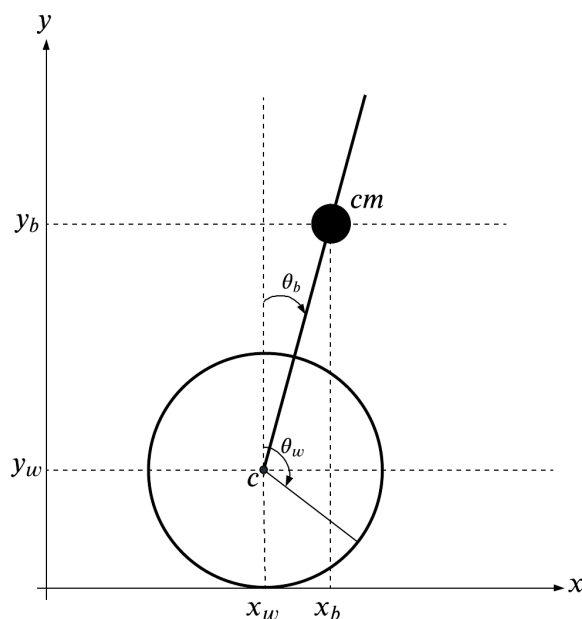


3.1 Consideraciones Previas

En esta sección se dejan en claro todos los detalles que el lector debería tener en consideración antes de iniciar de lleno en la lectura sobre la obtención de las ecuaciones de movimiento del sistema.

3.1.1 Nomenclatura Básica Utilizada

Para estudiar el movimiento del robot tendremos que trabajar con diversas variables, algunas de ellas describirán el movimiento de las ruedas (analogía con el carrito según la Fig. 4) mientras que otras describirán el movimiento del chasis (analogía con el péndulo según la Fig. 4), es por ello por lo que en la Tabla 1 establecemos la nomenclatura de estas variables y en la tabla debajo indicamos sus unidades en el sistema internacional junto con una breve descripción del aspecto físico que representa cada una de ellas.

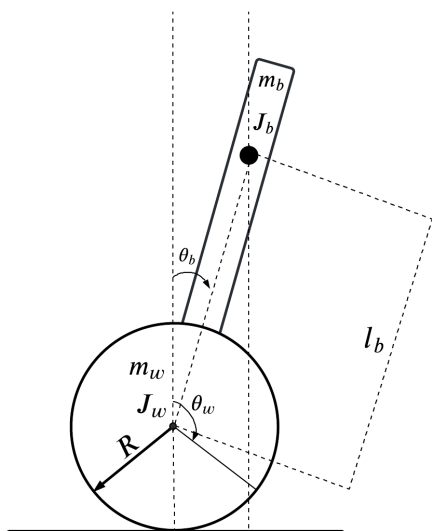


Símbolo			Unidad			Descripción
θ_b	$\dot{\theta}_b$	$\ddot{\theta}_b$	rad	$\frac{rad}{s}$	$\frac{rad}{s^2}$	Desplazamiento / Velocidad / Aceleración angular del cuerpo principal
θ_w	$\dot{\theta}_w$	$\ddot{\theta}_w$				Desplazamiento / Velocidad / Aceleración angular del sistema de rotación
x_b	\dot{x}_b	\ddot{x}_b	m	$\frac{m}{s}$	$\frac{m}{s^2}$	Desplazamiento / Velocidad / Aceleración lineal del centro de masa del cuerpo principal en la dirección x
y_b	\dot{y}_b	\ddot{y}_b				Desplazamiento / Velocidad / Aceleración lineal del centro de masa del cuerpo principal en la dirección y
x_w	\dot{x}_w	\ddot{x}_w				Desplazamiento / Velocidad / Aceleración lineal del centro de la rueda en la dirección x
y_w	\dot{y}_w	\ddot{y}_w				Desplazamiento / Velocidad / Aceleración lineal del centro de la rueda en la dirección y

Tabla 1: Unidades y descripción de las variables que describen el movimiento del Robot

3.1.2 Parámetros del Robot

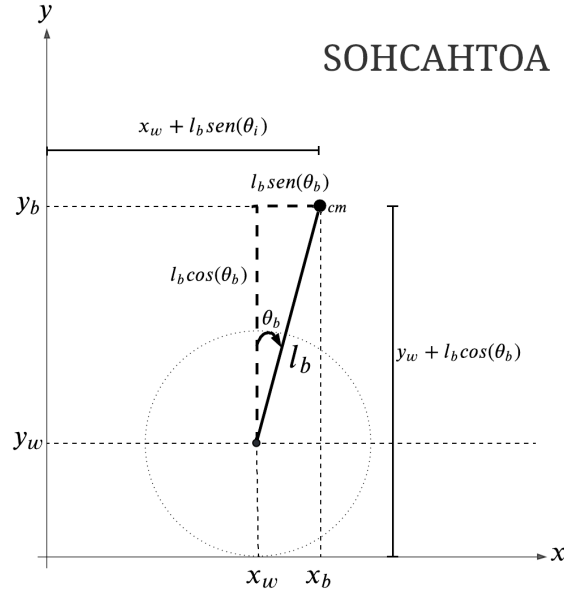
En la figura se indican los distintos parámetros del robot, donde podemos distinguir entre los que son propios del chasis y los que son propios de la rueda.



Símb	Uds.	Descripción
m_b	Kg	Masa del cuerpo principal
J_b	$Kg\ m^2$	Momento inercial del cuerpo principal
m_w	Kg	Masa del sistema de rotación
J_w	$Kg\ m^2$	Momento inercial del sistema de rotación
R	m	Radio de la rueda
l_b	m	Distancia entre el centro de la rueda y el centro de masa del chasis

3.1.3 Sistema de Coordenadas Utilizado

A la hora analizar la dinámica del chasis y de las ruedas será útil tener definido un sistema de coordenadas que permita avanzar en el desarrollo de las ecuaciones. La siguiente imagen muestra el análisis realizado al conjunto chasis – ruedas mediante reglas trigonométricas básicas aplicables con la regla SOHCAHTOA, es decir realizando una asociación del sistema del robot con el círculo unitario, esto nos permitirá trabajar con coordenadas polares.



En base a este análisis podemos observar que el desplazamiento lineal del chasis referenciado a su centro de masa tiene coordenadas $cm = (x_b, y_b)$ con:

$$x_b = x_w + l_b \sin(\theta_b) \quad (1)$$

$$y_b = y_w + l_b \cos(\theta_b) \quad (2)$$

Donde x_w representa la coordenada en la dirección x del centro de la rueda e y_w representa la coordenada en la dirección y de la rueda.

Aplicando las reglas de derivación para el producto de funciones y la regla de la cadena a (1) y (2) obtenemos:

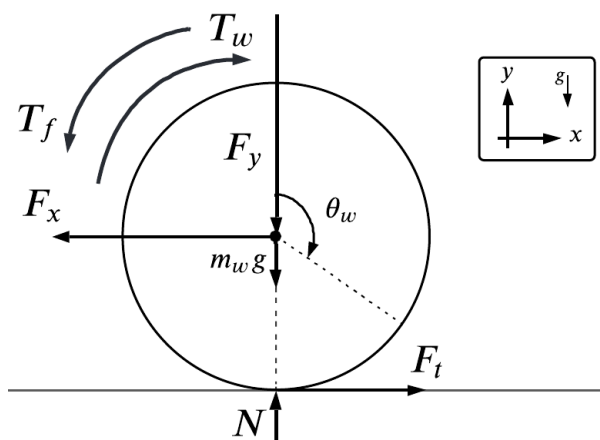
$$\begin{cases} \dot{x}_b = \dot{x}_w + \dot{\theta}_b l_b \cos(\theta_b) \\ \ddot{x}_b = \ddot{x}_w + \ddot{\theta}_b l_b \cos(\theta_b) - \dot{\theta}_b^2 l_b \sin(\theta_b) \\ \dot{y}_b = \dot{y}_w - \dot{\theta}_b l_b \sin(\theta_b) = -\dot{\theta}_b l_b \sin(\theta_b) \\ \ddot{y}_b = -\ddot{\theta}_b l_b \sin(\theta_b) - \dot{\theta}_b^2 l_b \cos(\theta_b) \end{cases} \quad (3)$$

3.2 Dinámica del Sistema de Rotación

Recordemos que cuando hablamos de sistema de rotación nos referimos al conjunto rueda-rotor-engranajes-eje.

Para el modelado se asumen las siguientes condiciones:

1. Ambas ruedas son idénticas.
2. Las ruedas son rígidas y de masa uniformemente distribuida.
3. Las ruedas giran sin deslizar.
4. La superficie en que se mueve el robot es plana y horizontal, cumpliéndose *que* $\dot{\mathbf{y}}_w = \mathbf{0}$



Símb	Uds.	Descripción
F_x	N	Fuerza horizontal rueda-chasis
F_y	N	Fuerza vertical rueda-chasis
F_t	N	Fuerza de tracción (rozamiento)
$m_w g$	N	Peso del sistema de rotación
N	N	Fuerza normal a la superficie
T_w	N m	Torque aplicado a la rueda
T_f	N m	Torque de fricción (despreciado)

Como hemos ya establecido al inicio se asume una superficie plana y horizontal de forma que $\dot{y}_w = 0$ y en consecuencia $\ddot{y}_w = 0$, teniendo esto en cuenta podemos aplicar la segunda ley de movimiento de Newton a las fuerzas en la dirección y (i.e. $\sum \vec{F}_y = m_w \ddot{y}_w$) tomando como referencia la Tabla 1:

$$N - m_w g - F_y = m_w \ddot{y}_w = 0 \quad (4)$$

Haciendo uso de la segunda ley de movimiento de Newton para las fuerzas en la dirección x (i.e. $\sum \vec{F}_x = m_w \ddot{x}$), se obtiene:

$$F_t - F_x = m_w \ddot{x}_w \quad (5)$$

Haciendo uso de la misma ley aplicada a la rotación (i.e. $\sum M_o = J_w \ddot{\theta}_w$), se obtiene:

$$T_w - R F_t = J_w \ddot{\theta}_w \quad (6)$$

Dado que se ha asumido que la rueda gira sin deslizar resulta válido plantear:

$$x_w = \theta_w R \rightarrow \theta_w = \frac{x_w}{R} \rightarrow \ddot{\theta}_w = \frac{\ddot{x}_w}{R} \quad (7)$$

Si ahora se despeja F_t de (5) y se la reemplaza junto a (7) en (6) se obtiene la expresión:

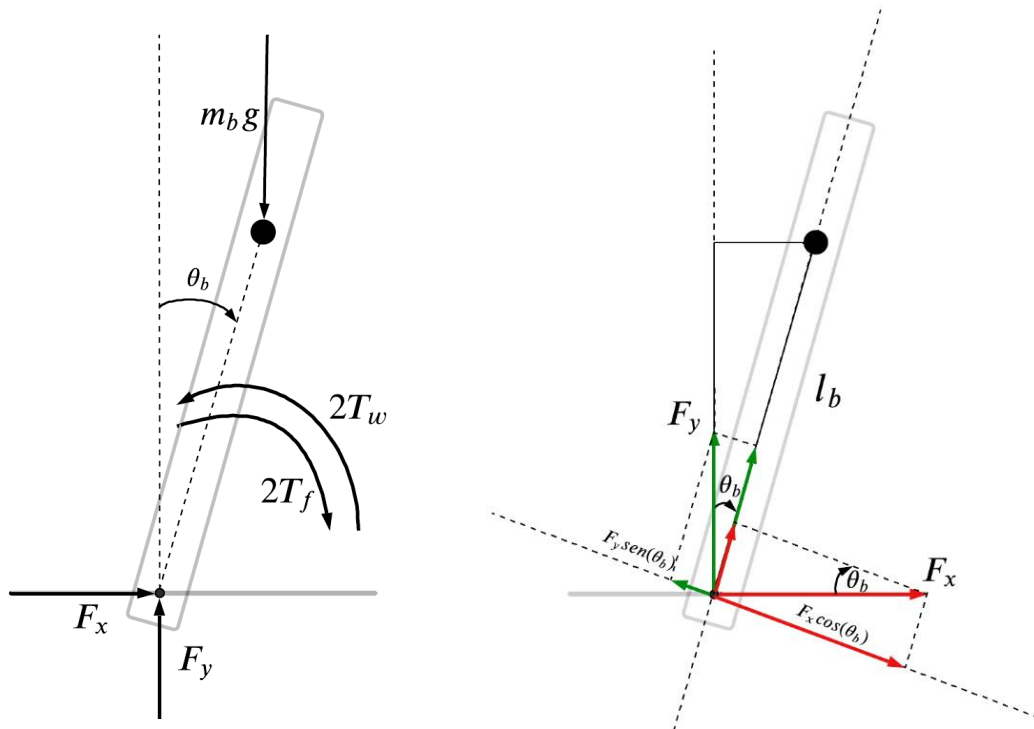
$$\ddot{x}_w \left(J_w \frac{1}{R^2} + m_w \right) = \frac{1}{R} T_w - F_x \quad (8)$$

Esta ecuación nos resultará útil mas adelante.

3.3 Dinámica del Cuerpo Principal

Hay que recordar que cuando hablamos de cuerpo principal nos referimos al conjunto chasis-estator-carcasa. Para simplificar el modelado se asume que el peso se encuentra uniformemente distribuido.

La única diferencia con las fuerzas externas de la rueda es la ausencia de fuerza de tracción F_t , esto es debido a que el eje del motorreductor solo impulsa a la rueda, no al chasis.



Consideraremos al giro en sentido horario como negativo y al giro en sentido anti-horario como positivo.

Haciendo uso de la segunda ley de movimiento de Newton en la dirección x (i.e. $\sum \vec{F}_x = m_b \ddot{x}_b$), se obtiene:

$$F_x = m_b \ddot{x}_b \quad (9)$$

Según los resultados de (3) podemos expresar (9) como:

$$F_x = m_b [\ddot{x}_w + \ddot{\theta}_b l_b \cos(\theta_b) - \dot{\theta}_b^2 l_b \sin(\theta_b)] \quad (10)$$

Haciendo uso de la segunda ley de movimiento de Newton en la dirección y (i.e. $\sum \vec{F}_y = m_b \ddot{y}_G$), se obtiene:

$$F_y - m_b g = m_b \ddot{y}_b \quad (11)$$

Según los resultados de (3) podemos expresar (11) como:

$$F_y = m_b g - m_b \left[\ddot{\theta}_b l_b \sin(\theta_b) + \dot{\theta}_b^2 l_b \cos(\theta_b) \right] \quad (12)$$

Haciendo uso de la segunda ley de movimiento de Newton aplicada a la rotación del centro de masas del chasis (i.e. $\sum M_{cm} = J_b \ddot{\theta}_b$), y teniendo en cuenta la descomposición de fuerzas presentada en 3.3 tenemos que:

$$-2T_w + F_y l_b \sin(\theta_b) - F_x l_b \cos(\theta_b) = J_b \ddot{\theta}_b \quad (13)$$

3.4 Obtención de las Ecuaciones de Movimiento

Sustituyendo (10) y (12) en (13) se obtiene la primer ecuación de movimiento para el conjunto rueda-chasis:

$$(J_b + m_b l_b^2) \ddot{\theta}_b = m_b g l_b \sin(\theta_b) - m_b l_b \cos(\theta_b) \ddot{x}_w - 2T_w \quad (14)$$

Mientras que si sustituimos (10) en (8) obtenemos la segunda ecuación de movimiento para el conjunto rueda-chasis:

$$\left[m_b + 2 \left(\frac{J_w}{R^2} + m_w \right) \right] \ddot{x}_w = -m_b l_b \cos(\theta_b) \ddot{\theta}_b + m_b l_b \sin(\theta_b) \dot{\theta}_b^2 + \frac{2T_w}{R} \quad (15)$$

3.5 Linealización de las Ecuaciones de Movimiento

Observando las ecuaciones de movimiento obtenidas podemos darnos cuenta de que no son lineales debido a la presencia de funciones seno, coseno y exponentes cuadráticos. Como ya sabemos, en la asignatura de Sistemas de Control nos hemos enfocado exclusivamente en el estudio de sistemas lineales e invariantes en el tiempo (LTI), por lo que un sistema no lineal necesitará ser linealizado para que podamos avanzar con su estudio. La clave para linealizar nuestras ecuaciones de movimiento es identificar en qué punto el sistema se encuentra en equilibrio ya que en torno a dicho punto existirá una región de operación en la cual puede establecerse una aproximación como sistema lineal de nuestro sistema no lineal.

Resulta intuitivo pensar que el robot se caerá si no es inicialmente colocado en una posición cercana al punto de equilibrio, esto significa que bajo funcionamiento normal el robot deberá encontrarse en una posición aproximadamente vertical respecto de la superficie de apoyo y el movimiento de rotación a lo largo del eje de la rueda será $\theta_b \approx 0 \text{ rad}$, entonces este será el punto de operación que utilizaremos para linealizar nuestro sistema.

Los términos de las ecuaciones de movimientos que deberemos linealizar son $\sin(\theta_b)$, $\ddot{x}_w \cos(\theta_b)$, $\ddot{\theta}_b \cos(\theta_b)$ y $\dot{\theta}_b^2 \sin(\theta_b)$, para linealizarlos utilizaremos el método de expansión en serie de Taylor considerando el primer término de esta y considerando el punto de operación ya indicado (i.e. punto de operación $\theta_i = 0 \text{ rad}$).

$$\sin(\theta_b) \approx \theta_b$$

$$\ddot{x}_w \cos(\theta_b) \approx \ddot{x}_w$$

$$\ddot{\theta}_b \cos(\theta_b) \approx \ddot{\theta}_b \quad (16)$$

$$\dot{\theta}_b^2 \approx 0$$

Sustituyendo lo planteado en (16) en (14) y (15) obtenemos las ecuaciones de movimiento linealizadas:

$$(J_b + m_b l_b^2) \ddot{\theta}_b = m_b g l_b \theta_b - m_b l_b \ddot{x}_w - 2T_w \quad (17)$$

$$\left[m_b + 2 \left(\frac{J_w}{R^2} + m_w \right) \right] \ddot{x}_w = -m_b l_b \ddot{\theta}_b + \frac{2T_w}{R} \quad (18)$$

Notar que se han linealizado las ecuaciones de manera previa a combinar las ecuaciones de la dinámica del sistema de rotación y el cuerpo principal con las ecuaciones de la dinámica del actuador, la razón por la que podemos hacer esto es porque de antemano sabemos utilizaremos un modelo lineal del motor, es decir que las ecuaciones que describen su comportamiento no requerirán la linealización de algunos de sus términos.

Las ecuaciones (17) y (18) pueden ser reescritas con el fin de expresarlas en el formato de espacio de estados, para eso sustituimos $\ddot{\theta}_b$ de (17) en (18) y \ddot{x}_w de (18) en (17) al hacer esto y reordenar las expresiones resultantes se obtiene:

$$\ddot{x}_w = \frac{1}{m_b + 2\left(\frac{J_w}{R^2} + m_w\right) - \frac{(m_b l_b)^2}{J_b + m_b l_b^2}} \left\{ -\frac{(m_b l_b)^2 g}{J_b + m_b l_b^2} \theta_b + 2 \left[\frac{1}{R} + \frac{m_b l_b}{J_b + m_b l_b^2} \right] T_w \right\} \quad (19)$$

$$\ddot{\theta}_b = \frac{1}{J_b + m_b l_b^2 - \frac{(m_b l_b)^2}{m_b + 2\left(\frac{J_w}{R^2} + m_w\right)}} \left\{ m_b l_b g \theta_b - 2 \left[R + \frac{m_b l_b}{m_b + 2\left(\frac{J_w}{R^2} + m_w\right)} \right] \frac{T_w}{R} \right\} \quad (20)$$

Definiendo las constantes:

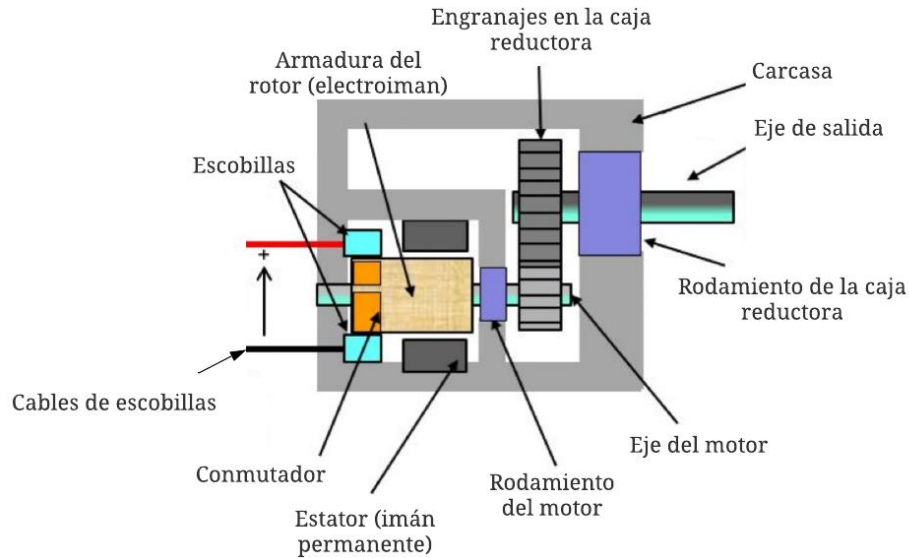
$$\begin{aligned} M &\equiv m_b + 2\left(\frac{J_w}{R^2} + m_w\right) - \frac{(m_b l_b)^2}{J_b + m_b l_b^2} \\ J &\equiv J_b + m_b l_b^2 - \frac{(m_b l_b)^2}{m_b + 2\left(\frac{J_w}{R^2} + m_w\right)} \\ C_1 &\equiv -\frac{1}{M} \frac{(m_b l_b)^2 g}{J_b + m_b l_b^2} \\ C_2 &\equiv \frac{2}{M} \left[\frac{m_b l_b}{J_b + m_b l_b^2} + \frac{1}{R} \right] \\ C_3 &\equiv \frac{1}{J} m_b l_b g \\ C_4 &\equiv -\frac{2}{RJ} \left(\frac{m_b l_b}{m_b + 2\left(\frac{J_w}{R^2} + m_w\right)} + R \right) \end{aligned} \quad (21)$$

Las ecuaciones (19) y (20) pueden ser expresadas como:

$$\begin{aligned} \ddot{x}_w &= C_1 \theta_b + C_2 T_w \\ \ddot{\theta}_b &= C_3 \theta_b + C_4 T_w \end{aligned} \quad (22)$$

3.6 Dinámica del Actuador

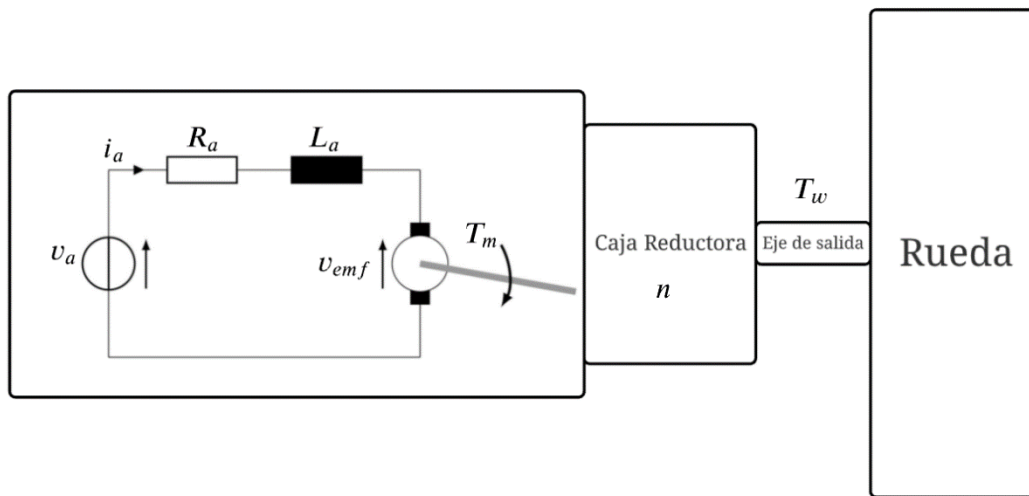
Los actuadores que se implementarán en el sistema son motorreductores, es decir motores DC con escobillas a los cuales se les adiciona una caja reductora con engranajes que permiten aumentar la cantidad de toque producido por el motor a costa de una disminución proporcional de la velocidad de giro del eje de salida de la caja reductora respecto de la velocidad de giro del eje motor. En la siguiente imagen se indican las partes básicas por las que se encuentra formado un motorreductor.



Antes de comenzar con el modelado del motorreductor se presenta la tabla, la cual indica la simbología que se utilizará para representar los parámetros del motorreductor, así como también indicará las unidades en el sistema internacional y una breve descripción de dichos parámetros.

Símb	Uds.	Descripción
θ_m	<i>rad</i>	Posición angular del eje del motor
$\dot{\theta}_m$	<i>rad/s</i>	Velocidad angular del eje del motor
b_f	<i>Nms/rad</i>	Coefficiente de amortiguamiento viscoso del rotor (despreciado)
n	-	Relación de transmisión
T_m	<i>Nm</i>	Torque producido por el motor
v_a	<i>V</i>	Voltaje de entrada del motor
v_{emf}	<i>V</i>	Voltaje de contra fuerza electromotriz
i_a	<i>A</i>	Corriente en la armadura del rotor
R_a	Ω	Resistencia del motor
L_a	<i>H</i>	Inductancia de la armadura (despreciada)
k_e	<i>Vseg/rad</i>	Constante de contra fuerza electromotriz de la armadura
k_t	<i>Nm/A</i>	Constante de torque del motor

Para el modelado del motor DC nos resulta de utilidad trabajar con el circuito eléctrico equivalente del rotor que se muestra en la imagen a continuación. El circuito equivalente de las bobinas en el rotor es representado por la conexión en serie de R_a y L_a . R_a representa la pérdida resistiva en la bobina mientras que L_a representa la autoinductancia del bobinado (el bobinado genera su propio campo magnético cuando se aplica tensión en los terminales del motor, por ello se comporta como un inductor). Cuando se aplica el voltaje v_m entre los terminales del motor, la corriente i_m fluye hacia el bobinado del rotor, entonces este se convierte en un electroimán e interactúa con los imanes permanentes en el estator para impulsar el rotor en sentido horario o antihorario. Los parámetros k_t y k_e son coeficientes que dependen de la construcción del motor DC.



Se considera el circuito de la armadura en esta imagen, donde el torque generado en el rotor (y posteriormente amplificado por la caja de transmisión y presentado en el eje de salida) es proporcional a la intensidad del campo magnético de la bobina del rotor que a su vez depende de la corriente i_a que fluye por la bobina (Ley de Biot-Savart de magnetostática). Este conjunto de relaciones se puede plasmar matemáticamente por medio de la fórmula:

$$T_m = k_t i_a \quad (23)$$

Otra relación de los motores DC es la que indica que la tensión de contra fuerza electromotriz dependerá de que tan rápido el rotor gire en torno al estator:

$$v_{emf} = k_e \dot{\theta}_m \quad (24)$$

Aplicando la ley de tensiones de Kirchoff al circuito de la armadura se tiene:

$$v_a = R_a i_a + \frac{d i_a}{dt} L_a + v_{emf} \quad (25)$$

Sin embargo, la inductancia del motor L_m se considera nula, esto se debe a que cuando el motor se encuentra en reposo la bobina se ha descargado completamente a través del circuito cerrado posteriormente a la anterior marcha del motor. Otra razón para considerar despreciable L_m es que generalmente su valor es muy pequeño comparado con el valor resistivo de R_m . Entonces tenemos que

$$v_a \cong R_a i_a + v_{emf} \quad (26)$$

Sustituyendo (24) en (26):

$$v_a \cong R_a i_a + k_e \dot{\theta}_m \quad (27)$$

Despejando i_a de (27) se tiene:

$$i_a \cong \frac{v_a}{R_a} - \frac{k_e \dot{\theta}_m}{R_a} \quad (28)$$

Teniendo en cuenta que hemos asumido $b_f = 0$ se cumplen las siguientes relaciones debidas a la presencia de la caja reductora entre la rueda y el motor

$$T_w = nT_m \rightarrow T_m = \frac{T_w}{n} \quad (29)$$

$$\theta_m = n\theta_w \quad (30)$$

Si ahora sustituimos (28) en (23):

$$T_m = \frac{k_t}{R_a} v_a - \frac{k_t k_e}{R_a} \dot{\theta}_m \quad (31)$$

Sustituyendo (29) en (31):

$$T_w = \frac{nk_t}{R_a} v_a - \frac{nk_t k_e}{R_a} \dot{\theta}_m \quad (32)$$

La ecuación (32) es válida si se asume que la carcasa del motor se encuentra en reposo, sin embargo los motorreductores estan rigidamente fijos al chasis, y θ_m es el ángulo relativo entre el eje del motor (o rotor) y el estator, por lo tanto deberemos considerar este detalle realizando la modificación $\dot{\theta}_m \rightarrow (\dot{\theta}_m - \dot{\theta}_b)$ en (32), de forma que resulta:

$$T_w = \frac{nk_t}{R_a} v_a - \frac{nk_t k_e}{R_a} \dot{\theta}_m + \frac{nk_t k_e}{R_a} \dot{\theta}_b \quad (33)$$

Finalmente, si recordamos (7), podemos reemplazar $\dot{\theta}_m$ por $\frac{\dot{x}_w}{Rn}$ resultando:

$$T_w = \frac{nk_t}{R_a} v_a - \frac{k_t k_e}{R_a R} \dot{x}_w + \frac{nk_t k_e}{R_a} \dot{\theta}_b \quad (34)$$

Y definimos las siguientes constantes del motor:

$$\begin{aligned} C_{m1} &\equiv \frac{nk_t}{R_a} \\ C_{m2} &\equiv -\frac{k_t k_e}{R_a R} \\ C_{m3} &= \frac{nk_t k_e}{R_a} \end{aligned} \quad (35)$$

De forma que podemos expresar (34) como

$$T_w = C_{m1}v_a + C_{m2}\dot{x}_w + C_{m3}\dot{\theta}_b \quad (36)$$

3.7 Incorporación de la Dinámica del Actuador a las EDM

Sustituyendo (36) en las expresiones de (22) obtenemos:

$$\begin{aligned} \ddot{x}_w &= C_1\theta_b + C_2C_{m1}v_a + C_2C_{m2}\dot{x}_w + C_2C_{m3}\dot{\theta}_b \\ \ddot{\theta}_b &= C_3\theta_b + C_4C_{m1}v_a + C_4C_{m2}\dot{x}_w + C_4C_{m3}\dot{\theta}_b \end{aligned} \quad (37)$$

3.8 Función de Transferencia

Respecto de esta cuestión la asignatura Sistemas de Control 1 ha presentado su foco principal en la obtención de funciones de transferencia utilizando como herramienta la transformada de Laplace ya que esta nos permite tratar ecuaciones diferenciales que describen el comportamiento de un sistema como simples ecuaciones algebraicas que resultan fáciles de manipular.

3.8.1 Modelado en Espacio de Estados

Aquí hemos decidido optar por utilizar no solo la transformada de Laplace como herramienta, sino también la representación en espacio de estados para la cual podemos distinguir entre tres tipos de variables:

- Variables de entrada: aquellas que proveen las excitaciones necesarias para el funcionamiento del sistema.
- Variables de estado: aquellas que evolucionan en el tiempo dependiendo de su estado actual y de los valores impuestos a las variables de entrada.
- Variables de salida: aquellas sobre las cuales se quiere tener control.

La representación en espacio de estados resulta especialmente útil cuando deseamos estudiar el comportamiento de sistemas del tipo MIMO (Multiple-input Multiple-output), es decir sistemas que tienen mas de una entrada y/o salida, sin embargo, también existen los sistemas del tipo SISO (single-input single output), es decir sistemas que solo tienen una entrada y una salida. En nuestro caso, el sistema del robot auto – balanceado puede plantearse tanto como sistema MIMO o como sistema SISO dependiendo del nivel de detalle y control que se quiera

lograr para el sistema. En principio, optaremos por pensar a nuestro sistema como SISO donde la variable de entrada es la tensión v_m aplicada a los motorreductores y la variable de salida es el ángulo de inclinación del robot, sin embargo, la razón por la que se utilizará la representación en espacio de estados es que durante el curso no se puede dar demasiada profundidad a este tema en concreto y resulta un reto interesante para este trabajo.

Para expresar el modelado del Robot en el espacio de estados debemos determinar la ecuación de estado y la ecuación de salida las cuales se expresan en su versión mas general como:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}\tag{38}$$

En (38) A es la matriz de control o matriz del sistema, B es la matriz de entrada, C la matriz de salida y D es la matriz de transmisión directa, es una matriz que indica un acoplamiento directo entrada-salida. Por su parte \dot{x} es lo que se denomina vector de estado, mientras que y es el vector de salidas y u es la variable de entrada. Para nuestro robot visto como sistema SISO la variable de entrada es la tensión de v_m aplicada a los motorreductores y la variable de salida es el ángulo de inclinación del chasis θ_b .

Podemos relacionar (37) con (38) especificando la parametrización de las variables de estado:

Variables de estado:

- Posición horizontal del Robot (en el eje x): $x_1 = x_w$
- Velocidad lineal del Robot (en el eje x): $x_2 = \dot{x}_w$
- Posición angular del Cuerpo Principal (ángulo de cabeceo del robot): $x_3 = \theta_b$
- Velocidad angular del Cuerpo Principal (velocidad de cabeceo del robot): $x_4 = \dot{\theta}_b$

$$\text{Vector de estado: } x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_w \\ \dot{x}_w \\ \theta_b \\ \dot{\theta}_b \end{bmatrix}$$

$$\text{Derivada del vector de estado: } \dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \dot{x}_w \\ \ddot{x}_w \\ \dot{\theta}_b \\ \ddot{\theta}_b \end{bmatrix}$$

$$\text{Matriz del sistema o de estado: } A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & C_2 C_{m2} & C_1 & C_2 C_{m3} \\ 0 & 0 & 0 & 1 \\ 0 & C_4 C_{m2} & C_3 & C_4 C_{m3} \end{bmatrix}$$

$$\text{Matriz de entrada: } B = \begin{bmatrix} 0 \\ C_2 C_{m1} \\ 0 \\ C_4 C_{m1} \end{bmatrix}$$

Matriz de salida: $C = [0 \ 0 \ 1 \ 0]$

Matriz de transmisión directa: 0

Entrada: $u = v_a$

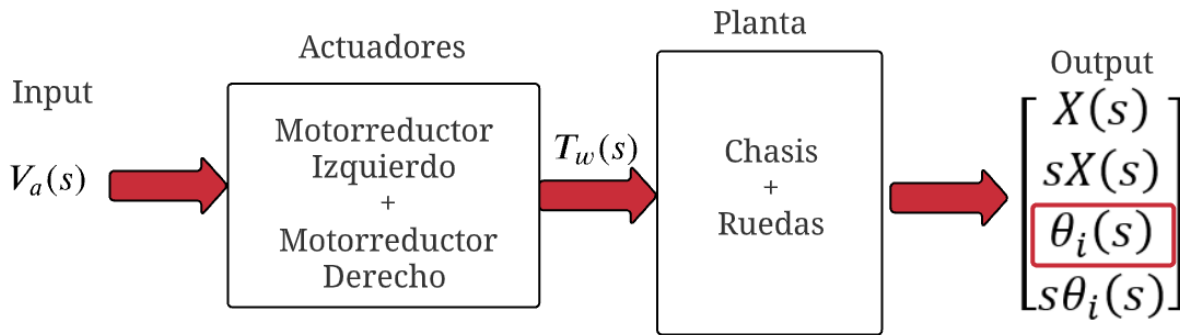
Salida: $y = \theta_b$

Combinando toda esta información podemos expresar a nuestro sistema matricialmente, ya sea en el dominio temporal o en el dominio de Laplace, de la siguiente manera:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & C_2 C_{m2} & C_1 & C_2 C_{m3} \\ 0 & 0 & 0 & 1 \\ 0 & C_4 C_{m2} & C_3 & C_4 C_{m3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ C_2 C_{m1} \\ 0 \\ C_4 C_{m1} \end{bmatrix} u(t) \xrightarrow{\mathcal{L}} s \begin{bmatrix} X_1 \\ sX_2 \\ X_3 \\ sX_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & C_2 C_{m2} & C_1 & C_2 C_{m3} \\ 0 & 0 & 0 & 1 \\ 0 & C_4 C_{m2} & C_3 & C_4 C_{m3} \end{bmatrix} \begin{bmatrix} X_1 \\ sX_2 \\ X_3 \\ sX_4 \end{bmatrix} + \begin{bmatrix} 0 \\ C_2 C_{m1} \\ 0 \\ C_4 C_{m1} \end{bmatrix} U(s) \quad (39)$$

O su equivalente si se prefiere no utilizar las variables de estado paramétricas:

$$\begin{bmatrix} \dot{x}_w \\ \ddot{x}_w \\ \dot{\theta}_b \\ \ddot{\theta}_b \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & C_2 C_{m2} & C_1 & C_2 C_{m3} \\ 0 & 0 & 0 & 1 \\ 0 & C_4 C_{m2} & C_3 & C_4 C_{m3} \end{bmatrix} \begin{bmatrix} x_w \\ \dot{x}_w \\ \theta_b \\ \dot{\theta}_b \end{bmatrix} + \begin{bmatrix} 0 \\ C_2 C_{m1} \\ 0 \\ C_4 C_{m1} \end{bmatrix} v_a \xrightarrow{\mathcal{L}} \begin{bmatrix} sX_w \\ s^2 X_w \\ s\Theta_b \\ s^2 \Theta_b \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & C_2 C_{m2} & C_1 & C_2 C_{m3} \\ 0 & 0 & 0 & 1 \\ 0 & C_4 C_{m2} & C_3 & C_4 C_{m3} \end{bmatrix} \begin{bmatrix} X_w \\ sX_w \\ \Theta_b \\ s\Theta_b \end{bmatrix} + \begin{bmatrix} 0 \\ C_2 C_{m1} \\ 0 \\ C_4 C_{m1} \end{bmatrix} V_a \quad (40)$$



3.8.2 Obtención de la FTLA

Llegados a este punto nos encontramos en condiciones de hallar la función de transferencia para nuestro sistema considerando como entrada la tensión suministrada al motorreductor y como salida el ángulo de inclinación del chasis de nuestro robot.

Para hallar la función de transferencia podemos utilizar la fórmula:

$$G(s) = \frac{Y(s)}{U(s)} = C(sI - A)^{-1}B + D \quad (41)$$

Esta es fácilmente deducible a partir de (38) si se está familiarizado con el álgebra matricial.

Utilizando Matlab para hallar la función de transferencia se obtiene:

$$\frac{\Theta_b(s)}{V_a(s)} = \frac{-C_4 C_{m1} s}{-s^3 + (C_2 C_{m2} + C_4 C_{m3})s^2 + C_3 s + C_1 C_4 C_{m2} - C_2 C_3 C_{m2}} = \frac{-90s}{(s+4.75)(s+5.7)(s-5.7)} \quad (42)$$

El resultado de (42) puede comprobarse en Matlab con el código “FT_simbolica.m” mientras que la expresión numérica puede comprobarse con el código “FT_numerica.m”. En caso de que el lector quiera verificar manualmente la función de transferencia numérica presentaremos a continuación los parámetros necesarios para calcular las distintas constantes definidas en (21) y (35).

Parámetros dados de los motorreductores (puede ver la hoja de datos en el anexo)

Parámetro	Variable en Matlab	Valor	Unidad
Torque máximo	T_{stall}	0.011	$Kg\ m$
Corriente máxima	I_{stall}	1.5	A
Corriente sin carga	I_{nl}	0.06	A
Velocidad sin carga	$RPM_{no\ load}$	230	RPM
Tensión nominal	V_{in}	6	V
Relación de reducción	n	48:1	-

Parámetros calculados de los motorreductores

Parámetro	Variable en Matlab	Valor	Unidad
Resistencia de armadura	R_a	$\frac{V_{in}}{I_{stall}} = 4$	Ω
Velocidad sin carga	w_{nl}	$\frac{RPM_{no\ load} 2\pi}{60} = 24.09$	rad
Constante de torque	k_t	$\frac{T_{stall}}{V_{in} n} R_a = 1.53e - 4$	$\frac{Nm}{A}$
Constante de contra emf	k_e	$\frac{V_{in} - R_a I_{nl}}{w_{nl} n} = 0.005$	$\frac{s\ V}{rad}$

Parámetros del sistema de rotación (valores utilizados justificados en)

Parámetro	Variable en Matlab	Valor	Unidad
Radio de la rueda	R	0.035	m
Radio considerado del rotor	r	0.005	m
Masa de la rueda	m_{rueda}	0.0551	Kg
Masa del rotor	m_{rot}	0.007	Kg
Masa de engranajes y eje	$m_{engranajesYeje}$	0.0025	Kg
Masa del sistema de rotación (suma de los 3 anteriores)	m_w	0.0646	Kg
Inercia de la rueda	$J_{rueda} = 0.5 * m_{rueda} * R^2$	$3.3749e - 5$	$Kg m^2$
Inercia del rotor	$J_{rot} = 0.5 * m_{rot} * r^2$	$8.75e - 8$	$Kg m^2$
Inercia del sistema de rotación (suma de los 2 anteriores)	J_w	$3.3836e - 5$	$Kg m^2$

Parámetros del Cuerpo Principal

Parámetro	Variable en Matlab	Valor	Unidad
Masa del chasis	m_{chasis}	0.5584	Kg
Masa del estator	$m_{estator}$	0.01	Kg
Masa de la carcasa	$m_{carcasa}$	0.0054	Kg
Masa del cuerpo principal (Suma de las 3 anteriores)	m_b	0.5738	Kg
Distancia del centro de la rueda al centro de masa	lb	0.07	m
Inercia del cuerpo principal	J_b	0.0024	$Kg m^2$

3.8.3 Estudio de Estabilidad Absoluta (LA)

Resulta lógico pensar que a lazo abierto nuestro robot no representa a un sistema estable ya que este nunca mantendrá el equilibrio por sí mismo (es decir sin acción de control) sino que se caería constantemente, significando esto que, en términos de estabilidad absoluta, nuestro sistema es inestable.

De lo estudiado en la materia sabemos que, dada una función de transferencia, si el polinomio denominador no está completo o si existen cambios de signo en sus coeficientes el sistema es inestable.

Entonces podemos comprobar rápidamente nuestra hipótesis inicial observando el polinomio denominador de (42), donde podemos ver que, si bien el polinomio está completo, se presenta un cambio de signo, esto confirma que la dinámica de la planta representa a un sistema inestable.

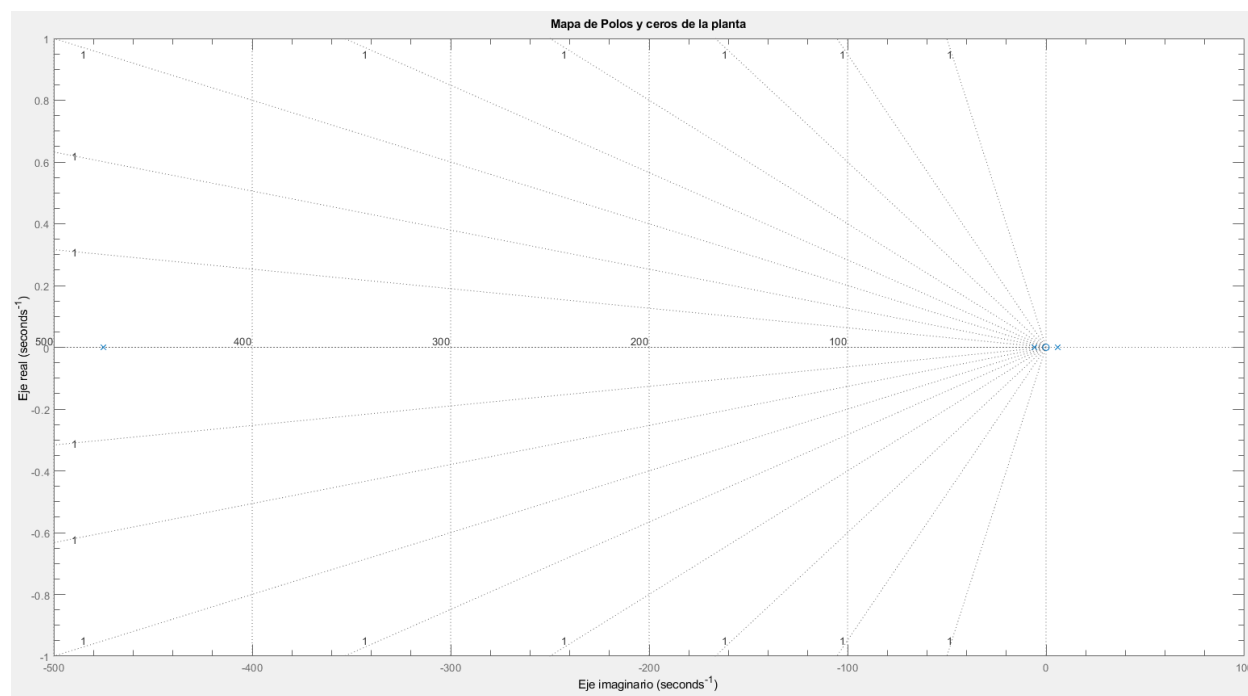
Una forma quizás más directa de probar lo mismo es obtener los polos del polinomio denominador que, con la ayuda de la función `pole()` de Matlab se puede verificar que son:

$$p_1 = 475$$

$$p_2 = 5.7$$

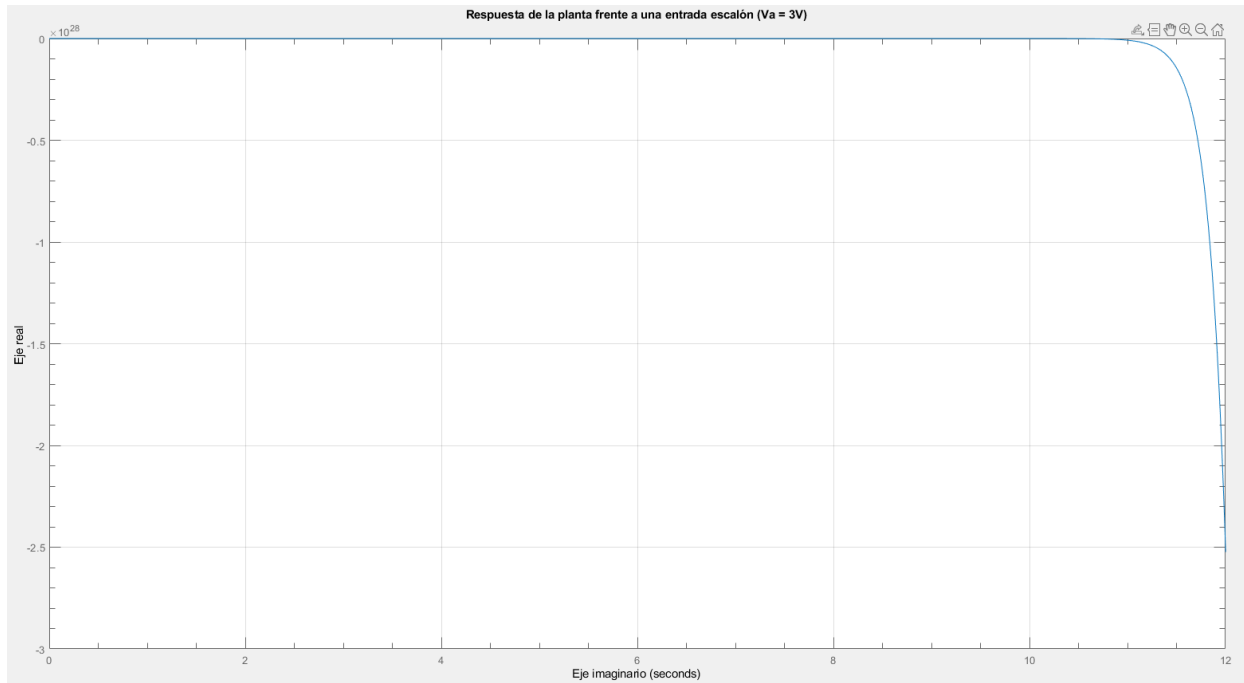
$$p_3 = -5.7$$

También podemos visualizar esto en un diagrama de polos y polos utilizando la función `pzmap()` disponible en Matlab.



Como se puede ver existen dos polos en el semi eje real positivo, lo cual nuevamente indica la inestabilidad de la planta.

Alternativamente podemos utilizar la función `step()` para ver cuál es la respuesta de nuestro sistema frente a una entrada del tipo escalón, hacer esto sería equivalente a ver que sucede con el ángulo de inclinación del robot cuando aplicamos una tensión constante de, por ejemplo, 3V a los motores.



Como se puede observar en la figura, cuando utilizamos una entrada del tipo escalón, la respuesta del sistema decrece infinitamente indicando la inestabilidad del sistema. Lo mismo sucede si en vez de una entrada del tipo escalón se utiliza un impulso. Todos los gráficos y funciones presentados en Matlab se pueden revisar mediante el programa “FT_numerica.m”.

4. Especificaciones de Diseño

En esta sección se definen las especificaciones de diseño en el dominio del tiempo, buscaremos definir los valores deseados de error en estado estable, sobre pasamiento máximo y tiempo de establecimiento para el sistema.

4.1 Error en Estado Estable

Para que el robot sea capaz de mantener el equilibrio será de vital importancia eliminar el error en estado estable, esto es $e_{ss} = 0$.

4.2 Sobre Pasamiento Máximo

Es deseable minimizar al máximo el sobrepaso del sistema durante la respuesta transitoria. Definiremos entonces el máximo sobrepaso aceptable para el correcto funcionamiento del robot.

En primer lugar, debemos notar el sobrepaso M_p estará dado en $[\circ]$ ya que la variable de salida del sistema que se desea controlar es el ángulo de inclinación θ_i .

El valor de M_p seleccionado deberá garantizar el funcionamiento del sistema, es decir deberá permitir que el robot sea capaz de volver al estado de equilibrio cuando se encuentra inclinado.

La capacidad del robot de recuperar su estado de equilibrio dependerá básicamente de que el torque aplicado por los motorreductores en un momento dado sea suficiente para contrarrestar la tendencia a caerse que tenga el robot en ese instante.

El torque necesario para que el robot recupere el equilibrio estará dado por:

$$T_{req} = mgl\sin(\theta_i)$$

Los motorreductores utilizados serán capaces de proporcionar un determinado torque T_{stall}

Tanto la masa m y la distancia l desde eje de los motores al centro de masa como la gravedad g serán constantes que dependerán del robot construido, por lo tanto, podemos decir que si se dispone de motores capaces de ejercer un torque máximo T_{stall} , el máximo valor de inclinación permitido para que el robot pueda recuperarse estará dado por:

$$\theta_{imax} = \sin^{-1}\left(\frac{T_{stall}}{mgl}\right)$$

El sobre pasamiento máximo M_p vendrá dado por la diferencia entre el ángulo de inclinación deseado y el máximo ángulo de inclinación con el que los motores pueden lidiar.

$$M_p = \frac{|\theta_{iset} - \theta_{imax}|}{\theta_{iset}}$$

O bien:

$$M_p(\%) = \frac{|\theta_{iset} - \theta_{imax}|}{\theta_{iset}} * 100$$

Un valor de sobrepaso mayor a este tendrá como resultado la caída del robot.

5. Diseño del Controlador

En esta sección buscaremos estabilizar la FT dada por la señal de entrada (voltaje del motor $v_a(s)$) y la señal de salida (ángulo de inclinación del cuerpo del robot θ_i).

5.1 Marco Teórico

5.1.1 Acción de control P

La acción proporcional busca que el error disminuya lo más rápido posible, normalmente a mayor K_p más rápido se minimiza el error, pero aumentarlo demasiado puede traer consecuencias negativas, como por ejemplo el aumento de las oscilaciones a la salida, es decir que el sistema se vuelva sub amortiguado, lo cual implicaría un tiempo de establecimiento mayor, y también podríamos ocasionar que el sistema salga de la zona de trabajo lineal.

Cuando se implementa una acción de control proporcional:

$$u(t) = K_p e(t)$$

$$U(s) = K_p E(s)$$

$$\frac{U(s)}{E(s)} = P(s) = K_p$$

5.1.2 Acción de control PI

En este caso se agrega una acción integral que se encarga de la acumulación del error a lo largo del tiempo, la acción integral no solo se basa en el valor actual del error, sino también en todos sus valores pasados y sumado a esto añade un polo en el origen al sistema, es decir aumenta en uno el orden del sistema. En conclusión, la acción proporcional busca minimizar rápidamente el error, mientras que la acción integral busca hacer nulo el error, tanto para frente a cambios en la entrada del sistema como frente a perturbaciones externas. Agregar la acción integral no aumentará el sobre pasamiento, pero si puede aumentar el tiempo de establecimiento ya que al poner un polo en el origen se acercan las raíces de la ecuación característica al eje imaginario.

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau \right) = K_p e(t) + K_i \int_0^t e(\tau) d\tau$$

$$U(s) = K_p \left(E(s) + \frac{E(s)}{T_i s} \right) = K_p \left(1 + \frac{1}{T_i s} \right) E(s)$$

$$\frac{U(s)}{E(s)} = PI(s) = K_p \left(1 + \frac{1}{T_i s} \right) = K_p + \frac{K_i}{s} = \frac{K_p \left(s + \frac{K_i}{K_p} \right)}{s}$$

De esta última fórmula vemos que un controlador PI se traduce en la presencia adicional de un cero ubicado en $-\frac{K_i}{K_p}$ y un polo en el origen, de forma que mediante las constantes K_i y K_p podemos mover el cero real y cambiar la ganancia general del sistema.

5.1.3 Acción de control PD

En este caso, al agregar la acción derivativa la salida del controlador pasara a ser proporcional a la razón de cambio de la señal de error respecto del tiempo, esto quiere decir que, si el error es constante, la acción derivativa es nula, es decir que la acción derivativa solo toma efecto durante los transitorios y cuando se producen perturbaciones. La acción derivativa busca minimizar los sobre pasamientos que se producen cuando la acción proporcional es muy alta, esto es posible ya que la acción derivativa se anticipa al estado futuro del sistema.

$$u(t) = K_p \left(e(t) + T_d \frac{de(t)}{dt} \right) = K_p e(t) + K_d \frac{de(t)}{dt}$$

$$U(s) = K_p (E(s) + s T_d E(s)) = K_p (1 + T_d s) E(s)$$

$$\frac{U(s)}{E(s)} = PD(s) = K_p (1 + T_d s) = K_p + K_d s$$

5.1.4 Acción de control PID

Combinando las tres leyes de control anteriores, podemos esperar que se produzca un rápido rechazo de las perturbaciones y la tendencia a hacer nulo el error en estado estable tanto por cambios en la entrada como por perturbaciones (acción integral), además por la acción proporcional esa tendencia deberá suceder rápidamente, por último, la acción derivativa actuará durante los transitorios para minimizar posibles oscilaciones ocasionados por un alto valor de K_p .

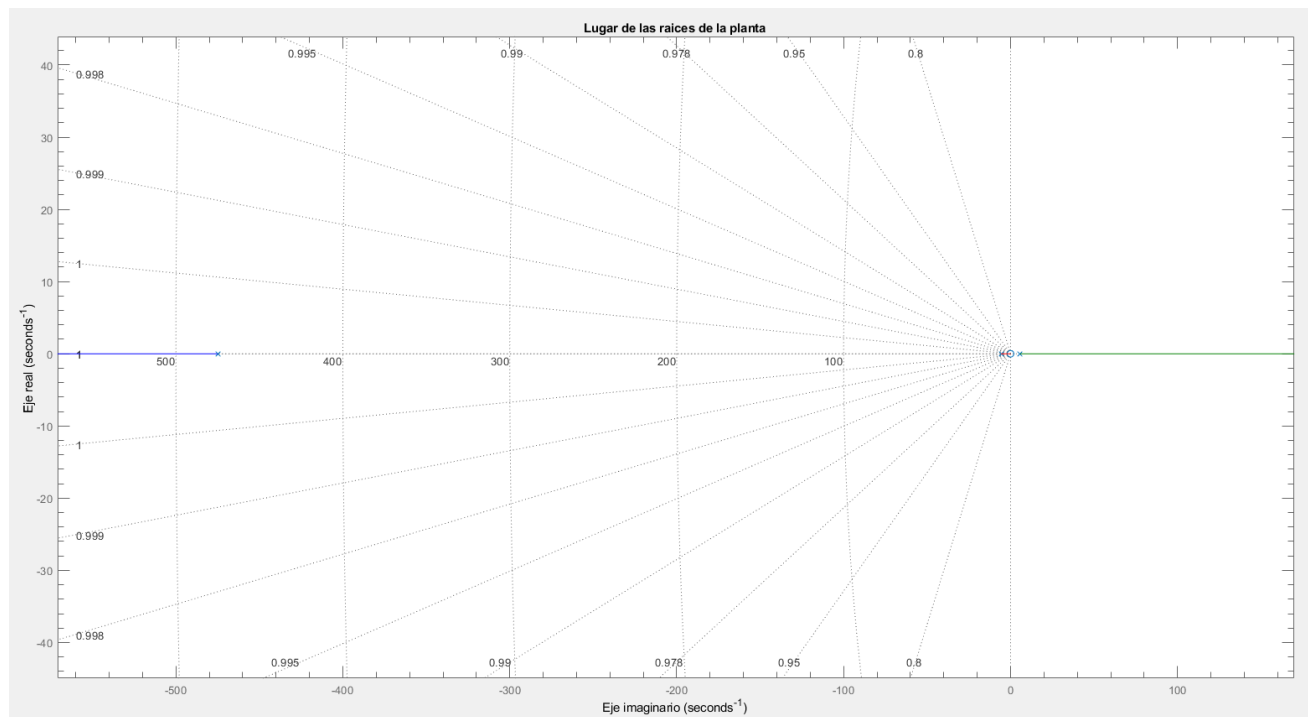
$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

$$U(s) = K_p \left(E(s) + \frac{E(s)}{T_i s} + s T_d E(s) \right) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) E(s) = K_p \left(\frac{T_d T_i s^2 + T_i s + 1}{T_i s} \right) E(s)$$

$$\frac{U(s)}{E(s)} = C(s) = K_p \left(\frac{T_d T_i s^2 + T_i s + 1}{T_i s} \right) = \frac{K_d s^2 + K_p s + K_i}{s}$$

5.2 Diseño

Comenzamos analizando el grafico del lugar de raíces del sistema (sin control PID):



Podemos apreciar que tal como ya habíamos anticipado, el grafico del lugar de raíces pone en evidencia nuevamente la inestabilidad de la planta sin controlar, una función de transferencia inestable causará que el control del robot falle. Es por ello por lo que diseñaremos a continuación un control PID, basándonos en los fundamentos teóricos ya mostrados en el marco teórico de esta sección. Del gráfico del lugar de raíces también podemos observar que no existe un camino entre los polos y el cero, esto significa que no existe ningún valor de ganancia proporcional K_p que pueda estabilizar el sistema por sí misma, siendo esto otra de las razones por las que no solo resulta conveniente, sino también necesario optar por un control PID.

El método de diseño utilizado será el de “ubicación de los polos”. La estabilidad se alcanza logrando que el polo inestable en $+5.7$ sea cancelado y colocado un nuevo polo convenientemente en el medio entre el origen y el polo estable en -5.7 , en este caso buscaremos ubicar este nuevo polo en -3 , donde es de esperar que no se genere interferencia con la dinámica del sistema por estar demasiado cercano al origen y, por lo tanto al eje imaginario y el plano inestable (semi plano derecho) y no demasiado cerca a otro polo que también podría interferir los efectos positivos del polo dominante generando la no amortiguación del sistema. Según hemos investigado, en teoría, podríamos mover el polo aún más lejos del origen para una mejor respuesta del sistema, sin embargo, si esto es llevado al extremo podría causar mucho estrés en los actuadores llevándolos a un punto de ruptura.

Teniendo en cuenta que nuestra FT es de la forma:

$$G(s) = \frac{ks}{(s - p_1)(s - p_2)(s - p_3)}$$

Combinando esta ecuación con la obtenida en el marco teórico para la acción de control PID:

$$\begin{aligned}
 P(s) &= \frac{G * (s)C(s)}{1 + G(s) * C(s)} = \frac{ks(K_d s^2 + K_p s + K_i)}{ks(K_d s^2 + K_p s + K_i) + s(s - p_1)(s - p_2)(s - p_3)} \\
 &= \frac{kK_d s^3 + kK_p s^2 + kK_i s}{kK_d s^3 + kK_p s^2 + kK_i s + s(s^3 - s^2(p_1 + p_2 + p_3) + s(p_1 p_2 + p_1 p_3 + p_2 p_3) - p_1 p_2 p_3)} \\
 &= \frac{kK_d s^3 + kK_p s^2 + kK_i s}{s(s^3 + s^2(-p_1 - p_2 - p_3 + kK_d)) + s(p_1 p_2 + p_1 p_3 + p_2 p_3 + kK_p) + (-p_1 p_2 p_3 + kK_i))}
 \end{aligned}$$

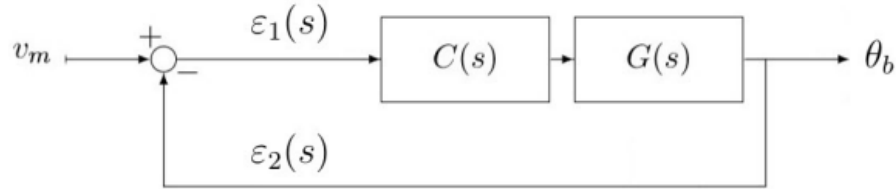


Diagrama en bloques de la FTLC del sistema del robot

Ahora el objetivo es obtener un denominador de la forma:

$$s(s - p'_1)(s - p'_2)(s - p'_3) = s(s - p_1)(s - p_2)(s - p'_3) = s(s - 475)(s - 5.7)(s - 3)$$

La razón de esto es que la nueva función de transferencia diseñada con PID tiene que ser equivalente a la anteriormente planeada en (42) con la diferencia de que un polo positivo es “intercambiado” por un polo negativo.

Con esta lógica tenemos entonces que:

$$\begin{cases} -p_1 - p_2 - p_3 + kK_d = -p_1 - p_2 - p'_3 \\ p_1 p_2 + p_1 p_3 + p_2 p_3 + kK_p = p_1 p_2 + p_1 p'_3 + p_2 p'_3 \\ -p_1 p_2 p_3 + kK_i = -p_1 p_2 p'_3 \end{cases}$$

Resultando así:

$$\begin{cases} K_p = \frac{p_1 p'_3 + p_2 p'_3 - p_1 p_3 + p_2 p_3}{k} \\ K_i = \frac{p_1 p_2 p_3 + p_1 p_2 p'_3}{k} \\ K_D = \frac{p_3 - p'_3}{k} \end{cases}$$

Reemplazando los valores de los polos y de la ganancia obtenemos:

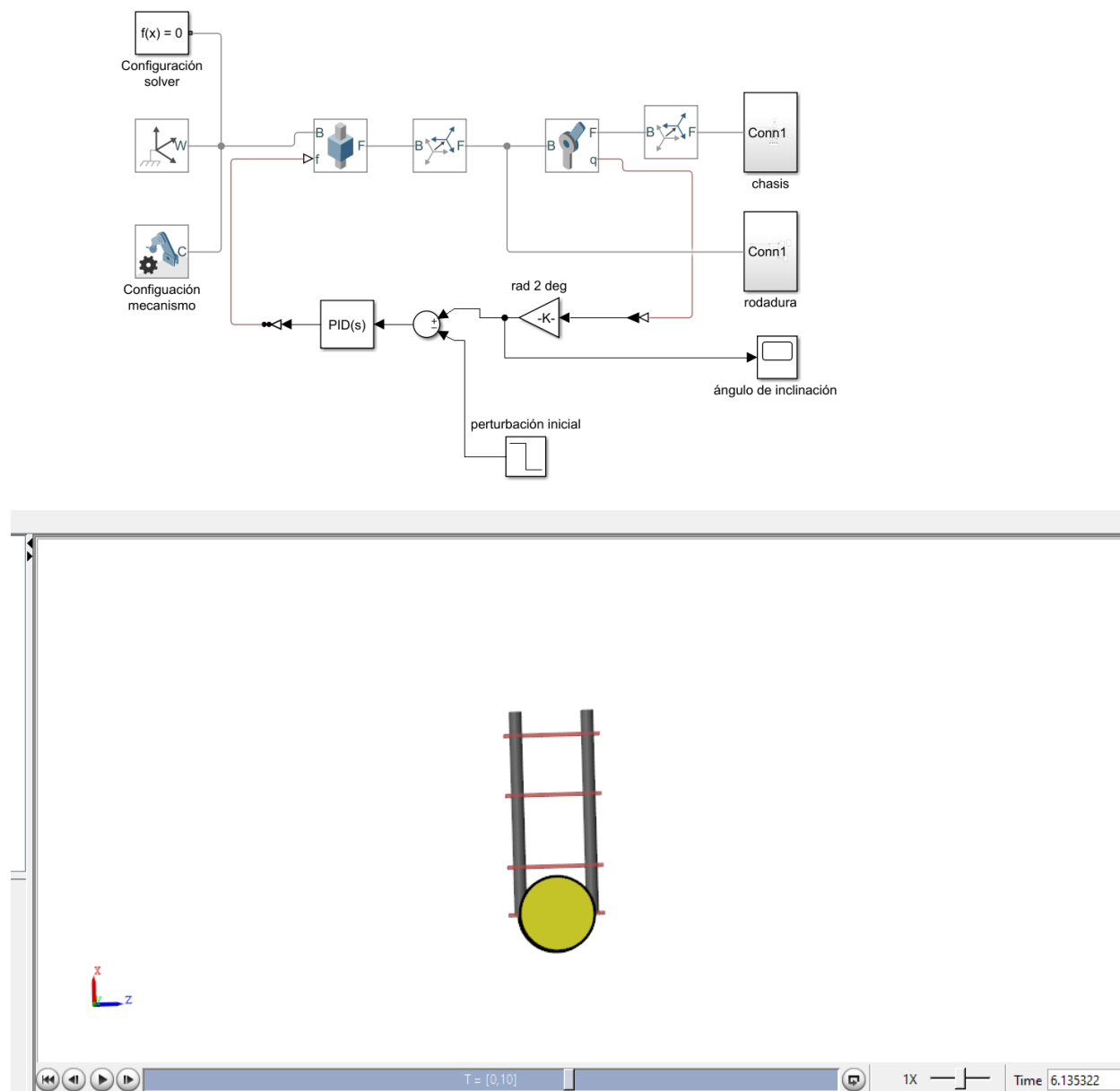
$$\begin{cases} K_p = -46.5603 \\ K_i = -260.2962 \\ K_D = -0.0969 \end{cases}$$

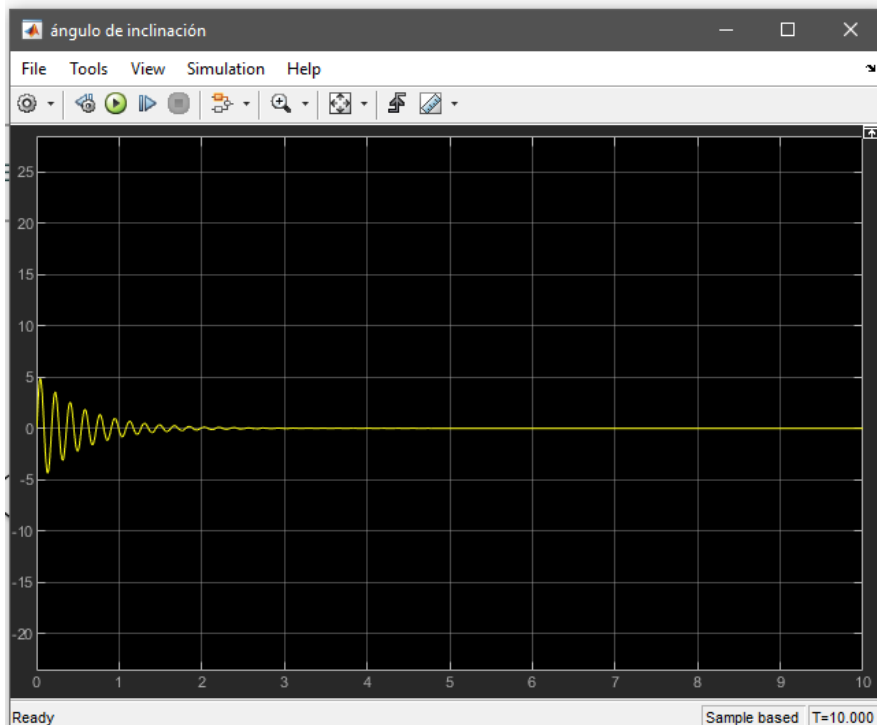
Esto ha sido una aproximación teórica a los valores de las constantes que deberíamos utilizar para lograr la estabilidad del sistema mediante un control PID, sin embargo según se ha investigado, para realizar la validación del modelo sería necesario convertir el control PID al dominio discreto, este tema no ha sido cubierto durante la cursada, por lo tanto se lo considera fuera del alcance del curso de sistemas de control 1, de hecho toda la implementación del control PID en tiempo continuo fue investigada y aprendida por cuenta propia, ya que durante la cursada no se alcanzan a cubrir estos contenidos. Por estas razones la validación final del sistema físico construido será mediante “prueba y error” en la programación de la constante de control PID.

6. Simulación en Matlab

6.1 Mecánica

Puede acceder a esta simulación mediante el archivo “simul_parte_mecanica.slx” disponible en el repositorio de GitHub.





7. Conclusiones

- Respecto a lo tratado en la sección 5.2, por intuición se podría argumentar que los valores negativos de las constantes son el resultado de un sistema inestable que intenta estabilizarse con ganancias negativas. Normalmente con un sistema estable estas ganancias deberían ser positivas y no se necesitaría un factor de compensación negativo K_p . Respecto a las simulaciones y scripts realizados en Matlab, la simulación muestra la interacción mecánica de los elementos principales del chasis la estructura planteada para el sistema de control. Vemos que, en la simulación, la cual es una simplificación del sistema real implementado, las oscilaciones en el ángulo de inclinación θ_b desaparecen por completo en aproximadamente $t = 3s$, llevando al sistema a la estabilidad.
- Respecto a la validación del sistema, hemos observado que tanto en la simulación por software del sistema como en el sistema físico implementado se logró alcanzar la estabilidad, sin embargo encontramos diferencias debido a las diversas simplificaciones realizadas para la simulación, el efecto mas notorio de discrepancia entre los dos casos es que en la simulación se alcanza un estado de equilibrio total o ideal, donde luego de la respuesta transitoria el robot ya no necesita desplazarse para mantener una inclinación de 0° respecto a la normal, mientras tanto, en el sistema físico real esto no sucede, el control PID se encuentra realizando correcciones en todo momento para mantener el robot “de pie”.
- Respecto a los potenciales efectos no lineales introducidos por el filtro creado por software para el MPU6050, se observa que no han resultado críticos al nivel de impedir el control del sistema, pero presumiblemente podría ser este uno de los factores que afecta introduciendo diferencias entre el comportamiento del sistema real y el comportamiento del sistema simulado que se ha mencionado en el punto anterior.
- Tanto el robot simulado como el robot físico pueden mantener el equilibrio (aunque de forma algo errática en el caso físico) con una influencia prácticamente nula de la constante k_i siempre y cuando k_p y k_d estén finamente ajustadas en el código fuente.

- Si bien la estabilidad del sistema puede lograrse mediante la implementación del control PID diseñado por el método de ubicación de polos, un controlador LQR ofrecería un control óptimo sobre el sistema de entrada a través de la llamada matriz de ponderación R, sin embargo, este tema debería ser profundizado y quizás implementado a futuro.
- El material y grosor de las ruedas afectan considerablemente el funcionamiento del sistema físico, se ha comprobado como las ruedas tomadas de un viejo auto a control remoto ayudan a lograr más fácilmente la estabilidad del sistema en comparación a las utilizadas inicialmente que fueron las típicas ruedas plásticas famosas para proyectos de robótica, las cuales además de ser más angostas, no logran un agarre a la superficie tan bueno como el de las ruedas recicladas del auto a control remoto.
- Una vez encendidos todos los componentes del sistema físico, el robot (con el sistema de control) demora aproximadamente 2 segundos en comenzar a funcionar o mostrar respuesta, lo cual resulta suficiente para que el robot comience a moverse sin caer, esto siempre y cuando la posición inicial en que se colocó el robot haya sido vertical, intentando aproximarse al punto de equilibrio, es decir 0° , esto no difiere sustancialmente de lo observado en la simulación. Si la posición inicial del robot no es vertical, el mismo cae, esto se debe a la incapacidad de los motores de contrarrestar el par generado por la naturaleza pendular del sistema.
- Dado que el driver controlador de los motores utilizado permite controlar cada uno de estos independientemente del otro, se pudieron realizar pruebas en las que la amplificación que se da mediante la señal PWM a cada uno de los motores difiere, esto gracias a la configuración de distintos factores de amplificación implementados por software para cada uno de ellos, lo cual resultó útil para compensar las diferencias de capacidad (respecto a las RPM) de cada motor. Si bien una mejor solución sería la implementación de un control de velocidad para cada rueda, esto no se ha realizado en este trabajo y queda como posible mejora a futuro.

8. Mejoras Futuras al Proyecto

Para lo que fue el alcance del curso de sistemas de control 1, los resultados de este proyecto fueron satisfactorios ya que se logró implementar y poner en funcionamiento el robot autobalanceado aplicando los aprendizajes de la materia y una gran cantidad de aprendizajes extra curriculares relacionados a la operación de los distintos componentes y la programación del sistema en general utilizando Arduino, en adición a esto la realización del proyecto ayudó a dimensionar lo complejo que pueden llegar a ser estos proyectos, en especial cuando se trabaja en equipos reducidos, la elaboración del proyecto ha consumido mucho tiempo y ha habido muchos momentos de frustración, sin embargo cada vez que se logró superar un problema, la motivación creció ayudando a seguir avanzando con la realización del proyecto.

El enfoque inicial dado para la realización de este trabajo es sumamente simplificado, ya que solo se ha implementado un control PID para controlar la inclinación del robot, sin embargo como se ha mencionado a lo largo de este informe, se podrían implementar adicionalmente controles PID para la velocidad y sincronización de giro de las ruedas, lo cual lograría un movimiento aún más estable en el sistema físico implementado, hacer esto requeriría del uso de algún tipo de encoder que permita calcular mediante software la velocidad de giro de las ruedas, hacer esto requerirá probablemente diseñar un nuevo chasis para el robot, ya que el diseño actual es sumamente compacto y prácticamente no admite la adición de más componentes. Otra mejora que se planea implementar a futuro, con un nuevo diseño es la modificación en tiempo real de las constantes del control PID, para poder realizar la configuración de estas directamente desde el hardware sin necesidad de tener conocimientos de programación, y haciendo así al robot una buena herramienta para demostrar a quienes no sepan nada del tema, como variando los parámetros se altera el comportamiento del sistema. En adición a esto se podría trabajar en mejorar el tiempo de respuesta del control bluetooth ya que la forma de control implementada actualmente es muy básica y solo permite indicar al robot cuando ir hacia delante o atrás, pero no cuando hacer giros a la izquierda o derecha, tardando además entre 2 y 3 segundos en surgir efecto el comando indicado.

9. Recursos

Bibliográficos

K. Ogata, 2010, “Ingeniería de Control Moderna”, 5ta Edición, Pearson Educación

T. Wescott, 2006, “Applied Control Theory for Embedded Systems”, Elsevier Inc.

Filminas Teóricas y Prácticas de la Cátedra de Sistemas de Control 1

Apuntes Personales de Cursado, Sistemas de Control 1, 2021

Web

<https://fkeng.blogspot.com/2019/03/>

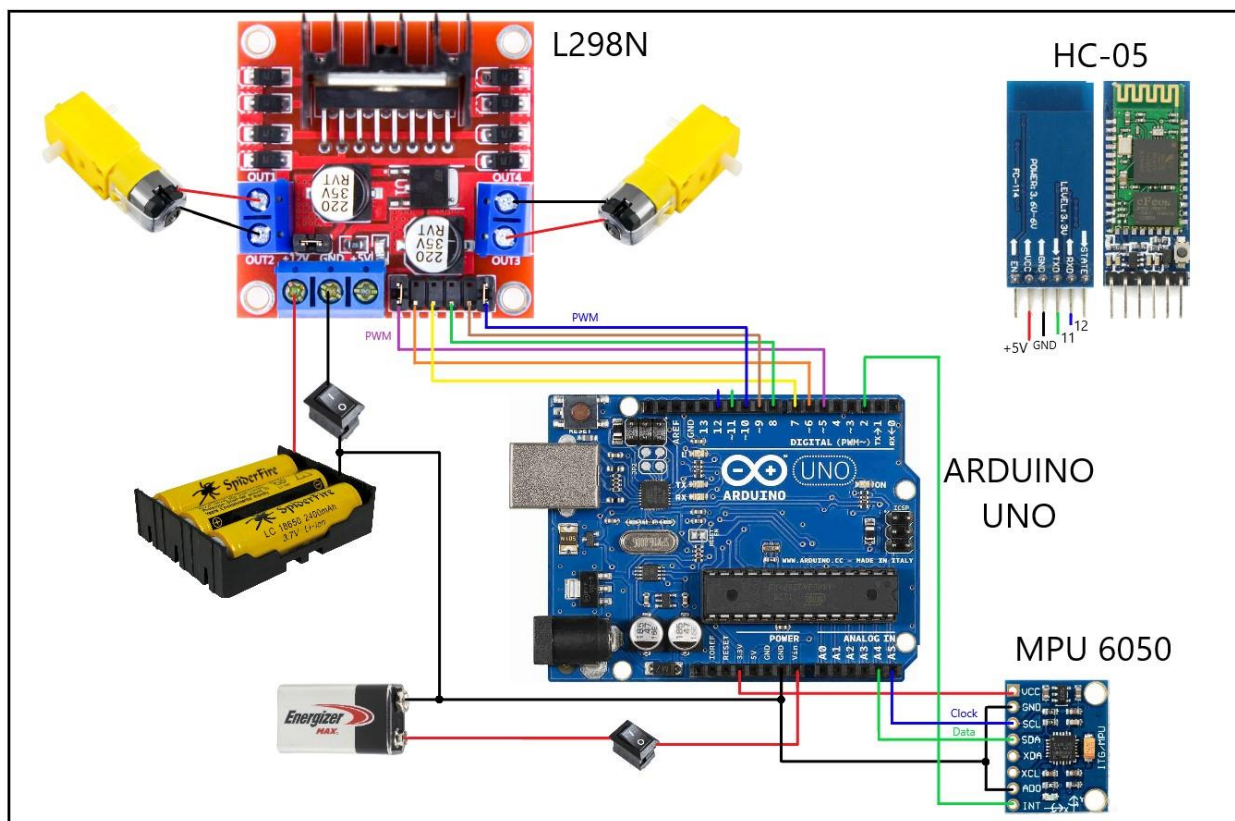
<https://sciendo.com/article/10.1515/ijame-2017-0046>

<http://www.ehcar.net/library/these/these028.pdf>

https://aaltodoc.aalto.fi/bitstream/handle/123456789/32398/master_Silfver_Toni_2018.pdf?sequence=1&isAllowed=y

10. Anexo – Implementación

10.1 Hardware



10.1.1 Driver L298N

Este módulo controlador de motores es ampliamente utilizado en proyectos de robótica y automatización que requieren el control de hasta dos motores DC como motorreductores o motores paso a paso de forma independiente. El 298N sirve para gobernar de alguna manera el rendimiento de un motor eléctrico. En el caso de este trabajo es de un motor de corriente continua. Este dispositivo nos permitirá:

- Suministrar la alimentación necesaria para el correcto funcionamiento del motor.
- Permite incluir un accionamiento manual o automático para iniciar y detener el motor.
- Da la posibilidad de cambiar fácilmente el sentido de giro y da también la opción de la regulación de la velocidad, así como regulación o limitación del par.
- Presenta protección contra sobrecargas y posibles fallas que se puedan presentar.

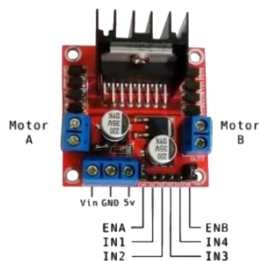
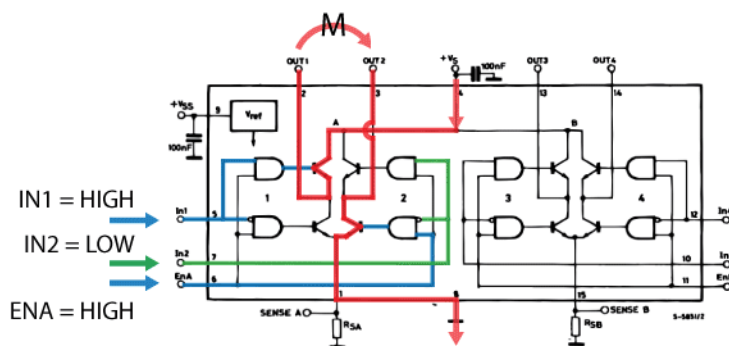


Figura: módulo L298N

En realidad, la denominación L298N corresponde al circuito integrado principal del módulo, el cual se encuentra montado de forma vertical sobre la placa PCB y tiene adosado un disipador pasivo de aluminio, además este circuito integrado incluye transistores de potencia para el manejo de hasta dos motores DC y se lo denomina doble puente H ya que se disponen cuatro transistores que conectados al motor forman una H, y con esta topología es posible controlar el giro del motor ya sea en sentido horario o antihorario.

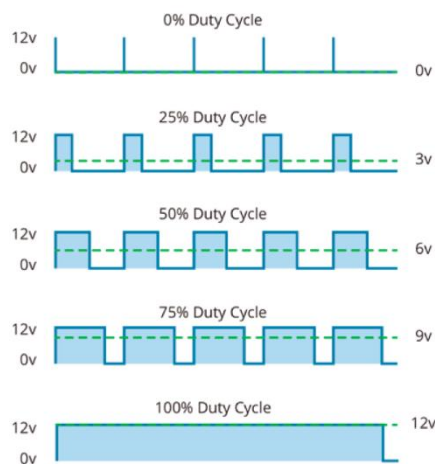


La conexión de los motores al módulo se realiza mediante las dos borneras ubicadas en extremos opuestos del mismo. Las salidas de los motores son controladas mediante los pines ubicados en la parte inferior derecha del módulo (3 pines izquierdos controlan el motor A y 3 pines derechos controlan el motor B), los pines ENA y ENB deben conectarse a pines PWM de Arduino, pues estos permiten habilitar los motores y realizar el control de velocidad, los mismos poseen jumpers (puentes que permiten contacto entre dos pines) que permiten que los pines ENA y ENB tengan un nivel lógico alto. Los pines IN1, IN2 y IN3, IN4 permiten establecer el sentido de giro de los motores A y B respectivamente haciendo uso de los mencionados puentes H, las tablas presentadas a continuación indican la relación entre los distintos niveles lógicos y el sentido de giro provocado por estos.

ENA	IN1	IN2	Motor A	ENB	IN3	IN4	Motor B
0	-	-	-	0	-	-	-
1	0	1	↻	1	0	1	↻
1	1	0	↺	1	1	0	↺

La alimentación del módulo se lleva a cabo utilizando los terminales Vin Y GND de la bornera, dado que estaremos trabajando con motores, y el consumo de corriente será considerable, este módulo requerirá una alimentación independiente con una tensión adecuada para el rango de funcionamiento del motor (en este caso para motorreductores de 3V a 12V) y con una provisión de corriente acorde. El terminal identificado como 5V es una salida que sirve para alimentar otros dispositivos, como por ejemplo sensores desde el mismo módulo, estos 5V son generados a partir de la tensión de alimentación que pasa por un circuito integrado regulador de tensión (LM7805) el cual no puede recibir tensiones mayores a los 12V, es decir que en caso de utilizar una alimentación con tensión superior a este valor sería necesario quitar el jumper del regulador para deshabilitarlo y así eliminar el riesgo de quemarlo, en nuestro caso la tensión de trabajo es inferior a los 12V y esto no supone ningún problema.

Para controlar la velocidad de los motores se debe controlar también el voltaje de entrada del motor y para lograr esto el módulo utiliza la modulación por ancho de pulsos (PWM). PWM es una técnica que permite ajustar el valor promedio de la tensión suministrada al motor utilizando secuencias de on-off. La tensión promedio depende del Duty Cycle y mientras mayor sea este, mayor tensión aplicada a los terminales de los motores, y a su vez mayor velocidad.



10.1.2 Motorreductores

Como ya se ha visto, para el proyecto estaremos utilizando motorreductores, es decir motores DC con escobillas que incluyen una caja reductora. La alimentación del motor se aplica mediante dos terminales conectados a las escobillas que hacen contacto con el rotor (parte giratoria). La circulación de corriente por las bobinas del rotor genera un campo magnético que interactúa con el de los imanes permanentes, generándose una fuerza que permite la rotación del eje conectado al rotor.



Figura: vista externa del motorreductor



Figura: vista interna del motorreductor

La caja reductora encierra a los engranajes que generan una reducción de la velocidad dada por la relación de transmisión o “gear ratio” y como consecuencia de dicha reducción se genera un aumento proporcional del torque en el eje de salida.

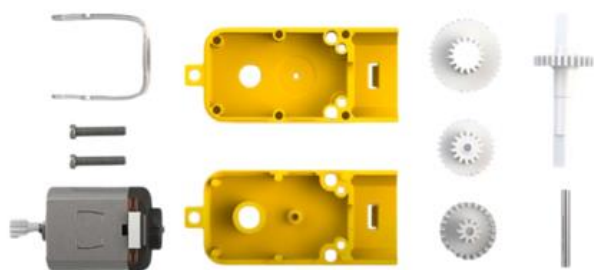


Figura: componentes del motorreductor

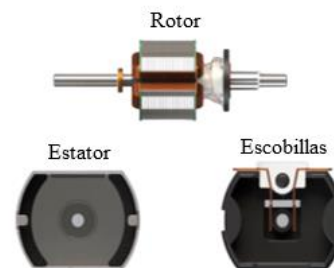


Figura: componentes del motor con escobillas

Respecto de la forma de uso de estos motores podemos aclarar que al igual que cualquier motor DC, requiere un controlador de corriente para controlarlo con un micro (en nuestro caso Arduino), esto es porque como mínimo este motor (y cualquier otro) consume más corriente de la que puede proveer el pin.

Estos motores de origen chino son de uso frecuente en proyectos de robótica y son genéricos, no incluyen ningún tipo de codificador incorporado, control de velocidad o retroalimentación posicional, básicamente su funcionamiento se resume en “tensión entra, rotación sale”. Los vendedores en tiendas online de estos motores ofrecen versiones con distintas relaciones de reducción, sin embargo en tamaño y peso todos son idénticos a simple vista, lo que dificulta saber si las especificaciones presentadas por el vendedor se corresponden realmente con el modelo pedido, según se ha investigado, en muchos casos los vendedores ofrecen un modelo con una determinada relación de reducción, y a la hora de recibir el producto se trata de una relación de reducción distinta, este tipo de situación se desprende del hecho de la poca información disponible sobre estos motores, de hecho, ni siquiera es posible saber con total certeza quien es el fabricante.

A causa de esto se ha realizado una ardua labor de búsqueda de las hojas de datos de estos motores hasta que eventualmente, mediante la contrastación de los datos en las hojas de datos halladas y las mediciones físicas realizadas en los ensayos con nuestros propios modelos (mediciones de corriente sin carga, corriente con carga máxima, RPM a 6V, peso y dimensiones físicas) y finalmente hemos dado con el que creemos puede ser modelo de nuestros motores. Se trata del modelo TGP01DA130 del fabricante TTMOTOR, más concretamente del 12215-48 (porque la relación de reducción es 1:48), este producto es el motor reductor visto en las imágenes anteriores, el cual incluye la caja reductora. Sin embargo, también hemos dado con el modelo de motor con escobillas que implementa este motor reductor, tratándose en este caso del TFE-130RA-12215 del fabricante Pololu Robotics and Electronics.

Habiendo realizado esta serie de aclaraciones el lector puede hallar las hojas de datos de los modelos mencionados en el anexo siendo avisado el lector de que estos parámetros de motores usados para el modelado matemático corresponden a modelos reales de motores, y probablemente (pero no necesariamente) corresponden o se asemejan a los de los motores utilizados en esta práctica. También estarán disponible en el anexo algunas imágenes que documentan y explican brevemente en que consistieron los ensayos realizados a los motores para realizar el contraste de los resultados obtenidos con las especificaciones de las hojas de datos.

10.1.3 Acelerómetro y Giroscopio MPU-6050

El sensor MPU-6050 es un IMU (Unidad de Medición Inercial) permite obtener aceleración lineal y velocidad de rotación sobre tres ejes, y en base a estos datos podemos obtener el ángulo de inclinación del robot.

Internamente el módulo IMU MPU-6500 está formado por un acelerómetro de 3 ejes, giroscopio de 3 ejes y un DPM (Procesador de Movimiento Digital) el cual toma los datos de los sensores y aplica un procesamiento para enviar información preprocesada y quitarle así carga de cálculo al microcontrolador de la placa Arduino. El dispositivo opera con una tensión de 3.3 V pero también incorpora un regulador que permite la alimentación directa a 5 V desde Arduino.

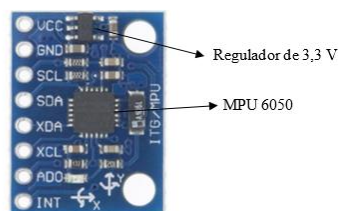
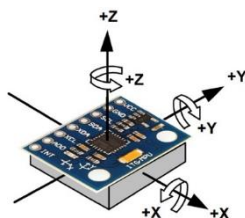


Figura: módulo IMU, con sensores de acelerómetro y giroscopio

El acelerómetro mide la aceleración inercial (cambio de velocidad en el tiempo). Los acelerómetros internamente tienen un MEMS (MicroElectroMechanical Systems) que de forma similar a un sistema masa resorte permite medir la aceleración lineal.



La aceleración es la variación de la velocidad por unidad de tiempo es decir la razón de cambio en la velocidad respecto al tiempo: $a = \frac{dv}{dt}$

Así mismo la segunda ley de Newton indica que en un cuerpo con masa constante, la aceleración del cuerpo es proporcional a la fuerza que actúa sobre él mismo: $a = \frac{F}{m}$

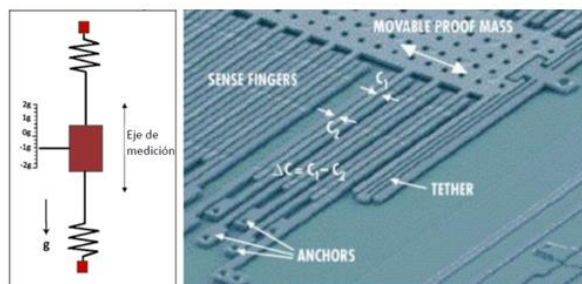


Figura: similitud entre sistema masa-resorte y sistema microelectromecánico

Con un acelerómetro podemos medir esta aceleración, teniendo en cuenta que a pesar que no exista movimiento, siempre el acelerómetro estará sensando la aceleración de la gravedad.

Con el acelerómetro podemos hacer mediciones indirectas, por ejemplo, si integramos la aceleración en el tiempo obtendremos la velocidad y si la integramos nuevamente obtendremos el desplazamiento, necesitando en ambos casos la velocidad y la posición inicial respectivamente.

Obtención de ángulos mediante el acelerómetro

La gravedad de la Tierra es perpendicular al suelo. Como es lógico, la IMU detecta la aceleración de la gravedad terrestre y gracias a ello se pueden usar las lecturas del acelerómetro para saber cuál es el ángulo de inclinación respecto al eje X o eje Y. Supongamos la IMU paralela con el suelo. Entonces, el eje Z marcará 9.8, y los otros dos ejes marcarán 0. Ahora girando 90 grados por el eje Y, es el eje X el que queda perpendicular al suelo, por lo tanto marcará la aceleración de la gravedad.

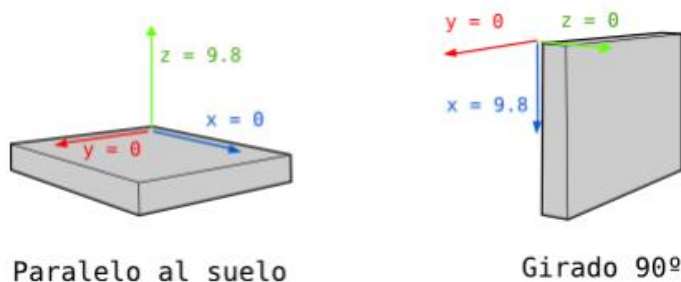


Figura 39: Obtención del ángulo de inclinación con el acelerómetro – fuente: https://robologs.files.wordpress.com/2014/10/imu_ac_ejemplos.png?w=300&h=119

Al saber que la gravedad es 9.8 m/s², y las medidas que dan los tres ejes del acelerómetro, por trigonometría es posible calcular el ángulo de inclinación de la IMU mediante las siguientes fórmulas:

$$\text{AnguloX} = \text{atan} \left(\frac{y}{\sqrt{x^2 + z^2}} \right)$$

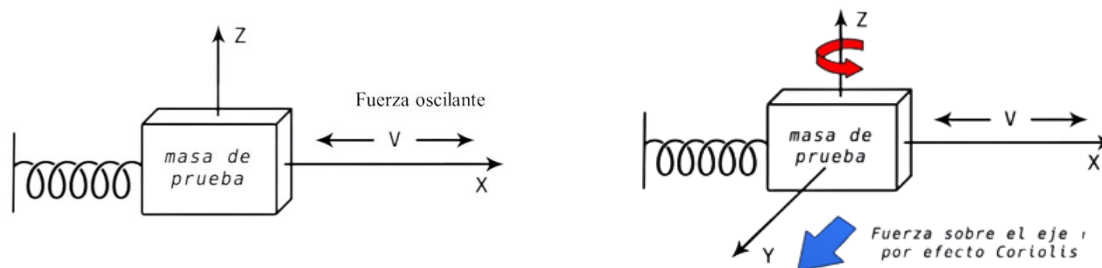
$$\text{AnguloY} = \text{atan} \left(\frac{x}{\sqrt{y^2 + z^2}} \right)$$

Figura 40: Fórmulas obtención ángulo del acelerómetro – fuente: <http://robologs.net/2014/10/15/tutorial-de-arduino-y-mpu-6050/>

Funcionamiento del Giroscopio

El giroscopio mide la velocidad angular alrededor de un eje (grados/s) (velocidad de giro o momento angular). La velocidad angular es la tasa de cambio del desplazamiento angular por unidad de tiempo, es decir que tan rápido gira un cuerpo alrededor de su eje:

Los giroscopios utilizan un MEMS (MicroElectroMechanical Systems) para medir la velocidad angular usando el efecto Coriolis. Con un giroscopio podemos medir la velocidad angular, y si se integra la velocidad angular con respecto al tiempo se obtiene el desplazamiento angular (posición angular si se sabe dónde se inició el giro).



Si sobre el eje z se aplica una fuerza de rotación, por efecto Coriolis aparecerá una fuerza sobre el eje y, y dado que se tratará de una aceleración lineal es posible medirla de la misma forma que en el caso del acelerómetro, es decir en este caso la fuerza en el eje y será proporcional a la velocidad de rotación sobre el eje z. Este razonamiento es similar para los 3 ejes.

Para la implementación en el robot, lo que nos interesará será obtener el ángulo de inclinación del robot respecto de una posición inicial.

Si se sabe el ángulo inicial de la IMU, se puede ir sumando el valor que marca el giroscopio para saber el nuevo ángulo a cada momento. Supongamos que iniciamos la IMU a 0°. Si el giroscopio realiza una medida cada segundo, y marca 3 en el eje X, tendremos el ángulo con esta sencilla fórmula:

$$\text{AnguloY} = \text{AnguloYanterior} + x\Delta t$$

Figura 41: Fórmula obtención ángulo del giroscopio - fuente: <http://robologs.net/2014/10/15/tutorial-de-arduino-y-mpu-6050/>

Consideraciones y Limitaciones del MPU 6050

Existen ciertas limitaciones para los sensores de acelerómetro y giroscopio que integra el MPU6050. Si bien con el acelerómetro podemos medir cualquier ángulo, los valores que detecta por sí mismo presentan problemas de ruido y efecto de jitter, mientras que los del giroscopio tienden a ser bastante precisos inicialmente, pero con el tiempo comienzan a presentar desviaciones debidas a la acumulación de errores que se propaga al realizar el cálculo del ángulo, en ambos casos esto puede afectar negativamente a la capacidad de nuestro robot para mantenerse equilibrado en posición vertical. Frente a esta dificultad tenemos dos opciones para evitar estas situaciones:

- Diseñar y utilizar un algoritmo de filtro complementario que estime el ángulo de inclinación en base al tratamiento matemático de los datos obtenidos tanto por el giroscopio como por el acelerómetro.
- Utilizar directamente la fusión de datos que permite el procesador de movimiento digital (DMP) incluido en el MPU6050. Si bien no hay información publicada acerca del algoritmo utilizado por el chip DMP para realizar los cálculos que fusionan las mediciones del giroscopio y del acelerómetro, el mismo sigue siendo útil porque presumiblemente tiene un algoritmo de fusión bastante bueno y porque los cálculos del DMP se realizan en el chip MPU-6050, lo que libera la potencia del procesador.

Para nuestro robot hemos decidido optar por la segunda opción, ya que desarrollar un algoritmo de filtro complementario es una tarea que demandaría tiempo extra y en base a la investigación realizada en blogs, foros y otros trabajos académicos, tenemos la certeza de que utilizar directamente el DPM proporciona mejores resultados que el uso de un filtro complementario.

Recursos externos de posible interés para el lector:

- Video comparativo de filtro complementario VS chip DPM: https://www.youtube.com/watch?v=2t-5CCyPJ74&ab_channel=geekmomprojects
- Enlace de descarga de documento hallado en la web, donde los autores hacen uso de un filtro complementario y muestran algunas conclusiones interesantes respecto del mismo (gráficos presentados en este documento adjuntos en el anexo):
https://www.researchgate.net/publication/324029194_A_new_Design_and_Control_of_a_Two-Wheel_Self-Balancing_Robot_using_the_Arduino_Microcontroller
- https://www.youtube.com/watch?v=qmd6CVrIHOM&t=1s&ab_channel=geekmomprojects

10.1.4 Alimentación

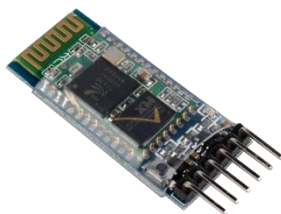
Los motores utilizados trabajan en el rango de 3 a 12 volts, por ello ha optado por utilizar tres baterías de iones de litio recargables de 3.7 V y 2200 mAh de autonomía, de forma que dispondremos de aproximadamente 11.1 V para la alimentación de los motores. Si bien este valor de tensión es excesivo para el tipo de motores utilizados el mismo será reducido vía software modificando el ciclo de trabajo de la señal PWM.

Por otra parte, se ha decidido alimentar la placa Arduino UNO y los demás módulos utilizados (MPU6050 y modulo bluetooth HC-05) con una batería de 9V.



10.1.5 Módulo Bluetooth HC-05

Este módulo es el encargado de establecer la comunicación bluetooth para controlar el robot, el mismo admite múltiples configuraciones vía software ingresando al modo de comandos AT, entre las configuraciones más útiles se encuentran el nombre del dispositivo, código para emparejamiento, baud rate, modo, etc. La configuración de mayor relevancia es el modo, pudiendo ser este maestro o esclavo, para este proyecto el HC-05 será utilizado en modo esclavo ya que será comandado por una app para dispositivos Android.



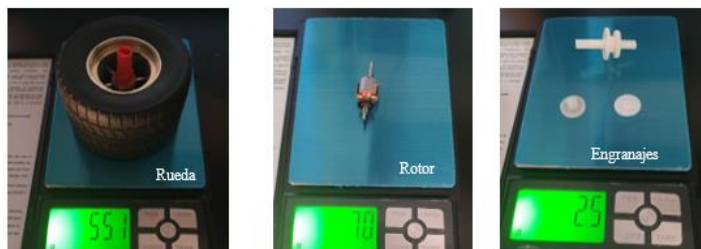
- Conexiones y configuración de comandos AT para el HC-05:
https://www.youtube.com/watch?v=ITvYbnr6aBo&list=WL&index=84&t=483s&ab_channel=INVOOTECHNICAGLOBALPVT.LTD
- Resumen de comandos AT: https://s3-sa-east-1.amazonaws.com/robocore-lojavirtual/709/HC-05_ATCommandSet.pdf

10.2 Mediciones

10.2.1 Masa del Sistema de Rotación

Como se vio en la sección de modelado matemático de la planta, la rodadura está formada por:

- Rueda
- Rotor del motor eléctrico
- Engranajes de la caja reductora
- Eje de salida de la caja reductora



10.2.2 Masa del Cuerpo Principal

Como se vio en la sección de modelado matemático de la planta, el chasis está por:

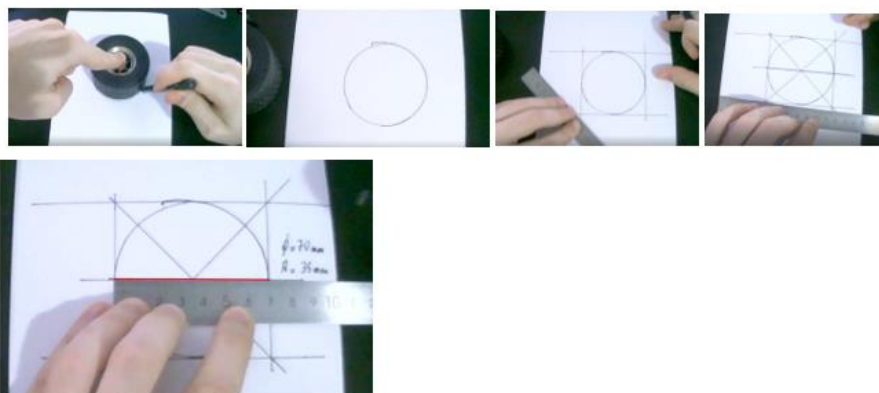
- Plataforma/s del robot y su contenido
- Estatores de ambos motores
- Rodamientos/bujes/rulemanes/bolilleros del eje del motor
- Carcasa de motorreductores



$$m_b = (610.6 \text{ g} - 2 * 26.1 \text{ g}) + 8.7 \text{ g} + 1.3 + 5.4 \text{ g} = 573.8 \text{ g} = 0.5738 \text{ Kg}$$

10.2.3 Radio de la Rueda

Radio de la Rueda R



$$R = 35\text{mm} = 0.035\text{m}$$

10.2.4 Longitud del Centro de la Rueda al Centro de Masa del Chasis

En primer lugar, se ha recurrido al modelo CAD aproximado del robot realizado mediante el software Fusion360 de AutoDesk, este permite visualizar la ubicación del centro de masas. El modelo realizado contiene los componentes principales (más pesados) del robot, sin embargo, esto se hace solo a nivel estimativo y principalmente visual, ya que además esta actividad nos permite introducirnos a la utilización de este software desconocido hasta el momento.



Figura: modelo CAD del Robot y su centro de masa determinado por el software Fusion360

Para hallar el centro de masa del chasis de una forma más confiable y rigurosa (pero aun aproximada) que la descripta anteriormente utilizaremos un método experimental utilizando el cuerpo principal o chasis del robot.

Recordemos primero algunos conceptos básicos, el centro de masa y el centro de gravedad de un cuerpo es el mismo punto cuando el cuerpo se encuentra en un campo gravitatorio uniforme (magnitud y dirección constante), a efectos prácticos esta coincidencia se cumple con precisión aceptable para casi todos los cuerpos sobre la superficie terrestre puesto que la disminución de intensidad gravitatoria es muy pequeña en toda la extensión de los cuerpos. Además, el centro geométrico de un cuerpo coincide con el centro de masa si el objeto es homogéneo o cuando se presentan ciertas propiedades como la simetría del cuerpo.

El método consistirá en suspender el chasis desde múltiples puntos de este hacia un pivot fijo mediante un hilo fino o tanza de forma que el objeto pueda moverse libremente. Se toman fotografías del robot pendiendo de distintos puntos, todas las fotografías estarán tomadas desde la misma posición, con un cuadro fijo que asegure los mismos ángulos y las mismas condiciones de luz en todas las fotografías. Una vez que se dispongan de varias fotografías (con dos es suficiente para hallar el CM) se recurre a software de diseño grafico y se siguen los siguientes pasos:

- Para ambas fotografías se prolongan las líneas de la tanza para que sea fácilmente visible.
- Se toma una de las fotografías de referencia.
- Se manipula la otra fotografía para que queden alineadas lo mejor posible con la fotografía de referencia y se la superpone a esta última.
- Se identifica el punto de intersección entre las líneas, el cual será el centro de masa.

Antes de seguir estos pasos notemos que el robot presenta un nivel aceptable de simetría en el plano zy (ver figura), entonces vamos a dividirlo imaginariamente mediante una especie de corte transversal en dos mitades iguales y asumiremos que el centro de masa se encuentra en el plano zy que contiene a la línea divisoria de la figura

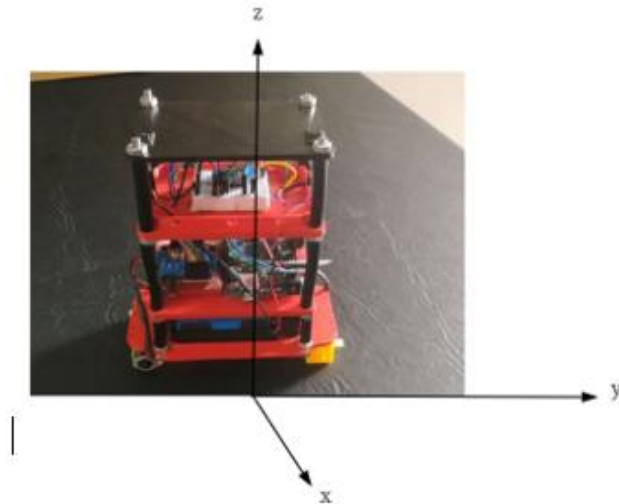


Figura: referencia de coordenadas



Figura: división imaginaria del robot en dos mitades

A continuación, se realiza el procedimiento descrito:

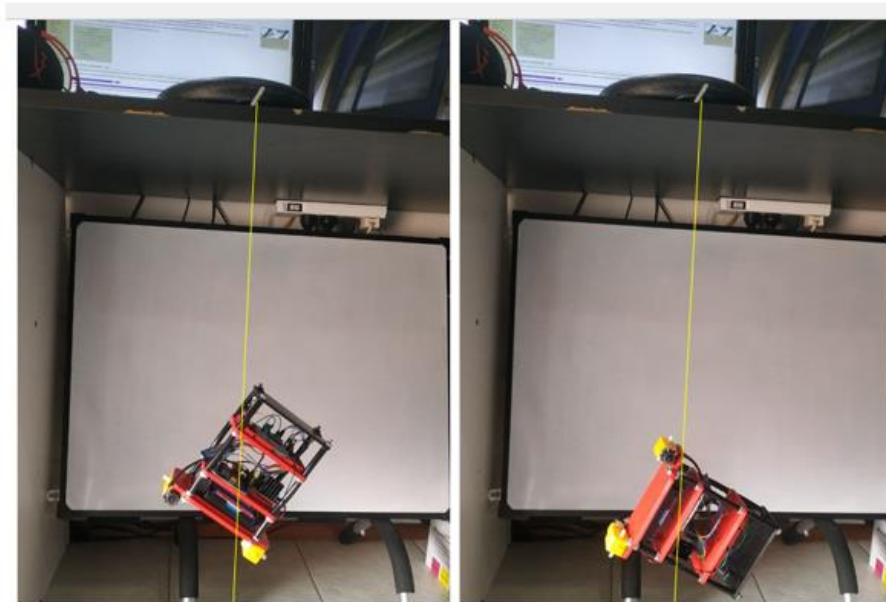


Figura: prolongación de las líneas de las que pende el chasis

Resultado de aplicar transparencia a una de las fotografías y luego superponerla con la fotografía de referencia:



Figura: fotografías superpuestas que muestran la ubicación aproximada del centro de gravedad del chasis

Podemos comprobar de esta forma que el centro de masa estimado mediante los dos métodos es a grandes rasgos muy similar y para los fines de este trabajo resulta suficiente. Tras realizar una simple medición con una regla de 0.5 mm de graduación se ha comprobado que la distancia desde el eje de salida de la caja reductora (o centro de la rueda) hasta el centro de masa es aproximadamente $l = 7\text{cm} = 0.07\text{m}$

10.2.5 Inercia Mecánica del Chasis J_b

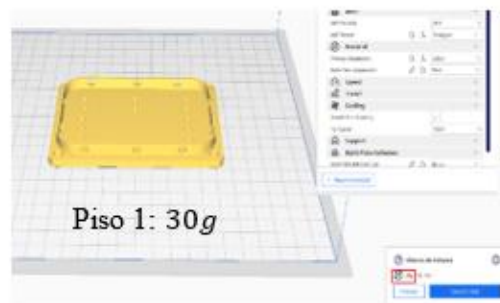
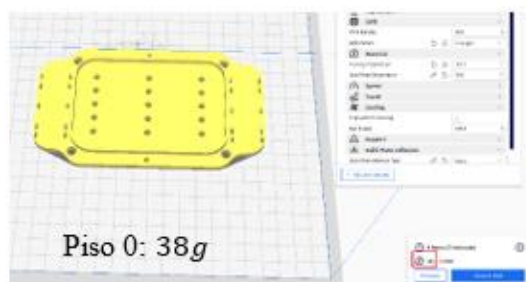
Para estimar este parámetro analizaremos el momento inercial de los elementos que lo componen considerándolos objetos rectangulares con masa distribuida uniformemente y los combinaremos utilizando el teorema de los ejes paralelos, también llamado teorema de Steiner.

Veamos primero la masa y dimensiones de los elementos que estaremos considerando:

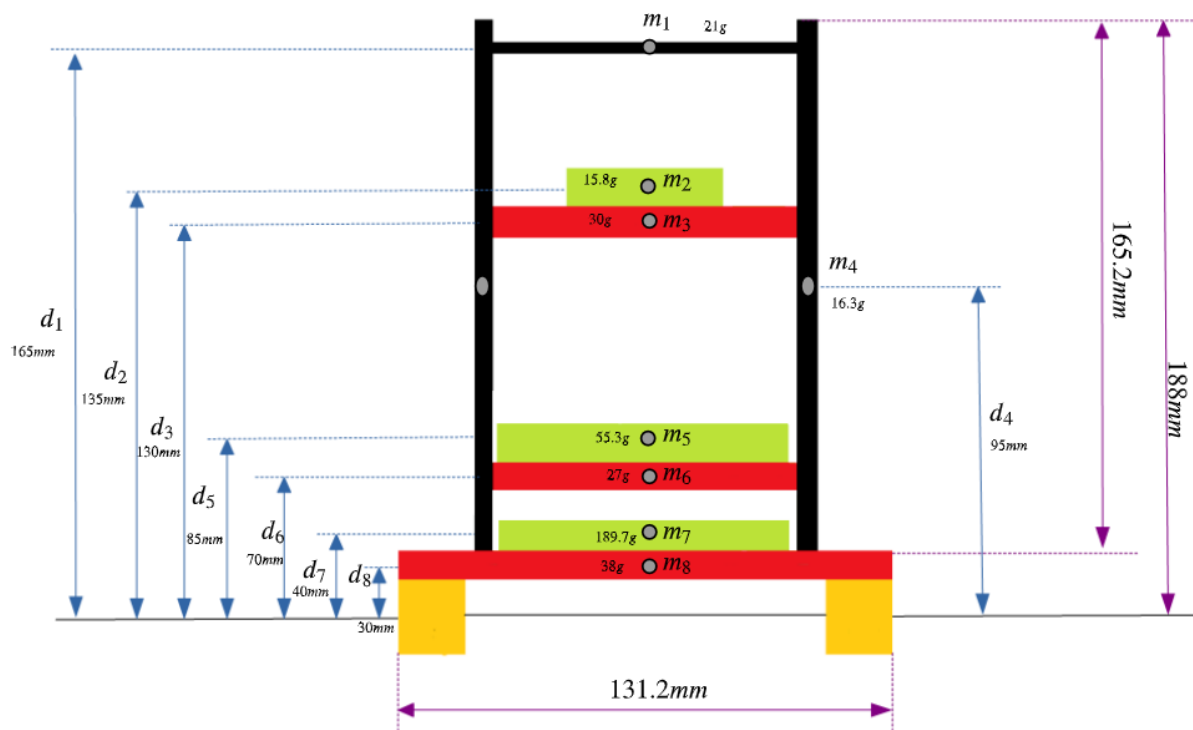


La varilla metálica fue pesada luego de construir el robot con lo que había sobrado, entonces considerando que la longitud de las varillas utilizadas en el robot es de 170mm , su peso es de aproximadamente **16.3g**.

Dado que las plataformas del robot fueron impresas en 3D podemos usar el software Ultimaker Cura el cual nos indica el peso que tendrán las piezas antes de imprimirlas en base al tipo de material utilizado y la cantidad de filamento empleada para el relleno de las piezas.



El modelo considerado, donde se plantean las distintas partes del chasis como objetos rectangulares simétricos y uniformes es el siguiente:

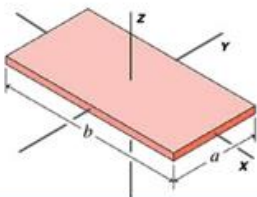


En este modelo hemos agrupado los componentes de la siguiente manera:

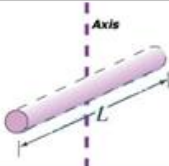
- El objeto rectangular de la base considera la masa de las baterías de iones de litio y de la batería de 9 V.
- El objeto rectangular en el primer piso considera la masa de la placa Arduino, el controlador de motores L298N y el módulo bluetooth HC-05.
- El objeto rectangular en el segundo piso considera la masa de una protoboard de 170 puntos y del sensor MPU6050.

A partir de aquí el cálculo de J_b lo realizaremos con ayuda de Matlab aplicando el teorema de Steiner y basándonos en la figura anterior y en la figura. El código utilizado estará disponible como anexo.

Moment of Inertia of Rectangular Plate Formula	
Mass moment of inertia about x axis	$I_x = (M/12) * a^2$
Mass moment of inertia about y axis	$I_y = (M/12) * b^2$
Mass moment of inertia about z axis	$I_z = (1/12) * M * (a^2 + b^2)$



Moment of Inertia of a Rod Formula	
Mass moment of inertia about axis	$I = (m/12) * L^2$



A continuación, se resumen las dimensiones y masas de las hipotéticas geometrías rectangulares que se utilizarán en Matlab:

	Elementos	Masa	Dimensiones
piso 2	Piso 3	$m_1 = 21g$	110 * 75.5 mm
	Objeto	$m_2 = 15.8g$	47 * 34 mm
	Piso 2	$m_3 = 30g$	110 * 75.5 mm
piso 1	Varillas	$m_4 = 16.3g$	165.2 mm
	Objeto	$m_5 = 55.3g$	100 * 65mm
	Piso 1	$m_6 = 27g$	110 * 75.5mm
base	Objeto	$m_7 = 189.7g$	110 * 75.5 mm
	Base	$m_8 = 38g$	150 * 85 mm

Programa realizado en Matlab:

```
close all, clear all; clc;
%Con este programa calcularemos el momento de inercia para el chasis
%utilizando el teorema de los ejes paralelos o teorema de Stainer.
% Utilizaremos las fórmulas de momento de inercia para placas
rectangulares
% y para varillas que se detallan en el informe en la sección de
mediciones

% Masas de los objetos considerados [Kg]
m1 = 0.021;
m2 = 0.0158;
m3 = 0.03;
m4 = 0.0163;
m5 = 0.0553;
m6 = 0.027;
m7 = 0.1897;
m8 = 0.038;
% Distancias de los objetos al eje de rotación del chasis [m]
d1 = 0.165;
d2 = 0.135;
d3 = 0.130;
d4 = 0.095;
d5 = 0.085;
d6 = 0.07;
d7 = 0.04;
d8 = 0.03;
%Lado a de las placas rectangulares [m]
a1 = 0.0755;
a2 = 0.034;
a3 = 0.0755;
a5 = 0.065;
a6 = 0.0755;
a7 = 0.0755;
a8 = 0.085;
%Longitud de la varilla [m]
l = 0.1652;

%Aplicamos el teorema de los ejes paralelos:  $J_i = J_{cm} + m_i \cdot l_i^2$  [Kg m2]
Jb = (m1/12)*(a1^2)+ m1*(d1^2) + (m2/12)*(a2^2)+ m2*(d2^2) +
(m3/12)*(a3^2)+ m3*(d3^2) + (m5/12)*(a5^2)+ m5*(d5^2) + (m6/12)*(a6^2)+
m6*(d6^2) + (m7/12)*(a7^2)+ m7*(d7^2) + (m8/12)*(a8^2)+ m8*(d8^2) + (m4/12)*l^2
```

Al ejecutar el programa obtenemos $J_b \cong 0.0024 \text{ Kg m}^2$

10.2.6 Inercia Mecánica de la Rodadura J_w

Para estimar este parámetro consideraremos únicamente la rueda y el rotor, ignorando los efectos de la caja reductora. Calcularemos los momentos inerciales por separado asumiendo distribución uniforme de las respectivas masas y luego los sumaremos.

Para estimar el momento inercial tanto de las ruedas como de los rotores consideraremos a estos elementos como cilindros sólidos:

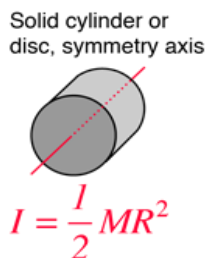


Figura: momentos de inercia de un cilindro solido

$$J_{rueda} = \frac{m_{rueda}}{2} R_{rueda}^2 = 0.5 * 0.0551 \text{ Kg} * (0.035\text{m})^2 = 0.000034 \text{ Kg m}^2$$

$$J_{rotor} = \frac{m_{rotor}}{2} R_{rotor}^2 = 0.5 * 0.007 \text{ Kg} * (0.005\text{m})^2 = 8.75 \times 10^{-8} \text{ Kg m}^2$$

$$J_w = J_{rueda} + J_{rotor} = 0.0000515 \text{ Kg m}^2$$

