



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería electrónica industrial y automática

**DISEÑO, IMPLEMENTACION Y CONTROL
DEL SISTEMA DE TRACCIÓN ELECTRICA
DE UN ROBOT AUTO-BALANCEADO**

Autor:

Sierra Valero, Daniel Camilo

Tutor:

**Herrero de Lucas, Luis Carlos.
Departamento de tecnología
electrónica.**

Valladolid, junio 2019

Resumen

En este proyecto se pretende desarrollar, implementar y controlar el sistema de tracción de un pequeño robot auto-balanceado. Dotándole la capacidad de mantener el equilibrio sobre dos ruedas y controlar su posición.

Para poder lograr lo todo lo anterior se utilizan herramientas como Matlab, Altium, Proteus y Android Studio.

Abstract

This project aims to develop, implement and control the traction system of a small self-balanced robot. Giving it the ability to maintain its' balance on two wheels and control its' position.

In order to achieve all of the above, tools such as Matlab, Altium, Proteus and Android Studio are used.

Palabras clave

Pendulo invertido, Arduino, robot auto-balanceado, Sistema de tracción eléctrica.





1 Índice

2	Índice de figuras	7
3	Índice de tablas	9
4	Índice de principales ecuaciones	9
5	Introducción	11
5.1	Transporte	11
5.1.1	Segway.....	11
5.2	Robótica	12
5.3	Robot auto-balanceado inteligente	13
6	Objetivos	15
7	Modelo matemático	19
7.1	Ecuaciones de modelo dinámico del sistema	20
7.2	Simulación de modelo matemático en Matlab	21
8	Construcción física del dispositivo.....	32
8.1	Motor	33
8.1.1	Rotor.....	33
8.1.2	Estatore	34
8.1.3	Relación de transmisión	34
9	Puentes en H	35
9.1	Introducción	35
9.2	Entradas y salidas de los puentes en H	35
9.3	Etapas del puente en H.....	36
9.3.1	Primera etapa: Puertas NOR.....	36
9.3.2	Segunda etapa: control de los MOSFET	37
9.3.3	Tercera etapa: MOSFET	40
9.3.4	Etapa auxiliar: CC/CC.....	40
9.4	Esquema de circuito.....	43
9.4.1	Primera etapa: Puertas NOR.....	44
9.4.2	Segunda etapa: control de los MOSFET.	45
9.4.3	Tercera etapa: MOSFET	45
9.4.4	Etapa auxiliar: convertidor CC/CC.....	46
9.5	Controlador	47
9.5.1	Texas Instruments.....	47
9.5.2	Arduino.....	48
9.6	Sensores	49
9.6.1	MPU6050.....	49
9.6.2	Encoder.....	50
9.7	Módulo de comunicación	51
10	Convertidor CC/CC	52
10.1	Interruptor en ON.....	53
10.2	Interruptor en OFF.....	54
11	Modelo de control	54
11.1	Entradas:.....	54
11.2	Salidas:	54



12	Estudio de control mediante Matlab	54
12.1	Lectura de encoder	54
12.2	Conclusiones.....	58
13	Estudio de control por medio del IDE Arduino	58
13.1	Cálculo de la velocidad	59
13.1.1	Conexión.....	59
13.1.2	Programación.....	59
13.2	Cálculo de ángulo.....	61
13.2.1	Conexión.....	61
13.2.2	Programación.....	62
13.3	Cálculo de PID	64
13.3.1	Programación.....	64
13.4	Bluetooth.....	65
13.4.1	Conexión.....	65
13.4.2	Programación.....	65
13.5	Enviar la señal a los motores	66
13.5.1	Conexión.....	66
13.5.2	Programación.....	67
13.6	Loop.....	68
13.6.1	Programación.....	68
14	Conexión eléctrica	69
15	Aplicación	71
15.1	Diseño.....	71
15.1.1	Dispositivos vinculados	71
15.1.2	Layout principal	72
15.2	Ficheros .java	73
15.2.1	Dispositivos vinculados	73
15.2.2	Layout principal	74
16	Valoración económica	78
17	Diseño estructural	79
18	Conclusiones	83
18.1	Líneas de continuación del proyecto.....	83
18.1.1	Cambio de motores.....	83
18.1.2	Estructura	85
19	Bibliografía	87
20	Anexos	89
20.1	Manual de usuario.....	89
20.2	Encender robot	89
20.3	Apagar robot.....	89
20.4	Conectar con Aplicación.....	89
20.5	Desconectar	90



2 Índice de figuras

Figura 1. Segway. Fuente: [1]	11
Figura 2. Funcionamiento de Segway. Fuente: [2].	12
Figura 3. Segway Robot. Fuente: [4].	12
Figura 4 Segway con brazos robótico. Fuente: [4].	13
Figura 5. Robot auto-balanceado Rocco [5].	13
Figura 6. Diagrama de Gantt.	17
Figura 7. Pendulo invertido. Fuente: [7]	19
Figura 8. Parametros del robot y matrices del sistema.	22
Figura 9. Visualización de matrices del sistema en la consola.	23
Figura 10. Comandos para definir el sistema, matriz de controlabilidad y el rango.	24
Figura 11. Matriz de controlabilidad y su rango.	24
Figura 12. Comando para calcular la matriz de observabilidad y su rango.	24
Figura 13. Matriz de observabilidad y su rango.	24
Figura 14. Diagrama de bloques que resuelve la ecuación 7	25
Figura 15. Salida del sistema en lazo abierto	25
Figura 16. Vector de polos y calculo del vector K.	26
Figura 17. Lazo cerrado del sistema.	26
Figura 18. Respuesta de las variables del sistema.	26
Figura 19. Señal de control.	27
Figura 20. Respuesta con polos 10 veces más alejados	27
Figura 21. Señal de control con polos 10 veces más alejados.	28
Figura 22. Calculo de vector K por regulación cuadrática lineal.	28
Figura 23. Respuesta del sistema con nuevo K.	29
Figura 24. Señal de control con nuevo K.	29
Figura 25. Lazo cerrado con referencia $x=1$	30
Figura 26. Respuesta del sistema con referencia $x = 1$	30
Figura 27. Lazo cerrado con mantenedores de orden 0.	31
Figura 28. Calculo de nuevo vector K para sistema con retenedores de orden 0.	31
Figura 29. Respuesta de control.	32
Figura 30. Respuesta del sistema.	32
Figura 31. Rotor.	33
Figura 32. Estator.	34
Figura 33. Relación de transmisión.	34
Figura 34. Diagrama de puente en H.	35
Figura 35. Diagrama lógico de puertas NOR.	36
Figura 36. Pines del integrado 74HC02D.	37
Figura 37. Pines del integrado L6384.	38
Figura 38. Tiempo muerto vs resistencia.	39
Figura 39. Montaje de L6384.	39
Figura 40. Diagrama de MC34063A.	41
Figura 41. Diagrama general de puente en H.	44
Figura 42. Puertas NOR.	44
Figura 43. Diagrama de conexión L6384D.	45
Figura 44. Diagrama de MOSFET.	45
Figura 45. Diagrama de CC-CC.	46
Figura 46. Puente en H.	46
Figura 47. LaunchPad [12].	47
Figura 48. MPU 5060.	49
Figura 49. Diagrama de funcionamiento de MPU 6050.	49
Figura 50. Encoder.	50
Figura 51. Módulo Bluetooth HC-05.	51
Figura 52. Reductor.	52



Figura 53. Buck [14].....	52
Figura 54. Gráfica de tensión y corriente en modo continuo [15].....	53
Figura 55. Diagrama de bloques de lectura de pulsos del encoder.	55
Figura 56. Conteo de pulsos en 10 segundos.	55
Figura 57. Retenedor de la suma de pulsos cada 10 segundos.....	55
Figura 58. Contador de pulsos y retenedor de la suma cada 10 segundos	56
Figura 59. Calculo de velocidad con un tiempo de muestreo $t = 0.25s$	56
Figura 60. Velocidad en r/min.	57
Figura 61. Calculo de velocidad con envío de señal PWM al motor.	57
Figura 62. Velocidad en r/min con la señal PWM enviada al motor.	58
Figura 63. Conexión Encoder.	59
Figura 64. Inicialización de todo lo necesario para obtener la velocidad.	60
Figura 65. Función “CalculoVelocidad”.	60
Figura 66. Diagrama de flujo del cálculo de la velocidad	61
Figura 67. Conexión de MPU5060.....	62
Figura 68. Inicialización de comunicación I2C.	63
Figura 69. Función “CalculoAngulo”.	63
Figura 70. Función “CalculoPID”.	64
Figura 71. Conexión de Bluetooth HC-05.....	65
Figura 72. Inicialización del Bluetooth.....	66
Figura 73. Función “RecepcionInformacion”	66
Figura 74. Conexión de Driver de los motores.	67
Figura 75. Función “EscribirSenalControl”.	67
Figura 76. Loop.....	68
Figura 77. Batería.	69
Figura 78. Esquema eléctrico general.	70
Figura 79. Lista de dispositivos vinculados.	71
Figura 80. Layout principal.	72
Figura 81. Diagrama de flujo de “Dispositivos Vinculados”.	73
Figura 82. Diagrama de flujo del botón “Subir”.	74
Figura 83. Diagrama de flujo del botón “Bajar”.	75
Figura 84. Diagrama de flujo del botón “Habilitar/Deshabilitar robot”.	76
Figura 85. Diagrama de flujo del botón “Enviar valores”.	77
Figura 86. Diagrama de flujo de botón “Desconectar”	77
Figura 87. Placa de plástico.	79
Figura 88. Tornillo espaciador.	79
Figura 89. Componentes de la base.....	80
Figura 90. Primera estructura.....	80
Figura 91. Segunda estructura.....	81
Figura 92. Tercera estructura.	81
Figura 93. Estructura final.	82
Figura 94. Piñones rectos [18].	83
Figura 95. Motor paso a paso [4].....	84
Figura 96. Piñones helicoidales [19].	84
Figura 97. Vincular dispositivo.	89
Figura 98. Menú principal.....	90



3 Índice de tablas

Tabla 1. Tareas.....	16
Tabla 2. Características de motor.....	33
Tabla 3. Tabla de verdad para las puertas NOR.....	37
Tabla 4. Características de LaunchPad.....	47
Tabla 5. Características de Arduino.....	48
Tabla 6. Características de MPU 5060.....	50
Tabla 7. Características de HC-05.....	51
Tabla 8. Características de SKL.....	52
Tabla 9. Costes de estructura.....	78
Tabla 10. Costes de electrónica.....	78
Tabla 11. Costes de componentes.....	79
Tabla 12. Costes de realización.....	79

4 Índice de principales ecuaciones

(1). Ecuación dinámica del péndulo invertido.....	20
(7). Ecuación de estado.....	20
(8). Ecuación de salida.....	21
(9). Ecuación de matriz de estados.....	21
(10). Ecuación de matriz de entrada.....	21
(18). Ecuación de condensador de salida.....	42
(19). Ecuación de tensión de salida.....	42





5 Introducción

En este proyecto se pretende desarrollar el diseño, implementación y control de un sistema de tracción para un robot auto-balanceado. Para ello se tendrá que definir robot auto-balanceado.

Un robot auto-balanceado es una máquina capaz de medir la inclinación que tiene con respecto al vector de la gravedad (entre otras variables) y actuar para estar lo más paralelo a este vector.

A partir de la información obtenida se tendrá que definir cuáles son los componentes que se tienen que utilizar para poder generar la estabilización del robot.

Cuando se tienen bien definidos los componentes del robot y se comprende cómo funcionan se procede a la implementación y control de cada una de las partes al sistema.

Se tiene que prestar especialmente atención en el diseño de la tracción eléctrica debido a es una parte fundamental de este robot y al enfoque electrónico que tiene este proyecto.

Por lo que se comienza viendo de forma general como en la actualidad se ha podido solucionar el problema de diseñar un robot que sea capaz de mantenerse erguido.

5.1 Transporte

5.1.1 Segway

Un Segway [1] es un vehículo de transporte eléctrico de dos ruedas con un control de balanceo. Este dispositivo, Figura 1, dispone de dos motores eléctricos dispuestos en los ejes de las ruedas donde se transmite la energía mecánica haciendo que cuando el usuario se inclina hacia adelante, los motores giran para producir el movimiento manteniendo el ángulo constante, Figura 2.



Figura 1. Segway. Fuente: [1]

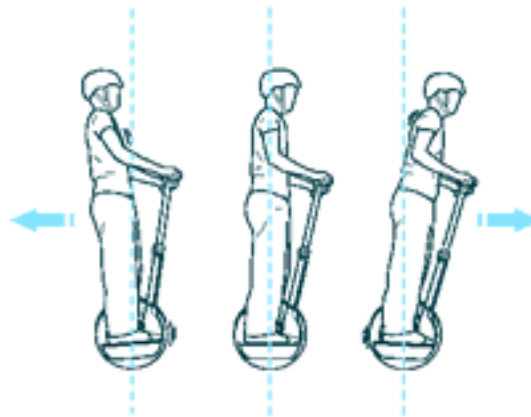


Figura 2. Funcionamiento de Segway. Fuente: [2].

5.2 Robótica

El dispositivo Segway que se ha visto en el apartado anterior se puede utilizar para añadirle un sistema inteligente como se realizó en el Segway Robot [3], Figura 3, que además de ser capaz de mantenerse perpendicular al suelo, se quiere que sirva de asistente para una persona.



Figura 3. Segway Robot. Fuente: [4].

Por lo que este dispositivo puede seguir al usuario gracias a un sistema de cámaras incorporadas y sensores de profundidad. También se le añade reconocimiento de voz y hasta brazos robóticos.



Figura 4 Segway con brazos robótico. Fuente: [4].

5.3 Robot auto-balanceado inteligente

Este robot [5] llamado Rocco, Figura 5, se basa en el mismo sistema estructural que los anteriores pero su funcionalidad está ligada a la interactividad con el usuario. Para ello tiene luces, altavoces y micrófonos.



Figura 5. Robot auto-balanceado Rocco. Fuente: [5].





6 Objetivos

Viendo como en la actualidad se está haciendo uso de este sistema eléctrico se plantea como objetivo desarrollar, implementar y controlar el sistema de tracción de un pequeño robot auto-balanceado.

Teniendo todo lo anterior en cuenta se fijan como objetivos:

1. Modelo matemático.
2. Selección de motores.
3. Selección de driver de motores.
4. Selección de controlador.
5. Selección de sensores.
6. Construcción y programación del robot.
7. Comunicación vía Bluetooth con dispositivo Android.

En capítulos posteriores se ira detallando cada uno de estos puntos para poder obtener el robot y se ira trabajando con diferentes Software dependiendo de las necesidades, por ejemplo: Matlab o Altium.

Se entiende que es necesario el uso de herramientas específicas de la rama de electrónica para poder desarrollar el proyecto, como un osciloscopio y un multímetro. También es necesario atornilladores y posiblemente un taladro.

Para poder tener un proyecto con objetivos concretos se tiene que plantear como ruta de programación, la establecida en el diagrama de Gantt dispuesto en la Figura 6 y sus tareas en la Tabla 1.



Id	Nombre	Duración	Comienzo	Fin	Predecesoras
0	Robot Autobalanceado	22,5 días	03 junio 2019 9:00	03 julio 2019 13:00	
1	Búsqueda del Robot Autobalanceado	5 horas	03 junio 2019 9:00	03 junio 2019 16:00	
2	Modelado del Sistema	10 horas	03 junio 2019 16:00	04 junio 2019 18:00	1
3	Simulación en Matlab	5 horas	04 junio 2019 18:00	05 junio 2019 13:00	2
4	Búsqueda de componentes	10 horas	05 junio 2019 15:00	06 junio 2019 17:00	3
5	Estudio del Motor	24 horas	06 junio 2019 17:00	11 junio 2019 17:00	4
6	Estudio del Puente en H	20 horas	11 junio 2019 17:00	14 junio 2019 11:00	5
7	Estudio del Microcontrolador	2 horas	14 junio 2019 11:00	14 junio 2019 13:00	6
8	Estudio del controlador mediante Matlab	15 horas	14 junio 2019 15:00	18 junio 2019 12:00	7
9	Estudio del control mediante el IDE de Arduino	25 horas	18 junio 2019 12:00	21 junio 2019 13:00	8
10	Construcción del Robot	25 horas	18 junio 2019 12:00	21 junio 2019 13:00	8
11	Aplicación Movil	10 horas	21 junio 2019 15:00	24 junio 2019 17:00	9;10
12	Valoración Económica	5 horas	24 junio 2019 17:00	25 junio 2019 12:00	11
13	Mejora del Proyecto	3 horas	25 junio 2019 12:00	25 junio 2019 17:00	12
14	Redactar proyecto	30 horas	25 junio 2019 17:00	01 julio 2019 13:00	13
15	Preparar presentación	16 horas	01 julio 2019 15:00	03 julio 2019 13:00	14

Tabla 1. Tareas.

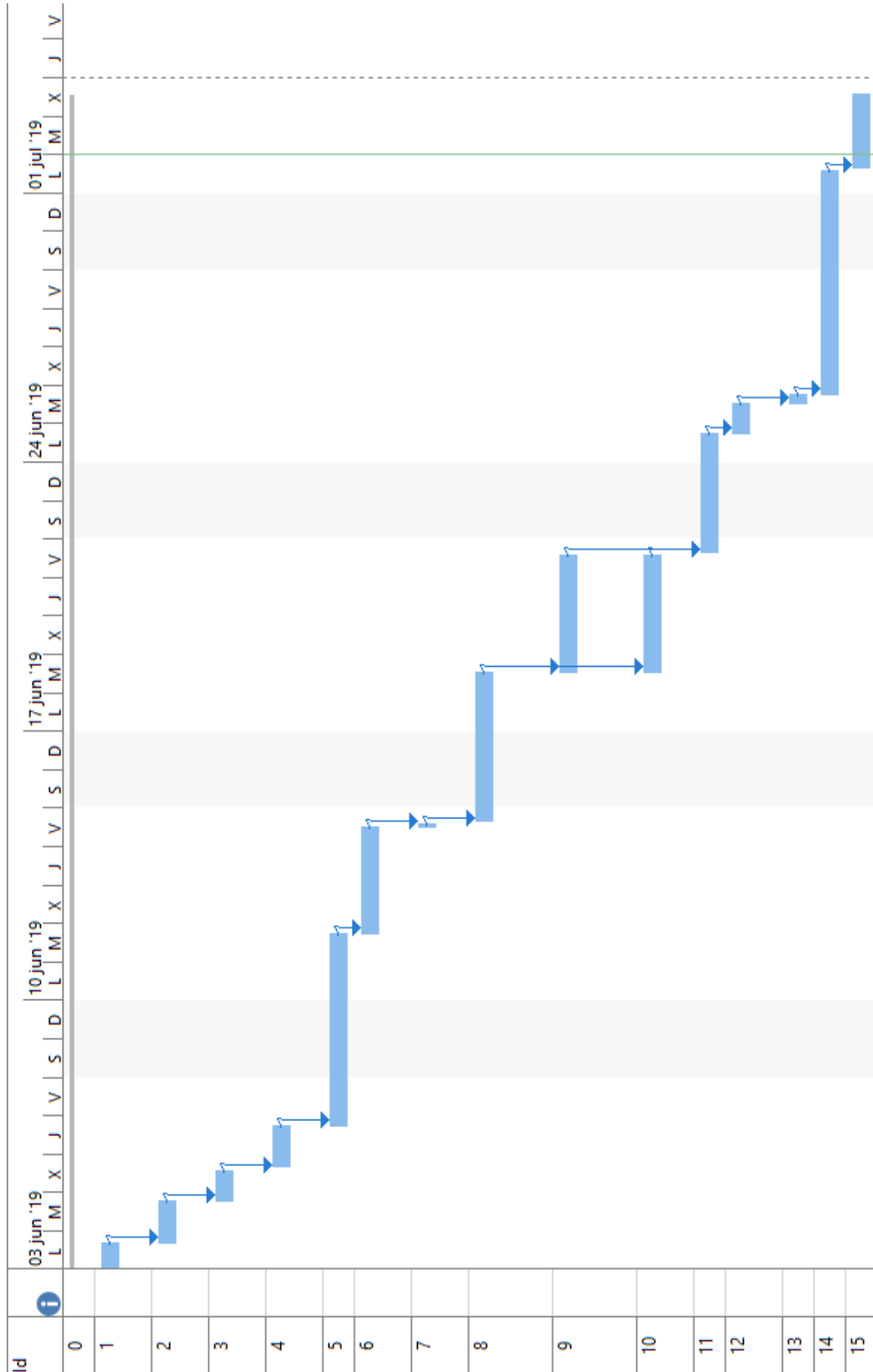


Figura 6. Diagrama de Gantt.



7 Modelo matemático

Para poder realizar este proyecto con éxito se tiene que realizar un estudio exhaustivo del proceso que se quiere controlar y para eso se van a utilizar los conocimientos que se tienen de la mecánica.

Este robot se puede asemejar de forma ideal a un péndulo invertido [6] debido a que este mecanismo tiene el centro de masa por encima de su punto de pivote al contrario que un péndulo tradicional.

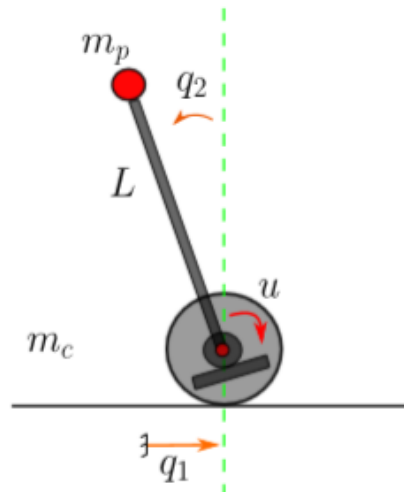


Figura 7. Péndulo invertido. Fuente: [7]

Los parámetros que se necesitan para definir el péndulo que se muestran en la Figura 7 son:

- m_p : masa total de la estructura.
- m_c : masa de las ruedas, motores y batería.
- q_2 : grados de giro con respecto a la vertical.
- L : distancia entre el centro de masa del robot del eje.
- q_1 : desplazamiento del robot.
- u : la fuerza que ejerce el motor.

En el caso del mecanismo, Figura 7, que se está estudiando es inestable debido a que si se deja en reposo la masa que se tiene de forma general caería hasta que el valor de q_2 valiera 90° por lo que se tiene que generar un momento que provoca un movimiento del punto de pivote para que el vector de la fuerza gravitatoria de la masa m_p sea lo más paralelo al vector que une al punto de pivote con el centro de masa de m_p .

Partiendo de la distribución geométrica que se tiene en la Figura 7, se definen tanto las ecuaciones [8] del sistema como la programación en Matlab que se tienen estudiadas con detenimiento por la universidad de Kaiserslautern [7].



7.1 Ecuaciones de modelo dinámico del sistema

Para desarrollar las ecuaciones dinámicas de este sistema se puede partir haciendo uso de diferentes métodos como el de Lagrange o el de Newton-Euler. Con cualquiera de ellos se puede llegar a la ecuación (1).

$$M(q)\ddot{q} + h(q, \dot{q}) = g_q u \quad (1)$$

Cada una de las variables de la (1) se definen con las ecuaciones:

$$M(q) = \begin{pmatrix} m_c + m_p & -Lm_p \cos q_2 \\ -Lm_p \cos q_2 & L^2 m_p \end{pmatrix} \quad (2)$$

$$h(q, \dot{q}) = \begin{pmatrix} Lm_p \dot{q}_2^2 \sin q_2 \\ -Lgm_p \sin q_2 \end{pmatrix} + \begin{pmatrix} d_1 \dot{q}_1 \\ d_2 \dot{q}_2 \end{pmatrix} \quad (3)$$

$$g_p = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (4)$$

Con una representación de espacios de estados se pretende tener de un modo compacto, conveniente para modelar y analizar este sistema ya que tiene varias entradas y salidas.

En (3) se tiene las variables d_1 y d_2 que representan la fricción del carro y del eje respectivamente. Haciendo uso de las ecuaciones (2-4) se va a representar el sistema en espacio de estados a partir de la ecuación (5).

$$x = \begin{pmatrix} q \\ \dot{q} \end{pmatrix} \quad (5)$$

Por lo tanto, se obtendría la ecuación (6).

$$\dot{x} = F(x) + G(x)u = \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ -M^{-1}(q)h(q, \dot{q}) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -M^{-1}(q)g_p \end{pmatrix} u \quad (6)$$

El objetivo es mantener el robot erguido por lo que el sistema se tiene que linealizar con q_1 y q_2 igual a 0.

La representación de espacios de estados de un sistema lineal se describe con las ecuaciones (7-8).

$$\dot{x} = Ax + Bu \quad (7)$$

$$y = Cx + Du \quad (8)$$



Siendo las variables:

- x es el estado del sistema.
- \dot{x} variación del estado del sistema.
- y es la salida del sistema.
- u es el vector de entrada (o control).
- A es la matriz de estados.
- B es la matriz de entrada.
- C es la matriz de salida.
- D es la matriz de transmisión directa.

Los matrices A y B son las que se definen en [\(9-10\)](#).

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{gm_p}{m_c} & -\frac{d_1}{m_c} & \frac{d_2}{Lm_c} \\ 0 & \frac{g(m_p + m_c)}{Lm_c} & -\frac{d_2}{Lm_c} & \frac{d_2(m_p + m_c)}{L^2m_cm_p} \end{pmatrix} \quad (9)$$

$$B = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{m_c} \\ \frac{1}{Lm_c} \end{pmatrix} \quad (10)$$

Los valores de la matriz C dependen de la variable que se pretende controlar por lo que en se hará una un estudio en el siguiente apartado para conocer su valor.

La matriz D se compone de 0 para este sistema debido a que este sistema no se va contemplar que tenga transmisión de información en el lazo de control.

7.2 Simulación de modelo matemático en Matlab

Para poder comprobar que el sistema descrito anteriormente se va a proceder a simularlo en la herramienta Matlab y comprobar que puede llegar a estabilizarse. Para ello se comienza declarando parámetros geométricos del robot y las matrices que componen el sistema en un Script y por último construimos el sistema con el comando “ss” con las matrices que definen el sistema en espacio de estados, Figura 8. Recordando que primero se va tomar como salida el valor q_2 . Esta simulación está basada en el estudio [9].

Los parámetros elegidos en está simulación se definieron con la intención de analizar de forma general si se puede llegar a controlar un dispositivo de estas características.



```

%% Parametros que definen al sistema
mc = 1.5; %Masa de las ruedas, motores y batería.
mp = 0.5; %Masa de la estructura.
g = 9.82; %Constante de gravedad.
L = 1; %Longitud del pendulo.
d1 = 1e-2; %Fricción del carro con el suelo.
d2 = 1e-2; %Fricción con el eje.

%% Las matrices que definen el sistema en diagrama de estados

% Siendo el sistema xpunto = Ax + Bu

A = [0,          0,          1,          0;
      0,          0,          0,          1;
      0,          (g*mp)/mc,  -d1/mc,      -d2/(L*mc);
      0, (g*(mp + mc))/(L*mc), -d1/(L*mc), -(d2*mp + d2*mc)/(L^2*mc*mp)];

B = [0; 0; 1/mc; 1/(L*mc)];

%% Salida

%C = [0; 1; 0; 0]; %La salida de q2
C = [1; 0; 0; 0]; %La salida de q1
D = 0;

%% Construir el sistema
sys = ss(A,B,C',D)

```

Figura 8. Parametros del robot y matrices del sistema.

Al ejecutar el código de la Figura 8, se obtiene las matrices A, B, C' y D con sus respectivos valores que dependen de las constantes geométricas del robot Figura 9.



```

sys =

A =
      x1      x2      x3      x4
x1      0      0      1      0
x2      0      0      0      1
x3      0      3.273 -0.006667 -0.006667
x4      0      13.09 -0.006667 -0.02667

B =
      u1
x1      0
x2      0
x3      0.6667
x4      0.6667

C =
      x1 x2 x3 x4
y1      0  1  0  0

D =
      u1
y1      0

Continuous-time state-space model.

```

Figura 9. Visualización de matrices del sistema en la consola.

Recordando que para poder controlar un sistema tiene que ser controlable y observable. Basándose en el estudio que realiza Ricardo J. Mantz [10]. Estos dos conceptos se definen como:

- Controlabilidad: un sistema es controlable cuando es posible transferir al sistema de un estado a otro estado deseado en un tiempo finito.
- Observabilidad: Un sistema es observable cuando un estado inicial $X(0)$ se puede predecir a partir de la observación de $y(kT)$ sobre un número de periodos de muestreo finito.

Se tiene que tener en cuenta que la controlabilidad y la observabilidad se asocian a entradas y salidas del sistema. Por lo tanto, si se mide otra variable, el sistema puede pasar de ser observable a no serlo. Este resultado se debe a que las matrices que conforman el modelo son el resultado de la elección de las variables de entrada y de salida seleccionadas.

Por lo que se tiene que comprobar la controlabilidad del sistema para ello se toma como salida q_2 , se halla la matriz de controlabilidad con el comando "ctrb". A partir de esta matriz se calcula su rango y si es igual que el tamaño del sistema se considera un sistema controlable, Figura 10.

Se dice que el rango es igual al orden del sistema es controlable por el Test de controlabilidad de Kalman [10].



```
%% Construir el sistema
sys = ss(A,B,C',D)
Sc = ctrb(sys)
RangoC = rank(Sc)
```

Figura 10. Comandos para definir el sistema, matriz de controlabilidad y el rango.

El resultado que se obtiene se encuentra en la Figura 11. Como se puede apreciar la matriz de controlabilidad tiene rango 4 por lo que es controlable.

```
Sc =

    0    0.6667   -0.0089    2.1824
    0    0.6667   -0.0222    8.7295
  0.6667  -0.0089    2.1824   -0.1455
  0.6667  -0.0222    8.7295   -0.5383

RangoC =

    4
```

Figura 11. Matriz de controlabilidad y su rango.

Para poder comprobar la observabilidad se genera la matriz de observabilidad Figura 12 y se comprueba que tiene el mismo rango que el sistema Figura 13. Debido a lo establecido en el test de observabilidad de Kalman [10]. Se comprueba que no es observable por lo que se va a comprobar con q_1 como salida.

```
So = obsv(sys)
RangoO = rank(So)
```

Figura 12. Comando para calcular la matriz de observabilidad y su rango.

```
So =

    0    1.0000    0    0
    0     0    0    1.0000
    0   13.0933  -0.0067  -0.0267
    0   -0.3710    0.0002   13.0941

RangoO =

    3
```

Figura 13. Matriz de observabilidad y su rango.



Cambiamos la matriz C para definir como q_1 como salida y obtenemos un sistema controlable y observable por podemos proseguir con la simulación recordando que se tiene que variar el ángulo de inclinación para estabilizar el sistema.

Ahora se tiene que crear un modelo en Simulink para que resuelva la ecuación (7) teniendo como entrada u y como salida la x e y y como se puede observar en Figura 14.

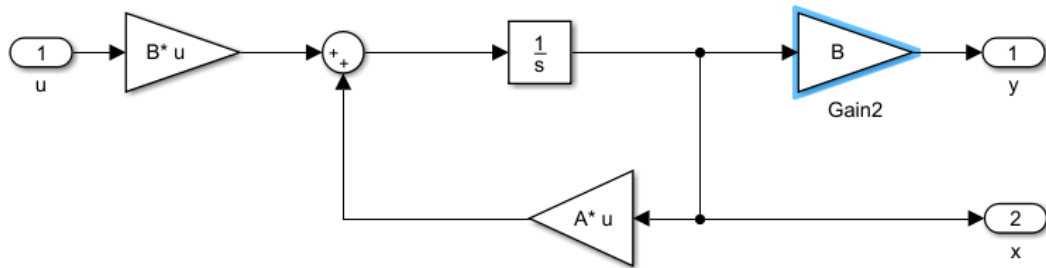


Figura 14. Diagrama de bloques que resuelve la ecuación 7.

Si se visualiza la salida que se obtiene en x se obtiene lo que se tiene en la Figura 15. En esta simulación el sistema es inestable.

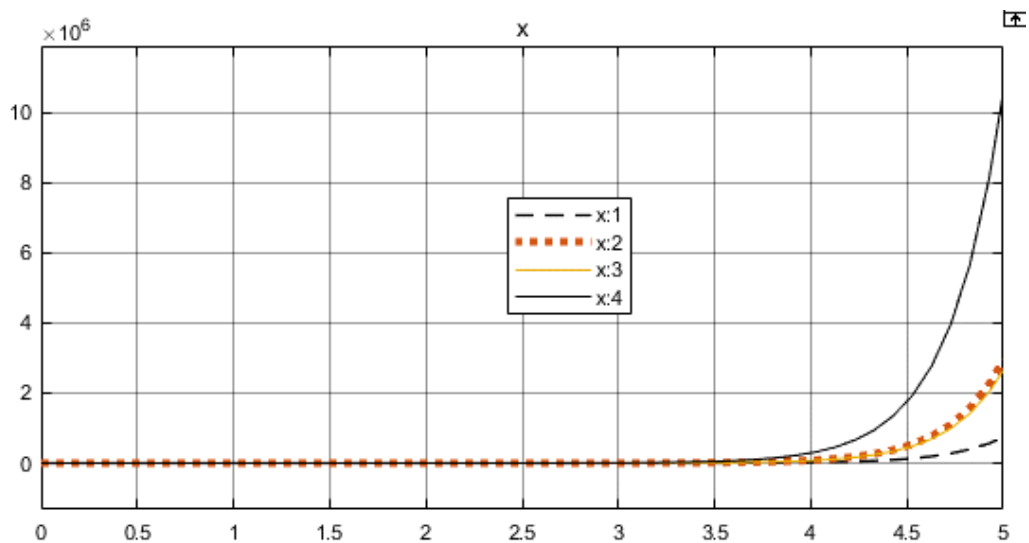


Figura 15. Salida del sistema en lazo abierto.

Se recuerda que el vector X son las variables estado del sistema siendo:

- $X(1): q_1$
- $X(2): q_2$
- $X(3): \dot{q}_1$
- $X(4): \dot{q}_2$

Por lo que se va a diseñar un controlador se va a definir, Figura 16, donde queremos tener los polos del sistema para luego utilizar el comando “acker” [11] que recibe las matrices A y B del sistema y calcula un vector K que al multiplicarse por la salida y retroalimentarlo al sistema lo lleva a los polos deseados “des_pole”.

```
%% Controlador
des_pole = [-3; -3; -3; -3];
K = acker(A,B,des_pole)
```

Figura 16. Vector de polos y calculo del vector K.

Después de generar el vector K que se necesita se introduce en Simulink la retroalimentación al sistema con un setpoint de 0. Como se muestra en la Figura 17.

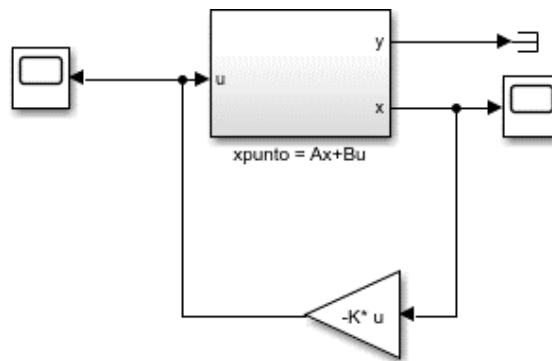


Figura 17. Lazo cerrado del sistema.

Partiendo de ese esquema se obtiene como salida la Figura 18 y la señal de control será la que se muestra en la Figura 19. Se puede ver que el sistema se estabiliza en lazo cerrado.

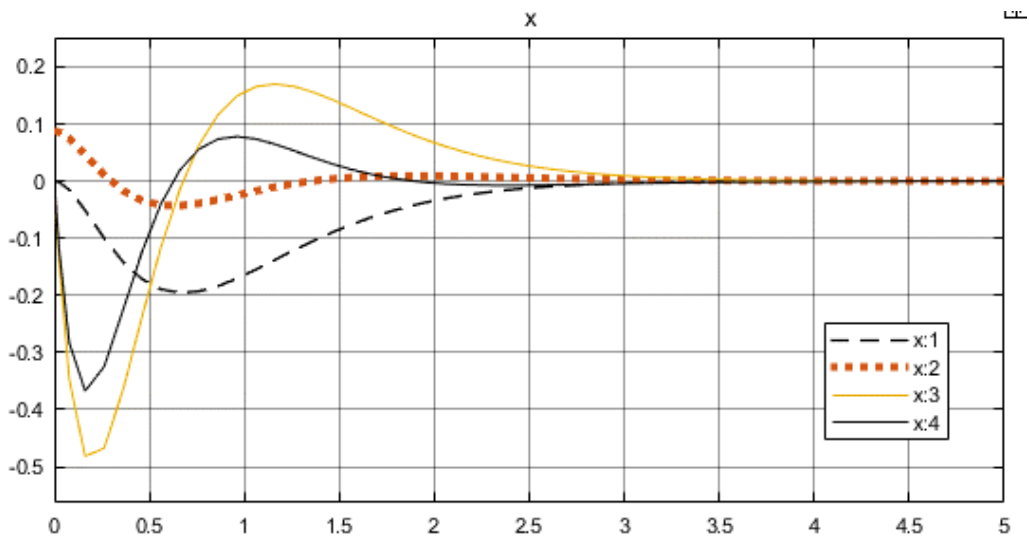


Figura 18. Respuesta de las variables del sistema.

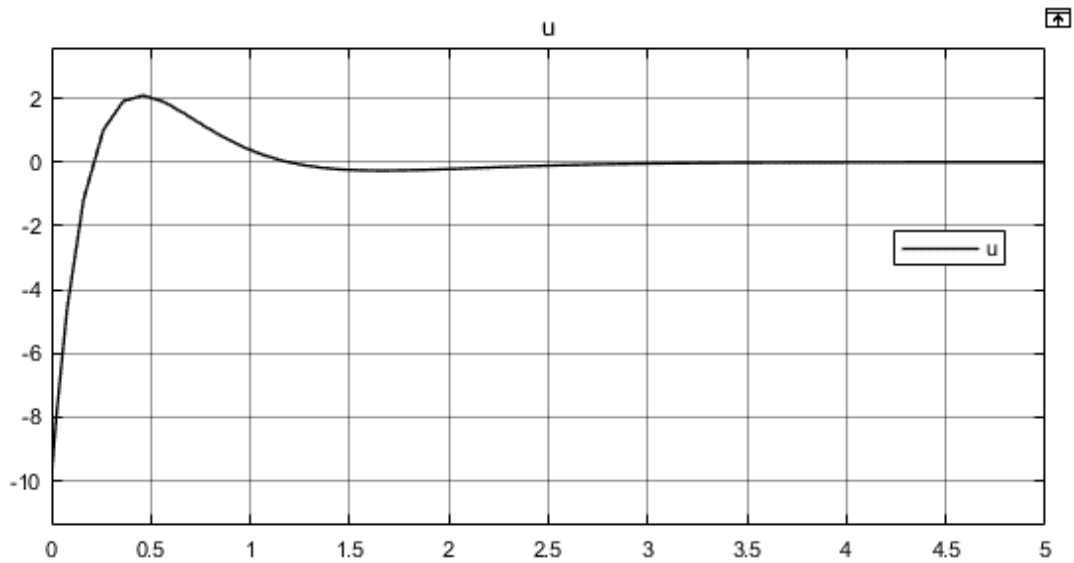


Figura 19. Señal de control.

Si se quiere un sistema más o menos rápido simplemente se va cambiando el lugar de los polos. Recordando que entre mayor sea la velocidad de estabilización, la señal de control tendrá que ser mayor. Como se puede observar en la Figura 20 (salida) y Figura 21 (señal de control) haciendo que los polos estén 10 veces más alejados del origen.

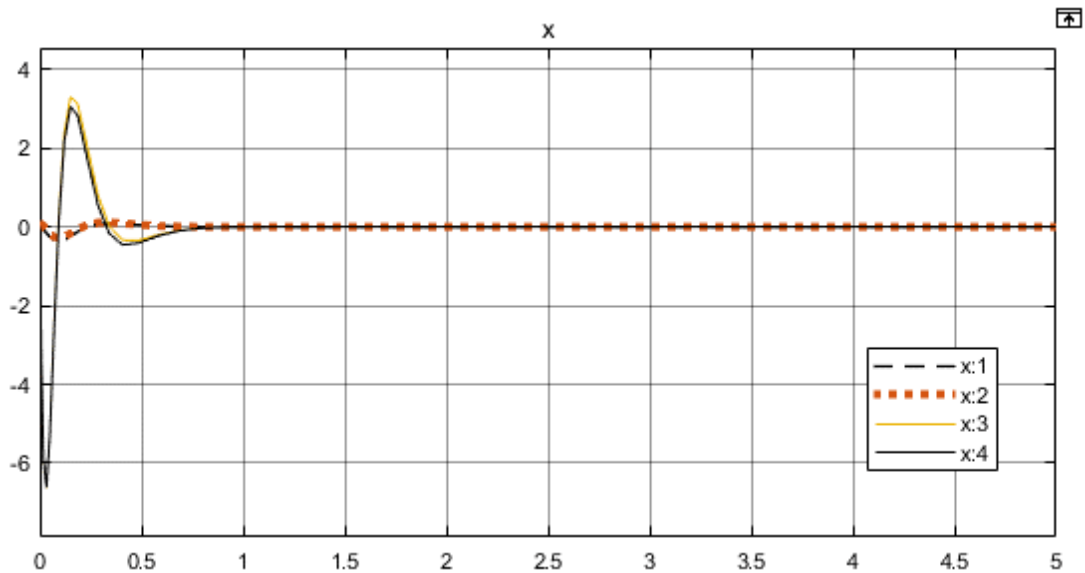


Figura 20. Respuesta con polos 10 veces más alejados.

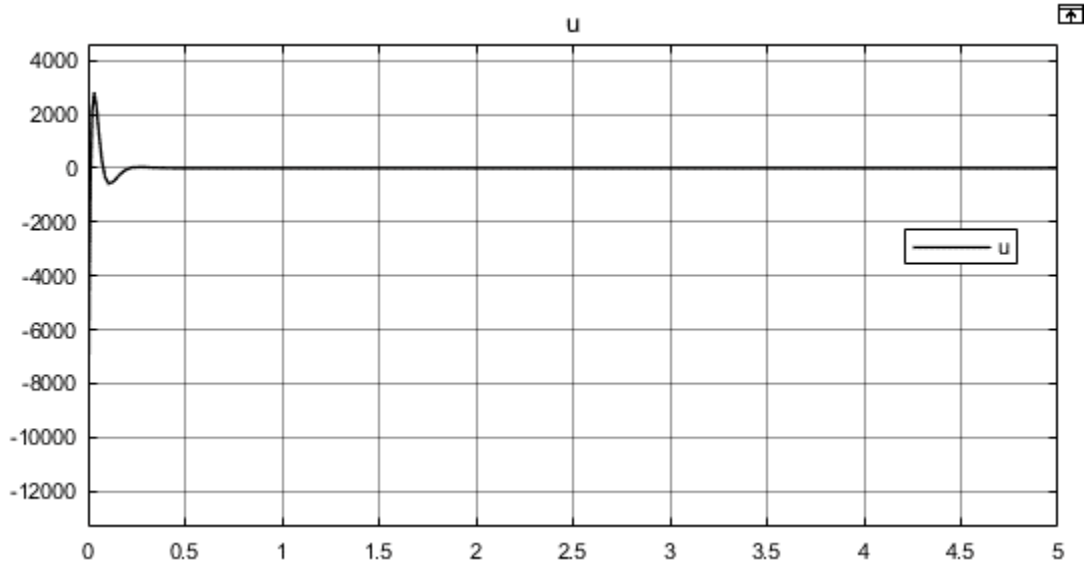


Figura 21. Señal de control con polos 10 veces más alejados.

En la Figura 21 se aprecia como la señal de control oscila entre números muy grandes por lo que se necesita de un actuador que tenga ese rango de acción, generando esa fuerza.

Para encontrar el mejor controlador se va a hacer uso del “lqr” es la función que utiliza el método de regulación cuadrática lineal y que devuelve la ganancia óptima del controlador suponiendo una planta lineal, una función de coste cuadrático y una referencia igual a 0. Dependiendo de los valores de Q y R se asigna prioridad a distintas variables de estado. Después de unas cuantas pruebas se programa como se puede ver en la Figura 22 y su resultado son la Figura 23 y la Figura 24.

```
Q = 20*eye(4);
R = 0.1;
K_lqr = lqr(A,B,Q,R)
```

Figura 22. Calculo de vector K por regulación cuadrática lineal.

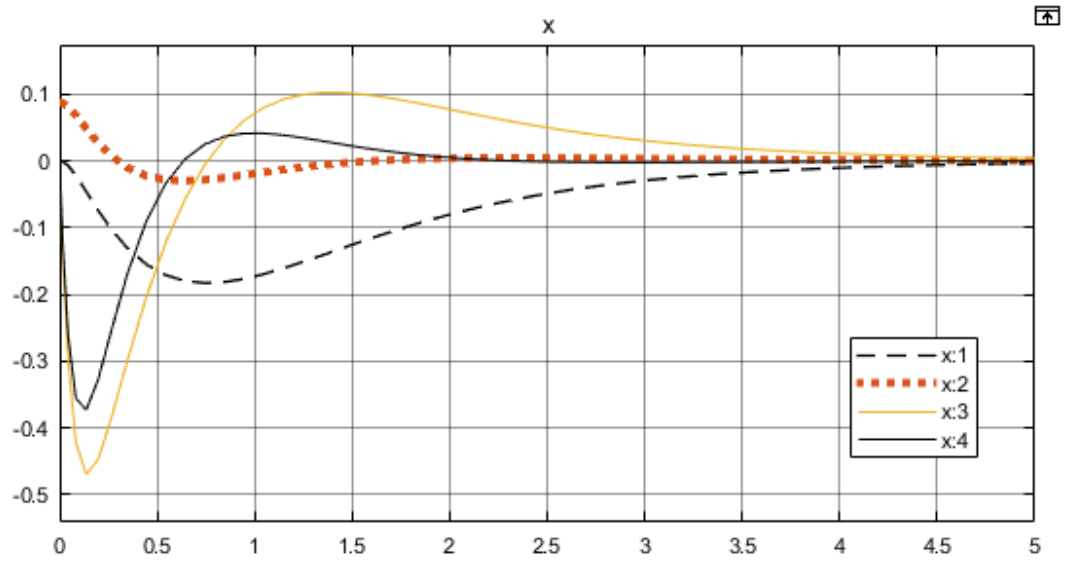


Figura 23. Respuesta del sistema con nuevo K.

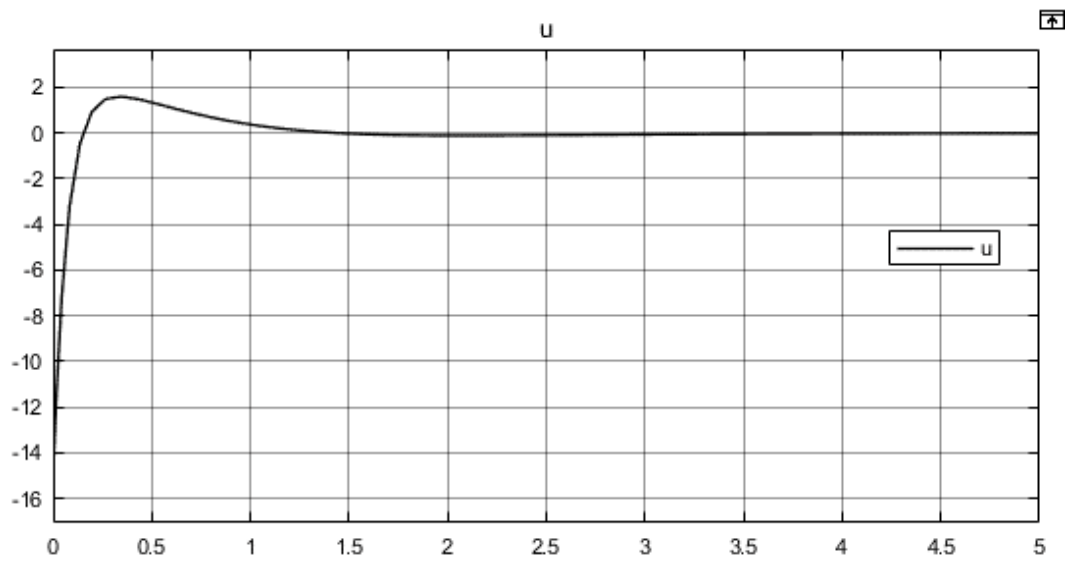


Figura 24. Señal de control con nuevo K.

Para poder introducir la posición deseada simplemente sumamos a la realimentación negativa el valor deseado y se obtiene el diagrama de bloques de la Figura 25.

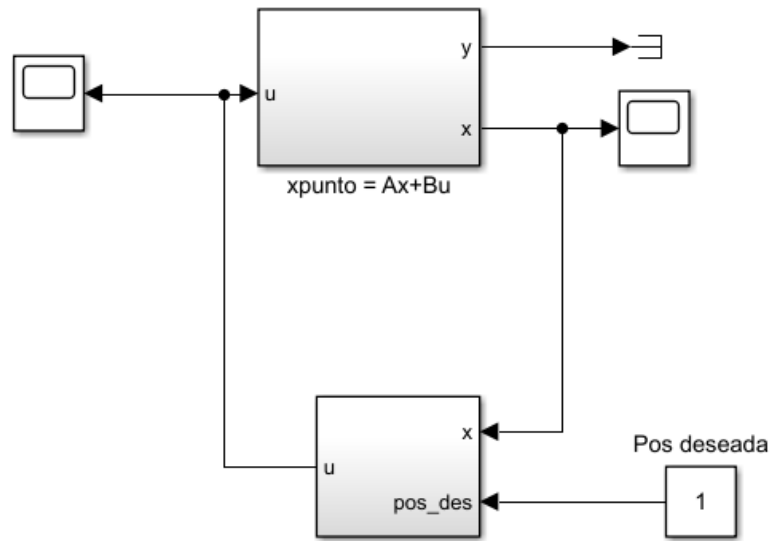


Figura 25. Lazo cerrado con referencia $x=1$.

Por lo simulando Figura 26 se puede ver como la posición se estabiliza en 1 como se había predicho.

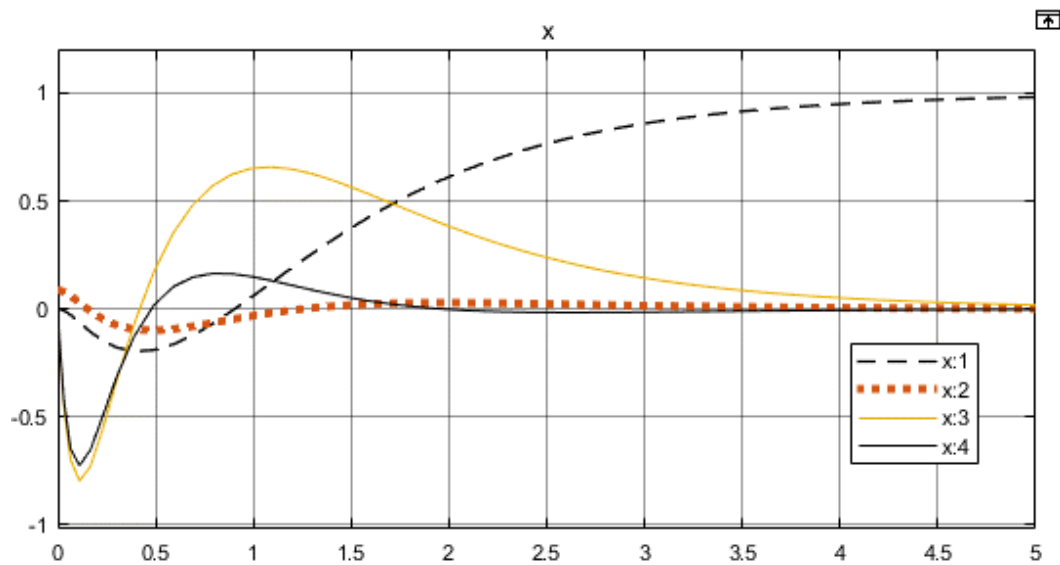


Figura 26. Respuesta del sistema con referencia $x = 1$.

Como se quiere utilizar un controlador digital tendremos que hacer un estudio discreto por lo que se tiene que añadir mantenedores de orden 0 para que transformen la señal continua en discreta y poder estudiar el comportamiento, como se observa en la Figura 27.

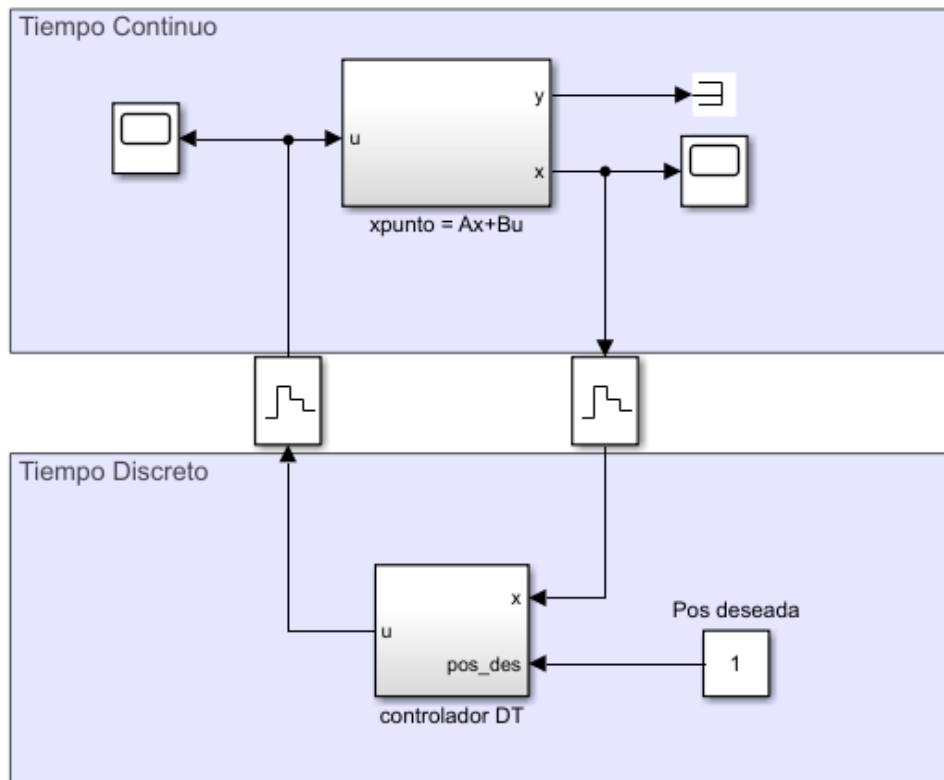


Figura 27. Lazo cerrado con mantenedores de orden 0.

Ahora se tiene que definir Figura 28 el tiempo de muestreo y obtener las matrices del sistema del tiempo continuo a discreto y poder generar la retroalimentación como se ha realizado anteriormente.

```
%% Sistema discreto
Ts= 0.1;
sys_d = c2d(sys,Ts)

Ad = sys_d.a;
Bd = sys_d.b;
Cd = sys_d.c;
Dd = sys_d.d;

des_pole_d = [0.3; 0.3; 0.3; 0.3]*1;
K_d = acker(Ad,Bd,des_pole_d)
```

Figura 28. Calculo de nuevo vector K para sistema con retenedores de orden 0.

Al simular el sistema con el controlador discreto se obtiene la Figura 29 y la Figura 30.

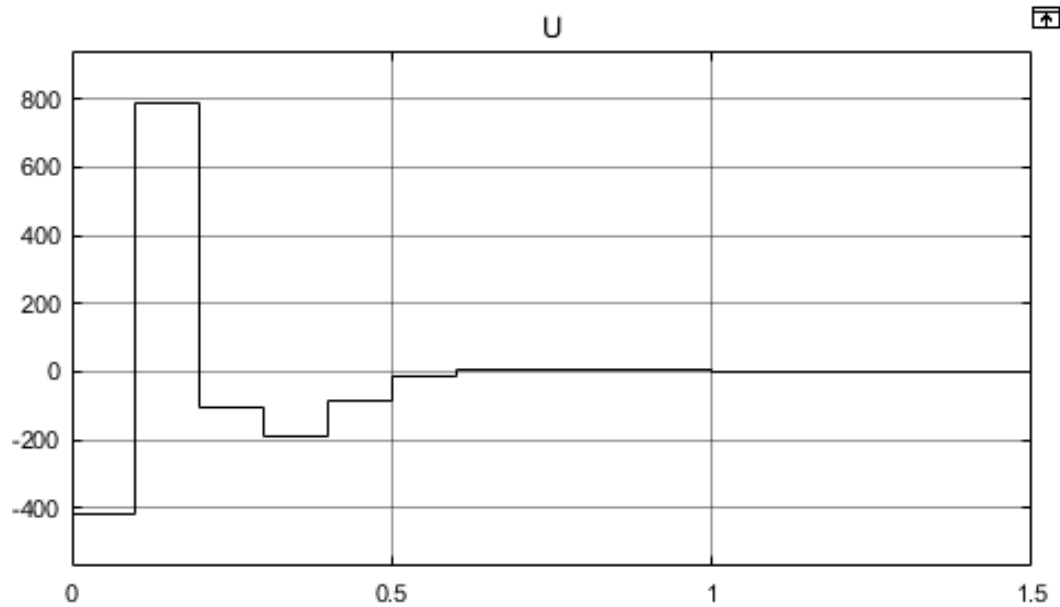


Figura 29. Respuesta de control.

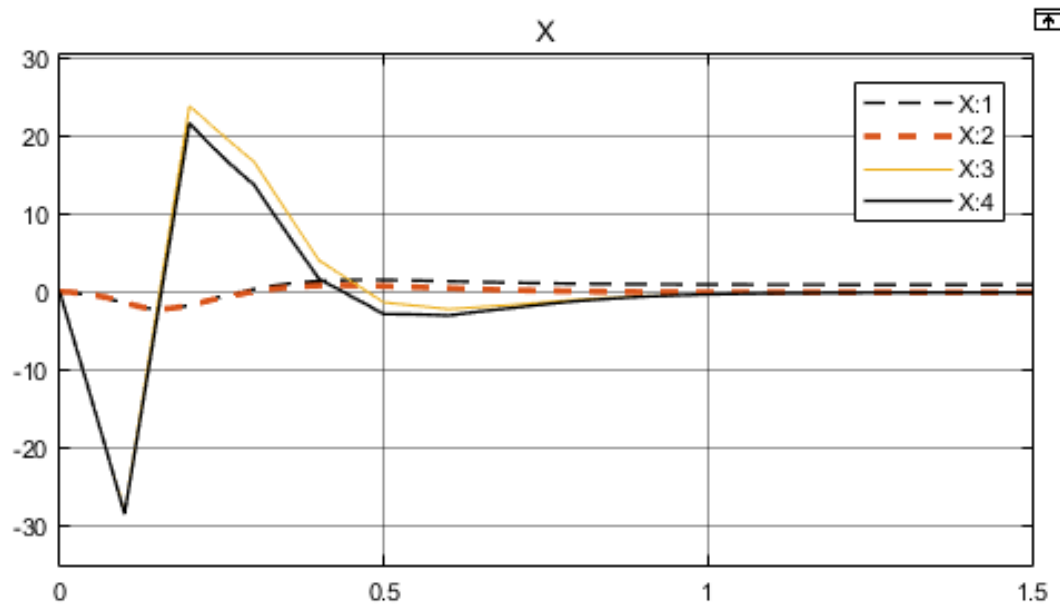


Figura 30. Respuesta del sistema.

8 Construcción física del dispositivo

Para poder realizar este proyecto se tiene que hacer un estudio de cada uno de sus componentes para poder utilizarlos en el robot, con la intención de hacer un pequeño robot auto balanceado. Claramente una de las partes más importantes de este robot es la tracción eléctrica por lo que se comienza por el estudio de los motores.

8.1 Motor

Para el desarrollo óptimo de este proyecto hace falta un dispositivo que convierta la energía eléctrica que se tiene almacenada en las baterías para convertirla en movimiento. Dentro de una gran variedad de motores eléctricos se escogen el motor de corriente continua debido a su controlabilidad principalmente.

Las características de este motor se observan en la Tabla 2.

Velocidad sin carga	350 r/min
Corriente nominal	0.65 A
Voltaje nominal	12 V
Potencia nominal	5 W
Peso	92 g

Tabla 2. Características de motor.

En el tema mecánico se tiene una relación de transmisión de 1:381, por tanto, por cada vuelta que da la rueda el motor necesita dar 381. Junto con esta relación de transmisión tenemos un encoder relativo solidario al eje del motor que se estudiará más adelante.

El motor de corriente continua que se utiliza tiene las siguientes partes:

8.1.1 Rotor

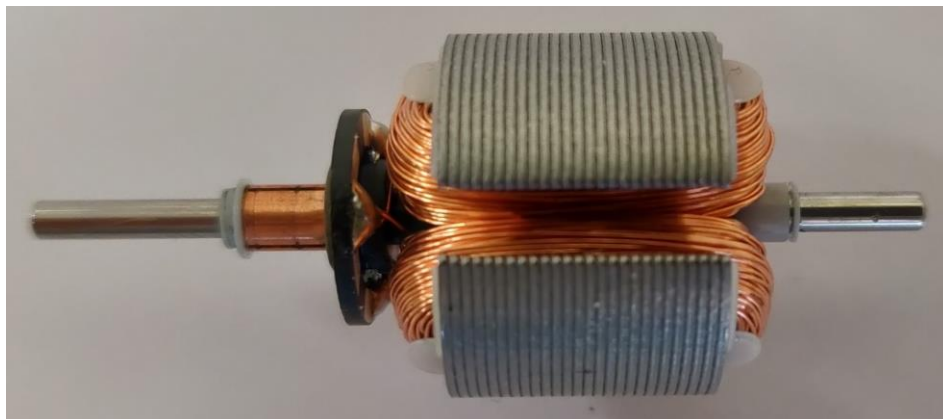


Figura 31. Rotor.

En la Figura 31 se puede observar el rotor que tiene las bobinas por las que pasa la corriente y generar el campo magnético. Situado en el eje, de color cobre, se observa el conmutador.

8.1.2 Estator



Figura 32. Estator.

En la Figura 32 se puede observar el estator del motor que se va a utilizar. Dentro de él se puede ver los imanes que generan el campo magnético permanente.

8.1.3 Relación de transmisión



Figura 33. Relación de transmisión.

En la Figura 33 se muestra la relación que mediante piñones rectos hace que el movimiento circular pierda velocidad, pero gane fuerza en la rueda.

Para poder realizar el control del motor de corriente continua es necesario utilizar un driver para poder suministrar la tensión necesaria.

9 Puentes en H

9.1 Introducción

Un puente en H es un tipo de circuito electrónico que se utiliza entre otras funciones para cambiar la polaridad de dos terminales para alimentar a un motor de corriente continua. Por lo tanto, se podrá variar entre dos valores de tensión definidos por la fuente de alimentación implementada (batería en este caso). Al ser capaz de conmutar con suficiente rapidez se podrá controlar tanto la velocidad del motor como el sentido de giro. Para este proyecto se necesitan de dos puentes idénticos para controlar cada una de las ruedas de forma independiente.

Para poder desarrollar la función descrita antes se necesitan de 4 interruptores, Figura 34, que se irán conmutando para que el motor tenga una tensión “positiva” cuando se tienen cerrados los interruptores S1 y S4; por lo que para que tenga una tensión negativa tendrá que estar cerrados los interruptores S3 y S2.

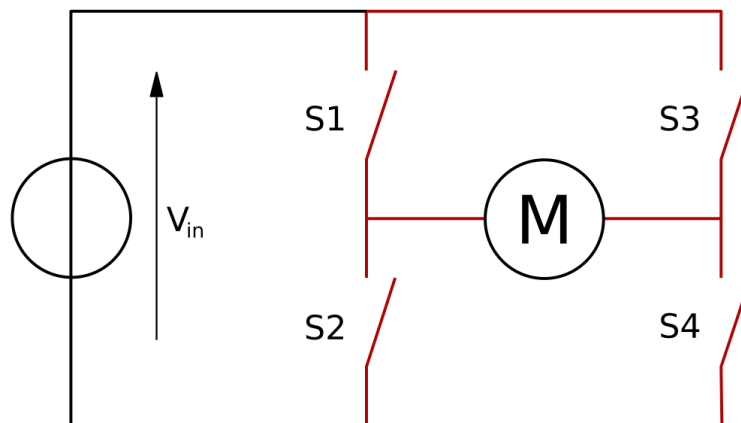


Figura 34. Diagrama de puente en H.

En este proyecto se utilizará el freno que consisten en cortocircuitar los dos bornes del motor. Esta técnica es conocida como fast-stop.

Se tiene que tener en cuenta que va a trabajar con dos sistemas idénticos para el control de los dos motores que trabajan de forma paralela e independiente por lo que se tratará de forma detallada uno de ellos y se podrá trasladar de forma idéntica al otro.

9.2 Entradas y salidas de los puentes en H

Estos sistemas tendrán como entrada una señal PWM que se encargará de indicar cuanto tiempo tienen que tener tensión entre los terminales y el sentido de giro mediante una señal booleana.

Como salida para estos sistemas tendremos 4 terminales que alimentarán y controlarán los dos motores de corriente continua que se utilizarán para este proyecto.

9.3 Etapas del puente en H

Para poder explicar de forma detallada el funcionamiento del puente en H se va a dividir en 3 etapas principales y una auxiliar. Estas etapas son las siguientes:

Primera etapa: Puertas NOR.

Segunda etapa: control de los MOSFET.

Tercera etapa: MOSFET.

Etapas auxiliares: convertidor CC/CC.

9.3.1 Primera etapa: Puertas NOR

En esta primera etapa se toma como entrada las señales PWM y la dirección de giro donde se codificarán para poder entregar a la segunda etapa cuando se tiene que tener tensión en cada uno de los terminales del motor. Para poder entender de forma exacta se tiene que recordar que para un determinado sentido de giro se ira conmutando entre un cortocircuito o una tensión distinta de 0 entre los dos terminales del motor.

Basándose en lo que se pretende obtener se desarrolla el circuito lógico de la Figura 35 para poder obtener las dos salidas que definen el comportamiento del puente:

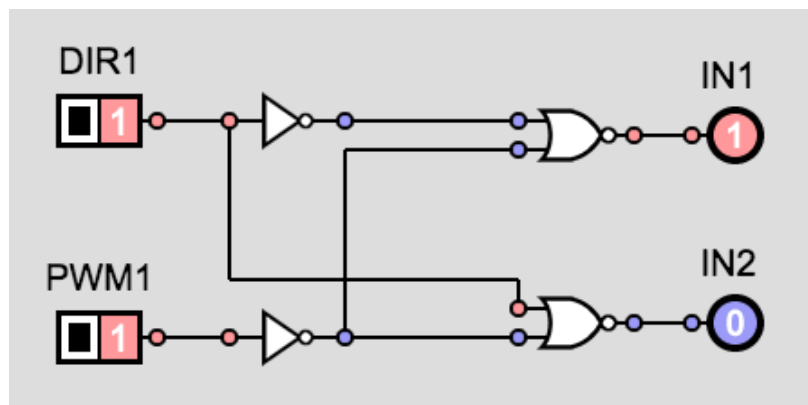


Figura 35. Diagrama lógico de puertas NOR.

La tabla de verdad que se obtiene de este circuito lógico será la Tabla 3.

DIR1	PWM1	IN1	IN2
0	0	0	0
1	0	0	0
0	1	0	1
1	1	1	0

Tabla 3. Tabla de verdad para las puertas NOR.

Para resumir esta etapa, cuando se tiene el sentido de giro definido, se fija el estado de una la mitad del puente, donde se tendría uno de los MOSFET abierto y otro cerrado. Mientras que en la otra mitad del puente se irá conmutando con la señal PWM el estado de los dos MOSFET, siendo siempre para uno de ellos el estado contrario del otro. Ya se verá en las siguientes etapas como se traduce este proceso.

Para realizar definir el comportamiento de esta etapa se utilizan las puertas NOR que se tiene en el integrado llamado 74HC02D que se tiene en la Figura 36.

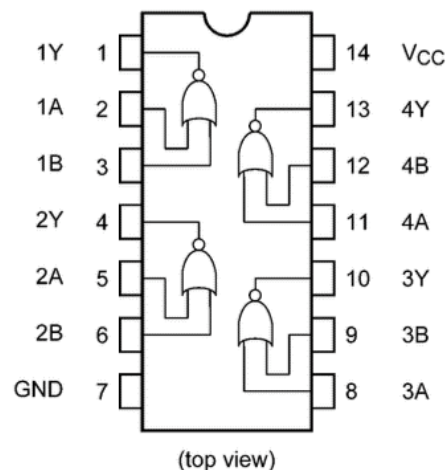


Figura 36. Pines del integrado 74HC02D.

Para poder desarrollar las puertas que niegan tanto la señal PWM y la DIR simplemente se cortocircuitan las entradas a las puertas NOR y se llevan a la señal PWM y DIR respectivamente. Este integrado necesita de una tensión de alimentación de 5V para poder trabajar.

9.3.2 Segunda etapa: control de los MOSFET

En esta etapa se tiene como entrada las señales IN1 y IN2 que cada una que van a controlar de forma independiente dos de los cuatro MOSFET que componen el puente en H. Por lo tanto, la salida de esta etapa serán las señales necesarias para controlar los cuatro MOSFET.



Ya teniendo claro lo anterior se va a proceder a ver como se controlan dos de los cuatro MOSFET que componen el puente y así se entenderá como se controlan los otros dos MOSFET.

Para controlar dos MOSFET se va a utilizar el integrado L6384 que tiene como entrada la señal booleana IN que al recibirla procederá a hacer que uno de los MOSFET esté cerrado y otro abierto dependiendo si IN es 0 o 1. Por lo que SIEMPRE se cumplirá que uno de los MOSFET tenga el estado contrario al otro.

Para poder hacer uso del integrado se tiene que tener en cuenta cual es el montaje y los componentes necesario para que funcione de forma óptima. Este integrado tiene las patillas que se muestran en la Figura 37.

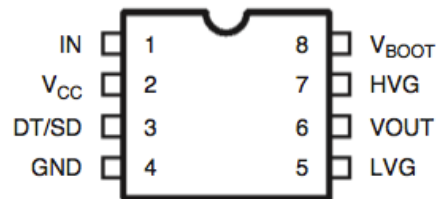


Figura 37. Pines del integrado L6384.

Esas patillas tienen el siguiente uso:

1. IN: entrada para poder definir el estado de los dos MOSFET.
2. V_{cc}: es la alimentación del circuito que en nuestro caso tendrá que ser 13.75V debido a que se necesita que entre las patillas HVG o LVG y OUT se tenga una tensión y así se pueda poner en saturación los MOSFET cuando sea necesario. Se coloca un condensador de 1.1 uF para el rizado de la entrada.
3. DT/SD: es un pin de alta impedancia que tiene dos funcionalidades:
 - Es necesario para cuando la tensión de alimentación cae por debajo de 0.5V apagará el integrado.
 - Un voltaje mayor que V_{dt} establece el tiempo muerto entre el controlador de puerta lateral alta y el controlador de puerta lateral baja que serán los que controlan los MOSFET. El valor del tiempo muerto se puede configurar forzando un cierto nivel de voltaje en el pin o conectando una resistencia entre el pin 3 y tierra.

En este caso se coloca una resistencia de 270KΩ porque necesitamos un tiempo muerto entre que los dos MOSFET cambien de estado en 2.5 us. Esta resistencia se puede definir a partir gráfica que se muestra en la Figura 38.

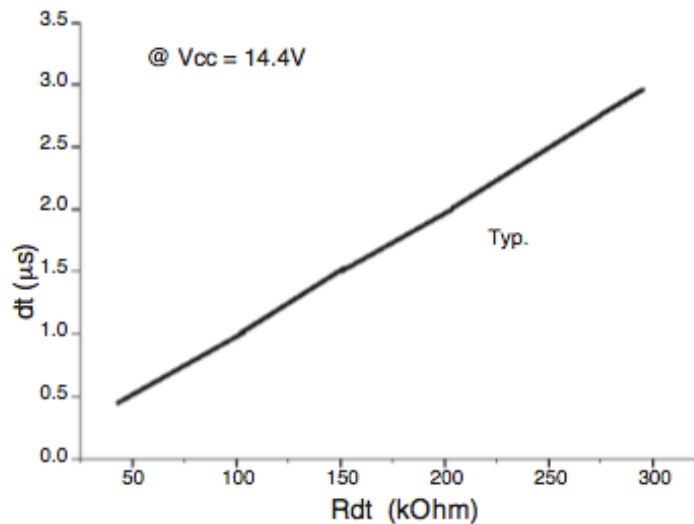


Figura 38. Tiempo muerto vs resistencia.

4. GND: es el pin donde se conecta el negativo del circuito.
5. LVG: es donde se coloca la puerta del MOSFET que está conectado a tierra para poder ser controlado.
6. Vout: es el pin que se encuentra cortocircuitado entre la puerta de un MOSFET y el surtidor del otro para poder controlarlos.
7. HVG: es el pin que se encuentra conectado a la puerta del MOSFET que está conectado a la tensión que tendrá que alimentar al motor.
8. Vboot: es el suministro flotante del controlador superior. El condensador de arranque conectado entre este pin y el pin 6 se puede alimentar mediante una estructura interna denominada "controlador de arranque". En este caso no se utiliza por lo que se necesita de un Dboot que se tiene entre la patilla Vcc y Vboot.

Para la función que se necesita para este proyecto se usará el montaje que se tiene en la Figura 39:

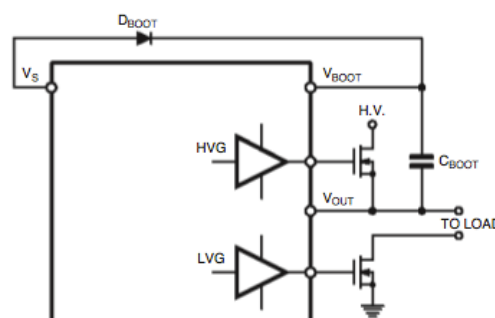


Figura 39. Montaje de L6384.

Por lo que también se tendrá que dimensionar el condensador Cboot que se encarga del control del MOSFET conectado a H.V. Para elegir el valor Cboot adecuado, el MOSFET externo se puede ver como un capacitor equivalente.



Este condensador está relacionado con la carga total de la puerta MOSFET. Para poder calcularlo se consulta el data sheet del integrado STP75NF75.

$$C_{ext} = \frac{Q_{puerta}}{V_{puerta}} = \frac{117nC}{3V} = 39nF$$

La relación entre los condensadores C_{ext} y C_{boot} es proporcional a la pérdida de tensión cíclica. Tiene que ser:

$$C_{boot} \gg \gg C_{ext}$$

Por lo que podrá ser $C_{boot} = 0.03\mu F$

9.3.3 Tercera etapa: MOSFET

Esta etapa se realiza para entregar la tensión necesaria en los bornes de motor de corriente continua. Para ello se disponen de cuatro MOSFET que se disponen en estructura de puente en H y se controlan mediante dos de los integrados de la segunda etapa que tienen como salidas HGV, Vout y LVG para cada pareja de MOSFET.

En este apartado se utiliza el integrado STP75NF75 en el cual se tiene que una tensión drenador – surtidor de 75V cuando tenemos una intensidad del drenador de 80A por lo que es más que suficiente para soportar una corriente máxima del motor de 0.41A y una tensión de 12V que lo alimentará.

Algo que también se tiene que tener muy en cuenta es el tiempo que tarda en entrar en saturación y cuánto tarda en entrar en corte que en este caso es de 100ns y 66ns respectivamente que en principio será más que suficiente ya que las frecuencias de control serán entorno a 16KHz.

9.3.4 Etapa auxiliar: CC/CC

Para el integrado de la etapa de control de los MOSFET es necesario tener una tensión de 13.75V para que se pueda llevar a saturación los MOSFET por lo que se necesita un CC/CC que pase de 5V con que se alimenta al puente en H y se entreguen los 13.75V.

Para esta función utilizamos el integrado MC34063A que se encargará de aumentar la tensión de entrada. Para poder realizar esta función tendremos que respetar el circuito de la Figura 40:

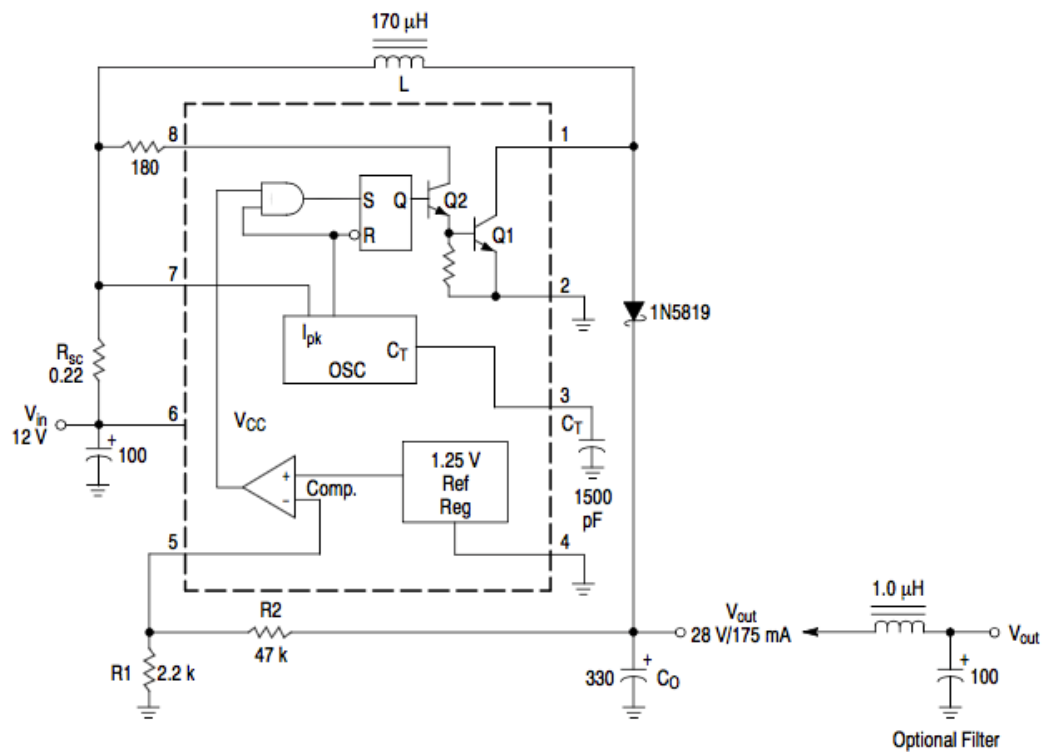


Figura 40. Diagrama de MC34063A.

Teniendo en cuenta que la tensión que se necesita para los integrados de la etapa anterior es de 13.45V y que cada uno demanda una intensidad máxima de 25mA por lo que se tendrá una intensidad máxima de salida para esta etapa de 100mA.

Cálculo de componentes necesarios para el integrado

Los componentes que se tienen que seleccionar son los siguientes:

- Ct: condensador utilizado para fijar la frecuencia de funcionamiento del oscilador interno.
- Rsc: valor de la resistencia de precisión.
- Lmin: valor mínimo de la inductancia del filtro de salida.
- Co: valor mínimo del condensador de filtro de salida.

Para calcular cada uno de estos valores el fabricante suministra las siguientes formulas:

$$\frac{ton}{toff} = \frac{Vout + Vf + Vin(min)}{Vin(min) - Vsat} \quad (10)$$

$$ton + toff = \frac{1}{f} \quad (11)$$



$$toff = \frac{ton + toff}{\left(\frac{ton}{toff}\right) + 1} \quad (12)$$

$$ton = (ton + toff) - toff \quad (13)$$

$$Ct = 4 * 10^{-5} ton \quad (14)$$

$$Ipk(software) = 2Iout(max)\left(\frac{ton}{toff} + 1\right) \quad (15)$$

$$Rsc = \frac{0.3}{Ipk(software)} \quad (16)$$

$$Lmin = \frac{Vin(min) - Vsat}{Ipk(software)} ton(max) \quad (17)$$

$$Co = 9 \frac{tonIout}{Vripple(pp)} \quad (18)$$

$$Vout = 1.25\left(1 + \frac{R2}{R1}\right) \quad (19)$$

Los parámetros que se tienen que tener en cuenta para poder dimensionar los componentes son:

- Vsat: tensión de saturación del conmutador externo. En este caso no necesita de un transistor por lo que será 0.
- Vf: caída de tensión del rectificador seleccionado.
- Vin: tensión de entrada a la etapa.
- Vout: tensión de salida de la etapa.
- Iout: corriente de la salida de la etapa.
- F: frecuencia del oscilador interno para realizar la conmutación.
- Vripple: tensión pico a pico de la salida.

El condensador en el pin de entrada se coloca de 10uF para que sirva como filtro. La frecuencia de conmutación se elige entorno a 2KHz.

Teniendo en cuenta lo que se tiene que calcular se establecen los valores que tiene que cumplir el convertidor.

- Vin: 5V
- Vout: 13.75V
- Iout: 100mA
- Vripple: 100mV
- Vf: 0.5V



Haciendo los cálculos se obtienen los siguientes parámetros:

$$\frac{ton}{toff} = \frac{Vout + Vf + Vin(min)}{Vin(min) - Vsat} = \frac{13.75 + 0.5 + 5}{5} = 3.85$$

$$ton + toff = \frac{1}{f} = \frac{1}{2 * 10^3} = 0.0005 \text{ s}$$

$$toff = \frac{ton + toff}{\left(\frac{ton}{toff}\right) + 1} = \frac{0.0005}{3.85 + 1} = 0.0001031 \text{ s}$$

$$ton = (ton + toff) - toff = 0.0005 - 0.0001031 = 0.000396 \text{ s}$$

$$Ct = 4 * 10^{-5} ton = 1.584 * 10^{-8} F$$

$$Ip k(\text{switch}) = 2Iout(\text{max}) \left(\frac{ton}{toff} + 1 \right) = 0.9681 \text{ A}$$

$$Rsc = \frac{0.3}{Ip k(\text{switch})} = 0.30986 \Omega$$

$$Lmin = \frac{Vin(min) - Vsat}{Ip k(\text{switch})} ton(\text{max}) = 0.002045 \text{ H}$$

$$Co = 9 \frac{tonIout}{Vripple(pp)} = 0.00356 \text{ F}$$

$$Vout = 1.25 \left(1 + \frac{R2}{R1} \right) = 13.75V$$

9.4 Esquema de circuito

Para poder realizar el esquema de todo el sistema se utilizará el programa Altium. Se ha escogido debido a su capacidad para realizar circuitos electrónicos con bastante profesionalidad. En este programa se han creado los integrados que no se han tenido en la librería oficial.

Se introduce una visión general del circuito, Figura 41, para poder introducir cada una de las partes que componen el sistema en general.

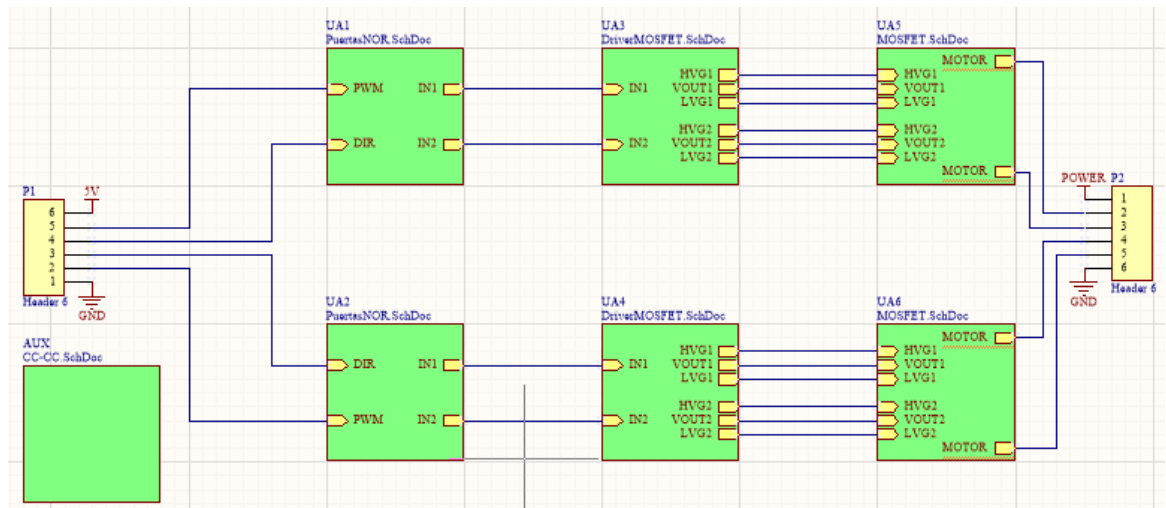


Figura 41. Diagrama general de puente en H.

Donde se tienen las siguientes etapas:

- Primera etapa: Puertas NOR.
- Segunda etapa: control de los MOSFET.
- Tercera etapa: MOSFET.
- Etapa auxiliar: convertidor CC/CC.

9.4.1 Primera etapa: Puertas NOR.

En esta etapa, Figura 42, se necesita de los pines donde se van a conectar las entradas y la alimentación llamado P2. También se colocan los integrados que contienen las puertas NOR. Se necesitan dos integrados, uno por cada motor.

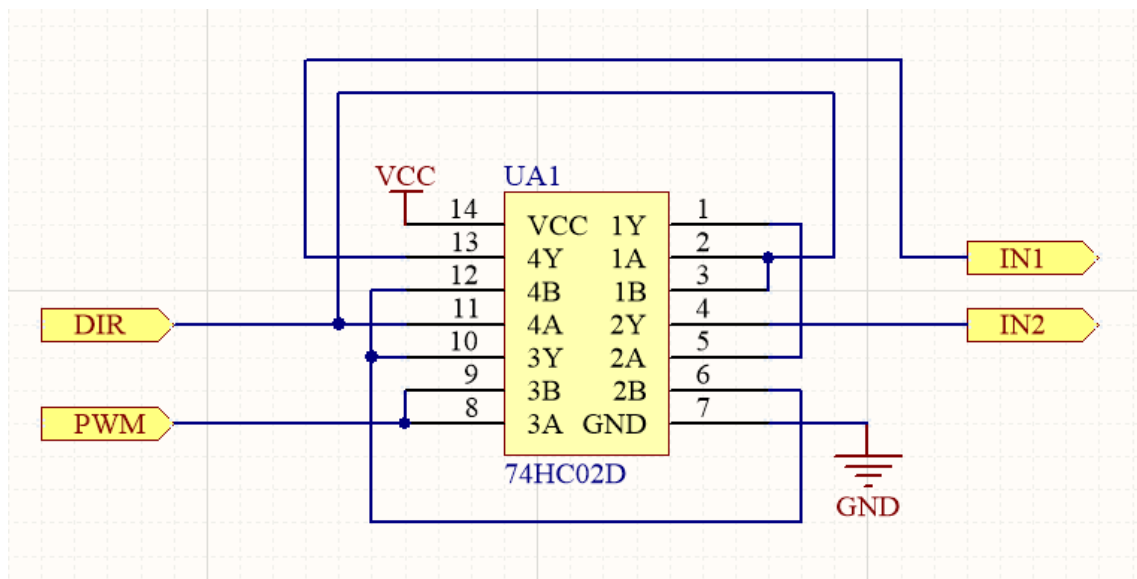


Figura 42. Puertas NOR.



9.4.2 Segunda etapa: control de los MOSFET.

En esta etapa se hace uso de los integrados L6384D, que como cada uno controla un semipunto y se tienen dos motores, se necesitan 4 de ellos.

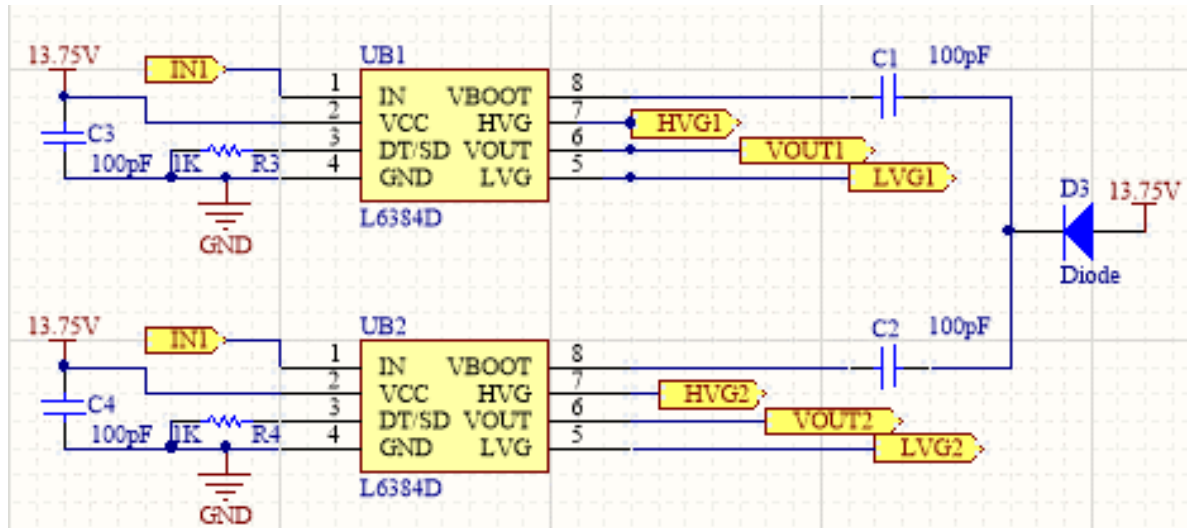


Figura 43. Diagrama de conexión L6384D.

9.4.3 Tercera etapa: MOSFET

Esta será la última etapa, Figura 44, en la que se colocan los MOSFET necesarios para poder controlar los dos motores, por lo que se necesitará de dos grupos de cuatro MOSFET.

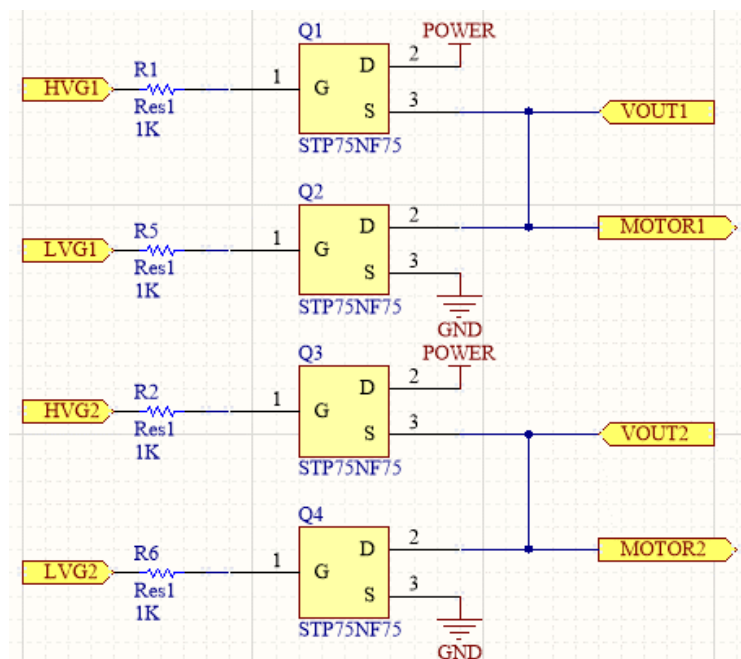


Figura 44. Diagrama de MOSFET.



9.4.4 Etapa auxiliar: convertidor CC/CC.

En esta etapa, Figura 45, auxiliar necesitamos del integrado MC34063 para poder construir el convertidor CC/CC.

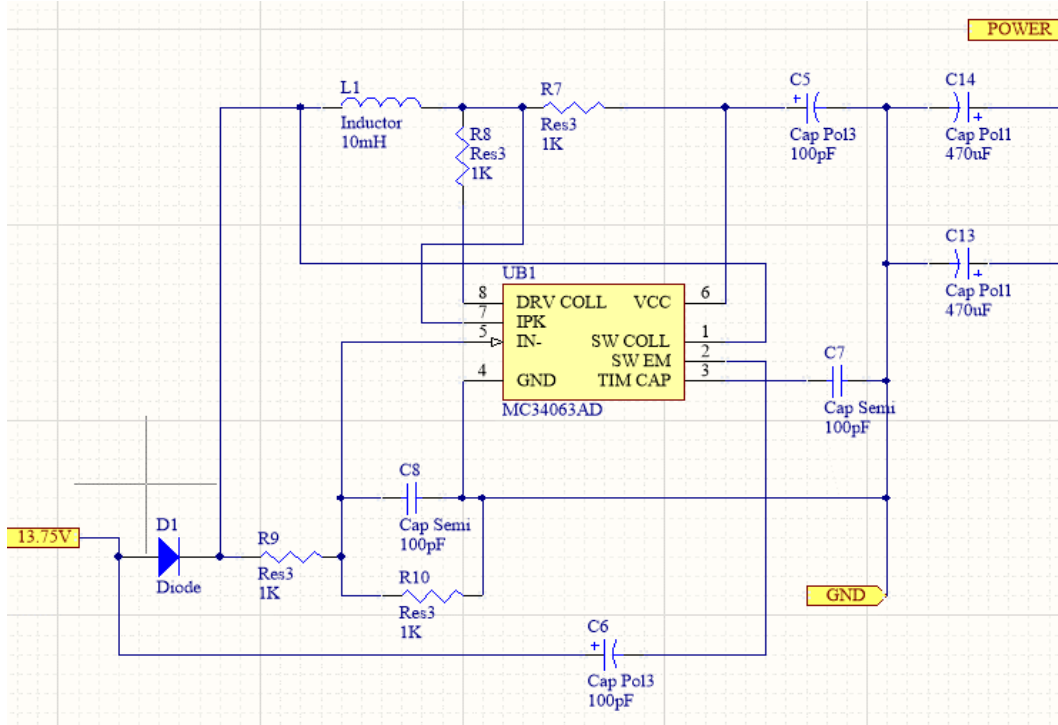


Figura 45. Diagrama de CC-CC.

Para poder enviar las señales para controlar el driver se necesita de un controlador con la suficiente velocidad para generar una señal PWM y demás señales.

La placa físicamente se puede observar en la Figura 46.

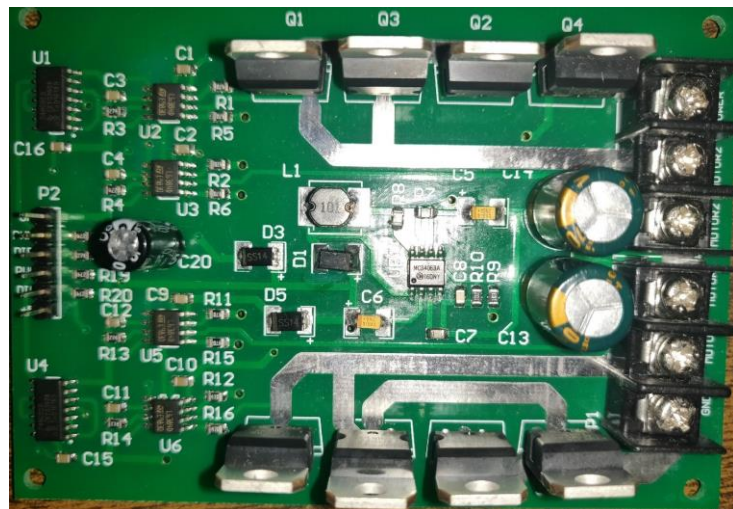


Figura 46. Puente en H.

9.5 Controlador

Desde un punto de vista general se necesita de un dispositivo capaz de obtener información de los sensores para después enviar las señales necesarias para poder ejecutar el correcto control sobre los motores y el robot pueda funcionar correctamente, por ello se introducen 2 de las marcas más importantes en el sector, que son Arduino y Texas Instruments.

9.5.1 Texas Instruments

Esta empresa de componentes electrónicos [12] nos ofrece una gama de placas, pero una de las más básicas LaunchPad MSP430G2, Figura 47, nos ofrece las características de la Tabla 4.

Microcontrolador	MSP430
Voltaje de operación	5V
Voltaje recomendado de entrada	5V
Límite voltaje de entrada	5V
Pines de entrada / salida	16
Pines PWM	7
Pines de entrada analógica	8
Memoria Flash	16KB
Velocidad de reloj	3KHz
I2C	SÍ

Tabla 4. Características de LaunchPad.

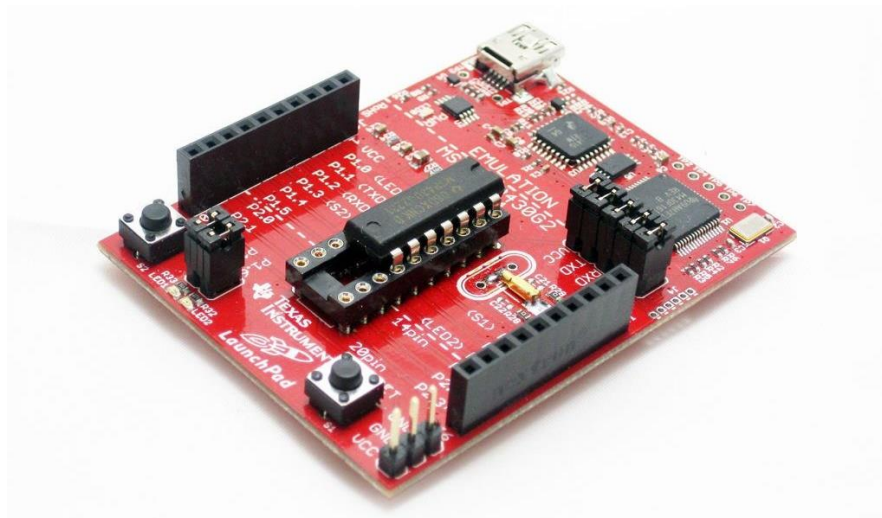


Figura 47. LaunchPad [12].



9.5.2 Arduino

Arduino se creó como una plataforma de creación electrónica de código abierto que tiene varios tipos de placas que sirven como controladores. Entre ellos se encuentra los más importantes Arduino Uno y Arduino Mega 2560.

9.5.2.1 Arduino Uno

Las características técnicas que ofrece el Arduino Uno están expuestas en la Tabla 5

Microcontrolador	ATmega328P
Voltaje de operación	5V
Voltaje recomendado de entrada	7 - 12 V
Límite de voltaje de entrada	20V
Pines digitales entrada / salida	14
Pines PWM	6
Pines entradas analógicas	6
Corriente DC en pines entrada / salida	20mA
Corriente DC en pin 3.3V	50mA
Memoria Flash	32 KB
SRAM	2KB
EPR0M	1KB
Velocidad de reloj	16 MHz
I2C	Sí

Tabla 5. Características de Arduino.

9.5.2.2 Conclusiones

Como ya se ha comentado este robot necesita de un controlador para poder procesar los datos que se generan en los sensores. Para ello se tiene que tener en cuenta que clase de sensores se va a utilizar que son 2 encoder relativos para los 2 motores y un giroscopio. También es necesario utilizar un módulo de comunicación Bluetooth por lo que ya se parte con bastantes restricciones.

A la hora de buscar información sobre las dos placas se puede apreciar de primera mano cómo la gama Arduino tiene mucha más información que la gama LaunchPad de Texas Instruments, por lo que es una parte fundamental debido a la complejidad del proyecto que se va a realizar.

Las características importantes que se tendrán en cuenta serán la velocidad de procesamiento del microcontrolador, la capacidad de generar señales PWM, si tiene comunicación I2C y la cantidad de información que se podrá encontrar sobre ella.

Haciendo un balance sobre todos los pros y contras de cada una se llega a la conclusión de que Arduino es la mejor opción debido a la cantidad de información que se tiene sobre él y si se necesitará mayores prestaciones se saltará a una placa Arduino superior como la Due o Mega 2560 sin ningún problema a la hora de reutilizar lo que se implementó en Arduino Uno.

Para poder medir tanto la velocidad como los grados necesitamos de sistemas que sean capaces de medir con la suficiente precisión todos estos datos por lo que estudiamos el MPU5060 para medir el ángulo de inclinación y el encoder relativo que tienen los motores.

9.6 Sensores

9.6.1 MPU6050

Como se ha introducido se necesita un dispositivo capaz de mantenerse paralelo al vector de gravedad, por lo que primero se necesita es un sensor que pueda ser sensible a ello por lo que se introduce el sensor MPU6050, Figura 48. Este sensor es un módulo de medición inercial con 6 grados de libertad que combina un acelerómetro de 3 ejes y un giroscopio de 3 ejes.



Figura 48. MPU 5060.

Este módulo está potenciado por el integrado MPU6050 que como se ha comentado antes combinan un giroscopio de 3 ejes y un acelerómetro de 3 ejes junto con un procesador digital de movimiento. El diagrama de funcionamiento del integrado es el que se puede observar en la Figura 49.



Figura 49. Diagrama de funcionamiento de MPU 6050.



Las características generales de este integrado se encuentran en la Tabla 6.

Sensibilidad de giroscopio	65.5 LSB/(°/s)
Sensibilidad de acelerómetro	8.152 LSB/g
Salida de información	I2C
Voltaje de salida lógica	1.8V
Voltaje de operación	2.375 - 3.46V
Tamaño	4 x 4 x 0.9 mm

Tabla 6. Características de MPU 5060.

Pero como se va hacer uso de este integrado con una PCB para soportarlo, esta tiene que ser alimentada a 5V debido a que tiene un CC/CC para transformar esa tensión a los 2.5V que necesita el integrado. También se tiene disponibles los pines SDA y SCL para la comunicación I2C.

Para poder definir la dirección de trabajo con el I2C se tendrá que seleccionar con el pin ADO la dirección 0x68 sin conectar o la dirección 0x69 colocándolo a 5V.

9.6.2 Encoder

En principio también se tendrá que hacer uso de sensores de posición del eje para poder medir de manera precisa la posición y velocidad relativa del robot. Por lo que se concluye que se necesita de un encoder relativo en el eje del motor.

Los encoder relativos simplemente envían un pulso cuando detectan una marca concreta mientras gira el eje del motor. Por lo general este tipo de sensores vienen integrados con el motor al que van a medir. En este caso se ha encontrado uno de estos conjuntos con un encoder relativo que genera 381 pulsos por vuelta.

En la Figura 50 se podrá ver el circuito que compone el encoder relativo, donde se tienen dos sensores desfasados 90° para poder saber si se está girando en un sentido u en otro.



Figura 50. Encoder.

Para poder alimentar a todos estos dispositivos se necesitan de dos sistemas importantes, la batería y el CC/CC.

9.7 Módulo de comunicación

Como se quiere hacer uso de una aplicación para móvil para poder conducir el robot, una de las mejores opciones sería un módulo Bluetooth debido a que la mayoría de los móviles actuales tienen este protocolo de comunicación.

Haciendo una búsqueda se ha llegado a la conclusión que podrá hacer uso del módulo HC-05 que se verá en la Figura 51.



Figura 51. Módulo Bluetooth HC-05.

Este módulo funciona con una tensión de alimentación de 5V, puede funcionar como esclavo o master y su protocolo de comunicación es vía puerto Serie con el controlador que se necesita de la conexión Bluetooth.

Las especificaciones generales de este módulo son las que se muestran en la Tabla 7.

Versión de Bluetooth	v2.0
Frecuencia	2.4 GHz, banda ISM
Potencia de emisión	≤ 4 dBm, Clase 2
Alcance	5 m a 10 m
Velocidad asincrónica	2.1 Mbps (máx.) / 160 kbps
Velocidad sincrónica	1 Mbps/1 Mbps
Consumo de corriente	50 mA
Voltaje de alimentación	3.6 V a 6 V
Dimensiones totales	1.7 cm x 4 cm aprox.
Temperatura de operación	-20 °C a +75 °C

Tabla 7. Características de HC-05.

10 Convertidor CC/CC

Como se puede intuir hace falta convertir los 12V de la batería a los 5V por lo que es necesario utilizar un reductor. Haciendo una búsqueda se encuentra el QSKJ, Figura 52 [13].



Figura 52. Reductor.

Este puede tener como entrada de 7-40V y de salida de 1.2-35V. Las características se pueden ver en la Tabla 8.

Voltaje de entrada	7-40 V
Voltaje de salida	1.2-35V
Corriente de salida	12 A
Rango de corriente constante	0,2-12 A
Corriente sin carga	0.02 A

Tabla 8. Características de SKL.

Este reductor hace uso del integrado XL4016 del fabricante XLSEMI que se encarga de reducir la tensión cambiando el ciclo de trabajo de un interruptor utilizando la técnica de Buck, Figura 53.

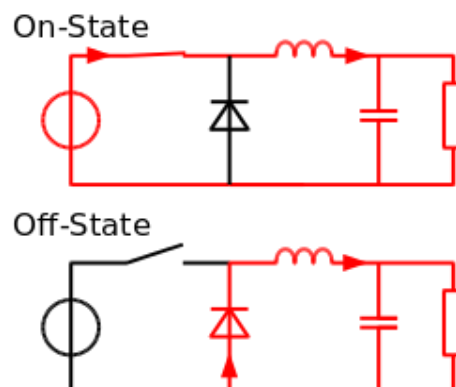


Figura 53. Buck [14].



El funcionamiento de este dispositivo se basa en ir conmutando un interruptor y junto con diodo se va conectando una bobina entre la fuente de alimentación y la carga. Cambiando el tiempo en el que el interruptor está cerrado se va aumentando o disminuyendo la energía transferida a la bobina y el condensador. Todo esto se traduce en una disminución de la tensión en los bornes de la salida del CC/CC.

En la Figura 54 se denota los dos estados principales en los que está el circuito, cuando es el interruptor abierto y cuando está cerrado. El condensador que se tiene en la salida tiene la labor de disminuir el rizado de la tensión.

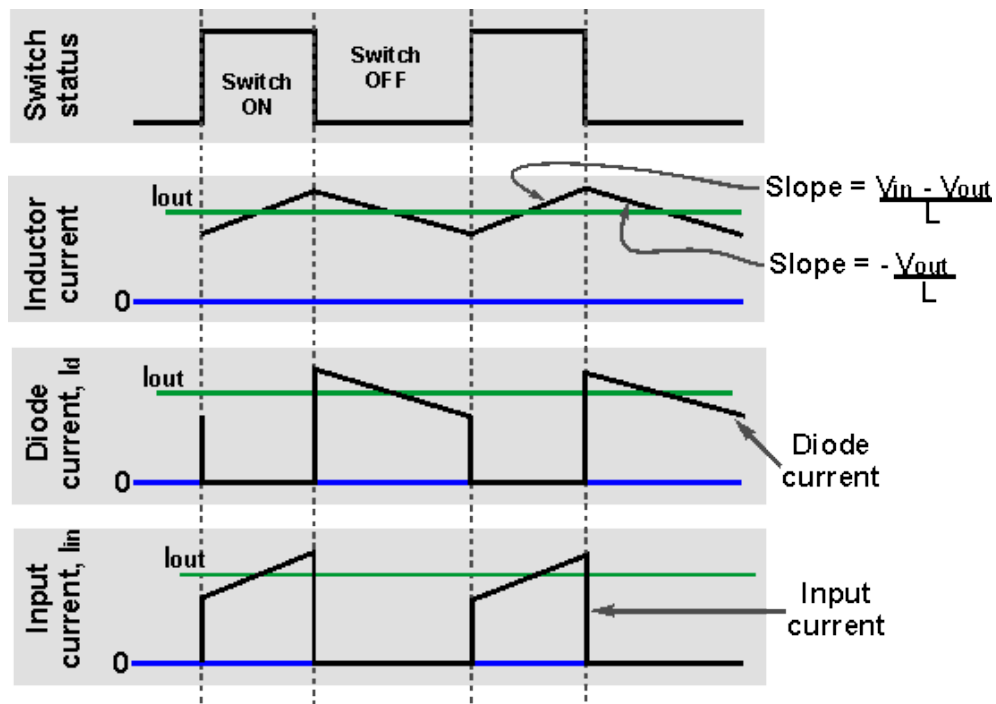


Figura 54. Gráfica de tensión y corriente en modo continuo [15].

10.1 Interruptor en ON

Al estar el interruptor cerrado la tensión de fuente de alimentación pone inversa al diodo y por lo tanto no circula corriente por él. Pero si se tiene un circuito cerrado entre la bobina y el paralelo de la carga con el condensador, haciendo que circule corriente. En el momento que circula corriente por ellos se comienza a cargar el condensador y la bobina.

Se tiene que recordar que una bobina funciona como un circuito “resistencia” en el transitorio debido a que va almacenando energía. En cuando se va cargando opone menos resistencia al paso de la corriente.

Como la bobina esta en serie con la fuente de alimentación cuando el interruptor esta en ON, corriente que pasa por la fuente de alimentación es la misma que la que pasa por la bobina.



10.2 Interruptor en OFF

Cuando el interruptor esta en OFF ya la fuente de alimentación no aporta potencia al circuito, es decir la corriente por la fuente de alimentación es 0 como la corriente por interruptor.

La corriente por la bobina va disminuyendo ya que al principio tiende a oponerse a la variación de corriente, aportando la energía que tiene almacenada, pero esta va disminuyendo.

Como la bobina tiene a mantener constante el módulo y signo de la corriente genera el cambio de estado de inversa a directa del diodo haciendo que circule por él la misma corriente que se tiene en la bobina debido a que están en serie.

11 Modelo de control

En el apartado matemático se ha detallado que este sistema es intrínsecamente inestable por lo que es necesario poder obtener el modelo de control para poder controlar el sistema y por lo tanto que se comporte como el usuario prefiera. Para desarrollar este modelo es necesario saber cuáles son las entradas y las salidas de este sistema.

11.1 Entradas:

- Angulo de inclinación
- Velocidad a la que giran las ruedas

11.2 Salidas:

- Momento en las ruedas del robot

12 Estudio de control mediante Matlab

12.1 Lectura de encoder

En la Figura 55 simplemente estamos contando los pulsos que se obtienen del encoder relativo y se llevan a un contador de pulsos que se va reiniciando cada 10 segundos.

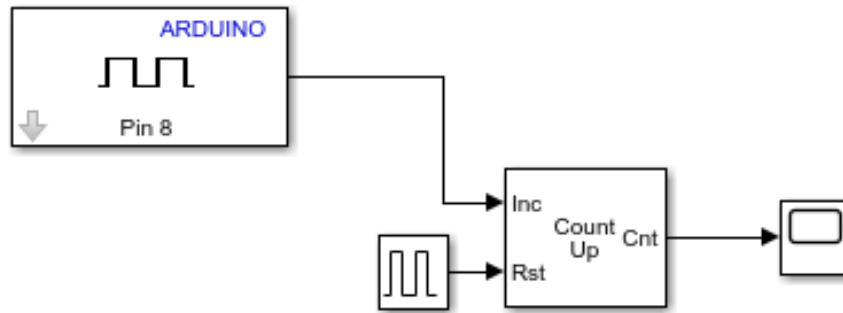


Figura 55. Diagrama de bloques de lectura de pulsos del encoder.

En la Figura 56 se puede apreciar cómo se va reiniciando cada 10 segundos. Se tiene que tener presente que el movimiento de la rueda se hace de forma manual.

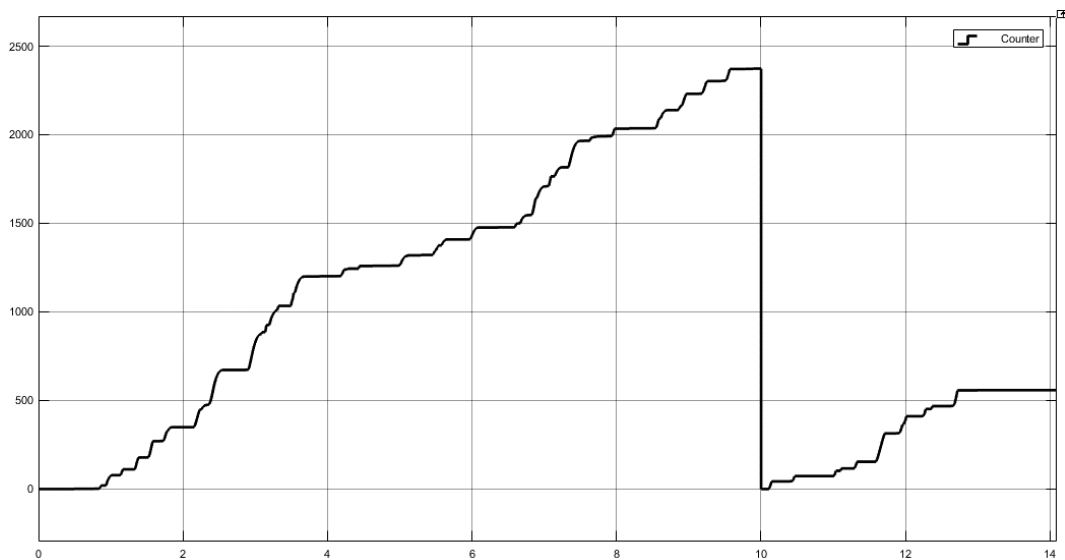


Figura 56. Conteo de pulsos en 10 segundos.

Ahora se añade en la Figura 57, el bloque para que mantenga el último justo después de terminar los 10 segundos.

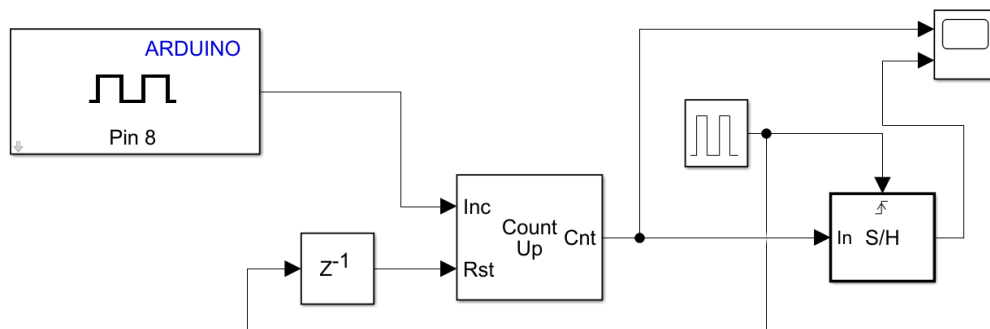


Figura 57. Retenedor de la suma de pulsos cada 10 segundos.

En la Figura 58 se puede observar la gráfica superior se obtiene del bloque contador mientras que la inferior corresponde al mantenedor.

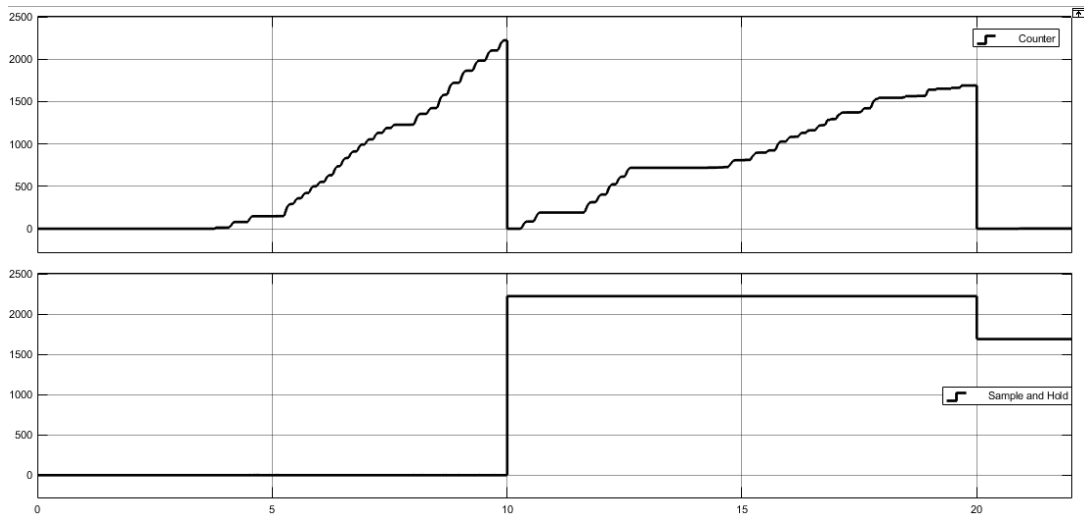


Figura 58. Contador de pulsos y retenedor de la suma cada 10 segundos

El siguiente paso es disminuir el tiempo de la cuenta hasta 0.25 s y luego multiplicar por $1/1.58$ para poder obtener la velocidad en revoluciones por minuto, como se ha argumentado antes el encoder genera 381 pulsos por una revolución de la rueda. El esquema en Simulink queda de la siguiente manera:

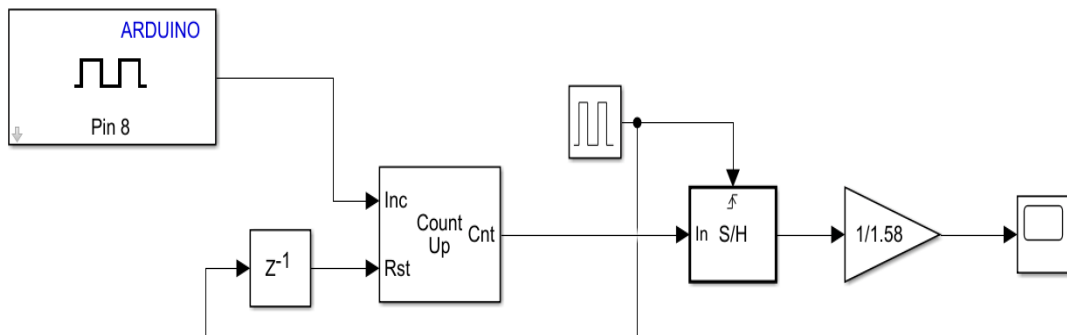


Figura 59. Cálculo de velocidad con un tiempo de muestreo $t = 0.25s$.

Por lo que en la Figura 60 se puede apreciar la velocidad en revoluciones por minuto con un muestreo de 0.25s.

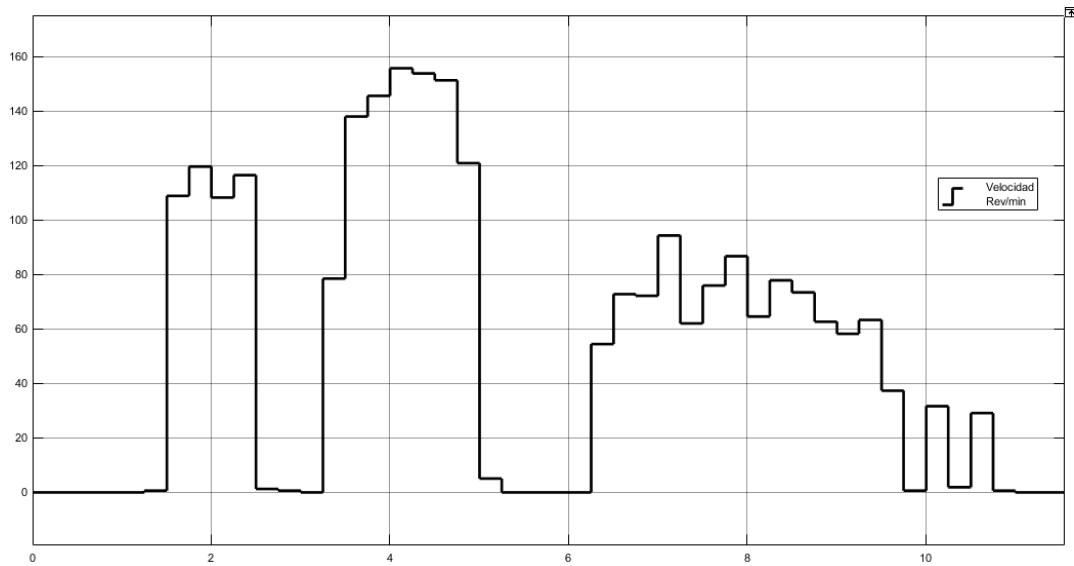


Figura 60. Velocidad en r/min.

Hasta ahora se ha medido la velocidad simplemente girando la rueda con la mano, ahora se va cambiando la velocidad de la rueda enviando una señal PWM al driver de control del motor, Figura 61.

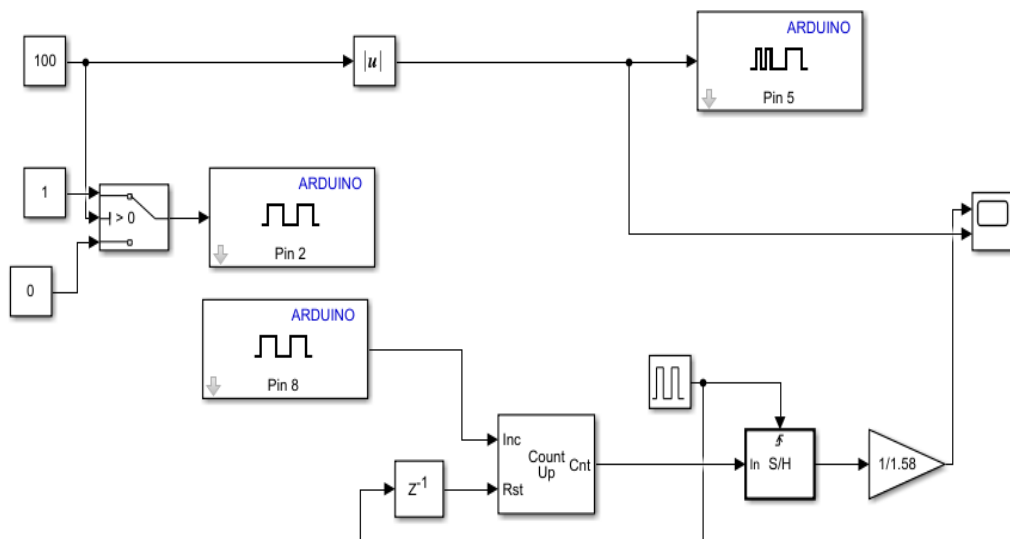


Figura 61. Calculo de velocidad con envío de señal PWM al motor.

Por lo que se obtiene las gráficas de la Figura 62, cabe resaltar que la señal PWM en el Arduino tiene un rango de 0 a 255. Cada uno de estos valores enteros del rango determinan el ciclo de trabajo con la que va a oscilar la señal PWM.

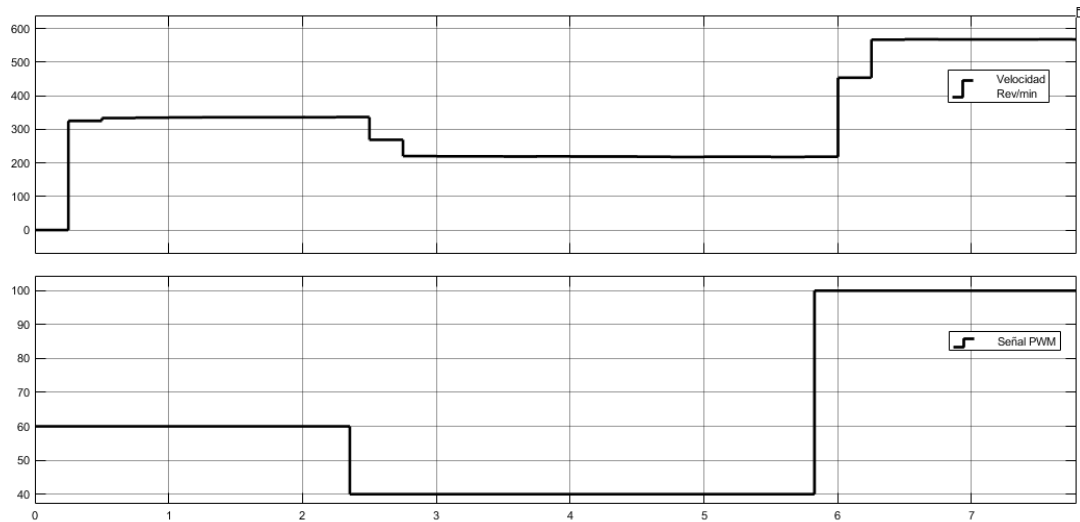


Figura 62. Velocidad en r/min con la señal PWM enviada al motor.

12.2 Conclusiones

En este apartado se pudo estudiar de manera general la conexión de Arduino con Matlab. Se pudo programar y ajustar valores como la señal PWM en tiempo real, pero para el cálculo efectivo de la velocidad de giro se necesita de una mayor velocidad de comunicación.

Al estar trabajando un poco más de tiempo con este sistema se pudo observar que la velocidad real no correspondía a la que se mostraba en las gráficas debido a que el Arduino no contaba todos los pulsos que se producían si se aumentaba la velocidad de giro de la rueda.

13 Estudio de control por medio del IDE Arduino

Como Matlab no ha sido suficiente para llegar programar el robot, se procede a utilizar el IDE de Arduino que proporciona el fabricante de la placa. Esta es una de las mejores formas para programar este tipo de placas debido a que se ha hecho por el mismo fabricante. También tiene una gran herramienta, el puerto serie que se utiliza para visualizar los datos que se necesitan mientras se ejecuta el programa en el procesador de la placa.

Los esquemas realizados en este apartado fueron realizados con Proteus por la facilidad que tiene para definir un esquema electrónico modular y en el que se puede mostrar de manera gráfica los pines del Arduino.

13.1 Cálculo de la velocidad

13.1.1 Conexión

Para poder leer los pulsos que se van generando en los sensores se necesita tener claro cómo se tiene que conectar con los encoders. Se necesita de la placa Arduino, el encoder que se ha estudiado en los apartados anteriores y una fuente de alimentación para alimentar el circuito del encoder con 5V, Figura 63.

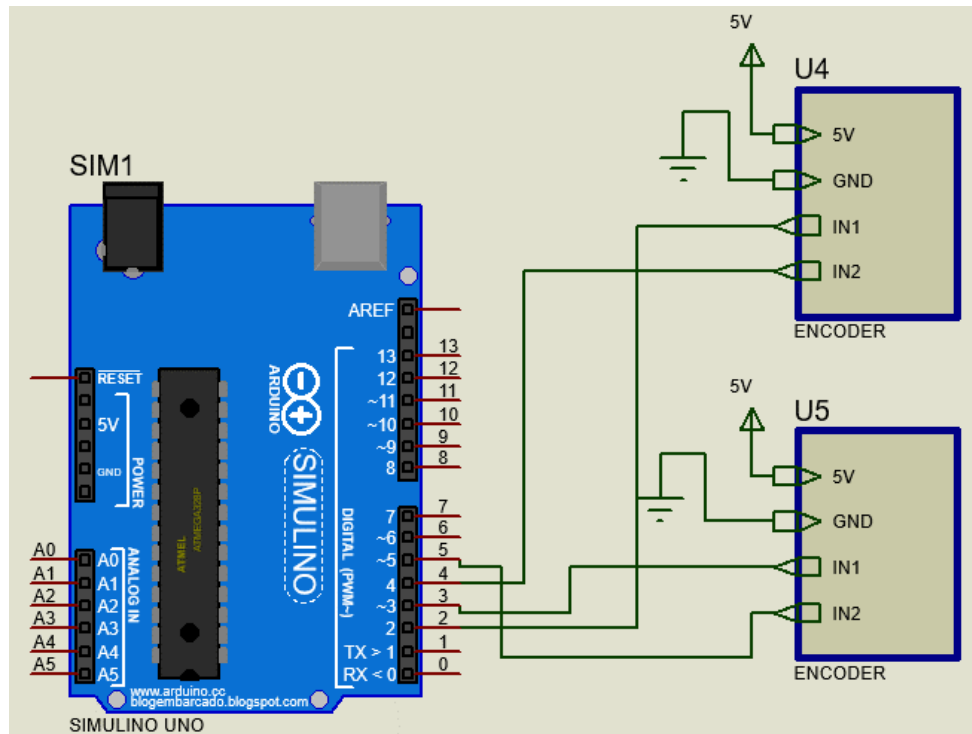


Figura 63. Conexión Encoder.

Los pines que se necesitan para conectar los encoders al Arduino son dos por cada rueda, pero uno de ellos necesita ir a uno de los pines de interrupciones que tenemos en el Arduino. En el caso del Arduino uno son los pines 2 y 3.

13.1.2 Programación

Para poder leer de forma rápida el estado de el pin del otro encoder se hace uso de la librería "DigitalWriteFast.h" y se recuerda que como se necesita hacer uso de las interrupciones se tienen que definir las interrupciones en la inicialización del programa, Figura 64.

Como ya se sabe el Arduino Uno solo tiene un núcleo por lo que solo puede ejecutar una línea de código a la vez por lo que si pretende contabilizar de manera efectiva los pulsos que se generan es necesario colocar una interrupción por cada rueda. Los otros encoder van a pines digitales tradicionales. El diagrama de flujo que describe la programación se puede observar en la Figura 66.



Por último, se tendría la función: “CalculoVelocidad” que se encargaría de a partir de la cantidad de pulsos y teniendo un tiempo estimado se calcula la velocidad, esta se define como se puede observar en la Figura 65.



Figura 64. Inicialización de todo lo necesario para obtener la velocidad.

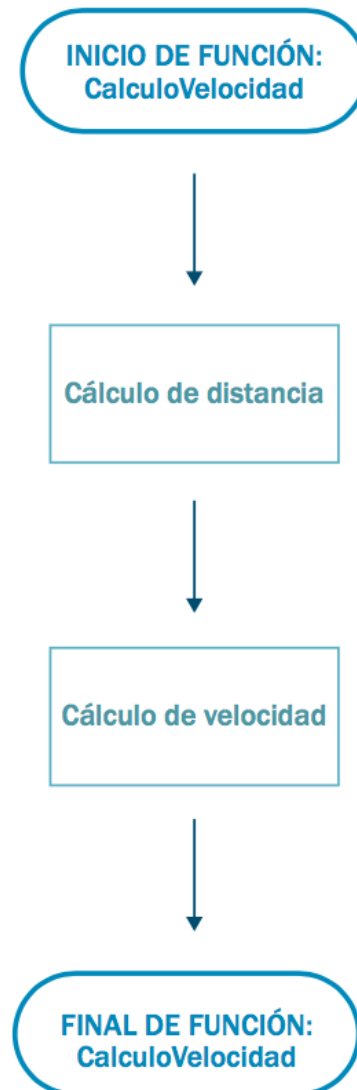


Figura 65. Función “CalculoVelocidad”.

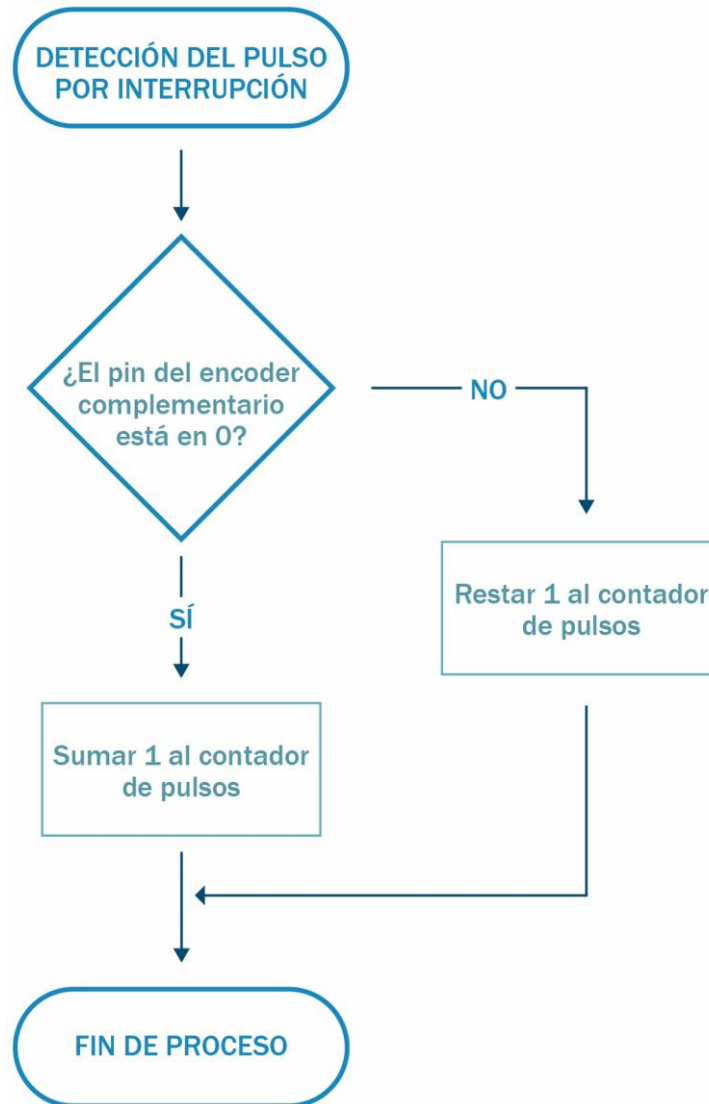


Figura 66. Diagrama de flujo de la interrupción.

13.2 Cálculo de ángulo

13.2.1 Conexión

Se tiene que tener en cuenta que para poder medir el ángulo de inclinación es necesario utilizar el MPU5060 por lo que se tiene que entablar comunicación con el dispositivo. Por lo tanto, se tiene que realizar la conexión de los pines definidos para el puerto I2C. Por último, simplemente se tiene que alimentar, Figura 67.

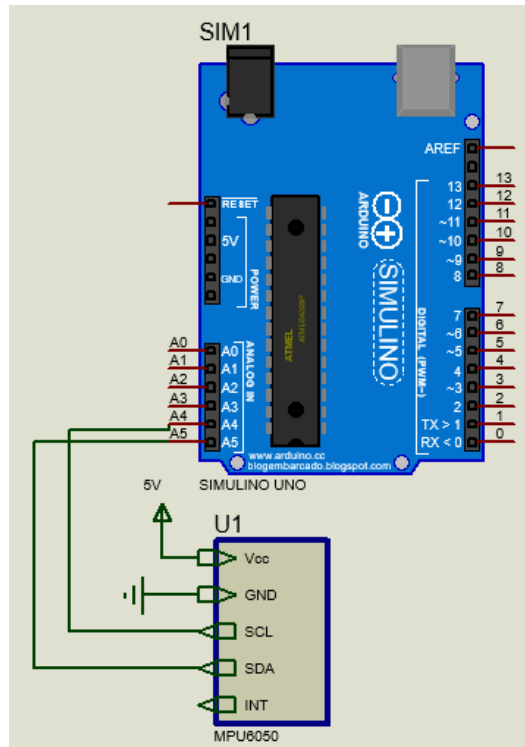


Figura 67. Conexión de MPU5060

13.2.2 Programación

Para poder utilizar el puerto I2C se necesita de la librería “Wire.h” y si se pretende usar el MPU5060 se necesita determinar la velocidad de reloj a 400000Hz y se envía un 0 para activar el módulo, Figura 68. [16]

Después de todo lo anterior se define la función “CalculoAngulo” que se encarga de solicitar al módulo los datos del giroscopio y acelerómetro para poder calcular el ángulo y filtrarlo, Figura 69.



Figura 68. Inicialización de comunicación I2C.



Figura 69. Función "CalculoAngulo".



13.3 Cálculo de PID

13.3.1 Programación

Con la entrada que se obtiene del sensor MPU5060 se calcula el error sabiendo que la referencia es 0 que corresponde con la vertical. Luego se va multiplicando por los constantes K_p , K_i y K_d . Por último, se realiza el anti-windup para que la salida no se vaya hasta valores superiores a 255, que es el valor máximo de salida PWM.

Todo lo anterior se coloca dentro de un “if” para que cuando el robot este con un ángulo muy grande o se haya deshabilitado no calcule nada, Figura 70.

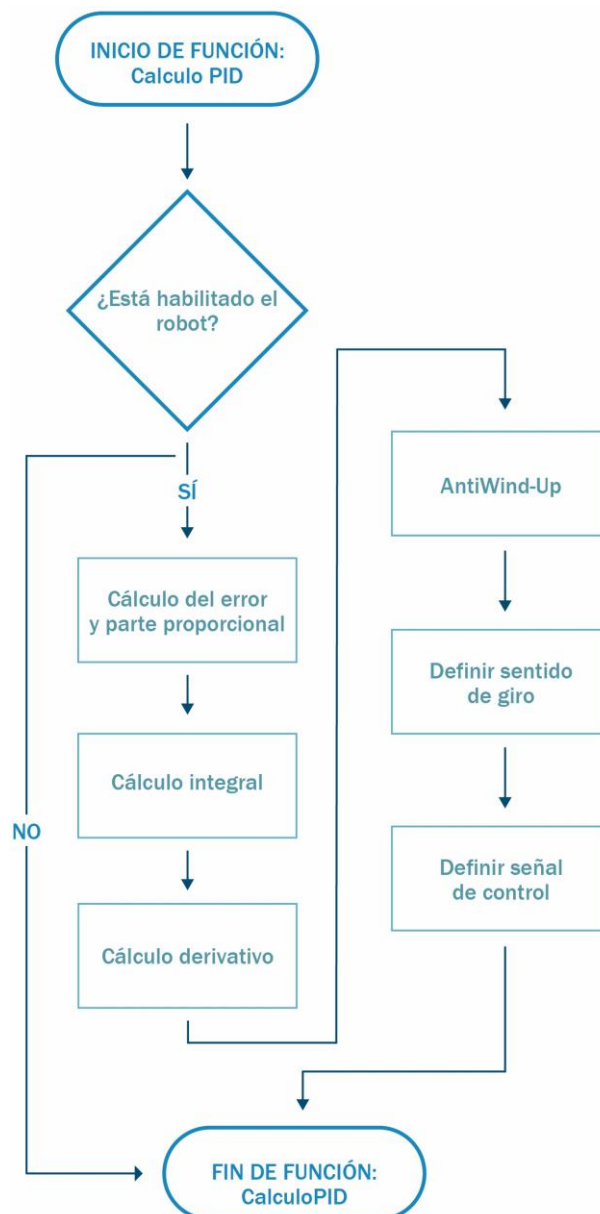


Figura 70. Función “CalculoPID”.

13.4 Bluetooth

13.4.1 Conexión

El módulo Bluetooth se comunica con el Arduino por el puerto serie por lo que se puede conectar con los pines 0 y 1 definidos por defecto como el puerto serie, pero no se hace uso de estos para esta función debido a que estos pines se utilizan internamente para cargar el programa. Por lo que se definen otro par de pines, Figura 71.

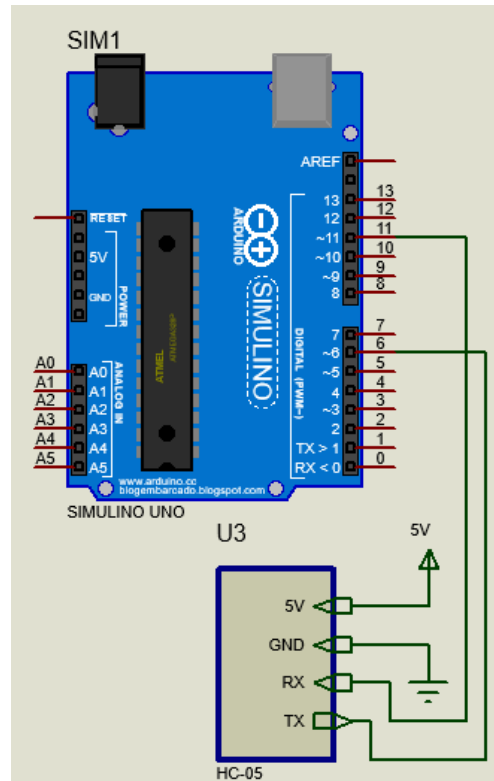


Figura 71. Conexión de Bluetooth HC-05

13.4.2 Programación

Para comenzar a utilizar este módulo simplemente se comienza definiendo la librería “SoftwareSerial.h” para poder crear puertos serie en otros de los pines que no sean los hechos de serie para el Arduino. Se definen cuáles van a ser y por último se define la velocidad de comunicación como 9600 baudios, Figura 72.

La recepción de información se realiza con la función “RecepcionInformacion” que se encarga de verificar si hay un nuevo dato enviado y si es así lo coloca en cada una de las variables a modificar, Figura 73.



Figura 72. Inicialización del Bluetooth.



Figura 73. Función "RecepcionInformacion".

13.5 Enviar la señal a los motores

13.5.1 Conexión

Para ello simplemente se conecta los pines que corresponden a los valores de PWM (9,10) de cada uno de los motores y los que corresponden a los pines del sentido de giro (7,6), Figura 74.

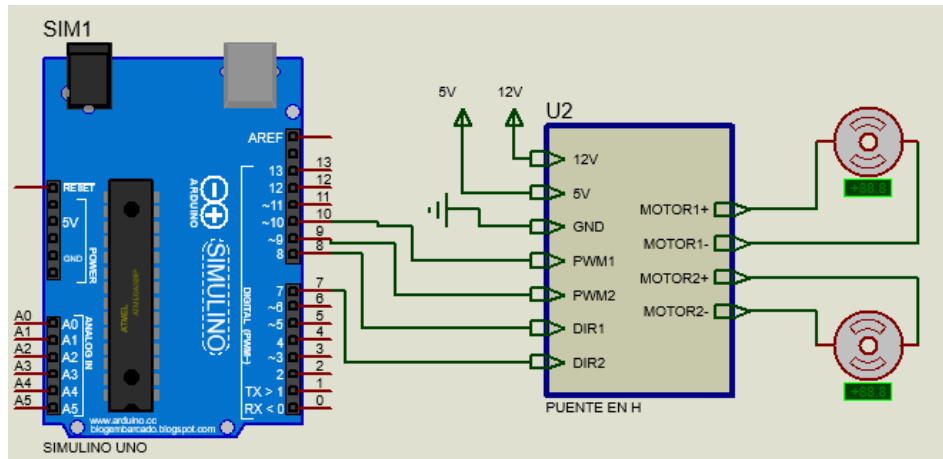


Figura 74. Conexión de Driver de los motores.

13.5.2 Programación

Simply se toma el valor que envía la función “CalculoPID” y si el valor es negativo se determina como 0 el sentido giro y de lo contrario 1. Por último, se escribe el valor PWM en los pines correspondientes, Figura 75.

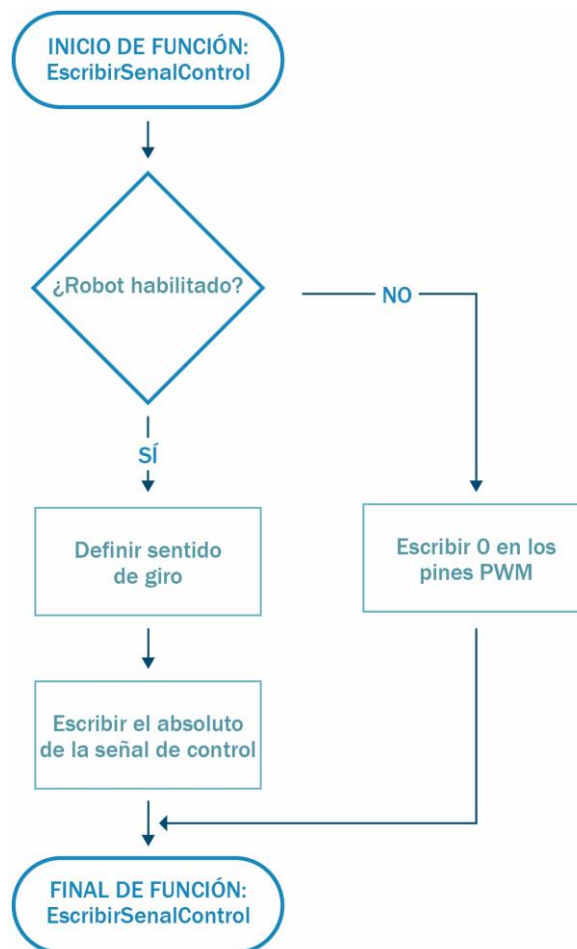


Figura 75. Función “EscribirSenalControl”.



13.6 Loop

13.6.1 Programación

En este bucle se va a ir haciendo las llamadas a las funciones SIEMPRE respetando el tiempo de muestreo como se muestra en la Figura 76.

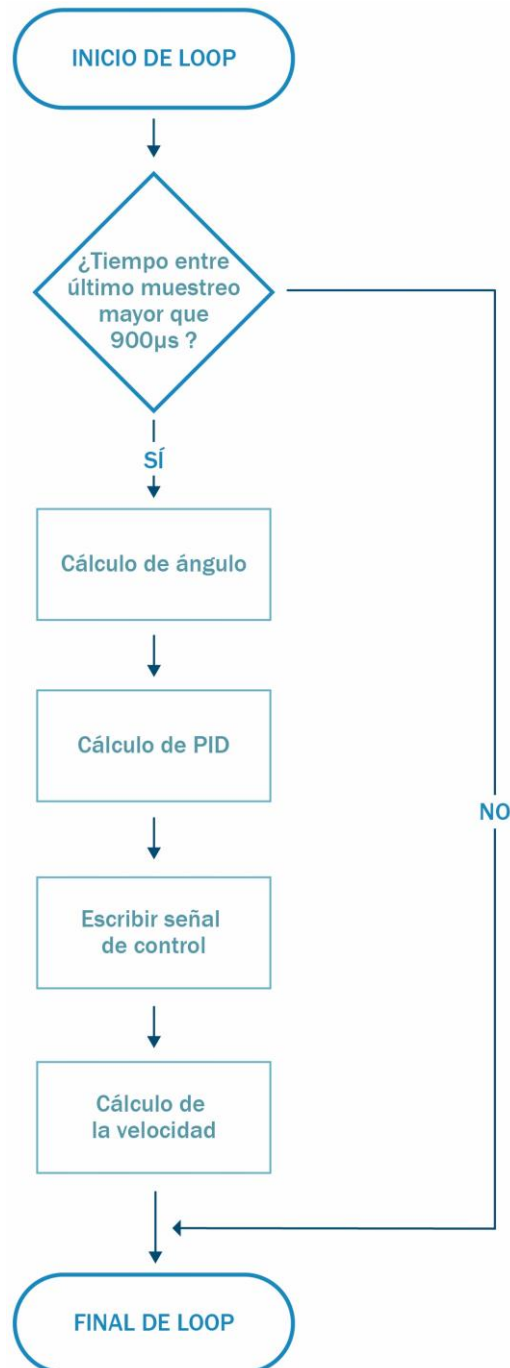


Figura 76. Loop.

14 Conexión eléctrica

La conexión en conjunto de todo el proyecto se define en la Figura 78. Para esto se realizó un estudio de carga para poder definir la batería. Para esto se mide el consumo de cada uno de los módulos y se suma para obtener la carga máxima que se necesita suministrar.

- Puente en H: 0.02 A
- Módulo Bluetooth: 0.05 A
- MPU6050: 0.038 A
- Motor: 0.41 A
- CC/CC: 0.02 A

Si se hace la suma de los valores anteriores se obtiene una carga máxima de 0.948 A. Por lo tanto, se necesita una batería como mínimo de 1000mAh para poder suplir esta demanda.

Con la información anterior se busca una batería de 12V y tiene la capacidad de 3000mAh para poder estar por encima del mínimo que se necesita. [17].



Figura 77. Batería.

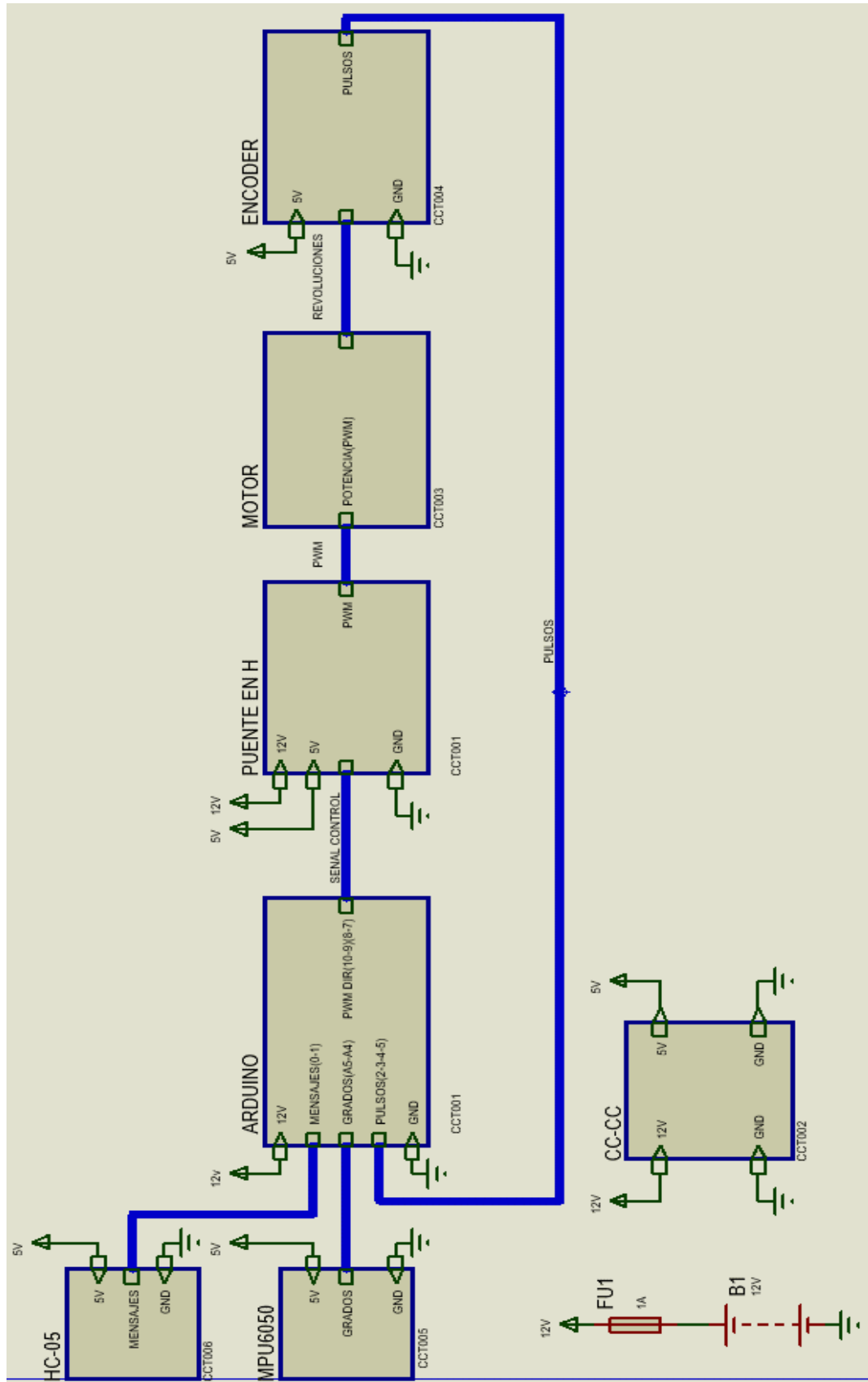


Figura 78. Esquema eléctrico general.



15 Aplicación

Es necesario realizar una aplicación para el móvil debido a que se pretende ir cambiando los valores de controlador PID y deshabilitar al robot.

Para ello se utiliza el software Android Studio con el que se pretende crear una app que se conectará vía Bluetooth para enviar los parámetros Kp, Ki y Kd del controlador y la señal para deshabilitarlo.

15.1 Diseño

Hace falta dos layout una en donde se mostrará el listado de dispositivos vinculados al móvil y otra en donde se introducirán los datos necesarios para enviar al robot.

15.1.1 Dispositivos vinculados

En este layout simplemente se tendrá un título afirmando que en ese layout se tiene la lista de dispositivos vinculados y una lista con ellos, Figura 79. Después de seleccionar el Robot se accede el layout principal.

Dispositivos vinculados	
Robot	20:16:10:25:54:71
MID	2C:4D:79:10:9E:3D
G-311	78:76:0E:FB:48:6C
Beats Solo ³	D4:90:9C:2D:C0:B2
HC-05	00:21:13:04:37:BF

Figura 79. Lista de dispositivos vinculados.



15.1.2 Layout principal

En esta layout, Figura 80, se tienen 3 pares de espacios para introducir texto, cada par de espacios se utilizan para la modificación un parámetro del controlador. En uno se introduce el parámetro que se pretende tener y en el otro sería el incremento por si se quiere ir aumentando o disminuyendo ese valor con ayuda de los botones “subir” y “bajar”.

También se tiene un botón para habilita y deshabilita el robot cuando el robot este inclinado más de 20 grados y el botón de “desconectar” para desactivar la conexión Bluetooth.

The screenshot displays the main layout of the TFGApp. At the top, there is a dark green header with the text "TFGApp". Below the header, the title "Ajuste de parametros PID" is centered. The interface is organized into three vertical sections, each corresponding to a PID parameter: Kp, Ki, and Kd. Each section contains a text input field for the parameter value (all set to 0), a text input field for the increment (all set to 1), and two buttons labeled "BAJAR" and "SUBIR" for adjusting the values. At the bottom of the interface, there are three large, light gray buttons: "ENVIAR VALORES", "HABILITAR/DESHABILITAR ROBOT", and "DESCONECTAR".

Figura 80. Layout principal.



15.2 Ficheros .java

En estos ficheros se dispondrá toda la lógica que tendrán que seguir al realizar alguna acción. A partir de lo que se tiene en el diseño, se comienza a trabajar.

15.2.1 Dispositivos vinculados

En este apartado se tendrá que encender el Bluetooth, seleccionar el dispositivo que previamente estuviera vinculado. En la realizamos el diagrama de flujo que se sigue.

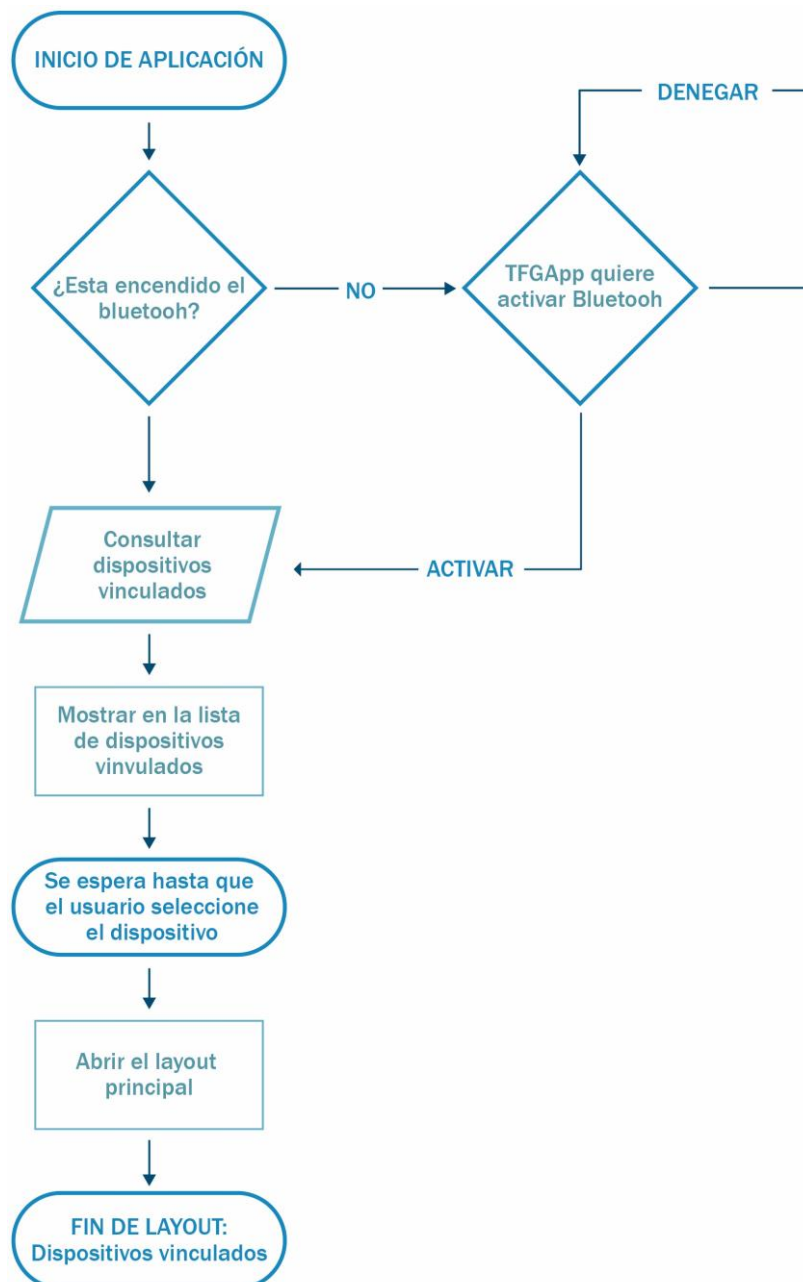


Figura 81. Diagrama de flujo de "Dispositivos Vinculados".



15.2.2 Layout principal

En este layout se va a encargarse de enviar los datos que el usuario quiere introducir en el controlador PID del robot haciendo uso de campos de texto y botones. Como se puede ver en la Figura 80, se hace uso de campos para introducir los valores de las constantes y si se quiere se pueden modificar utilizando los botones “subir” o “bajar” con los que se suma o se resta el valor del incremento al valor de la constante en ese instante.

Los botones por tanto funcionan como interrupciones que se activan en el momento en el que son presionados.

15.2.2.1 Botón de subir

Este botón tiene el mismo funcionamiento, Figura 82, en todas las constantes. Con él, se toma el valor de la constante escrito en el instante de ser presionado y se le suma el incremento que se encuentra en el campo de texto “incremento”.



Figura 82. Diagrama de flujo del botón “Subir”.



15.2.2.2 Botón “Bajar”

Este botón tiene el mismo funcionamiento, Figura 83. Diagrama de flujo del botón “Bajar”, en todas las constantes. Con él, se toma el valor de la constante escrito en el instante de ser presionado y se le resta el incremento que se encuentra en el campo de texto “incremento”.



Figura 83. Diagrama de flujo del botón “Bajar”.

15.2.2.3 Botón “Habilitar/Deshabilitar robot”

Como su nombre indica este botón envía un mensaje al robot para que se habilite o se deshabilite. Dependiendo de lo que se haya enviado previamente enviara el mensaje para desactivarlo o activarlo, Figura 84.

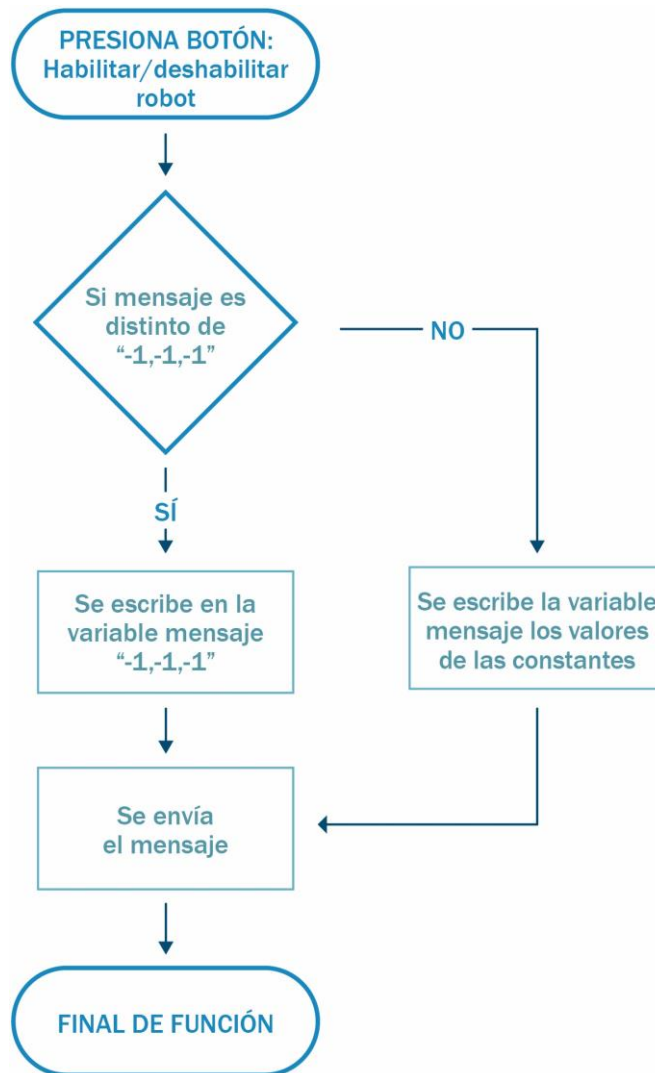


Figura 84. Diagrama de flujo del botón "Habilitar/Deshabilitar robot".

15.2.2.4 Botón "Enviar valores"

Con este botón se va a hacer uso de la conexión Bluetooth que se tenía previamente para enviar los valores que se tienen definidos en los campos de las constantes K_p , K_i y K_d , Figura 85.

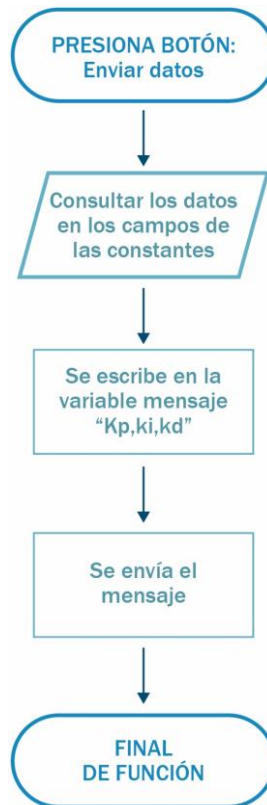


Figura 85. Diagrama de flujo del botón "Enviar valores".

15.2.2.5 Botón "Desconectar"

Este botón se utiliza para realizar la desconexión del dispositivo que se eligió previamente el layout de "Dispositivos vinculados", el funcionamiento de este botón se encuentra en la Figura 86.



Figura 86. Diagrama de flujo de botón "Desconectar".



16 Valoración económica

Este apartado se ha realizado para informar del coste económico que tiene este proyecto. Teniendo en cuenta todo lo que compone al robot se divide el proyecto en dos partes:

- Estructura, Tabla 9.
- Electrónica, Tabla 10.

Nº	Descripción	Cantidad	Coste Unitario	Coste
1	Rueda con esponja 6cm	2	0,65 €	1,30 €
2	Soporte de motor	2	0,50 €	1,00 €
3	Placa fibra de vidrio alta resistencia 12x6.5x0.15 cm	1	0,30 €	0,30 €
4	Placa estructura 12x6.5x0.5 cm	2	0,40 €	0,80 €
5	Tornillo espaciador cabeza hexagonal M3x25 6mm	8	0,20 €	1,60 €
6	Tornillo Allen M3 6mm	10	0,15 €	1,50 €
7	Tornillo Phillips M4 5mm	8	0,10 €	0,80 €
8	Tornillo Phillips cabeza plana M3 5mm	4	0,15 €	0,60 €
9	tuerca M3	10	0,18 €	1,80 €
10	Placa 2.5x8.5 cm	4	1,00 €	4,00 €
			Subtotal:	13,70 €

Tabla 9. Costes de estructura.

Nº	Descripción	Cantidad	Coste Unitario	Coste
1	Arduino Uno	1	20,00 €	20,00 €
2	Puente en H	1	10,00 €	10,00 €
3	MPU6050	1	10,00 €	10,00 €
4	Convertidor Buck CC/CC	1	3,48 €	3,48 €
5	Motor DC 12V 0,41A	2	3,50 €	7,00 €
6	Cables macho-macho	8	0,05 €	0,40 €
7	Cable macho-hembra	5	0,05 €	0,25 €
8	Batería Li-on 12V 3000mAh	1	8,00 €	8,00 €
9	HC-05	1	2,50 €	2,50 €
			Subtotal:	61,63 €

Tabla 10. Costes de electrónica.

Por lo tanto, los costes de todos los componentes sería los expuestos en la Tabla 11.



Nº	Descripción	Coste
1	Estructura	13,70 €
2	Electrónica	61,63 €
	Subtotal:	75,33 €

Tabla 11. Costes de componentes.

Así mismo, para realizar el proyecto en su totalidad además de los componentes se debe tener en cuenta la mano de obra y el alquiler de las herramientas, Tabla 12.

Nº	Descripción	Coste
1	Estructura	13,70 €
2	Electrónica	61,63 €
3	Mano de obra	2.000,00 €
4	Alquiler de herramienta	100,00 €
	TOTAL:	2.175,33 €

Tabla 12. Costes de realización.

17 Diseño estructural

Los componentes estructurales de este robot que sostienen los componentes sobre la base son básicos, placas de plástico y tornillos espaciadores.

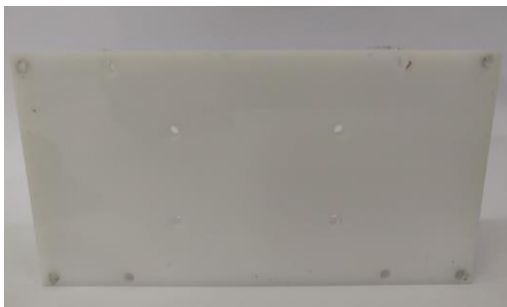


Figura 87. Placa de plástico.



Figura 88. Tornillo espaciador.

La base, Figura 89, se compone de una placa de fibra de vidrio con soporte para los motores.

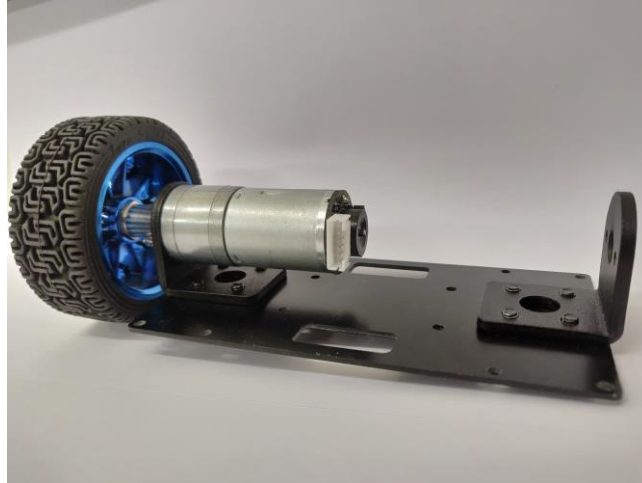


Figura 89. Componentes de la base.

El diseño físico que se consiguió fue un proceso debido a que se estuvieron probando distintas estructuras. Primero se intentó conseguir una estructura con 4 plataformas donde en cada una de ellas se encontraban los componentes necesarios, colocando la batería arriba para que sirviera de contra peso ante las vibraciones, Figura 90. Primera estructura.

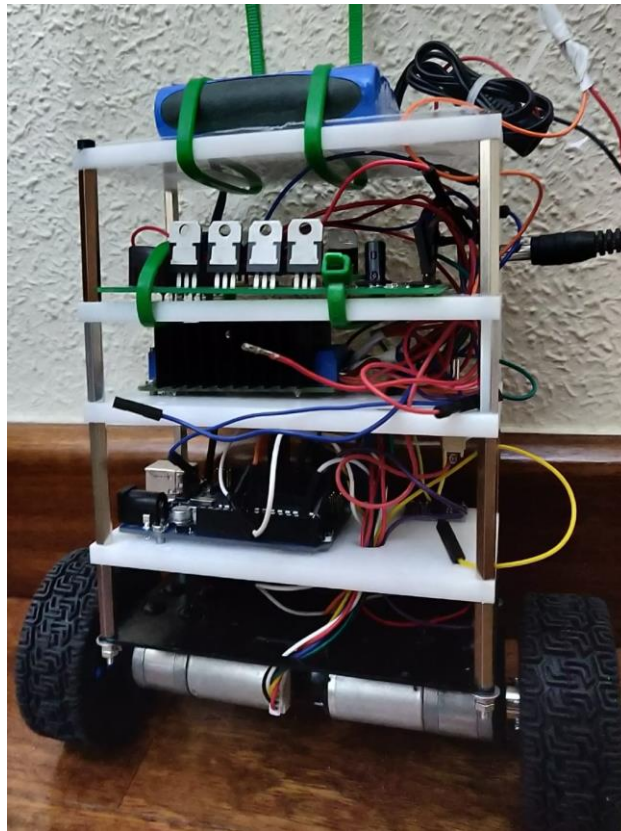


Figura 90. Primera estructura.

Luego al ver que no se estabilizaba se incrementó la altura y se le colocó un peso en la parte superior (de prueba) para comprobar que aumentando la altura se podía estabilizar más, Figura 91. Lo hizo, pero esta estructura necesitaba de un gran torque por parte de los motores haciendo que vibraran mucho y junto con la holgura que tiene los piñones de la relación de transmisión terminaba cayendo.

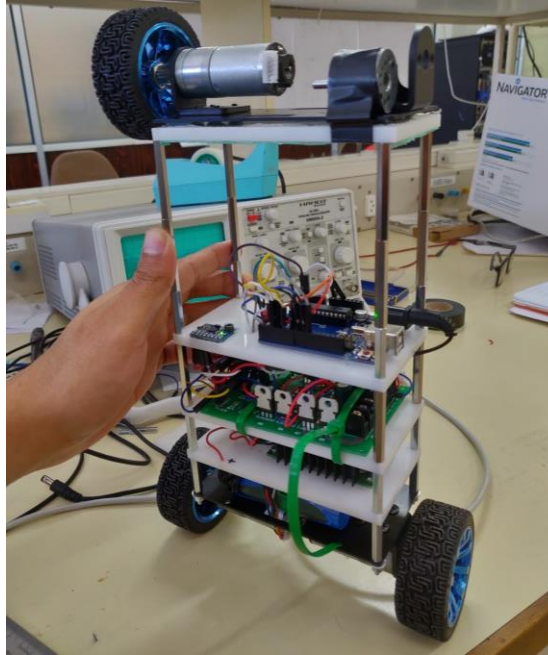


Figura 91. Segunda estructura

Por lo tanto, se decide bajar la altura de la estructura lo máximo posible y se decide realizar la estructura de la Figura 92. Tercera estructura. Esta estructura mejora, pero se denota que vibra mucho.

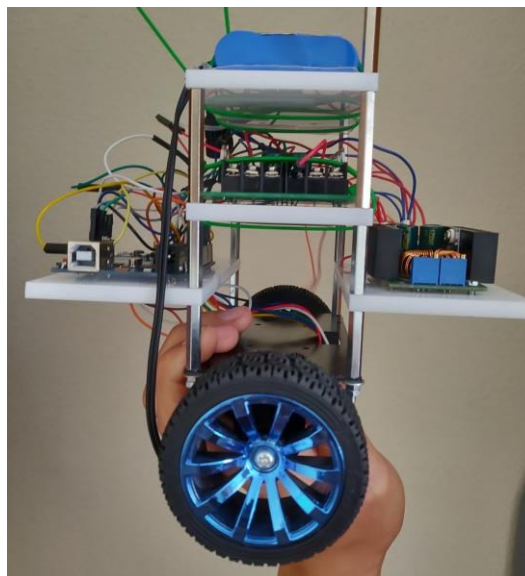


Figura 92. Tercera estructura.

Se decide seguir disminuyendo la altura, pero ahora se tiene que agrupar lo máximo que se pueda los componentes, Figura 93. Esta fue la última estructura más estable.

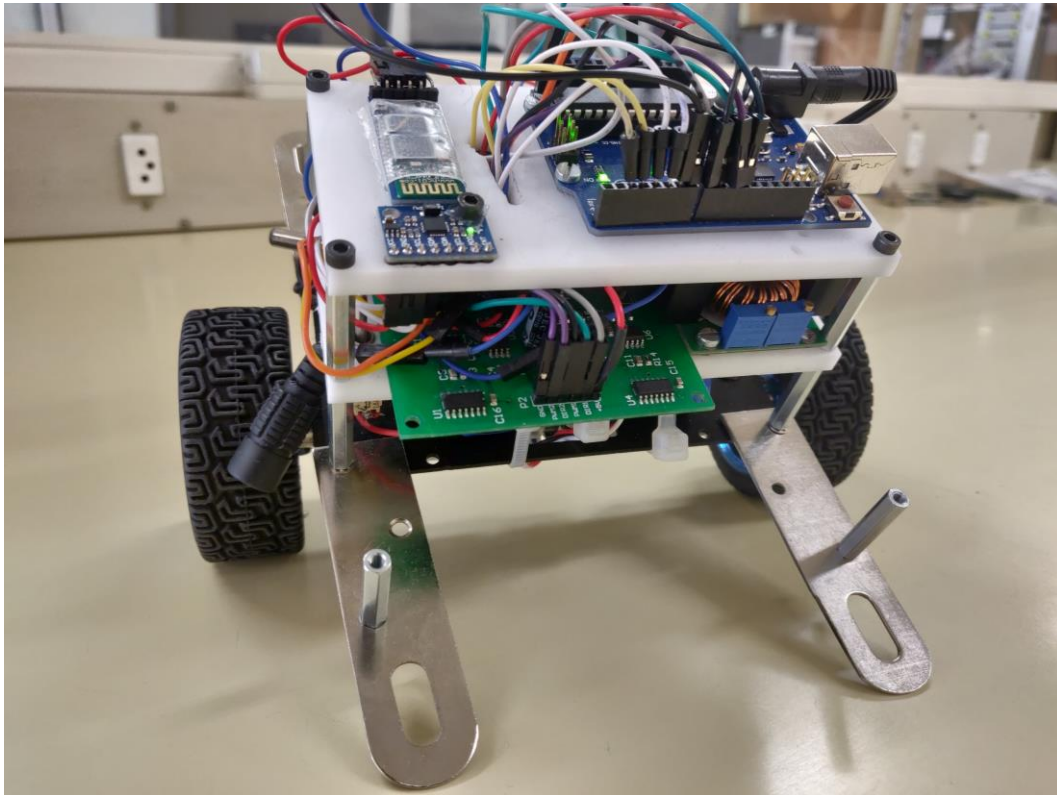


Figura 93. Estructura final.



18 Conclusiones

En este proyecto se ha diseñado y creado el robot auto balanceado propuesto mediante el estudio de un modelo matemático lo más cercano a la realidad utilizando para ello la herramienta Matlab.

Después de seleccionar los componentes que mejor se acercaban a los objetivos, se pretendió programar el robot con Simulink obteniendo una velocidad de muestreo insuficiente. Para su mejora se pasó a la programación en el IDE de Arduino.

Después de invertir bastante tiempo en mejorar la estructura y la programación del robot, se llegó a la conclusión de que se necesitaban componentes con otras características para poder tener un comportamiento mucho más estable tal y como se detalla a continuación

Este proyecto es muy importante debido a que presenta unas conclusiones sólidas sobre el camino que se tiene que seguir para continuar mejorando. Por esa razón, se plantea el siguiente apartado.

18.1 Líneas de continuación del proyecto

18.1.1 Cambio de motores

18.1.1.1 Problema

Uno de las grandes limitaciones que se tuvo y que genero MUCHOS problemas fue el utilizar los motores de corriente continua con relación de transmisión con piñones rectos, Figura 94, estos piñones no eliminan el juego entre sus dientes por lo que al cambiar el sentido de giro se produce un golpe entre los dientes que, entre mayor sea la tensión en los bornes del motor se traduce en una vibración mayor.



Figura 94. Piñones rectos [18].

Según el circuito utilizado para alimentar a los motores, se utiliza onda PWM para alimentar los motores que va variando la tensión media en los bornes de los motores, pero la tensión máxima de esa onda viene dada por la tensión en los bornes de la batería. Por lo tanto, dependiendo de si la batería está más o menos cargada, le llegará más o menos tensión máxima. Si la tensión máxima varía la señal de control genera distintos comportamientos y en consecuencia los valores de PID no serán los mismos.

18.1.1.2 Solución

La primera solución que eliminaría todos los problemas referidos al motor sería colocar motores paso a paso, Figura 95. Estos motores son muy precisos, no necesitan relación de transmisión y ya no tendría dependencia de la tensión de la batería debido al driver de control de los motores.



Figura 95. Motor paso a paso [4].

Si no se pretende hacer ese salto a motores paso a paso, otra solución sería colocar un CC/CC para regular la tensión máxima que llega a los bornes del motor y escoger motores que tengan relaciones de transmisión con piñones helicoidales, Figura 96, que tienen menos tolerancia entre dientes.



Figura 96. Piñones helicoidales [19].



18.1.2 Estructura

18.1.2.1 Problema

El peso ha sido uno de los problemas encontrados debido a que entre más peso tiene más inercia tiene y más lento es el sistema para estabilizarse. Se necesita realizar la estructura lo más liviana posible.

18.1.2.2 Solución

Se puede realizar la estructura con materiales como la madera por su poco peso y tiene la rigidez suficiente, realizando la fabricación con corte laser debido a su exactitud.

También se podría realizar la estructura con un diseño impreso en 3D, este diseño es liviano y si se realiza adecuadamente puede llegar a ser muy compacto.





19 Bibliografía

- [1] Wikipedia, «Wikipedia,» [En línea]. Available: <https://es.wikipedia.org/wiki/Segway>.
- [2] «Segway,» [En línea]. Available: <http://host.web-print-design.com/segway/your-first-ride/move-forward-back2.htm>.
- [3] «xataka,» [En línea]. Available: <https://www.xataka.com/robotica-e-ia/segway-monta-un-robot-sobre-su-conocido-sistema-de-transporte-personal>.
- [4] «TdRobotica,» [En línea]. Available: https://www.google.com/url?sa=i&source=images&cd=&ved=2ahUKEwj-uNzToPXiAhU4A2MBHaHvCjMQjRx6BAgBEAU&url=http%3A%2F%2Fdrobotica.co%2Fmotor-pap-nema-17-bipolar-200-pasosrev37kg-cm42x38mm%2F191.html&psig=A0vVaw0nK_nWJO3sN9rhq-lqnyyb&ust=1561023690888282.
- [5] Rocco, «Rocco,» [En línea]. Available: https://www.amazon.es/Rocco-Giocattoli-Juguetes-88044-Interactivo/dp/B01N2322I5/ref=pd_sbs_21_27?_encoding=UTF8&pd_rd_i=B01N2322I5&pd_rd_r=75aef078-8bae-11e9-9a9a-49585dea68e8&pd_rd_w=2KZWn&pd_rd_wg=GJOGg&pf_rd_p=f9384d3f-fa3d-4e25-8bc3-b0c7853cd8a6&pf_rd_r=EWDZN798FM5G72NXG4KK&psc=1&refRID=EWDZN798FM5G72NXG4KK.
- [6] «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Inverted_pendulum.
- [7] P. D.-I. S. Liu, «Univerisdad de Kaiserslautern,» [En línea]. Available: <https://www.eit.uni-kl.de/liu/>.
- [8] u. d. Kaiserslautern. [En línea]. Available: https://github.com/turnwald/CAE_Exercise/blob/master/Uebung_CAE.pdf.
- [9] P. D.-I. S. Liu. [En línea]. Available: <https://www.eit.uni-kl.de/liu/en/english.html>.
- [10] U. N. d. I. Plata. [En línea]. Available: https://catedra.ing.unlp.edu.ar/electrotecnia/controlm/electronica/archivos/apuntes/Controlabilidad_y_observabilidad.pdf.
- [11] «ece,» [En línea]. Available: <http://www.ece.northwestern.edu/local-apps/matlabhelp/toolbox/control/ref/acker.html>.
- [12] «Texas Instruments,» [En línea]. Available: <http://www.ti.com/>.
- [13] Aliexpress. [En línea]. Available: <https://es.aliexpress.com/item/32964118989.html?storeId=4460045>
- [14] «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Buck_converter.
- [15] «Electronics Notes,» [En línea]. Available: <https://www.google.com/url?sa=i&source=images&cd=&ved=2ahUKE>



wixp-

WGv_biAhUG3xoKHSIDBpsQjRx6BAgBEAU&url=https%3A%2F%2Fwww.electronics-notes.com%2Farticles%2Fanalogue_circuits%2Fpower-supply-electronics%2Fswitching-step-down-buck-regulator-dc-dc-converter.php&psig=AOvVaw3HMpBleVXhzj-2anbCyydN&ust=1561066175509258.

- [16] «Robologs,» [En línea]. Available: <https://robologs.net/2014/10/15/tutorial-de-arduino-y-mpu-6050/>.
- [17] Aliexpress. [En línea]. Available: https://es.aliexpress.com/item/32903092862.html?storeId=3101021&spm=a219c.search0104.3.1.758f613f0rdZ2m&ws_ab_test=searchweb0_0%2Csearchweb201602_2_10065_10068_10547_10546_319_10059_10884_10548_317_10887_10696_321_322_453_10084_454_10083_10103_10618_10307_10820_10303_537_536%2Csearchweb201603_53%2CpccSwitch_0&algo_expid=e9133a81-ea55-4fb5-a8d4-f176c017367a-0&algo_pvid=e9133a81-ea55-4fb5-a8d4-f176c017367a.
- [18] «Engranajes de usto,» [En línea]. Available: https://www.google.com/url?sa=i&source=images&cd=&ved=2ahUKEwicjJOcoPXiAhUvBGMBHbiND-IQjRx6BAgBEAU&url=https%3A%2F%2Fwww.engranajesdeusto.com%2F&psig=AOvVaw3m8VJVrUrRDB65pFFUJtTt_&ust=1561022620833829.
- [19] [En línea]. Available: <https://www.google.com/url?sa=i&source=images&cd=&ved=2ahUKEwicrvzXpPXiAhWOERQKHT94DUsQjRx6BAgBEAU&url=https%3A%2F%2Fwww.metalmeccanica-facil.mahtg.com%2Fcalculo-de-engranajes-cilindricos-helicoidales-sistema-milimetrico%2F&psig=AOvVaw07GkSdZAhQT7ED23XiDXhr&ust=1561024437528263>.
- [20] «Wikipedia,» [En línea]. Available: <https://es.wikipedia.org/wiki/Segway>.



20 Anexos

20.1 Manual de usuario

Se tiene que recordar que el robot está programado para enviar tensión a los motores cuando el ángulo del robot es menor de 20° con respecto a la vertical. Para el uso adecuado del robot se tiene que realizar los pasos para encenderlo y apagarlo.

20.2 Encender robot

1. Conectar el terminal hembra de la batería con el macho que está conectado al interruptor.
2. Conectar el Arduino a la batería con el terminal macho que sale de la batería.
3. Cerrar el circuito con el interruptor para poder alimentar los módulos.
4. Si no actúan los motores a pesar de que el ángulo con la vertical es menor de 20° se tendrá que reiniciar el Arduino con el botón "Reset".

20.3 Apagar robot

1. Abrir el circuito con el interruptor en el lateral del robot.
2. Desconectar el Arduino de la batería.
3. Desconectar la batería del interruptor.

20.4 Conectar con Aplicación

1. Asegurarse de que el robot y el módulo Bluetooth estén encendidos.
2. Si es la primera vez comunicar con un móvil, se tiene que instalar la aplicación "AppTFG".
3. Vincular el módulo Bluetooth con el móvil yendo a Ajustes \Rightarrow Bluetooth y conexión de dispositivos \Rightarrow vincular nuevo dispositivo.
4. Abrir "AppTFG" \Rightarrow vincular con el módulo del Bluetooth seleccionándolo en la lista Figura 97.



Figura 97. Vincular dispositivo.



5. Colocar valores deseados y presionar “enviar valores”, Figura 98. Para aumentar o disminuir el valor con los botones “subir” o “bajar” respectivamente.

TFGApp

Ajuste de parametros PID

Kp: 0

Incremento 1

BAJAR SUBIR

Ki: 0

Incremento 1

BAJAR SUBIR

Kd: 0

Incremento 1

BAJAR SUBIR

ENVIAR VALORES

HABILITAR/DESHABILITAR ROBOT

Dato:

DESCONECTAR

Figura 98. Menú principal.

Se recuerda para que la información enviada se admitida por el robot en el momento en el que ángulo sea mayor de 20° y lo vuelvan a poner de forma manual a la posición vertical.

También se puede habilitar o deshabilitar el robot presionando el botón “Habilitar/deshabilitar robot”

20.5 Desconectar

Para desconectar el móvil del módulo Bluetooth simplemente se tiene que presionar el botón “Desconectar”.



Universidad de Valladolid