E.T.S. de Ingeniería Industrial, Informática y de Telecomunicación

DISEÑO E IMPLEMENTACIÓN DE ROBOT SEGWAY

Grado en Ingeniería Eléctrica y Electrónica

Trabajo Fin de Grado

Ion Irigoyen Martínez

Ignacio Del Villar Fernández

Pamplona, 29 de Junio de 2015

upna Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

ÍNDICE

1. INTRODUCCION

En el presente Trabajo Fin de Grado se desarrolla el diseño y la implementación de un robot segway. Se presentan las dos alternativas entre las muchas que pueden adoptarse para la construcción del mencionado robot:

- Robot segway basado en sensor infrarrojo y control realizado a través de Labview
- Robot segway basado en giroscopio y acelerómetro y control realizado a través de Arduino

Entre las alternativas modernas de transporte de pasajeros orientadas a tramos cortos se cuenta un tipo de vehículo cuyo funcionamiento está basado en el equilibrio humano. Se trata de un vehículo eléctrico que aunque no haya alcanzado introducirse en la sociedad actual como medio de transporte, presenta unas características muy adecuadas al uso por la ciudad debido a su tamaño y maniobrabilidad. Además se encuentra su componente ecológica, ya que funciona íntegramente con energía eléctrica, lo cual disminuiría en consideración la contaminación producida por los coches en las grandes ciudades.

Fig. 1: Segway utilizado como medio de transporte urbano

Fuente: http://www.legatraveler.com/2015/03/madrid-en-segway-tour-segway-trip

Este trabajo presenta todas las fases de diseño y construcción de un vehículo de este tipo, consistente en una base con dos ruedas laterales accionadas por motores DC, los que responden en función de la inclinación y velocidad de inclinación del pasajero.

A partir de la idea del vehículo de transporte se construirán dos robots que simulen el funcionamiento de dicho vehículo. Para ello contaran con sensores que den la información necesaria de inclinación. Estos datos serán recibidos y tratados para la obtención de información útil necesaria en los sistemas de control utilizados (Labview y Arduino). A partir de dicha información se construirá el algoritmo de control que dé la respuesta necesaria para que el robot segway consiga mantenerse en equilibrio.

El trabajo fin de grado contiene en su desarrollo diferentes áreas de la ingeniería, en las que destacan:

- Mecánica: tener en consideración el soporte mecánico del que se compone el robot

- Electrónica: para la adecuada elección de los componentes electrónicos

- Potencia: para la transmisión de la acción de control a los motores que harán moverse al robot

- Control: parte esencial del robot ya que un adecuado algoritmo de control permitirá el funcionamiento óptimo del robot segway

- Programación: necesaria para trasladar los conceptos de control a un software y que dé cómo resultado la acción de control necesaria

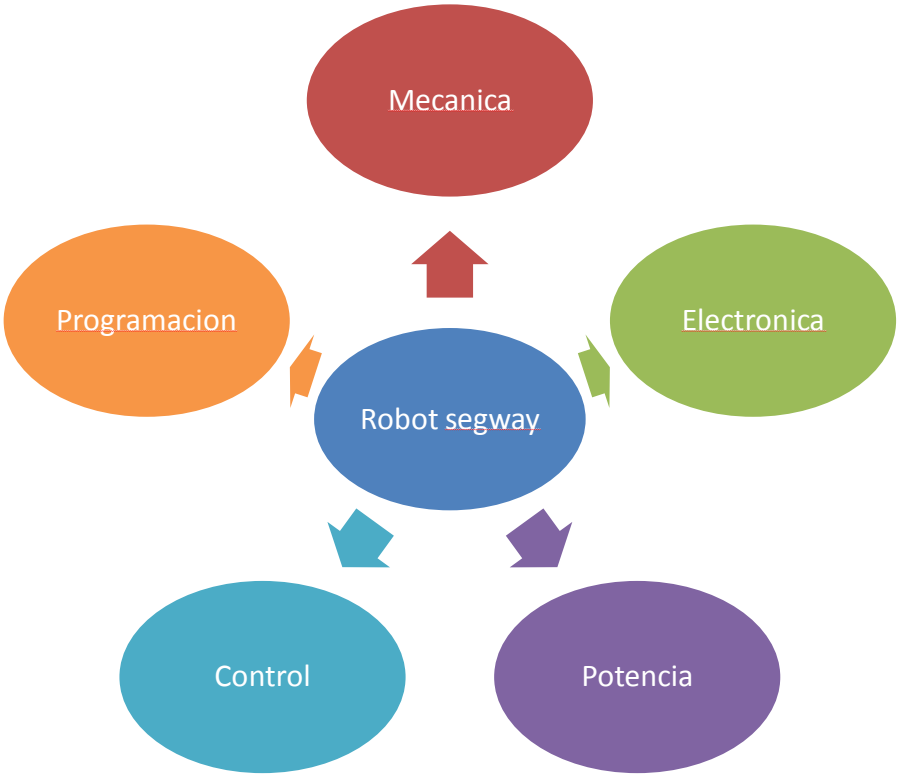


Fig. 2: Áreas presentes en el desarrollo de un robot segway

2. OBJETIVOS Y JUSTIFICACION

El objetivo principal del proyecto es la construcción de dos robots segway (auto balanceado) mediante dos tecnologías diferentes y la comparación de los mismos. Para ello se deben alcanzar unos objetivos menores que ayuden al desarrollo del proyecto. Estos son:

- Alternativas para la construcción mecánica (chasis)
- Componentes necesarios
- Obtención de datos a través de los diferentes sensores
- Tratamiento de los datos recibidos
- Programación del software empleado en cada caso
- Ensayos de prueba y ensayos de verificación

3. ANTECEDENTES

En el siguiente apartado se muestra la base teórica para comprender el funcionamiento del robot segway.

Posteriormente, se citarán los diferentes tipos de sensores que podrían emplearse a la hora de obtener los datos de inclinación y velocidad de inclinación. Además se deberá escoger el software en el que se programará el algoritmo de control. Este último también puede ser de varios tipos, que también se mostraran en el presente apartado.

3.1 PRINCIPIO DE FUNCIONAMIENTO

Un péndulo invertido es un péndulo que tiene su centro de masa por encima de su punto de pivote.

El péndulo invertido es conocido por ser uno de los problemas más importantes y clásicos de la teoría de control. Considerando que un péndulo normal es estable cuando se cuelga hacia abajo, un péndulo invertido es inestable, y debe ser equilibrado activamente con el fin de permanecer en posición vertical. El sistema está compuesto por un carro en el cual se coloca una barra que puede girar libremente. El objetivo es que el carro se desplace para compensar el movimiento de caída al que tendera la barra, y como consecuencia esta debe mantenerse en equilibrio.

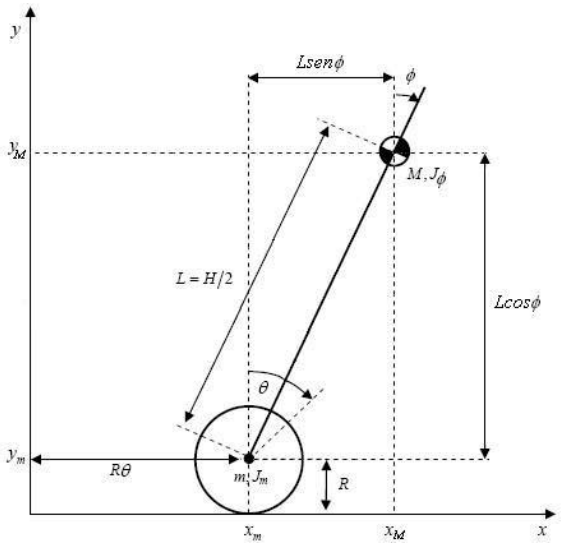


Fig. 3: Diagrama de funcionamiento del péndulo invertido

Fuente: http://webdelprofesor.ula.ve/ingenieria/apatete/resumen_hector.html

Se trata de un problema inestable y además no lineal. Suele utilizarse como ejemplo para mostrar las diferencias entre un control en lazo cerrado y en lazo abierto. Existen diferentes técnicas a la hora de diseñar un controlador que sea capaz de estabilizar el péndulo. El tipo de control que se emplee dependerá de diversos factores, entre los que se encuentran la linealidad de la planta, estabilidad etc. Hasta hace pocos años lo habitual era el empleo de controles proporcional, integral o derivativo o una combinación de ellos. Actualmente cada vez es más común utilizar los denominados controles inteligentes. [1]

3.2 SENSORES

Los sensores nos proporcionan la información necesaria para conocer el estado del robot en este caso. Mediante los mismos podemos conocer el valor del error que la planta del sistema tiene respecto a la consigna establecida. A partir de ello podremos diseñar e implementar el algoritmo de control para conseguir un sistema estable que se controle por sí mismo.

Existe una variedad inmensa de sensores. Por ello, a continuación solo se muestran algunos de ellos.

3.2.1 SENSOR INFRARROJO

El detector de proximidad por infrarrojo es quizás uno de los elementos de mayor aplicación en el automatismo electrónico**.**

Lo encontramos en dispensadores de agua automáticos, los secadores de mano automáticos y con algunas variantes lo encontramos en las puertas automáticas de los grandes almacenes.

Principio de funcionamiento

Generamos una ráfaga de pulsos de alta intensidad a baja frecuencia y los transmitimos por el led de chorro infrarrojo**.**

Luego los recibimos en un fototransistor colocado de tal manera que solo los reciba cuando un objeto refleje los pulsos.

Luego procesamos esa señal para poder utilizarla en el encendido-apagado de nuestros aparatos.

Para ello colocamos un fototransistor de tal manera que cuando haya una superficie que refleje los pulsos, bien sea una mano, un objeto, etc. Este los pueda recibir y enviar a un amplificador de corriente. [2]

3.2.2 SENSOR DE ULTRASONIDOS

Los sensores de ultrasonidos son detectores de proximidad que trabajan libres de roces mecánicos y que detectan objetos a distancias que van desde pocos centímetros hasta varios metros. El sensor emite un sonido y mide el tiempo que la señal tarda en regresar. La señal enviada rebota en el objeto, el sensor recibe el eco producido y lo convierte en señales eléctricas, las cuales son elaboradas en el aparato de valoración. Estos sensores pueden detectar objetos con diferentes formas, colores, superficies y de diferentes materiales. Los materiales pueden ser sólidos, líquidos o polvorientos. Sin embargo han de ser deflectores de sonido. Los sensores trabajan según el tiempo de transcurso del eco, es decir, se valora la distancia temporal entre el impulso de emisión y el impulso del eco. [3]

3.2.3 GIROSCOPIO

El giróscopo o giroscopio (del griego "skopeein = ver" y "gyros = giro") es un dispositivo mecánico que sirve para medirla orientación en el espacio de algún aparato o vehículo.

Está formado esencialmente por un cuerpo con simetría de rotación que gira alrededor del eje de dicha simetría. Cuando el giróscopo se somete a un momento de fuerza que tiende a cambiar la orientación de su eje de rotación, tiene un comportamiento aparentemente paradójico, ya que cambia de orientación (o experimenta un momento angular en todo caso, si está restringido) girando respecto de un tercer eje, perpendicular tanto a aquel respecto del cual se lo ha empujado a girar, como a su eje de rotación inicial. Si simplemente gira libre en el espacio, el giróscopo conserva la orientación de su eje de rotación ante fuerzas externas que tiendan a desviarlo mejor que un objeto no giratorio; se desvía mucho menos, y en una dirección diferente. [4]

3.2.4 ACELERÓMETRO

Se denomina acelerómetro a cualquier instrumento destinado a medir aceleración. Esta no es necesariamente la misma que la aceleración de coordenadas (cambio de la velocidad del dispositivo en el espacio), sino que es el tipo de aceleración asociada con el fenómeno de peso experimentado por una masa de prueba que se encuentra en el marco de referencia del dispositivo. Un ejemplo en el que este tipo de aceleraciones son diferentes es cuando un acelerómetro medirá un valor sentado en el suelo, ya que las masas tienen un peso, a pesar de que no hay cambio de velocidad. Sin embargo, un acelerómetro en caída gravitacional libre hacia el centro de la Tierra medirá un valor de cero, ya que, a pesar de que su velocidad es cada vez mayor, está en un marco de referencia en el que no tiene peso. [5]

3.2 SOFTWARES

Del mismo modo que sucede con los sensores, la variedad de softwares que se pueden utilizar para la programación del robot es amplia. Por esta razón a continuación se detallan algunos de los posibles controladores que se pueden llegar a utilizar.

3.3.1 LABVIEW

LabView (acrónimo de Laboratory Virtual Instrumentation Engineering Workbench) es una plataforma y entorno de desarrollo para diseñar sistemas, con un lenguaje de programación visual gráfico. Recomendado para sistemas hardware y software de pruebas, control y diseño, simulado o real y embebido, pues acelera la productividad. El lenguaje que usa se llama lenguaje G, donde la G simboliza que es lenguaje gráfico.

Su principal característica es la facilidad de uso, válido para programadores profesionales como para personas con pocos conocimientos en programación, que pueden hacer programas relativamente complejos, imposibles para ellos de hacer con lenguajes tradicionales. También es muy rápido hacer programas con LabView y cualquier programador, por experimentado que sea, puede beneficiarse de él. Los programas en LabView son llamados instrumentos virtuales (VIs). Para los amantes de lo complejo, con LabView pueden crearse programas de miles de VIs (equivalente a millones de páginas de código texto) para aplicaciones complejas, programas de automatizaciones de decenas de miles de puntos de entradas/salidas, proyectos para combinar nuevos VIs con VIs ya creados, etc. Incluso existen buenas prácticas de programación para optimizar el rendimiento y la calidad de la programación. [6]

3.3.2 ARDUINO

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares.

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168 , Atmega328, Atmega1280 , y Atmega8. El gran uso de estos microcontroladores se debe a su sencillez y bajo coste, lo que permite el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque que es ejecutado en la placa.

La plataforma Arduino se programa mediante el uso de un lenguaje propio basado en el lenguaje de programación de alto nivel Processing que es similar a C++. [7]

3.3.3 MATLAB

MATLAB (abreviatura de *MATrixLABoratory*, "laboratorio de matrices") es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows, Mac OS X y GNU/Linux .

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI). Además, se pueden ampliar las capacidades de MATLAB con las cajas de herramientas (toolboxes); y las de Simulink con los paquetes de bloques (blocksets).

Es un software muy usado en universidades y centros de investigación y desarrollo. En los últimos años ha aumentado el número de prestaciones, como la de programar directamente procesadores digitales de señal o crear código VHDL. [8]

3.3 SISTEMAS DE REGULACION Y CONTROL

Los sistemas de regulación y control se clasifican en dos tipos:

- Sistemas de control en lazo abierto.
- Sistemas de control en lazo cerrado.

La exactitud de estos sistemas depende de su programación previa. Es preciso prever las relaciones que deben darse entre los diferentes componentes del sistema, a fin de tratar de conseguir que la salida alcance el valor deseado con la exactitud prevista. [9]

La siguiente figura muestra un control en lazo abierto. Únicamente se dispone de la variable de entrada que se asignara como consigna. El controlador calculara la acción de control que posteriormente se aplicara sobra la planta, es decir, sobre el sistema que queremos controlar. Finalmente la planta responderá ante el control produciendo una variable de salida que si el control está bien diseñado deberá ser igual o similar a la variable de entrada.

El diagrama de bloque de un sistema en lazo abierto es:

Fig. 4: Diagrama de bloques de un control en lazo abierto

La principal diferencia que hay con el lazo cerrado, es que en este último se realiza una medición de la variable de salida y se calcula el error respecto a la entrada. Se trata de un sistema realimentado.

El diagrama de bloques correspondiente a un sistema de control en lazo cerrado es:

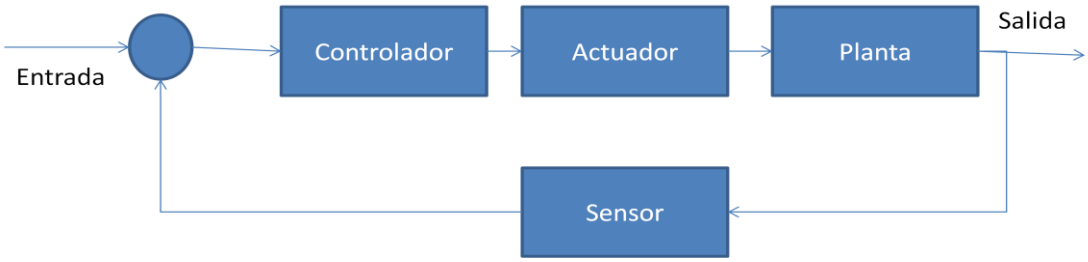


Fig. 5: Diagrama de bloques de un control en lazo cerrado

Los sistemas en lazo cerrado presentan las siguientes ventajas frente a los de lazo abierto.

- Más exactos en la obtención de los valores requeridos para la variable controlada.
- Menos sensibles a las perturbaciones.
- Menos sensibles a cambios en las características de los componentes.
- Mayor rapidez de respuesta

Aunque tienen las siguientes desventajas:

- Son más caros.
- Al ser más complejos son más propensos a tener averías, y presentan mayor dificultad en su mantenimiento.

Podemos afirmar que los tres principales tipos de controles son los tres siguientes [10]:

Proporcional

La parte proporcional consiste en el producto entre la señal de error y la constante proporcional para lograr que el error en estado estacionario se aproxime a cero, pero en la mayoría de los casos, estos valores solo serán óptimos en una determinada porción del rango total de control, siendo distintos los valores óptimos para cada porción del rango.

La fórmula del proporcional está dada por:

$$P_{\mathrm{sal}} = K_p\, e(t)$$

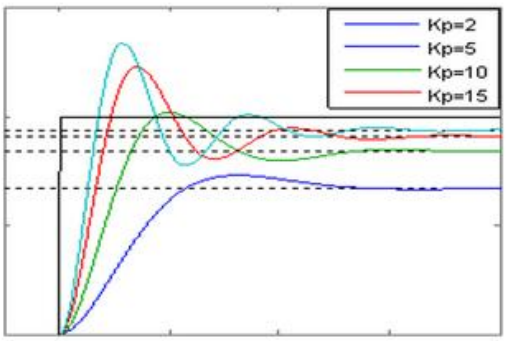


Fig. 6: Respuesta de un controlador P ante diferentes constantes de ganancia proporcional

Fuente: http://www.lra.unileon.es/es/book/export/html/268

Integral

El modo de control Integral tiene como propósito disminuir y eliminar el error en estado estacionario, provocado por el modo proporcional. El control integral actúa cuando hay una desviación entre la variable y el punto de consigna, integrando esta desviación en el tiempo y sumándola a la acción proporcional. El *error* es integrado, lo cual tiene la función de promediarlo o sumarlo por un período determinado

La fórmula del integral está dada por:

$$I_{\text{sal}} = K_i \int_0^t e(\tau)\, d\tau$$

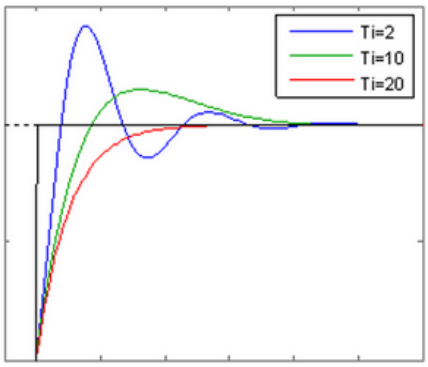


Fig. 7: Respuesta de un controlador I ante diferentes constantes de ganancia integral

Fuente: http://www.lra.unileon.es/es/book/export/html/268

Derivativa

La acción derivativa se manifiesta cuando hay un cambio en el valor absoluto del error; (si el error es constante, solamente actúan los modos proporcional e integral).

El error es la desviación existente entre el punto de medida y el valor consigna, o "*Set Point*".

La función de la acción derivativa es mantener el error al mínimo corrigiéndolo proporcionalmente con la misma velocidad que se produce. De esta manera se evita que el error se incremente.

La fórmula del derivativo está dada por:

$$D_{\text{sal}} = K_d \frac{de}{dt}$$

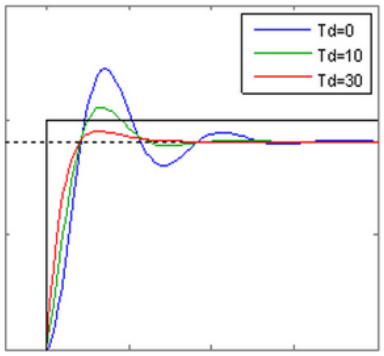


Fig. 8: Respuesta de un controlador D ante diferentes constantes de ganancia derivativa

Fuente: http://www.lra.unileon.es/es/book/export/html/268

En controles industriales es muy común encontrar los siguientes 5 tipos de reguladores que se forman combinando los términos expuestos anteriormente [11]:

• Dos posiciones (ON-OFF).

Es la forma más simple de control por realimentación. Es un control que únicamente permite dos estados de funcionamiento, On y Off. Cuando la variable controlada se desvía del valor deseado el control solo ocupa una de las dos posiciones posibles. Es un tipo de control que sirve para aplicaciones sencillas, ya que la acción de control no es proporcional a el error detectado.

Ejemplo: encendido y apagado de luces, apertura y cierre de válvulas

• Proporcional (P).

La salida obtenida es proporcional al error del sistema. Es el sistema más sencillo después del control On-Off y consiste en amplificar la señal del error antes de aplicarla.

Teóricamente, si la señal de error es cero, también lo será la salida del controlador y la respuesta, en teoría es instantánea. En la práctica, no ocurre esto. Si la variación de la señal de entrada es muy rápida, el controlador no puede seguir dicha variación y presentará una trayectoria exponencial hasta alcanzar la salida deseada. En general los controladores proporcionales presentan un error permanente incapaz de corregirse.

Ejemplo: la cisterna del aseo, donde el caudal que entra a la cisterna cuando está vacía es el máximo, y conforme el nivel del agua va subiendo, la válvula se va cerrado proporcionalmente accionada por el flotador.

• Proporcional-Integral (PI).

En estos reguladores el valor de la acción de control es proporcional a la integral de la señal de error, por lo que en este tipo de control la acción varía en función de la desviación de la salida y del tiempo en el que se mantiene esta desviación. En realidad no existen controladores que actúen únicamente con acción integral, siempre actúan en

combinación con reguladores de una acción proporcional, complementándose los dos tipos de reguladores, primero entra en acción el regulador proporcional (instantáneamente) mientras que el integral actúa durante un intervalo de tiempo.

Ejemplo: sistema de enfriado de un camión refrigerado, donde se irá regulando la temperatura para conseguir la adecuada, disminuyendo el error que provocaría una acción únicamente proporcional.

• Proporcional-Derivativo (PD).

El controlador derivativo se opone a desviaciones de la señal de entrada, con una respuesta que es proporcional a la rapidez con que se producen éstas.

La ventaja de este tipo de controlador es que aumenta la velocidad de respuesta del sistema de control. Al actuar el controlador derivativo con el proporcional, se consigue una mejora de la velocidad de respuesta. Por el contrario se pierde precisión en la salida mientras actúa la parte derivativa del controlador.

Ejemplo: control de temperatura. Debido a la inercia del sistema es importante saber hacia dónde se está evolucionando. La acción de calentamiento tiene que pararse a tiempo. Una conducción lenta de calor puede significar que, incluso después de desconectar el sistema de calentamiento, la temperatura continúe aumentando durante mucho tiempo. Durante este período la temperatura puede sobrepasar considerablemente su punto de consigna si no se ejerce una acción de control cuidadosa.

• Proporcional Integral Derivativo (PID).

Es un sistema de regulación que trata de aprovechar las ventajas de cada uno de los controladores de acciones básicas, de manera que si la señal de error varía lentamente en el tiempo, predomina la acción proporcional e integral y, mientras que si la señal de error varía rápidamente, predomina la acción derivativa. Tiene la ventaja de ofrecer una respuesta muy rápida y una compensación de la señal de error inmediata en el caso de perturbaciones. Presenta el inconveniente de que este sistema es muy propenso a oscilar y los ajustes de los parámetros son mucho más difíciles de realizar.

Ejemplo: Control de crucero del automóvil.

4. METODOLOGIA

En el siguiente apartado se desarrolla la metodología empleada para la construcción de los dos tipos de robot segway mencionados con anterioridad. Se trata de obtener el mismo fin (mantener en equilibrio el robot) pero mediante la utilización de distintos componentes. Las principales diferencias son el tipo de sensor utilizado y el software de programación empleados en el diseño de los robots.

El primero de ellos está construido con piezas Lego Mindstorm NXT 2.0. Este robot utilizará un sensor de proximidad infrarrojo y como software para realizar el control se empleara Labview.

El segundo robot está basado en sensor giroscópico y acelerómetro incluidos en la IMU 6050. Como plataforma para implementar el control se utilizara la placa Arduino UNO. En definitiva la IMU 6050 y el Arduino UNO sustituyen al Lego Mindstorms NXT 2.0 y su sensor de infrarrojos.

4.1 SEGWAY BASADO EN MINDSTORM NXT 2.0

El NXT es un equipo de robótica que parte de la idea de construir un robot mediante la unión de piezas y de la programación de acciones, de forma sencilla, a través del ensamblado de bloques con instrucciones concretas (LEGO Grupo, 2012). [12]

Fig. *9: Accesorios que conforman el kit LEGO Mindstorms NXT*

https://www.google.es/search?q=lego+mindstorms+nxt+2.0+components

NXT es el segundo robot de la línea LEGO Mindstorms. Sus principales características técnicas son:

- Mini-computadora o *brick* con una capacidad de almacenamiento en memoria RAM de 64 Kb e incluye cuatro puertos de entrada utilizados para conectar los diferentes sensores y tres puertos de salida usados para interconectar los motores.

- Seis motores que permiten movimientos precisos y ser controlados mediante el sensor de rotación, así como una sincronización en el tiempo de ejecución de una acción (rotación) con otros motores.

- Sensores que proporcionan información del mundo o contexto donde se ubica el robot, esto es, información de los objetos externos que rodean al robot NXT. Los sensores básicos en un robot NXT son: luz, sonido, tacto, color, ultrasónico y de rotación (integrado en los motores).

- Software que permite detectar objetos, determinar su ruta de desplazamiento e identificar su ubicación o localización.

4.1.1 BLOQUE NXT

En la tabla que se muestra a continuación se muestran los tres tipos de conjuntos que Lego ofrece para la construcción de robots y otras aplicaciones. Cada uno de ellos ofrece distintas características, siendo RCX el que ofrece menos prestaciones y EV3 el que más.

Entre medio de las dos gamas se encuentra el NXT, un brick utilizado para el desarrollo del robot basado en el sensor de luminosidad infrarrojo.

Las principales diferencias entre cada uno de los modelos de la gama son el modo de comunicación y el sistema operativo. En cuanto a comunicación se puede observar como cada brick de gama superior utiliza un tipo de comunicación basado en tecnologías más actuales. En el sistema operativo, la distinción la da el brick EV3, ya que ofrece un sistema operativo abierto basado en Linux, ya que ninguno de sus antecesores dispone de ello. En cuanto a procesadores EV3 y NXT utilizan unos más completos comparados con RCX.

NXT ofrece la conexión tipo USB que será la utilizada para comunicarse con el software Labview. En dicho brick posteriormente se conectara el sensor y los dos motores que serán accionados dependiendo de la señal dada por el sistema de control diseñado en Labview.

A) RCX **B) NXT** **C) EV3**

	RCX	NXT	EV3
Procesador	Hitachi H8/3292 10 – 16 MHz, 16 KB–ROM, 32 KB-RAM	Atmel 32-Bit - ARM7 48 MHz, 256KB flash, 64 KB RAM.	ARM 9 300 MHz, 16 MB – flash, 64 MB RAM
Sistema operativo	Propietario	Propietario	Abierto (basado en Linux)
Puertos	3 puertos para motores 3 puertos para sensores	3 puertos para motores 4 puertos para sensores	4 puertos para motores 4 puertos para sensores
Comunicación	Puerto IR en el frente del *brick* utilizado para comunicación con el equipo de cómputo y con otro RCX	USB 12 Mbps Bluetooth	Bluetooth v2.1 Wi-Fi mediante el puerto USB
Almacenamiento extra	N/A	N/A	Ranura Micro SD
Comunicación con dispositivos móviles inteligentes	N/A	Android	Android / iOS
Pantalla	LCD	LCD monocromática, 100 x 64 pixeles	LCD monocromática, 178 x 128 pixeles
Otras características.	-	Co-procesador Atmel 8-Bit AVR 8MHz, 4KB flash, 512 Byte RAM.	Auto-detección de dispositivos conectados al robot EV3. Control remoto (IR). Hilos de procesos. Compatible con los motores y sensores del robot NXT 2.0

Fig. 10: Tabla resumen de las características de los bricks Lego

Fuente: http://www.educagratis.org/moodle/course/view.php?id=1058

4.1.2 SENSORES

El set Lego Midstorm Nxt trae contiene entre sus sensores cuatro tipos diferentes que se describen a continuación:

- Sensor de luz: mide el nivel de brillo o luminosidad, y tiene la capacidad de distinguir entre los colores negro blanco y gris. El sensor tiene en su parte frontal un led emisor de luz que permite medir tanto el reflejo de la luz como la luz ambiente. Este mismo sensor se puede configurar como sensor de color, el cual permite diferenciar entre seis colores (negro, azul, verde, amarillo, rojo y blanco). Este sensor genera un valor numérico que se corresponde con los colores mencionados.

- Sensor de sonido: mide el nivel de sonido que hay alrededor. Este sensor únicamente es capaz de medir la intensidad del sonido que está recibiendo, siendo incapaz de diferenciar tonos distintos. La intensidad se mide en un rango de 0 a 100 siendo 0 el más suave y 100 el más alto. Puede llegar a medir hasta 90 dB.
- Sensor táctil: detecta cuando está presionado el botón delantero del sensor. Determina el estado del sensor basándose en una señal lógica de verdadero o falso. Este sensor puede trabajar con tres estados que son: pulsado (presionar y liberar), liberado o presionado.

- Sensor de ultrasonidos: mide la distancia entre el sensor y un objeto. EL sensor emite ondas sonoras de alta frecuencia y mide el tiempo que tarda en retornar cada onda. A partir del tiempo y conociendo la velocidad de propagación del sonido se determina la distancia a la que se encuentra el objeto. Este sensor tiene algunos inconvenientes, ya que los objetos a detectar pueden tener forma o texturas que dificulten al sensor realizar su medición. La superficies planas y solidas son las más fáciles de detectar. El sensor de ultrasonidos llega a detectar entre un rango de 0 a 254 cm, con un margen de error de +/- 3 cm.

4.1.3 ACTUADORES

El actuador que se empleara para la construcción del robot es el que proporciona Lego. Es un servo motor que incorpora un sensor de rotación que mide velocidad exacta y distancia e informa de ello al ladrillo inteligente Lego Mindstorms NXT. Esto permite el control del motor con 1 grado de precisión. Es posible poner varios motores funcionando en paralelo a la misma velocidad. Cada motor incluye un engranaje reductor y un sensor de rotación. Esto permite controlar los movimientos del robot con exactitud gracias a él podremos sincronizar varios motores para que funcionen a la misma velocidad o a diferentes velocidades al controlar por software el nivel de potencia de cada uno de ellos.

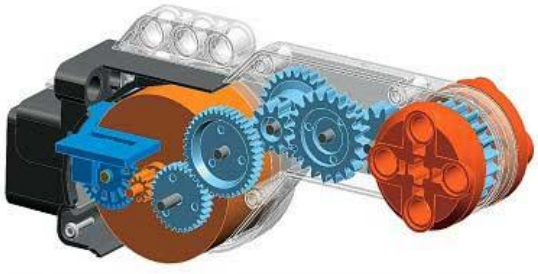


Fig. 11: Servomotor Lego

Fuente: http://www.taringa.net/posts/hazlo-tu-mismo/13004991/LEGO-Mindstorm-NXT-el-servomotor.html

4.1.4 CONFIGURACION

El kit Lego Midstorms Nxt ofrece una serie de piezas que permite la construcción de múltiples variantes que tengan un mismo fin. En el caso del robot segway se pueden encontrar varias configuraciones. Las mostradas a continuación tienen en común que están construidas en el eje vertical, aunque pueden encontrarse otras configuraciones en los que los elementos se disponen de forma más horizontal sobre el eje de las ruedas.

Aquí se pueden ver diferentes opciones.

Fig. 12: Configuraciones mecánicas para segway basado en Lego

https://www.google.es/search?q=robot+lego+segway

Todas las configuraciones que se pueden ver en la imagen superior guardan una misma estructura general. Se trata de una estructura construida sobre el eje de giro de las ruedas motoras, intentando concentrar toda la masa del robot sobre dicho plano. Las figuras 1 y 4 muestran el robot con sensor de luminosidad y las figuras 2 y 3, robots basados en el giroscopio que Lego también puede suministrar.

Se ha escogido la primera de las configuraciones, por tratarse de una estructura que posibilita el control del segway con un tercer motor adicional colocado en el "conductor", que puede resultar interesante estudiar al finalizar el segway básico.

4.1.5 LABVIEW COMO CONTROLADOR

En este apartado del proyecto se ha utilizado Labview como sistema controlador. Es una herramienta que ofrece múltiples variante a la hora de realizar el control. En este caso se ha optado por utilizar el módulo de LabView para LEGO MINDSTORMS que permite controlar y programar los sistemas basados en Lego. El módulo de Lego tiene el nombre de:

"LabView Module for LEGO® MINDSTORMS®"

Este software incluye las siguientes características diseñadas específicamente para LEGO MINDSTORMS [13]:

- **Editor de Control Remoto:** Permite configurar y controlar visualmente el NXT usando un control de palanca o un teclado
- **Terminal NXT:** Permite administrar la memoria y los programas del NXT desde su pantalla.
- **Visualización Remota:** Para ver todas las pantallas y botones del NXT en el monitor de su PC.
- **Editor de Esquemáticos:** Configurar gráficamente y probar conexiones de motor y sensores.
- **Visualizador de Sensores:** Ver los datos desde sus sensores desde su proyecto en tiempo real.
- **Visualizador de Datos:** Registrar y analizar fácilmente los datos que adquiere desde su NXT.
- **Editor de Imágenes:** Cree sus propias imágenes para visualizar en la pantalla del NXT.

4.2 SEGWAY BASADO EN ARDUINO

El segundo de los robots segway es el basado en Arduino con la utilización de un giroscopio y acelerómetro. A continuación se describen los principales elementos que componen el robot, como son el software Arduino, los sensores y el equipo de potencia empleado.

4.2.1 ARDUINO

Arduino es una placa con un microcontrolador Atmel e incluye toda la circuitería de soporte necesaria (reguladores de tensión, resistencias de pull-up, etc.) Además incluye un puerto USB que permite programar el microcontrolador desde cualquier PC de manera cómoda y sencilla, además de poder monitorizar datos provenientes de la placa [14].

La placa escogida para la realización del trabajo fin de grado se trata de Arduino UNO, una de las placas incluidas en la gama que oferta Arduino. Se trata de una de las placas con unas características muy buena para el precio de la misma. Dispone de un microcontrolador lo suficientemente potente para el desarrollo del trabajo y además cuenta con unos pines que se adaptan a la perfección al uso que se les dará. En concreto se trata de los pines que permiten la comunicación I2C además de los pines digitales que permitirán controlar los motores.

La mencionada placa dispone 14 pines configurables como entrada o salida y que permiten conectarse a cualquier dispositivo compatible con señales digitales de 0-5 V.

También dispone de entradas analógicas. Mediante las mismas se pueden obtener datos en forma continua a partir del voltaje.

Aparte de estas entradas y salidas, Arduino UNO ofrece en su placa una serie de pines con funciones especiales:

- RX y TX: Se usan para transmisiones serie de señales TTL.

- Interrupciones externas: Los pines 2 y 3 están configurados para generar una interrupción en el Atmega.

- PWM: Arduino dispone de 6 salidas destinadas a la generación de señales PWM de hasta 8 bits.

- SPI: Los pines 10, 11, 12 y 13 pueden utilizarse para llevar a cabo comunicaciones SPI, que permiten trasladar información full dúplex en un entorno Maestro/Esclavo.

- I2C: Permite establecer comunicaciones a través de un bus I2C. El bus I2C es un producto de Phillips para interconexión de sistemas embebidos. Actualmente se puede encontrar una gran diversidad de dispositivos que utilizan esta interfaz, desde pantallas LCD, memorias EEPROM, sensores...

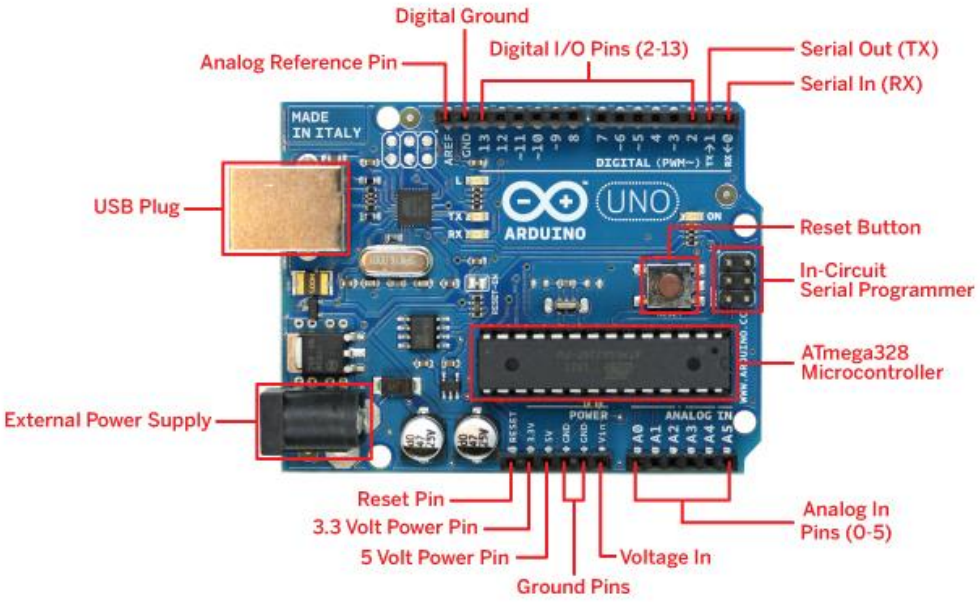


Fig. 13: Esquema de pines y conexiones de la placa Arduino UNO

Fuente: http://www.taringa.net/posts/hazlo-tu-mismo/17868737/Termistor-como-sensor-de-temperatura.html4.2.2 SENSORES

Microcontrolador	Atmega328
Voltaje de operación	5V
Voltaje de entrada (Recomendado)	7 – 12V
Voltaje de entrada (Límite)	6 – 20V
Pines para entrada- salida digital.	14 (6 pueden usarse como salida de PWM)
Pines de entrada analógica.	6
Corriente continua por pin IO	40 mA
Corriente continua en el pin 3.3V	50 mA
Memoria Flash	32 KB (0,5 KB ocupados por el bootloader)
SRAM	2 KB
EEPROM	1 KB
Frecuencia de reloj	16 MHz

Fig. 14: Tabla de características generales de Arduino UNO

Fuente: http://www.arduino.cc/en/Main/ArduinoBoardUno

4.2.2 MOTORES LEGO

Los motores empleados en este robot serán los mismos utilizados en el anterior (segway basado en infrarrojos), tal y como se recoge en el apartado 4.1.3.

4.2.3 EQUIPO POTENCIA

En este caso se necesitara de un equipo de potencia que suministre la energía necesaria a los motores, ya que la placa Arduino no es capaz de suministrarla directamente. Para ello se puede escoger entre varias opciones entre las que se encuentran:

- Transistores
- Relés
- Puentes en H
- Interfaces electrónicas de control

En este caso se escogió trabajar con el puente en H. Se trata de una configuración sencilla y que permite manejar los servomotores de manera precisa y sencilla. En concreto se escogió el integrado L293D que contiene cuatro circuitos que pueden configurarse de diferentes modos para obtener cuatro medios puentes H o bien dos completos. En este caso se utilizara como puente en H completo, que permitirá el control bidireccional de los dos motores así como frenado rápido en caso de ser necesario [15].

El integrado tiene 16 pines que se utilizan de la siguiente manera.

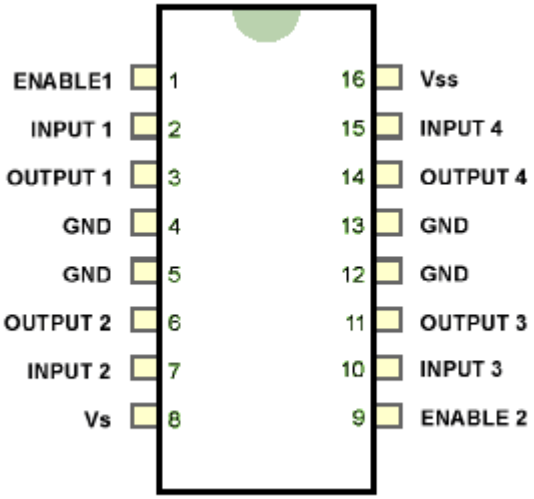


Fig. 15: Diagrama de pines del integrado L293D

Fuente: http://roboexperiments.com/l293d-h-bridge-dc-motor/

PIN 1 y 9: Habilitan la sección izquierda y derecha del puente respectivamente

PIN 2, 7, 15, 10: Permiten determinar el sentido de giro de los motores según la siguiente tabla siempre que los pines 1 y 9 estén activados.

PINES 2 o 15	PINES 7 o 10	FUNCION
H	L	GIRO EN SENTIDO HORARIO
L	H	GIRO EN SENTIDO ANTIHORARIO
H	H	DETENCION RAPIDA
L	L	DETENCION RAPIDA

Fig. 16: Tabla de funcionamiento del integrado L293D

PIN 3 y 6: Pines para conexión del motor 1

PIN 14 y 11: pines para conexión del motor 2

PIN 4, 5, 13 y 12: GND

PIN 8: alimentación de los motores (4.5 – 36 V)

PIN 16: Voltaje lógico (5V)

4.2.4 SENSOR IMU MPU 6050

Los sensores IMU (inertial measurement unit) son un tipo de sensor que se pueden encontrar en una gran cantidad de aparatos electrónicos de la actualidad. Por ejemplo en teléfonos, videojuegos… Estos sensores nos ayudan a conocer la orientación, velocidad y fuerzas gravitacionales que le afectan. Por esa gran versatilidad se le encuentran gran cantidad de aplicaciones en el mercado [16].

Existe una gran variedad de sensores IMU, pero uno de los más fiables y precisos es el MPU 6050. Además de tener unas conexiones de fácil conexión para Arduino es un sensor económico y de grandes prestaciones.

Los sensores IMU generalmente constan de dos o más partes. Pueden llegar a tener hasta cuatro sensores tales como acelerómetro, giroscopio, magnetómetro y altímetro. En concreto el MPU 6050 es un sensor de 6 grados de libertad o seis ejes, que nos proporciona seis variables de salida. Contiene un giroscopio y un acelerómetro incrustados dentro de un mismo chip. Tanto el acelerómetro como el giroscopio utilizan un protocolo de comunicación I2C que será explicado en el siguiente apartado.

Una vez explicado el contenido del IMU 6050 se describirá a continuación el funcionamiento del acelerómetro y del giroscopio.

Acelerómetro

Un acelerómetro funciona basándose en el principio del efecto piezoeléctrico. El funcionamiento del acelerómetro puede asemejarse a una caja cubica con una bola en su interior. Cuando se inclina el sensor, la bola contenida en el cubo se mueve en la dirección de la inclinación debido a la gravedad. La bola al chocar con las paredes de la caja crea unas pequeñas corrientes eléctricas. Estas corrientes creadas en cada una de las paredes de la caja se corresponden con cada uno de los ejes del espacio X, Y, Z. Dependiendo de la cantidad de corriente eléctrica producida en cada una de las paredes, se podrá determinar la inclinación del sensor y su magnitud.

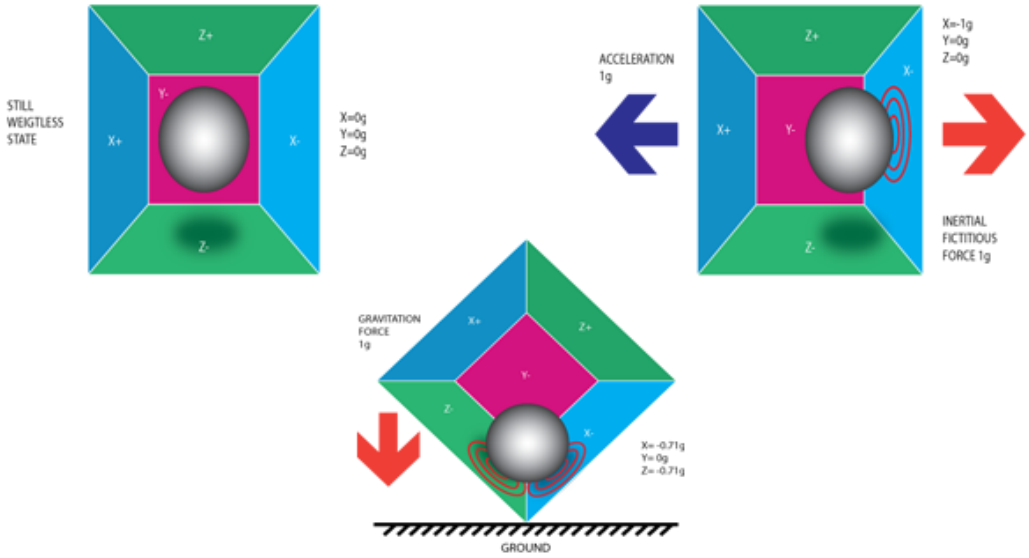


Fig. 17: Funcionamiento de acelerómetro

Fuente: http://diyhacking.com/arduino-mpu-6050-imu-sensor-tutorial/

Giroscopio

El giroscopio que incorpora el sensor IMU 6050 en un giroscopio denominado MEMS (Microelectromechanicalsystems).Son sensores pequeños y baratos que miden la velocidad angular. Las unidades de velocidad angular se miden en grados por segundo (°/s) o revoluciones por segundo (RPS). La velocidad angular es simplemente la medición de la velocidad de rotación.

Un giroscopio MEMS de tres ejes, similar a la foto que aparece posteriormente, puede medir la rotación alrededor de tres ejes X, Y, Z. Existen giroscopios de un solo eje o de dos, pero el de tres ejes es el más extendido en la actualidad.

El giroscopio MEMS son giroscopios de estructura vibrante que funcionan basándose en el principio de Coriolis. Contienen una masa micro-mecanizada conectada a una carcasa exterior mediante resortes, y a su vez la carcasa está conectada a la circuitería.

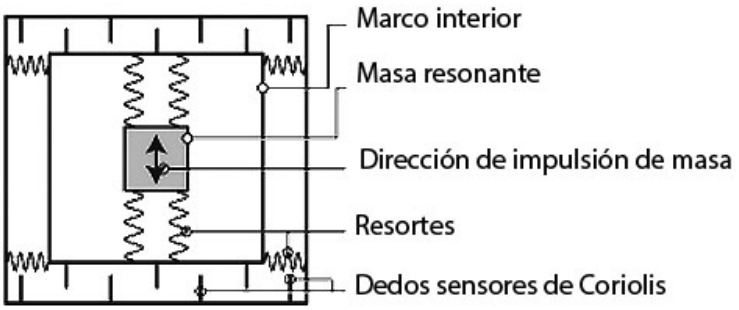


Fig. 18: Esquema de la estructura interior del giroscopio

Fuente: http://www.tecnoficio.com/electricidad/electricidad_del_automotor19.php

La fuerza de coriolis es detectada por los dedos sensores y será enviada la señal para ser procesada y obtener los valores tanto en magnitud como en dirección.

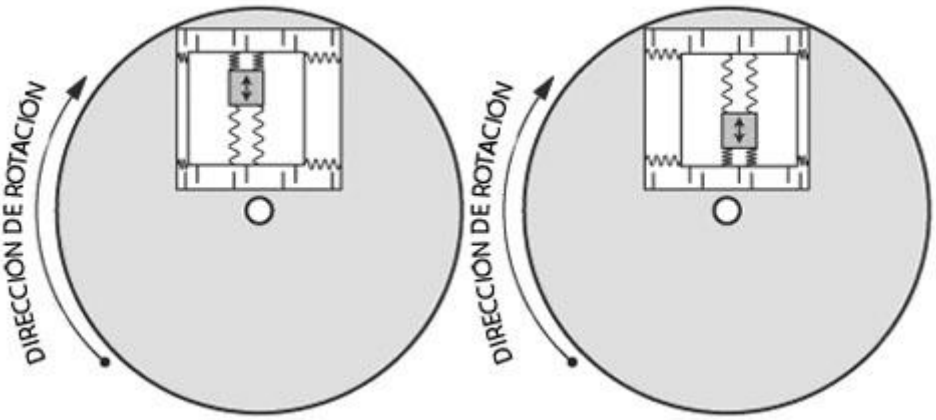


Fig. 19: Funcionamiento de giroscopio MEMS

Fuente: http://www.tecnoficio.com/electricidad/electricidad_del_automotor19.php

5. RESULTADOS

5.1 SEGWAY BASADO EN SENSOR DE LUMINOSISDAD

A continuación se desarrollaran los diferentes apartados que han servido para la construcción del robot segway basado en Lego.

5.1.1 CONFIGURACION MECÁNICA

La configuración escogida para la construcción de este primer robot ha sido la mostrada en la siguiente figura,

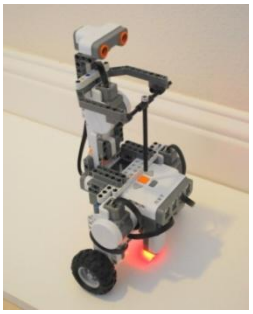


Fig. 20: Configuración inicial del robot segway

Para la construcción del mismo se han seguido los pasos que se encuentran en la referencia bibliográfica [16]

Se trata de una configuración que no presenta una distribución de la masa simétrica, lo que complicaría el control ya que el momento de inercia hacia los dos lados del eje de rotación de las ruedas no sería el mismo. Para intentar igualarlo y obtener una mejor distribución y por consiguiente un mejor control del sistema se ha realizado una modificación en la construcción del robot. Esta modificación se encuentra en la parte delantera del robot, en la cual se han añadido una serie de piezas lego con el único fin de conseguir equilibrar la distribución del peso del robot a lo largo del eje mostrado en la siguiente figura.

La configuración final se puede ver en la siguiente imagen:

Fig. 21: Configuración final del robot segway

5.1.2 SENSOR DE LUZ NXT

Para la obtención de datos del sistema, se ha optado en este primer caso por la utilización del sensor de luz del LEGO NXT. Este sensor permite utilizarse para distintos fines como son distinguir colores, seguir líneas o calcular una distancia mediante infrarrojos. Esta última aplicación es la que se utilizará para el cálculo de las distancia del sensor hasta el suelo.

Figura 22: Sensor de luz Lego

Fuente: http://www.lejos.org/forum/viewtopic.php?t=3636

El sensor infrarrojo utilizado se encuentra dentro del grupo de tipo difuso o auto-reflex, en donde el transmisor y receptor se encuentran encapsulados en el mismo sitio alineados. De este modo cuando un objeto se interpone en el camino del haz que emite el transmisor, la luz se ve reflejada y el receptor es capaz de recibirla. Por eso es de gran importancia que el objeto o la superficie en la que se refleja la luz sean suficientemente reflexivos y estén dentro de la distancia de actuación del sensor.

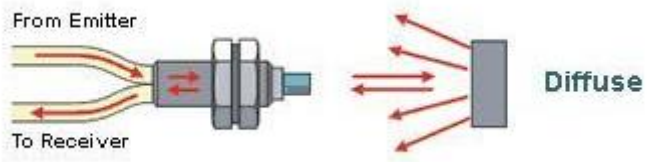


Fig. 23: Sensor infrarrojo difuso

Fuente: http://mci-automation.blogspot.com.es/2011/07/tipos-de-fotoceldas-deteccion-de.html

Cuando un objeto se interpone en el camino del haz de luz emitido, esta rebota hacia el receptor. La forma de medir la distancia a la que se encuentra el objeto es mediante la medición del tiempo que tarda el haz en realizar el recorrido desde que sale del emisor hasta que es recibido por el receptor. A partir de ese tiempo y conociendo la velocidad de propagación del haz de luz infrarroja se puede determinar la distancia a la que se encuentra el objeto.

5.1.3 ALGORITMO DE CONTROL

Se ha implementado un control PID en el entorno de programación de Labview. Este programa permite la interactuación entre el PC y el bloque NXT con la simple instalación del paquete para Lego Mindstorm NXT.

El programa está dividido en 3 partes diferenciadas:

a. Adquisición de la posición de equilibrio: el sensor infrarrojo realiza 4 medidas de la posición de equilibrio y realiza la media aritmética para obtener la posición vertical. Además emite un sonido cada vez que realiza una medición. De este modo se calcula el Set-Point del controlador.

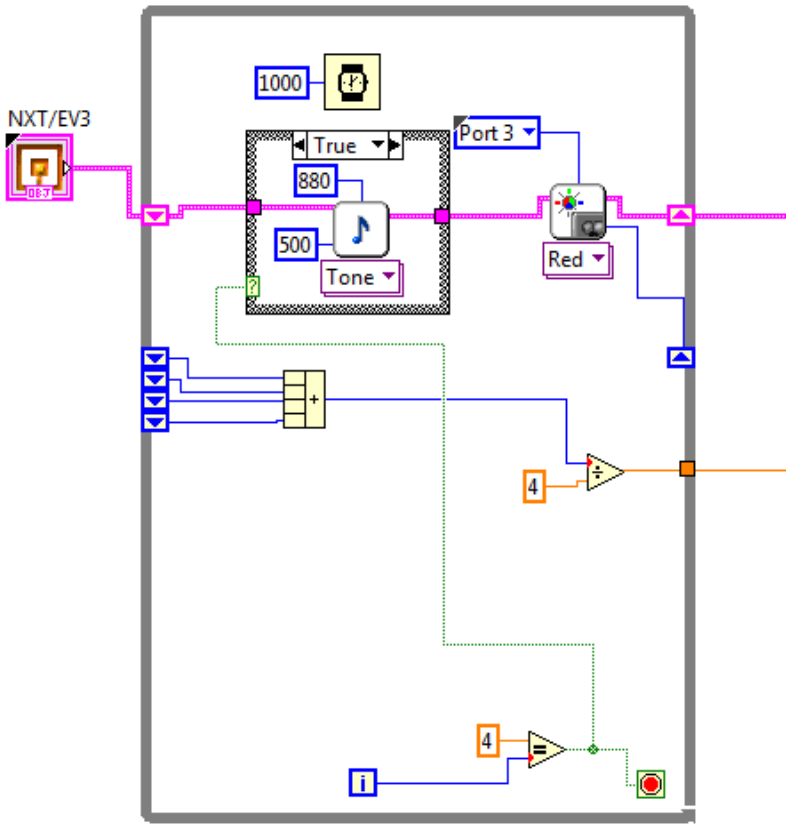


Fig. 24: Calculo del set-point en Labview

b. Calculo de la acción de control mediante PID: Se calcula la acción de control mediante la comparación de la distancia inicial y la que se va midiendo continuamente mediante el citado sensor infrarrojo

Parte Proporcional

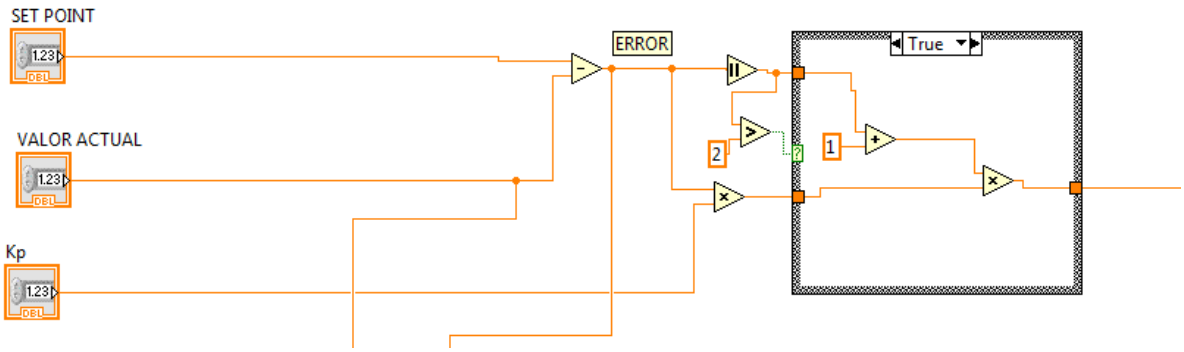


Fig. 25: Control proporcional en Labview

En la figura anterior se puede observar la configuración clásica de un control proporcional que responde a la formula siguiente:

$$Accion\ proporcional = error * Kp$$

Siendo el error,

$$error = setpoint - valor\ actual$$

Además de esto, el programa incorpora un bucle para conseguir una mayor rapidez de respuesta cuando el error se hace muy grande. Esto se consigue incrementando el valor de la acción proporcional cuando el robot se encuentra en dicha situación. El límite establecido se sitúa cuando el error es mayor a 2.

Parte Integral

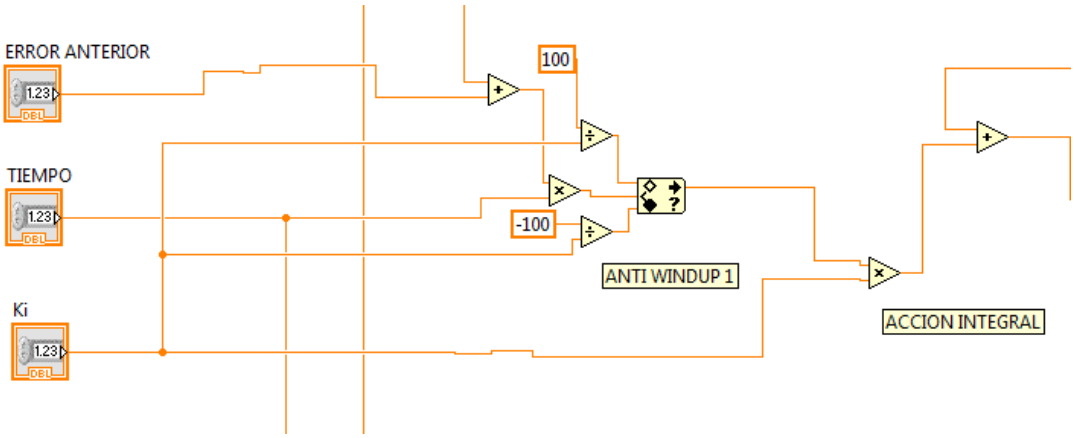


Fig. 26: Control integral en Labview

En la figura anterior se puede observar la parte integral del controlador PID. En el esquema se sigue la formula siguiente correspondiente a la acción integral,

$$Accion\ integral = (error\ anterior + error) * tiempo * Ki$$

En los controladores integrales se puede dar un efecto llamado windup. Esto ocurre cuando la acción de los motores se satura (el motor tiene un límite de velocidad), ya que aunque esta se encuentre saturada, el controlador integral sigue integrando el error haciéndose este cada vez mayor. Como consecuencia de que el error se prolongue en el tiempo, puede darse la situación de que aunque este se reduzca la acción derivativa nos lleve a dar una respuesta mayor de la necesaria [17].

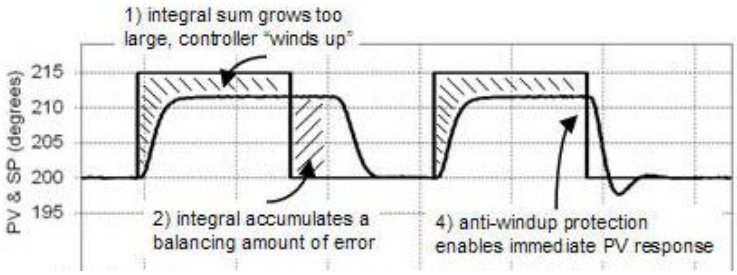


Fig. 27: Explicación grafica del efecto windup

Fuente: http://www.controlguru.com/2008/021008.html

Para evitar este efecto se pueden seguir tres métodos:

- Limitación del término integral
- Integración condicional
- Seguimiento integral

En este caso se ha optado por colocar un limitador que sature la acción integral entre los valores 100/Ki y -100/Ki. El valor 100 corresponde al valor máximo que puede alcanzar el motor NXT, referido a la potencia que se le aplica.

Parte derivativa

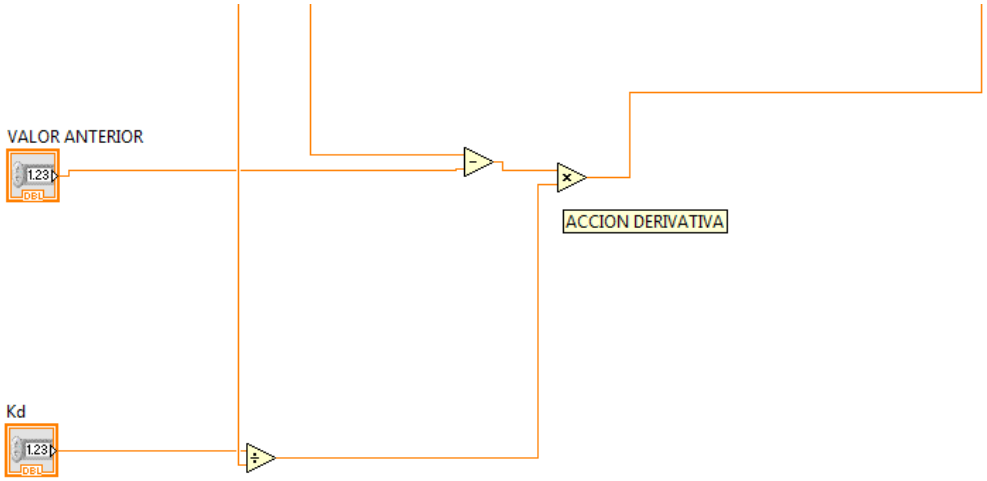


Fig. 28: Control derivativo en Labview

La acción derivativa se anticipa a lo que va a suceder para conseguir estabilizar el robot. Sigue la siguiente fórmula:

$$Accion\ derivativa = (valor\ actual - valor\ anterior) * (Kd/tiempo)$$

Obtención de la acción de control a partir de las acciones PID

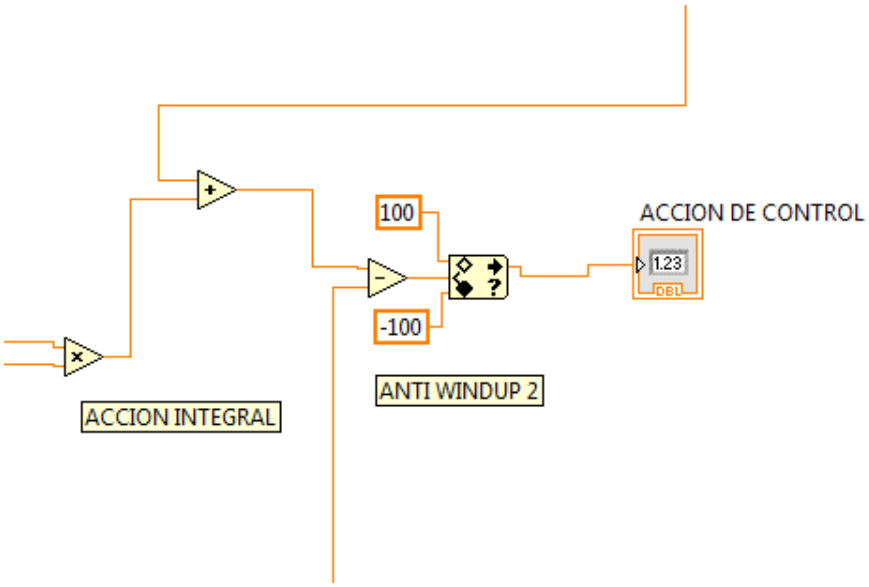


Fig. 29: Combinación de las acciones del PID para obtener la respuesta

Para conseguir la acción final del PID se combinan las tres acciones previamente descritas que forman la acción de control siguiente:

$$Accion\ de\ control = Accion\ proporcional + Accion\ integral - Accion\ derivativa$$

Además de esto, al igual que se ha introducido un limitador en la parte integral, también se ha introducido un limitador en la parte final para saturar la acción al valor máximo que pueden dar los motores. (Véase Fig. 22)

c. Aplicación de la acción de control a los servomotores

Una vez que la acción de control ha sido generada, se procede a aplicarla a los actuadores. Dependiendo del signo del error o si es 0 se siguen distintos caminos.

Error > 0 (el robot está inclinado hacia adelante)

El bloque que acciona los motores los hace girar hacia adelante para compensar la caída hacia ese lado.

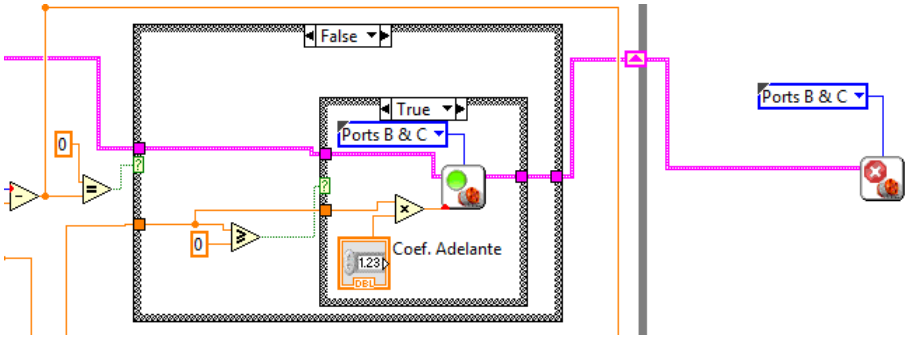


Fig. 30: Control de avance de los motores en Labview

Error < 0 (el robot está inclinado hacia atrás)

El bloque que acciona los motores los hace girar hacia atrás para compensar la caída hacia ese lado.

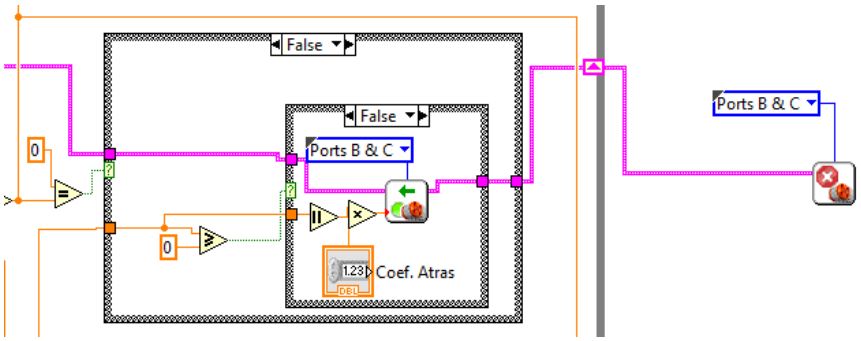


Fig. 31: Control de retroceso de los motores en Labview

Error =0

Los motores permanecen parados.

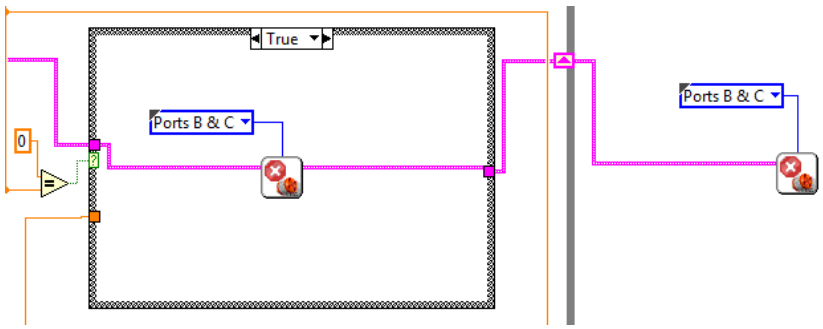


Fig. 32: Detención de los motores en Labview

Además de actuar sobre los motores, el programa también realiza acciones complementarias para poder visualizar en tiempo real tanto el error que se está produciendo, como la acción de control y también una gráfica en la que se puede seguir la referencia y el valor actual en el que se encuentra el robot.

Por otro lado también se han introducido dos coeficientes para poder controlar las acciones de avance y retroceso de los motores en caso de que hacia uno de los lados caiga más rápido que hacia el otro.

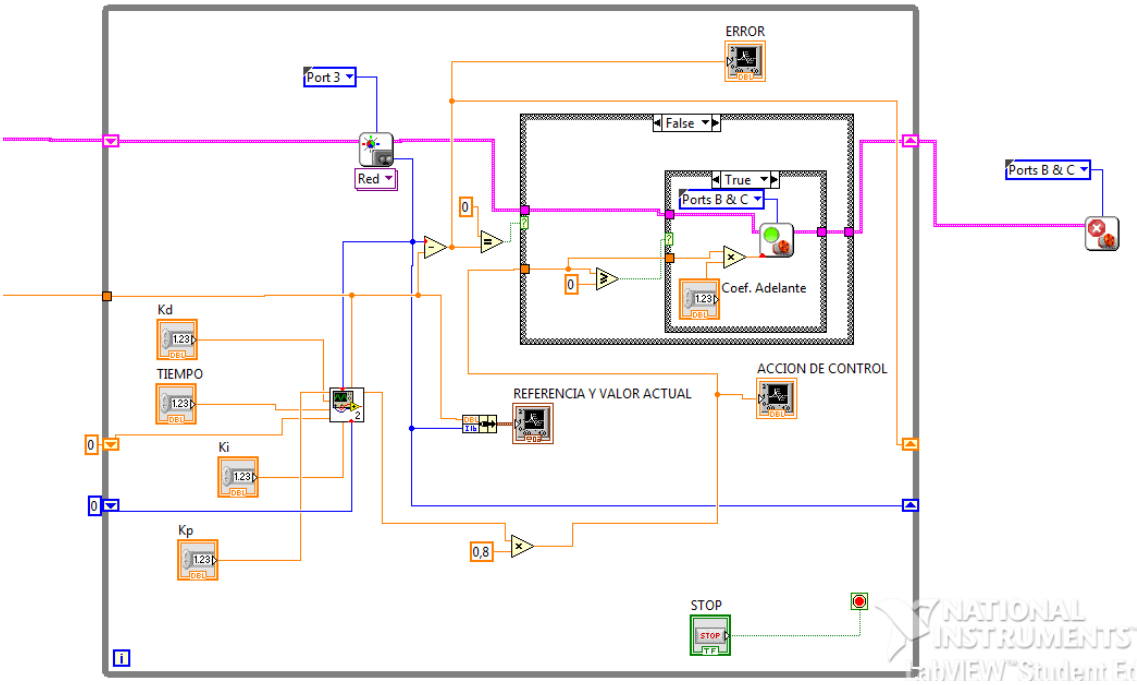


Fig. 33: Esquema para monitorización de datos en Labview

5.1.4 ENSAYOS

Determinación de los parámetros del PID

Para la determinación de los parámetros del PID diseñado anteriormente se utilizaron dos métodos.

El primero de ellos se realizó utilizando el bloque pre-establecido en Labview con nombre “Auto-tuning PID” y se obtuvieron los siguientes resultados.

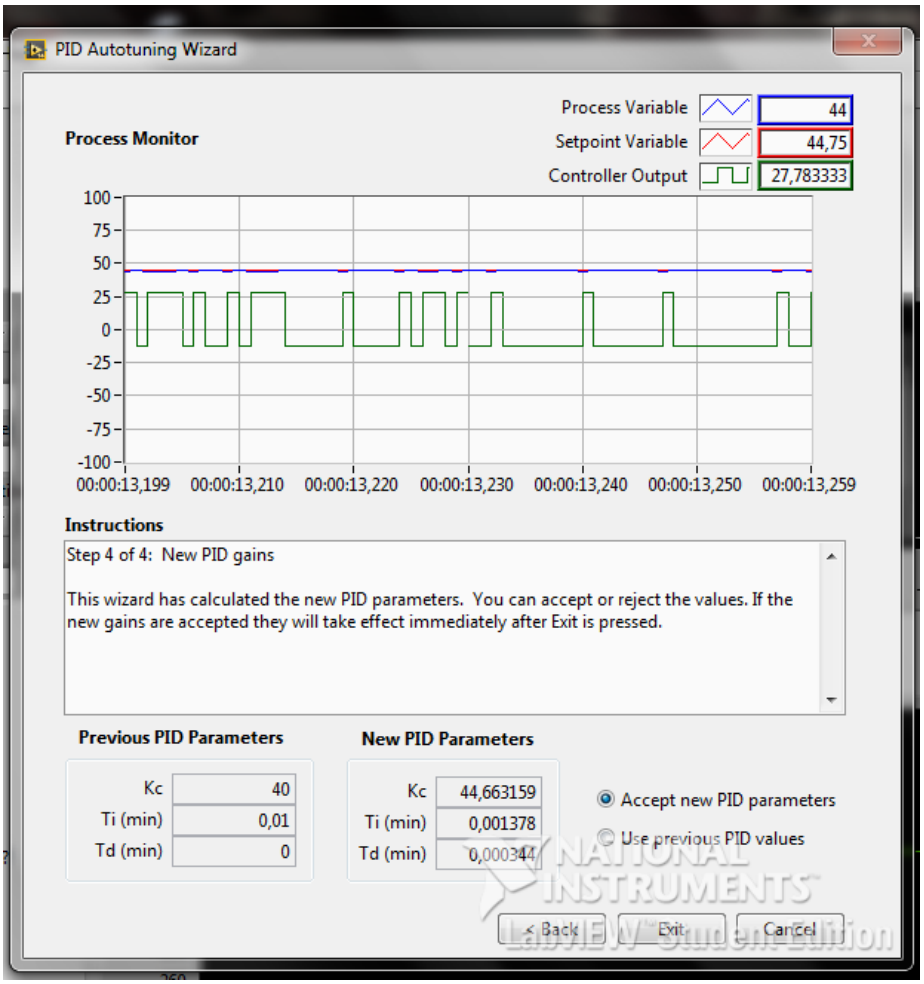


Fig. 34: Pantalla final del sintonizador automático del PID en Labview

Una vez obtenidos los parámetros del bloque, se ajustan los parámetros de forma manual llegando a los siguientes parámetros para el PID:

- Kp = 40

- Ki = 80

- Kd= 5

5.1.5 PROBLEMAS Y SOLUCIONES

Colocación del sensor infrarrojo. El principal problema que se ha tenido con el robot Lego ha sido el relacionado con el sensor infrarrojo. Se trata de un sensor que varía mucho su precisión dependiendo de la distancia a la que se encuentre la superficie que reflectara el haz de luz. En un principio se coloco a una distancia de 2 cm sobre el suelo, la cual era demasiado grande para poder establecer un buen control, ya que los

datos que se recibían eran muy aleatorios. La solución alcanzada fue tan sencilla como reducir la distancia a 1 cm. Se mejoro de forma muy clara el funcionamiento del segway.

Otro de los problemas es la superficie en la que se refleja el haz de luz. Se ha llegado a la conclusión de que la mejor superficie es de color blanca y de color uniforme, aunque el robot puede funcionar en otras superficies siempre que sean reflectantes del haz.

5.2 SEGWAY BASADO EN ACELERÓMETRO Y GIROSCOPIO

5.2.1 CONFIGURACION MECANICA

En el caso del robot basado en Arduino se ha optado por una disposición horizontal de los elementos que lo componen. En un principio se escogió una configuración vertical, la cual debido a su construcción complicaba la configuración del programa de control. Por ello se dispusieron todos los elementos más pesados sobre el eje de giro de las ruedas en un plano horizontal, tal como se muestra en las siguientes figuras.

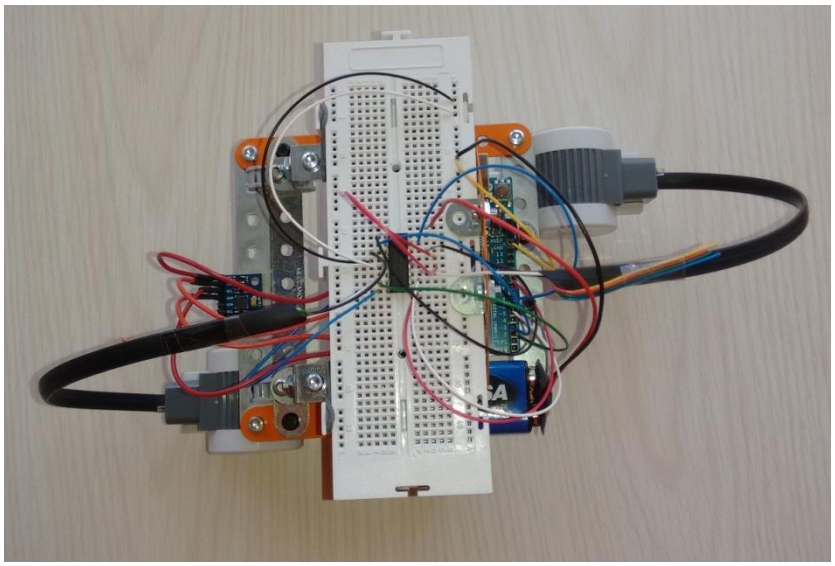


Fig. 35: Vista en planta del robot segway basado en Arduino

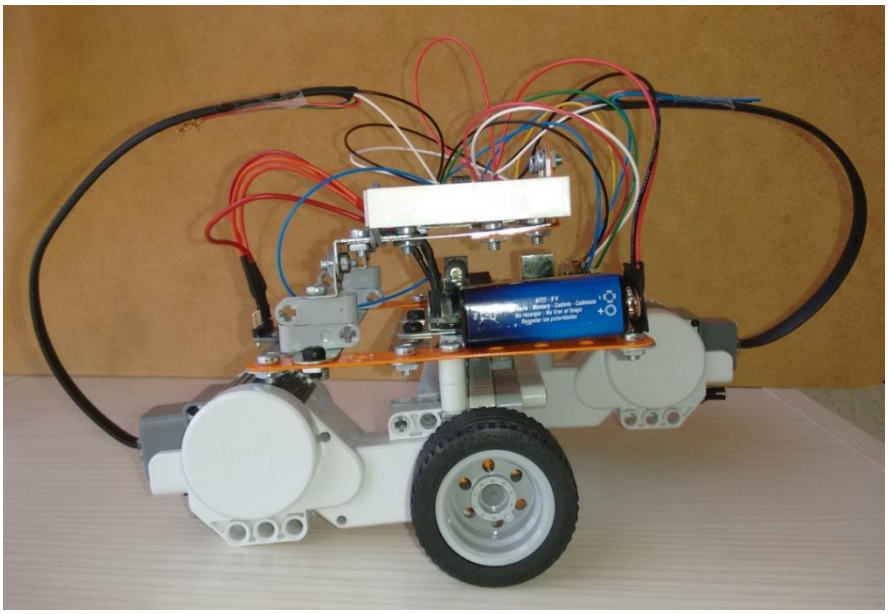


Fig. 36: Vista en alzado del robot segway basado en Arduino

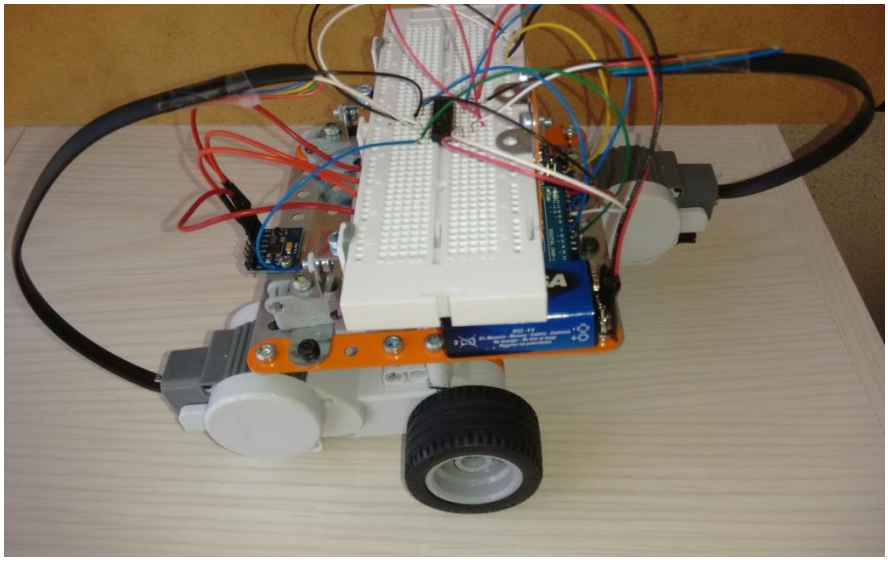


Fig. 37: Vista general del robot segway basado en Arduino

El chasis está construido mediante el ensamblaje de piezas Mecano, creando así una base para poder colocar el sensor, los motores y la circuitería electrónica necesaria para el funcionamiento del robot.

Se ha tratado de centrar todo el peso sobre el eje de giro para facilitar el control, además de disponer un motor hacia cada lado para compensar el peso por igual en ambos lados.

Como se puede observar se han creado dos alturas con el fin de conseguir centrar el peso y además se separan la parte de control (tarjeta Arduino) y la de potencia (L293D)

5.2.2 PROTOCOLO I2C

La comunicación entre el sensor MPU 6050 y Arduino se realiza mediante el protocolo I2C (Inter Integrated Circuit). Se trata de un protocolo muy utilizado para realizar comunicación en circuitos integrados, siendo uno de sus mayores usos e utilizado para comunicar un microcontrolador y un sensor periférico.

El bus I2C cuenta con dos líneas SDA (datos) y SCL (clock) además de la masa. Un aspecto a tener en cuenta en I2C es que solo el maestro o maestros pueden controlar la línea SCL, lo que implica que solo un maestro puede iniciar la transmisión. El esclavo deberá esperar a ser preguntado por el maestro para poder enviar un dato.

Los dispositivos conectados al bus I²C tienen una dirección única y pueden ser *maestros* o *esclavos*. El dispositivo *maestro* inicia la transferencia de datos y además genera la señal de reloj, pero no es necesario que el *maestro* sea siempre el mismo dispositivo, esta característica se la pueden ir pasando los dispositivos que tengan esa capacidad. Esta característica hace que al bus I²C se le denomine bus multimaestro.

Para usar el bus I2C en Arduino hay que incluir la librería Wire con #include<Wire.h>. A continuación en el setup hay que usar la función Wire.begin () esto inicia el bus I2C y nos define como maestros. Dentro del begin se puede poner una dirección aunque sí solo hay un máster no es necesario ya que nadie nos va a solicitar datos sino que seremos nosotros los que siempre iniciemos la comunicación [18].

Una vez iniciado el bus podemos empezar a transmitir. Para ello se usan 3 instrucciones:

1. Wire.beginTransmission(dirección): Inicia el bus y ponemos con qué dirección vamos a comunicarnos.
2. Wire.write (bytes): Esta función envía uno o varios bytes a la dirección anterior.
3. Wire.endTransmission(): Finaliza la comunicación con un STOP y deja el bus libre.

5.2.3 MOTORES LEGO

Los motores LEGO MINDTORM NXT son motores eléctricos denominados servomotores. Esto implica que además de controlar la velocidad de giro del motor también se puede controlar su posición. En este trabajo fin de grado se ha utilizado únicamente el control de velocidad, ya que no se ha requerido de la posición de los motores para realizar el control.

Los citados servomotores incluyen un cable de conexión RJ12 al ladrillo NXT, que en este caso deberá ser modificado para poder utilizarse con Arduino [19].

Pin number of original nxt-cable	cable color	function	signal / polarity	signal form
1	white	motor power supply	DC 9V + or -	DC
2	black	motor power supply	DC 9V - or +	DC
3	red	rotation detector supply	ground (0V)	DC
4	green	rotation detector supply	+ 4.3 (to 5.0 V)	DC
5	yellow	rotation detector output 1	0 / 4.3 V against 3, -4.3V / 0V against 4	rectangle
6	blue	rotation detector output 2	0 / 4.3 V against 3, -4.3V / 0V against 4	rectangle

Fig. 38: Tabla de asignación de funciones de cada uno de los cables del motor Lego

Fuente: http://trivox.tripod.com/lego-nxt-motor-input-output.html

Siguiendo la tabla del código de colores que utiliza Lego podremos se determina el uso de cada cable que llega hasta el motor. Los cables que determinan el sentido de giro y su velocidad son los correspondientes a los colores blanco y negro, que posteriormente se conectaran al puente en H para suministrar la potencia deseada a los motores.

5.2.4. CONEXIONADO

En la siguiente figura se muestra todo el conexionado empleado para la implementación del robot segway basado en Arduino.

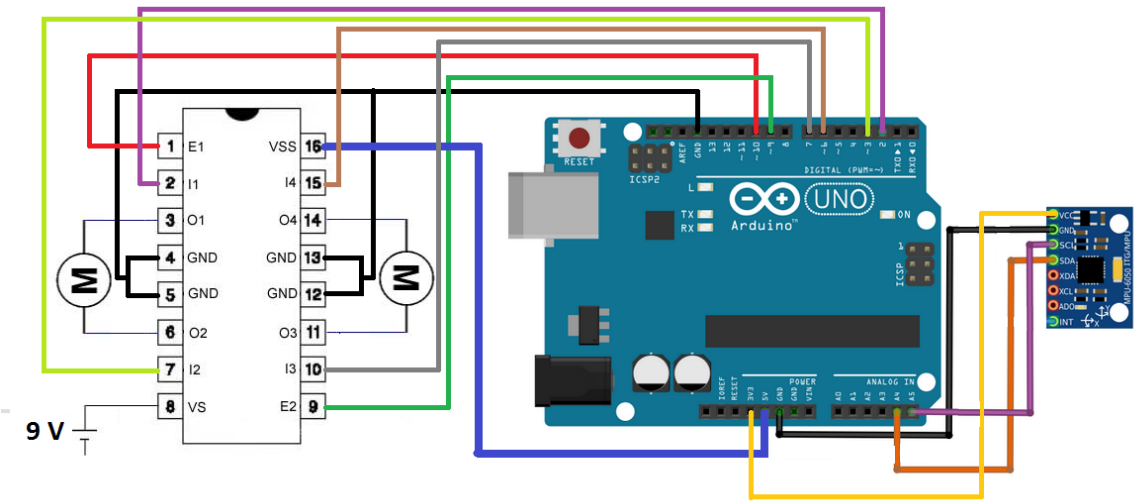


Fig. 39: Conexionado de los componentes electrónicos que conforman el segway

La parte más destacable de las conexiones se sitúa en los pines 9 y 10 de la placa Arduino (cables rojo y verde). Dichos pines son aptos para emitir una señal PWM, la cual habilitará o deshabilitará los motores consiguiendo así un control de la potencia que se les aplica.

Por otro lado también esta los pines 2-7 y 15-10 del integrado L293D. Mediante la configuración de esos pines se consigue manejar los motores para que giren en un sentido o en otro.

5.2.5. ALGORITMO DE CONTROL

El algoritmo utilizado para el control del robot está basado en la misma estructura que el anterior. Se realiza la medición del ángulo de inclinación y a partir de ese dato se implementa un controlador PID que dará la señal de control para realizar la compensación.

Para obtener el Angulo de inclinación se procesa la señal obtenida del anteriormente mencionado sensor MPU 6050. Dicho sensor da unos valores brutos que se refinan para obtener el valor del Angulo de inclinación en el que se encuentra el robot.

Para el cálculo del Angulo a partir de los datos del acelerómetro se utiliza la expresión:

$$Angulo\ Y\ acel = \operatorname{atan}\frac{x}{\sqrt{y^2 + z^2}}$$

Para el cálculo del Angulo a partir de los datos del giroscopio se utiliza la expresión:

$$Angulo\ Y\ giro = \frac{y}{G\_R}$$

Donde G_R es una constante de conversión que corresponde a 131.

Una vez convertidos los valores del sensor a valores en grados de inclinación se emplea un filtro. El uso del filtro es muy conveniente ya que se conseguirá eliminar el ruido que interfiere principalmente en el acelerómetro y a su vez se reducirá el error que se produce con el giroscopio. Se ha empleado un filtro complementario, que surge de la unión entre un filtro paso-alto para el giroscopio y un filtro paso-bajo para el acelerómetro. El resultado del filtro es el siguiente:

$$Angulo = 0.98 * (Angulo + Angulo\ giro * 0.01) + 0.02 * Angulo\ acel$$

A partir de este punto ya se puede implementar el control PID. En el caso de Arduino también se ha implementado una restricción anti-windup en el integrador a la salida de la variable de control “u” tal y como se realizó en el robot basado en sensor de luminosidad.

5.2.6 SINTONIZACION DEL PID

Para la sintonización del PID en un principio se ha recurrido al método de Ziegler-Nichols.

El procedimiento es el siguiente:

1. Aplicar a la planta sólo control proporcional con ganancia Kp pequeña.

2. Aumentar el valor de Kp hasta que el lazo comience a oscilar. La oscilación debe ser lineal y debe detectarse en la salida del controlador (u (t)).

3. Registrar la ganancia crítica Kp = Ku y el período de oscilación Tu de u (t), a la salida del controlador.

4. Ajustar los parámetros del controlador PID de acuerdo al apartado "PID" del siguiente cuadro.

Método Ziegler-Nichols			
TIPO DE CONTROL	**Kp**	**Ki**	**Kd**
P	0,5*Ku	******	******
PI	0,45*Ku	1,2*(Ku/Tu)	******
PD	0,8*Ku	******	Kp*Tu/8
PID	0,6*Ku	2*Kp/Tu	Kp*Tu/8

Fig. 40: Ajuste de parámetros de PID

Los resultados obtenidos fueron los siguientes:

K_u= 50

P_c=0.66

K_p= 33

K_i= 151.51

K_d= 7.26

Los valores obtenidos no daban como resultado una estabilización del robot por lo que posteriormente se inició un barrido de las constantes partiendo de las anteriormente conseguidas.

Finalmente se obtuvieron los siguientes valores:

K_p= 30

K_i= 10.

K_d= 40

Se realizó una reducción drástica de la constante integradora para conseguir eliminar el efecto wind-up. También se incremento en consideración la variable correspondiente a la parte derivativa del PID, para lograr una gran estabilidad en el robot.

5.2 DISCUSION DE LOS RESULTADOS OBTENIDOS

La implementación de los dos segways ha supuesto un trabajo en diferentes áreas de la ingeniería, pero principalmente en la electrónica, donde se ha podido comprobar la complejidad y el estudio que requiere un pequeño proyecto para poder sacarlo adelante. Dicho esto cabe destacar la satisfacción que supone alcanzar el objetivo propuesto a principio del trabajo fin de grado, tanto personal como a nivel de ingeniería.

En cuanto a los resultados obtenidos a partir de la realización de los dos diferentes robots podemos citar varias conclusiones.

Robot basado en Lego Mindstorm NXT

- Se puede comprobar la versatilidad que ofrece lego a la hora de realizar robots automatizados a pequeña escala.

- Se observa las deficiencias que acarrea el utilizar unos servomotores que no son de uso profesional, como puede ser la holgura mecánica que presenta el acoplamiento de las ruedas con el eje de giro del motor.

- También se ha comprobado el amplio uso que se le puede dar a la herramienta Labview, que proporciona soporte para poder programar en multitud de soportes conseguir una comunicación con distintos protocolos.

Robot basado en Arduino

- El diseño completo del robot ha supuesto un reto mayor al que el del anterior robot, ya que tenía en consideración más secciones que en Labview o Lego ya venían implementadas.

En cuanto a la comparación en el funcionamiento de los dos robots, cabe destacar la gran estabilidad que alcanza el robot diseñado con Arduino. Esto se puede deber a que se trata de una programación más sofisticada en cuanto a que la señal recibida por el sensor MPU 6050 es más precisa que la que ofrece el sensor de luminosidad. Este último se ve muy afectado por la iluminación de su alrededor y también de forma considerable de la superficie en la que está proyectando el haz de luz. Además de esto se puede decir que el segundo robot se trata de una construcción más ingenieril que la primera y como consecuencia de elegir los componentes más apropiados para este robot, se obtienen unos resultados mejores que los anteriores. Además de estas razones técnicas, existe también la razón económica, ya que los materiales y componentes utilizados para su elaboración son mucho más baratos.

Otro de los puntos que cabe destacar es la precisión que tiene cada uno de los robots en cuanto al valor de consigna. Mientras en el primero de los robots (Lego) se trata de un ajuste manual cada vez que se pone en funcionamiento y depende de la superficie en la que se encuentre, el robot Arduino tiene ya programada la señal de setpoint, que es válida para cualquier tipo de superficie.

6. PRESUPUESTO

6.1 SEGWAY BASADO EN MINDSTORM NXT 2.0

SEGWAY BASADO EN LEGO MINDSTORM NXT 2.0			
CONCEPTO	**CANTIDAD**	**PRECIO**	**TOTAL**
KIT LEGO MINDSTORM NXT 2.0	1	380 €	380 €
MANO DE OBRA	100	10 €	1.000 €
BASE IMPONIBLE			**1.380 €**
IVA 21 %			**290 €**
TOTAL			**1.670 €**

6.2 SEGWAY BASADO EN ARDUINO

SEGWAY BASADO EN ARDUINO			
CONCEPTO	**CANTIDAD**	**PRECIO**	**TOTAL**
MOTORES LEGO	2	25 €	50 €
PLACA ARDUINO UNO	1	25 €	25 €
PILA 9 V	1	1,5 €	1,5 €
INTEGRADO L293D	1	2,7 €	2,7 €
IMU 6050	1	10 €	10 €
ESTRUCTURA MECANO	1	5 €	5 €
MANO DE OBRA	80	10 €	800 €
TOTAL			**894,2**
IVA 21 %			**188 €**
TOTAL			**1.082 €**

7. CONCLUSIONES Y LÍNEAS FUTURAS

Como conclusión general se puede decir que el robot basado en Lego ha servido como introducción a la hora de diseñar un robot segway, y ha contribuido a detectar los errores y dificultades que se pueden dar en su construcción y programación. Como consecuencia ha sido más fácil elaborar un segundo robot segway por completo, escogiendo los componentes necesarios, y de ese modo se ha logrado un robot con mejores prestaciones que el construido anteriormente.

Después del trabajo realizado se podría trabajar en las siguientes tareas:

- Control de posición del "conductor" en el robot basado en Lego para dirigir el segway hacia adelante o hacia atrás.

- Añadir la posibilidad de comunicarse de forma inalámbrica mediante un mando de control

- Optimización del valor recibido por el sensor de luminosidad, descartando posibles valores erróneos que se produzcan debido al reflejo o luz ambiental. Conseguiríamos que el PID no diese valores de control erróneos.

- Ajuste de los parámetros del PID en Arduino para que sean modificados según la inclinación en la que se encuentre el robot. Se ganaría estabilidad y mejoraría la rapidez de respuesta del sistema.

- Proveer al robot Arduino de autonomía, incorporando un regulador de tensión de 9 a 5V para alimentar la placa y poder ser desconectado del ordenador.

- Incorporar al robot una batería de mayor duración y que pueda controlarse el nivel de carga de la misma.

- Elaboración de un chasis más ligero y personalizado para mejorar la distribución de la masa y por consiguiente el funcionamiento del robot.

8. BIBLIOGRAFIA

[1]https://en.wikipedia.org/wiki/Inverted_pendulum

[2]http://www.unicrom.com/cir_detector-proximidad-infrarrojo.asp

[3]http://es.wikipedia.org/wiki/Sensor_ultras%C3%B3nico

[4]http://es.wikipedia.org/wiki/Gir%C3%B3scopo

[5]http://es.wikipedia.org/wiki/Aceler%C3%B3metro

[6]https://books.google.es/books?id=n45jCAAAQBAJ&pg=PA407&lpg=PA407&dq

[7]http://www.somoslibres.org/modules.php?name=News&file=article&sid=5581

[8]https://books.google.es/books?id=n45jCAAAQBAJ&pg=PA463&lpg=PA463&dq

[9]http://educativa.catedu.es/44700165/aula/archivos/repositorio/4750/4925/html/4_tipos_de_sistemas_de_control.html

[10]http://es.wikipedia.org/wiki/Controlador_PID

[11 http://web.udl.es/usuaris/w3511782/Control_de_procesos/Unidades_files/apuntes_10-11.pdf

[12]http://bibliotecadigital.icesi.edu.co/biblioteca_digital/bitstream/10906/71683/5/revision_plataforma_robotica.pdf.txt

[13] http://www.ni.com/labview/esa/

[14] http://www.arduino.cc/

[15] http://robots-argentina.com.ar/MotorCC_L293D.htm

[16] http://www.nxtprograms.com/NXT2/segway/

[17] http://diyhacking.com/arduino-mpu-6050-imu-sensor-tutorial/

[18] http://csd.newcastle.edu.au/SpanishPages/clase_slides_download/C16.pdf

[19] http://www.quadruino.com/guia-2/sensores/protocolo-i2c-twi

[20] http://es.scribd.com/doc/24698262/Rj12-and-Lego-Nxt

9. INDICE DE FIGURAS

10. ANEXOS

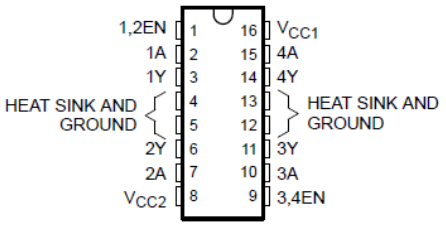
10.1 DATASHEET L293 D

L293, L293D
QUADRUPLE HALF-H DRIVERS

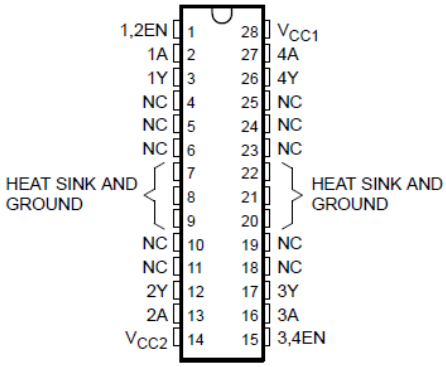
SLRS008B – SEPTEMBER 1986 – REVISED JUNE 2002

- **Featuring Unitrode L293 and L293D Products Now From Texas Instruments**
- **Wide Supply-Voltage Range: 4.5 V to 36 V**
- **Separate Input-Logic Supply**
- **Internal ESD Protection**
- **Thermal Shutdown**
- **High-Noise-Immunity Inputs**
- **Functional Replacements for SGS L293 and SGS L293D**
- **Output Current 1 A Per Channel (600 mA for L293D)**
- **Peak Output Current 2 A Per Channel (1.2 A for L293D)**
- **Output Clamp Diodes for Inductive Transient Suppression (L293D)**

N, NE PACKAGE
(TOP VIEW)



DWP PACKAGE
(TOP VIEW)



description

The L293 and L293D are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

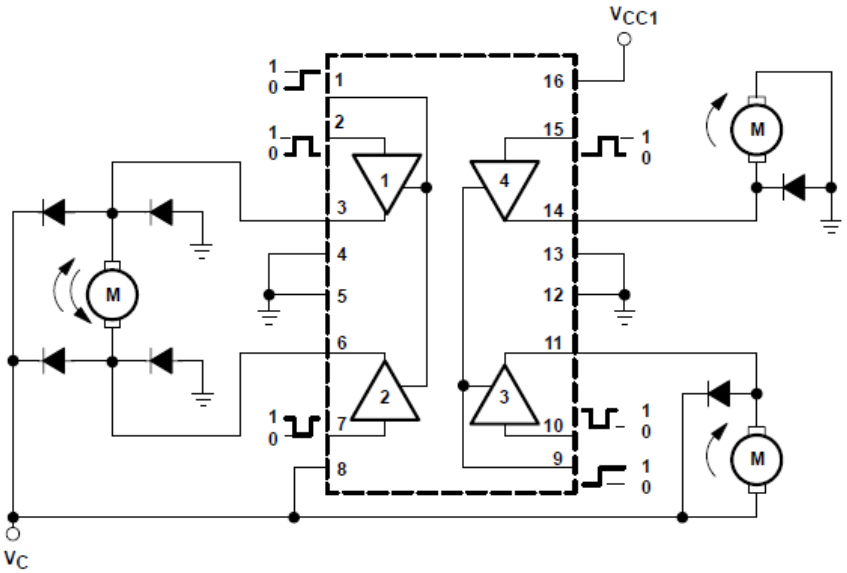
On the L293, external high-speed output clamp diodes should be used for inductive transient suppression.

A V_{CC1} terminal, separate from V_{CC2}, is provided for the logic inputs to minimize device power dissipation.

The L293and L293D are characterized for operation from 0°C to 70°C.

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

block diagram



NOTE: Output diodes are internal in L293D.

TEXAS INSTRUMENTS AVAILABLE OPTIONS

T_A	PACKAGE
	PLASTIC DIP (NE)
0°C to 70°C	L293NE L293DNE

Unitrode Products from Texas Instruments

AVAILABLE OPTIONS

T_A	PACKAGED DEVICES	
	SMALL OUTLINE (DWP)	PLASTIC DIP (N)
0°C to 70°C	L293DWP L293DDWP	L293N L293DN

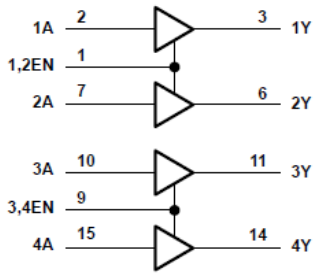
The DWP package is available taped and reeled. Add the suffix TR to device type (e.g., L293DWPTR).

FUNCTION TABLE
(each driver)

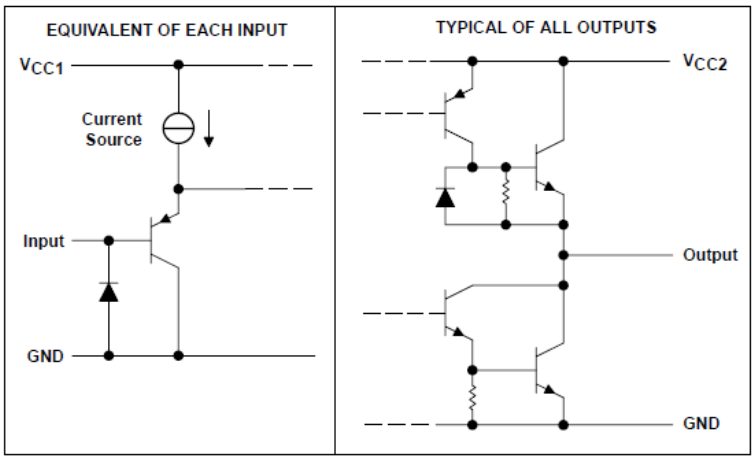
INPUTS†		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high level, L = low level, X = irrelevant,
Z = high impedance (off)
† In the thermal shutdown mode, the output is in the high-impedance state, regardless of the input levels.

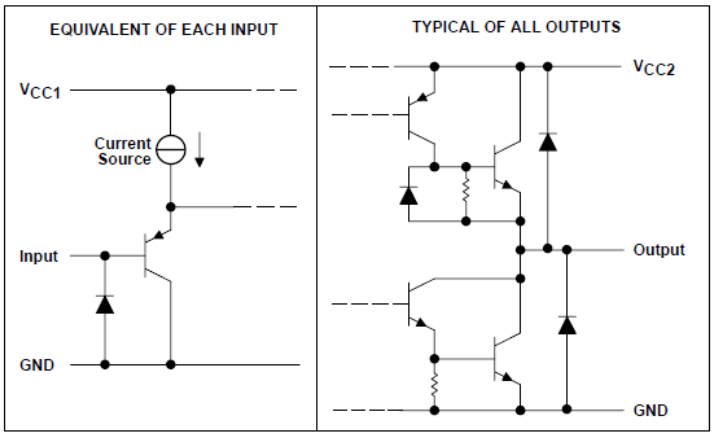
logic diagram



schematics of inputs and outputs (L293)



schematics of inputs and outputs (L293D)



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage, V_{CC1} (see Note 1)	36 V
Output supply voltage, V_{CC2}	36 V
Input voltage, V_I	7 V
Output voltage range, V_O	–3 V to V_{CC2} + 3 V
Peak output current, I_O (nonrepetitive, t ≤ 5 ms): L293	±2 A
Peak output current, I_O (nonrepetitive, t ≤ 100 µs): L293D	±1.2 A
Continuous output current, I_O: L293	±1 A
Continuous output current, I_O: L293D	±600 mA
Continuous total dissipation at (or below) 25°C free-air temperature (see Notes 2 and 3)	2075 mW
Continuous total dissipation at 80°C case temperature (see Note 3)	5000 mW
Maximum junction temperature, T_J	150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C
Storage temperature range, T_{stg}	–65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltage values are with respect to the network ground terminal.
2. For operation above 25°C free-air temperature, derate linearly at the rate of 16.6 mW/°C.
3. For operation above 25°C case temperature, derate linearly at the rate of 71.4 mW/°C. Due to variations in individual device electrical characteristics and thermal resistance, the built-in thermal overload protection may be activated at power levels slightly above or below the rated dissipation.

recommended operating conditions

			MIN	MAX	UNIT
	Supply voltage	V_{CC1}	4.5	7	V
		V_{CC2}	V_{CC1}	36	
V_{IH}	High-level input voltage	$V_{CC1} \le 7$ V	2.3	V_{CC1}	V
		$V_{CC1} \ge 7$ V	2.3	7	V
V_{IL}	Low-level output voltage		–0.3†	1.5	V
T_A	Operating free-air temperature		0	70	°C

† The algebraic convention, in which the least positive (most negative) designated minimum, is used in this data sheet for logic voltage levels.

electrical characteristics, V_{CC1} = 5 V, V_{CC2} = 24 V, T_A = 25°C

PARAMETER			TEST CONDITIONS		MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage		L293: I_{OH} = –1 A L293D: I_{OH} = –0.6 A		V_{CC2}–1.8	V_{CC2}–1.4		V
V_{OL}	Low-level output voltage		L293: I_{OL} = 1 A L293D: I_{OL} = 0.6 A			1.2	1.8	V
V_{OKH}	High-level output clamp voltage		L293D: I_{OK} = –0.6 A			V_{CC2} + 1.3		V
V_{OKL}	Low-level output clamp voltage		L293D: I_{OK} = 0.6 A			1.3		V
I_{IH}	High-level input current	A	V_I = 7 V			0.2	100	µA
		EN				0.2	10	
I_{IL}	Low-level input current	A	V_I = 0			–3	–10	µA
		EN				–2	–100	
I_{CC1}	Logic supply current		I_O = 0	All outputs at high level		13	22	mA
				All outputs at low level		35	60	
				All outputs at high impedance		8	24	
I_{CC2}	Output supply current		I_O = 0	All outputs at high level		14	24	mA
				All outputs at low level		2	6	
				All outputs at high impedance		2	4	

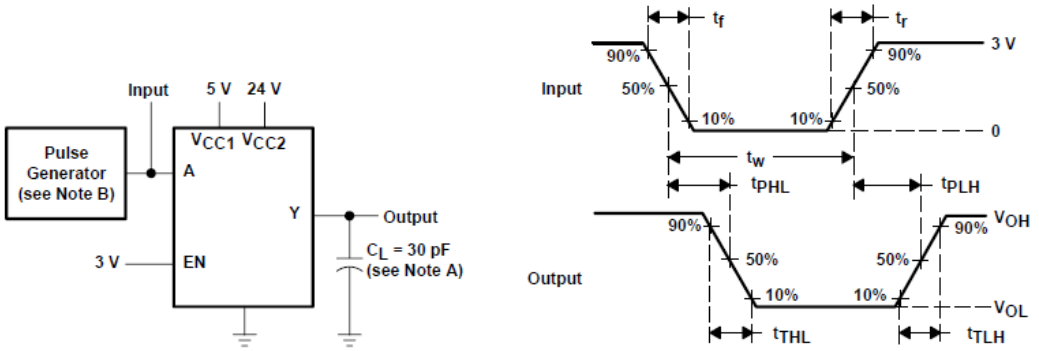
switching characteristics, V_{CC1} = 5 V, V_{CC2} = 24 V, T_A = 25°C

PARAMETER		TEST CONDITIONS	L293NE, L293DNE			UNIT
			MIN	TYP	MAX	
t_{PLH}	Propagation delay time, low-to-high-level output from A input	C_L = 30 pF, See Figure 1		800		ns
t_{PHL}	Propagation delay time, high-to-low-level output from A input			400		ns
t_{TLH}	Transition time, low-to-high-level output			300		ns
t_{THL}	Transition time, high-to-low-level output			300		ns

switching characteristics, V_{CC1} = 5 V, V_{CC2} = 24 V, T_A = 25°C

PARAMETER		TEST CONDITIONS	L293DWP, L293N L293DDWP, L293DN			UNIT
			MIN	TYP	MAX	
t_{PLH}	Propagation delay time, low-to-high-level output from A input	C_L = 30 pF, See Figure 1		750		ns
t_{PHL}	Propagation delay time, high-to-low-level output from A input			200		ns
t_{TLH}	Transition time, low-to-high-level output			100		ns
t_{THL}	Transition time, high-to-low-level output			350		ns

PARAMETER MEASUREMENT INFORMATION



TEST CIRCUIT VOLTAGE WAVEFORMS

NOTES: A. C_L includes probe and jig capacitance.
B. The pulse generator has the following characteristics: $t_r \le 10$ ns, $t_f \le 10$ ns, $t_w = 10$ µs, PRR = 5 kHz, $Z_O = 50$ Ω.

Figure 1. Test Circuit and Voltage Waveforms

APPLICATION INFORMATION

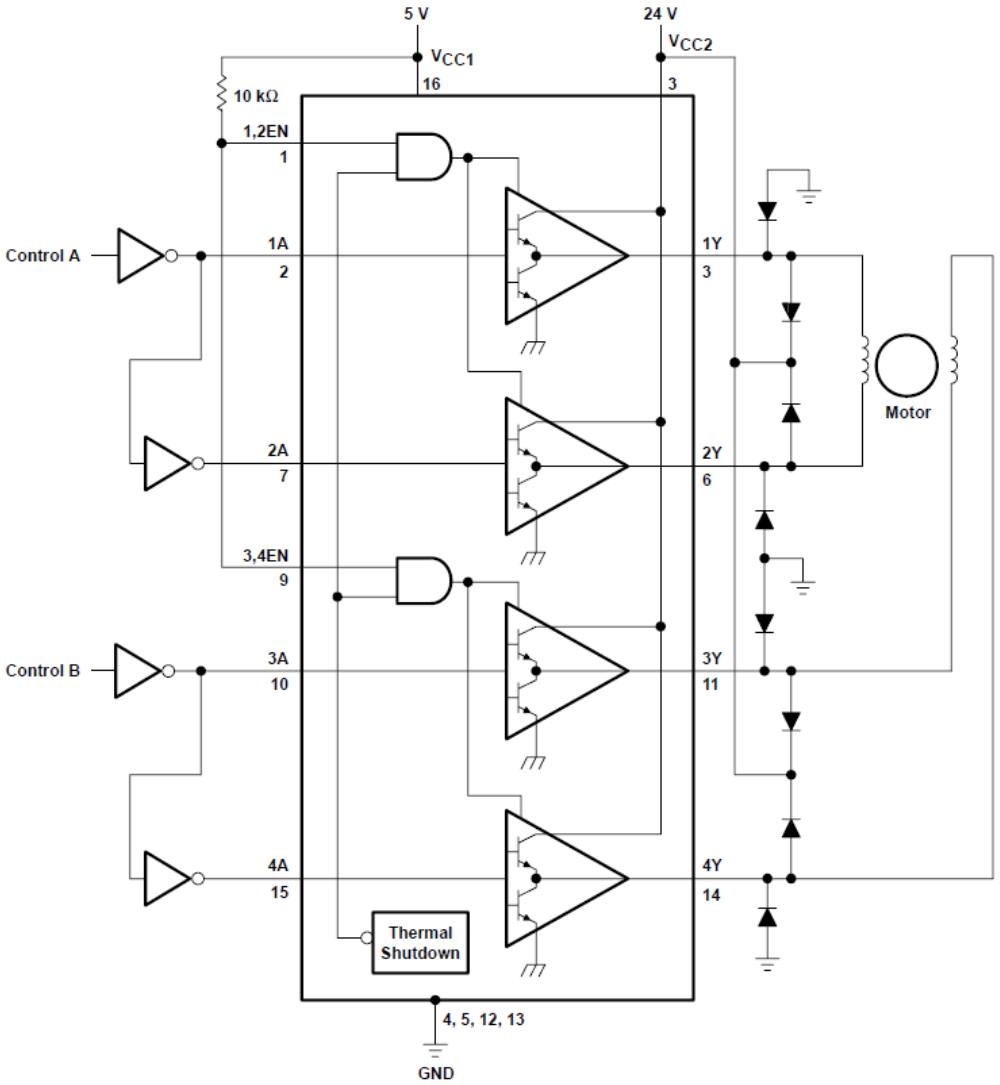


Figure 2. Two-Phase Motor Driver (L293)

APPLICATION INFORMATION

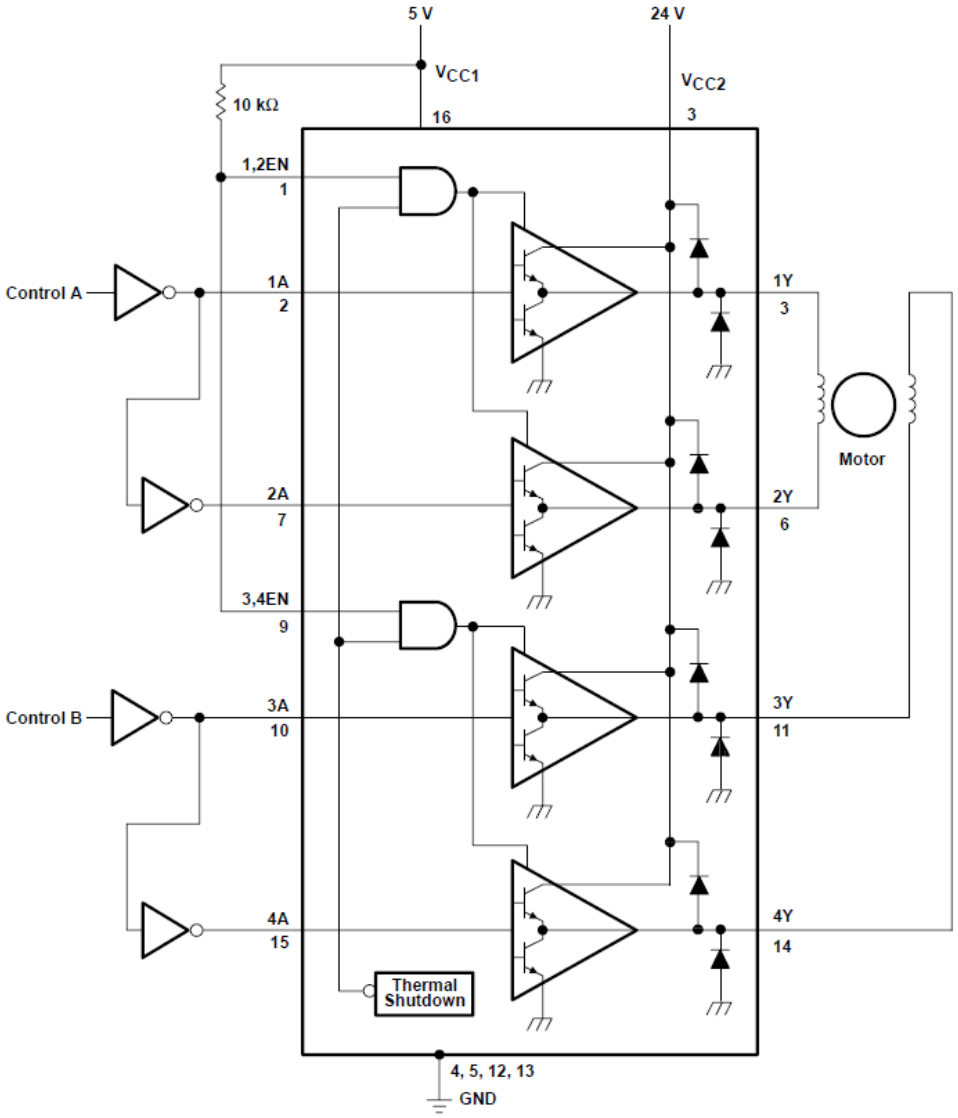


Figure 3. Two-Phase Motor Driver (L293D)

APPLICATION INFORMATION

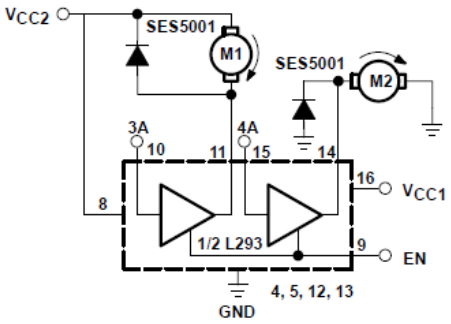


Figure 4. DC Motor Controls (connections to ground and to supply voltage)

EN	3A	M1	4A	M2
H	H	Fast motor stop	H	Run
H	L	Run	L	Fast motor stop
L	X	Free-running motor stop	X	Free-running motor stop

L = low, H = high, X = don't care

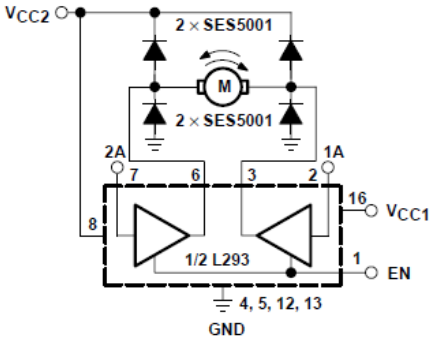


Figure 5. Bidirectional DC Motor Control

EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

L = low, H = high, X = don't care

APPLICATION INFORMATION

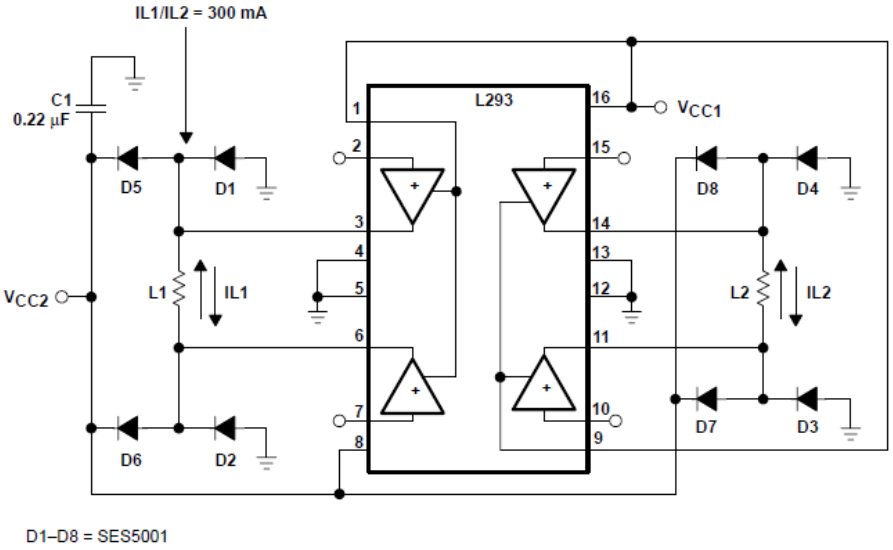


Figure 6. Bipolar Stepping-Motor Control

mounting instructions

The Rthj-amp of the L293 can be reduced by soldering the GND pins to a suitable copper area of the printed circuit board or to an external heatsink.

Figure 9 shows the maximum package power P_{TOT} and the θ_{JA} as a function of the side ℓ of two equal square copper areas having a thickness of 35 µm (see Figure 7). In addition, an external heat sink can be used (see Figure 8).

During soldering, the pin temperature must not exceed 260°C, and the soldering time must not be longer than 12 seconds.

The external heatsink or printed circuit copper area must be connected to electrical ground.

APPLICATION INFORMATION

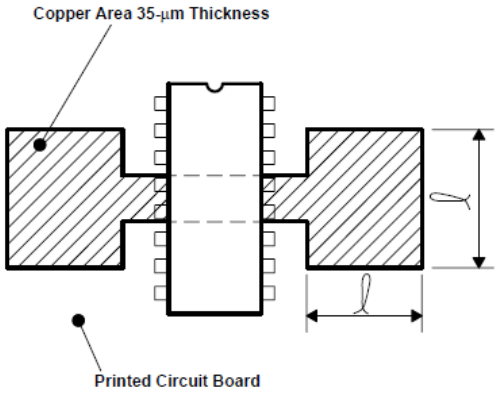


Figure 7. Example of Printed Circuit Board Copper Area (used as heat sink)

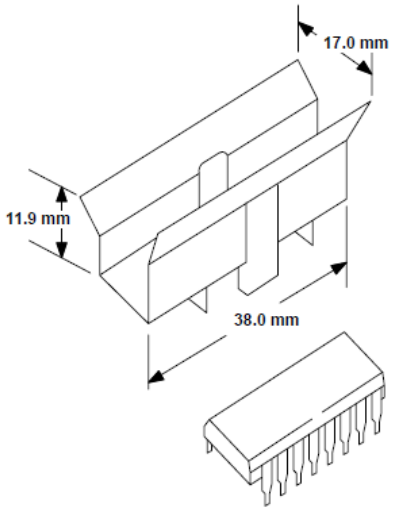


Figure 8. External Heat Sink Mounting Example (θ_{JA} = 25°C/W)

APPLICATION INFORMATION

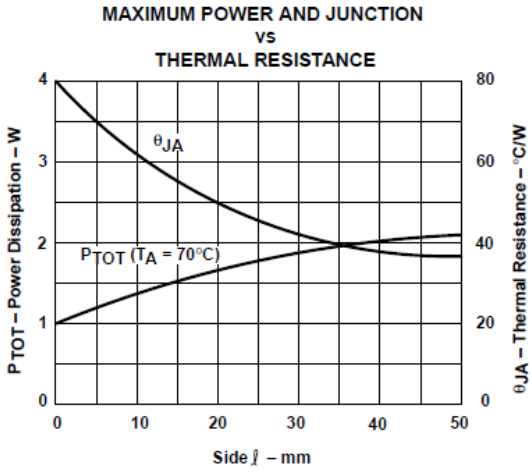


Figure 9

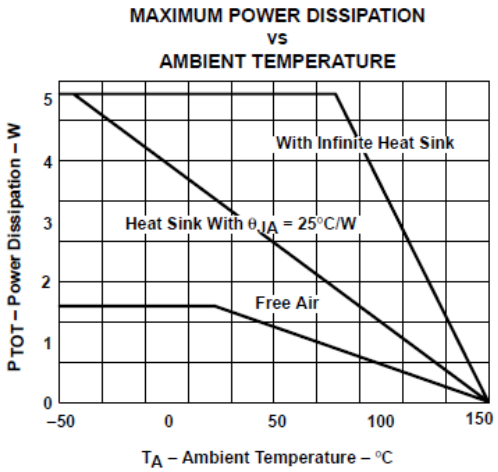


Figure 10

10.2 PROGRAMA PRUEBA DE MOTORES ARDUINO

```
int control;
void setup()
{
pinMode(2,OUTPUT);
pinMode(3,OUTPUT);
pinMode(6,OUTPUT);
pinMode(7,OUTPUT);
pinMode(9,OUTPUT);
pinMode(10,OUTPUT);
}
void loop()
{
control = 1;
if (control == 1)
{
analogWrite(9,100);
analogWrite(10,100);
digitalWrite(3,HIGH);
digitalWrite(2,LOW);
digitalWrite(7,HIGH);
digitalWrite(6,LOW);
}
if (control == 0)
{
analogWrite(9,100);
analogWrite(10,100);
digitalWrite(5,HIGH);
digitalWrite(4,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
}
```

10.3 PROGRAMA ARDUINO PARA SEGWAY BASADO EN GIROSCOPIO Y ACELEROMETRO

```
#include<Wire.h>

//Direccion I2C de la IMU
#define MPU 0x68

//Constantes de conversion
#define A_R 16384.0
#define G_R 131.0

//Conversion de radianes a grados 180/PI
#define RAD_A_DEG = 57.295779

//MPU-6050 da los valores en enteros de 16 bits
//Valores sin refinar
int16_t AcX, AcY, AcZ, GyX, GyY, GyZ;

//Angulos
floatAcc[2];
floatGy[2];
float Angle[2];
floatTiempo[200];
int
error,kp,ki,kd,angulo_ref,time,PID_prop,PID_int,PID_der,dT,error_int,error_int_last,error_der,ang_last,u,i,u
ant;
unsigned long tiempo;
void setup()
{
Wire.begin();
Wire.beginTransmission(MPU);
Wire.write(0x6B);
Wire.write(0);
```

```
Wire.endTransmission(true);
Serial.begin(9600);
}
void loop()
{
tiempo =0;
pinMode(2,OUTPUT);
pinMode(3,OUTPUT);
pinMode(6,OUTPUT);
pinMode(7,OUTPUT);
pinMode(9,OUTPUT);
pinMode(10,OUTPUT);
  //Leer los valores del Acelerometro de la IMU
Wire.beginTransmission(MPU);
Wire.write(0x3B); //Pedir el registro 0x3B -  AcX
Wire.endTransmission(false);
Wire.requestFrom(MPU,6,true); //A partir del 0x3B, se piden 6 registros
AcX=Wire.read()<<8|Wire.read(); //Cada valor ocupa 2 registros
AcY=Wire.read()<<8|Wire.read();
AcZ=Wire.read()<<8|Wire.read();

//Se calculan los angulos Y, X respectivamente.

Acc[1] = atan(-1*(AcX/A_R)/sqrt(pow((AcY/A_R),2)+pow((AcZ/A_R),2)))*RAD_TO_DEG;
Acc[0] = atan((AcZ/A_R)/sqrt(pow((AcX/A_R),2) + pow((AcY/A_R),2)))*RAD_TO_DEG;

//Leer los valores del Giroscopio
Wire.beginTransmission(MPU);
Wire.write(0x43);
Wire.endTransmission(false);
Wire.requestFrom(MPU,4,true);
GyX=Wire.read()<<8|Wire.read();
GyY=Wire.read()<<8|Wire.read();
```

```
//Calculo del Angulo del Giroscopio
Gy[0] = GyX/G_R;
Gy[1] = GyY/G_R;

  //Aplicar el Filtro Complementario
Angle[0] = 0.98 *(Angle[0]+Gy[0]*0.010) + 0.02*Acc[0];
Angle[1] = 0.98 *(Angle[1]+Gy[1]*0.010) + 0.02*Acc[1];

   //PID
kp = 30;
ki = 10;
kd = 40;
time = 2;

// Ángulo de consigna y obtención del error
   //
angulo_ref = 1.5;
error =angulo_ref - Angle[1]; // Solamente se utiliza el Angulo Y, que corresponde a la inclinacion en ese eje
   // Cálculo termino proporcional
   //
PID_prop = kp * error;
   // Cálculo termino integral
   // con implementación anti-WindUp
//
error_int = error_int_last + error * time / 1000;


if (error_int> 50/ki){
error_int = 50/ki;
   }
if (error_int< -50/ki){
error_int = -50/ki;
```

```
}
PID_int = ki * error_int;

   // Cálculo termino derivativo

error_der = Angle[1] - ang_last ;
PID_der = kd * error_der;

   // Generación de la acción de control
//
   u = PID_prop + PID_int - PID_der;
   // condición anti-WindUp 2

if (u > 250){
u = 250;}

if (u < -250){
   u = -250;}
error_int_last = error_int;
ang_last = Angle[0];
uant = u;
if (Angle[1] >angulo_ref)
 {
analogWrite(9,u);
analogWrite(10,u);
digitalWrite(2,HIGH);
digitalWrite(3,LOW);
digitalWrite(7,HIGH);
digitalWrite(6,LOW);
 }
if (Angle[1] <angulo_ref)
{
analogWrite(9,u);
```

```
analogWrite(10,u);
digitalWrite(3,HIGH);
digitalWrite(2,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
}
 }
```