Planificación

- Introducción a la Inteligencia Artificial
- Inteligencia Artificial (IA).
- Introducción a la Robótica.
- Herramientas Matemáticas para la localización espacial y Simulación en Matlab.
- Dinámica del robot.
- Control cinemático.
- ***Control Dinámico.***
- Robótica Embedded, Compiladores y Lenguajes de programación.
- Diseño y Aplicaciones de Robots.
- Cobots.
- Impresoras 3D y 4D

Control Dinámico

- **Control mono-articular: Control PID.**
- Control Multiarticular: Control PID.
- Índices de Comportamiento en sistemas de Control.
- Control adaptativo: Con planificación de ganancia, y con modelo de referencia. Modelado e identificación del control de movimiento del robot.

Control Dinámico

Breve Repaso:

Por lo visto hasta ahora, el ***Control Cinemático (Cinemática del Robot) selecciona las trayectorias que debe seguir el robot para ajustarse a las especificaciones del usuario***. En la práctica, este ajuste resulta algo ***difícil de realizar*** pues las ***características del robot tales como inercias, rozamiento y holguras***, ***suelen ser desconocidas***, lo cual impiden que la trayectoria calculada coincida con la deseada.

Es decir que, para hacer un análisis completo del **movimiento del Robot**, tenemos que tener en cuenta tanto la **Cinemática** como la **Dinámica** del mismo dado que tenemos que tener en cuenta las **fuerzas y torques** que actúan en las articulaciones.

Control Dinámico

Por esta razón, mediante el ***Control Dinámico*** vamos a estudiar estudiar que ***las trayectorias seguidas por el robot, sean lo más parecidas posibles a las propuestas por el usuario***, es decir, que las **perturbaciones** que puedan ocurrir en el Robot, se vean lo más **atenuadas** posibles.

Control Dinámico

Sistemas de Control

Por todo lo visto hasta ahora, si tomamos el caso de un **Brazo Robótico**, este puede tener múltiples utilidades, en el caso de la Industria puede servir para pintar las partes de un automóvil, trasladar objetos pesados, etc. Para estos casos, el Robot debe **controlar la fuerza** con la que trabaja, ya que de lo contrario puede **soltar el objeto levantado** o, en el caso extremo, darle mucha fuerza y romperlo.

Entonces, para estos casos, el Robot necesita un **Controlador** que pueda darle la **"orden" al actuador** (pueden ser motores), para que el mismo pueda darle un cierto grado de fuerza al objeto que desea mover.

Control Dinámico

Lo anterior comentado se llaman ***Estrategias de Control***, dado que logran los comportamientos descriptos. Algunas de estas estrategias pueden ser: ***Control de Movimiento, Control de Fuerza y Control de Impedancia.***

Entonces, ***¿cómo sabemos cuál de estos métodos usar?***

Control Dinámico

Eso ***depende del ambiente y la tarea*** que se quiere llevar a cabo, por ejemplo, en el caso de una ***pinza*** en el extremo del Robot, se puede usar un **Control de Fuerza** dado que no se tiene que controlar su movimiento en el espacio libre.

Control Dinámico

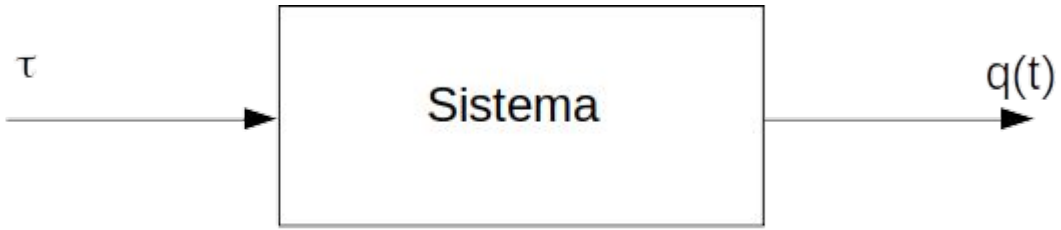
En base a esto, vemos entonces que la pinza del robot, ***no puede imponer una fuerza*** pues depende de lo que quiera abrazar (o levantar), es decir, el entorno va a ***limitar tanto la fuerza como los movimientos de la pinza.***

Dicho de otra forma, si a la pinza le damos la **"orden"** de que **tome un objeto con cierta fuerza** y dicho objeto **es más pesado**, podemos decir que la pinza se va a ver **perturbada** pues **no estaba preparada para el peso de ese objeto**.

https://www.quanser.com/products/2-dof-serial-flexible-link/

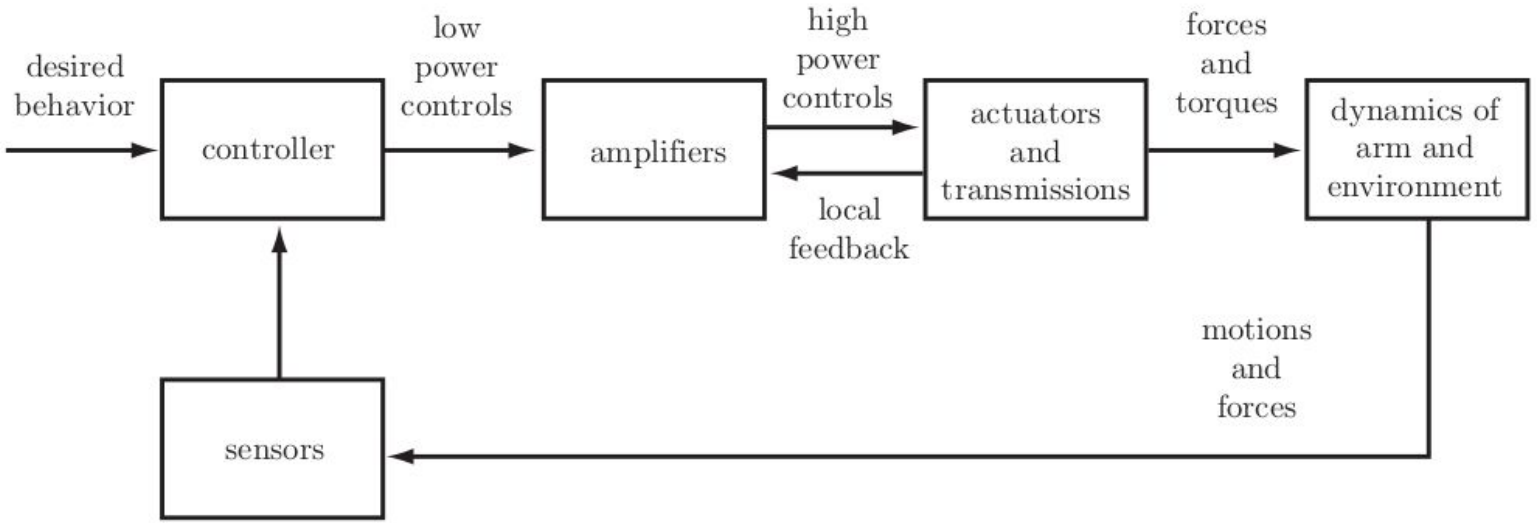
Control Dinámico

El diagrama anterior, es lo que se llama un sistema en ***lazo abierto***, con lo cual, ***cualquier perturbación que pudiera ocurrir, afectaría notablemente a la salida deseada q(t)***.



Para evitar esta problemática, se recurre a ***cerrar el lazo*** o, lo que se llama habitualmente, una **Realimentación**. Lo podemos ver en el siguiente diagrama:

Control Dinámico



[14]

Control Dinámico

Del diagrama anterior, podemos ver lo siguiente:

Comportamiento deseado: Aquí se ingresa la variable que queremos controlar, en el valor deseado. Por ejemplo, si se quiere controlar la velocidad, pueden ser 100 RPM. El sistema debería, en su salida, llegar a ese valor. Muchas veces se llama a esta entrada, Set Point o Referencia.

Controlador: Aquí es donde se programa la acción de Control. Se puede usar cualquiera de los controladores estudiados.

Amplificadores: Elevan los niveles de corrientes y/o tensión para poder accionar a los actuadores

Control Dinámico

Actuadores: Son los encargados de generar las dinámicas del Robot, por ejemplo, pueden ser motores, transmisores, etc.

Sensores: Son encargados de medir la variable que se quiere controlar, pueden ser sensores de posición, de fuerza, de ángulo, etc.

Del diagrama anterior se puede ver entonces que el **Controlador** recibe **"órdenes"** de la **Referencia** y de los **sensores**. En base a estas **señales** y al **algoritmo de Control programado**, genera una **"orden"** para controlar la **dinámica del Robot.**

Control Dinámico

En particular, a la **diferencia** entre la **señal de Referencia** y la de los **sensores**, se llama **señal de error.** Viene dada por

$$e = R - S$$

Siendo:

e: Señal de error

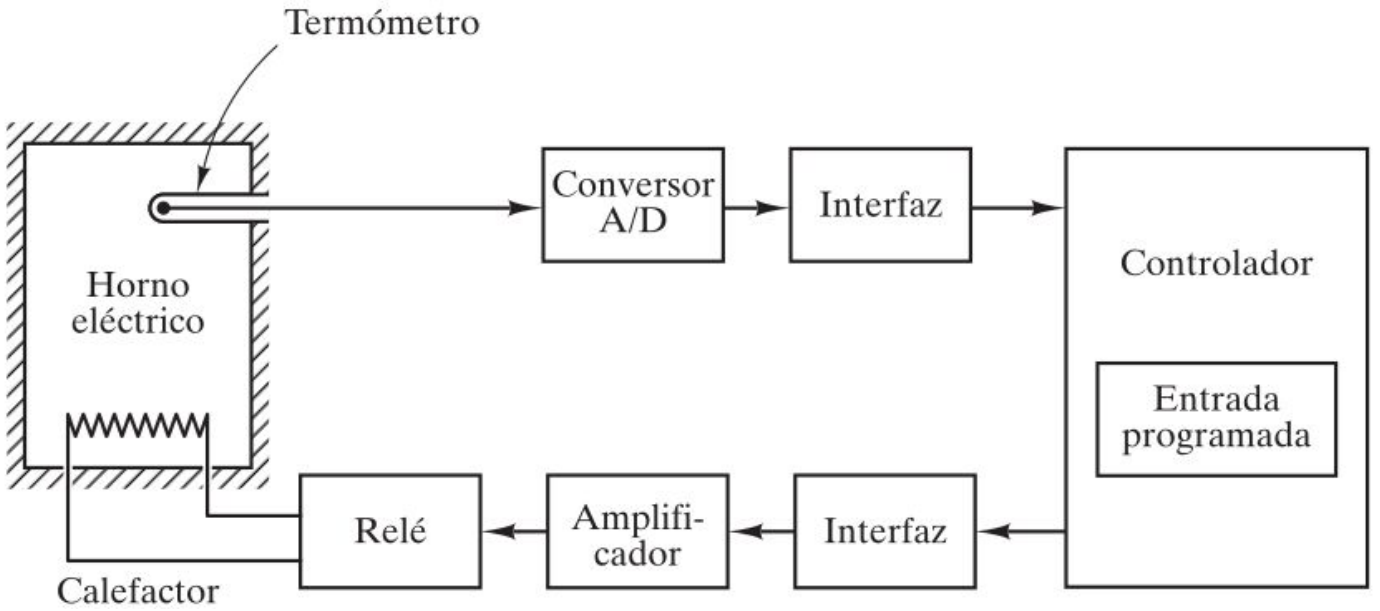
R: Señal de Referencia

S: Señal de los sensores

Es de gran importancia esta señal pues, en base a ella, el **Controlador** podrá **ejecutar los algoritmos y tomar una "Decisión".**

Control Dinámico

Ejemplo de Sistema de Control



[18]

Control Dinámico

En nuestro caso, si queremos controlar una posición angular, podemos definir de la siguiente forma:

$$\theta_e = \theta_d - \theta$$

Siendo:

θ_e: Ángulo de error

θ_d: Ángulo deseado

θ: Ángulo a controlar

Control mono-articular

Lo anterior descripto, lo estaremos llevando a un caso simple para un Robot con **una sola articulación**, lo que se llama el **Control Mono-Articular**. Luego se podrá extender esta idea a **múltiples articulaciones**.

Para esto, conviene hacer un breve repaso de la **Dinámica del Robot**.

Control mono-articular

Recordando de la **Dinámica del Robot**, sabemos que l***as características mecánicas*** (dimensiones, peso, tipo de actuadores, etc), ***influyen notablemente.*** Por eso podríamos pensar en dos tipos de Dinámicas:

Dinámica Directa:

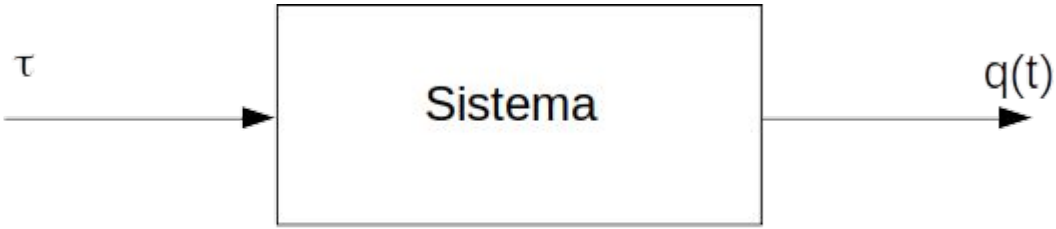
$$\ddot{\theta} = M^{-1}(\theta)\left(\tau - h(\theta, \dot{\theta})\right)$$

Dinámica Inversa:

$$\tau = M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}),$$

Control mono-articular

De la **Dinámica Directa** sabemos que podemos obtener la **aceleración del movimiento** en base al **torque** (o fuerzas) que se le está imprimiendo. De esta forma, se puede bosquejar un diagrama en bloques que queda de la siguiente forma:



Control PID

El sistema anterior se encuentra en **lazo abierto**, con lo cual, por lo estudiado anteriormente, resulta adecuado **cerrar el lazo** (Realimentar) y agregar un **Controlador**. Para esto, existe uno en particular de gran uso en la Industria, llamado **PID.**

Este tipo de Controlador se compone de tres partes: **Proporcional** - **Integral** - **Derivativa**. Ajustando cada uno de estos parámetros, ***podemos obtener distintas dinámicas sobre la articulación.***

https://www.quanser.com/products/2-dof-serial-flexible-link/

Control PID

Para la comprensión de este **Controlador**, es muy importante tener en cuenta a la **señal de error** comentada antes, la cual se obtiene como:

$$\theta_e = \theta_d - \theta$$

o también

$$e = R - S$$

A partir de esta señal, veremos las tres partes de este controlador:

Control PID

Proporcional: Ejecutar una acción de control así

$$u = Kp.\ e$$

Siendo:

u: Acción de control. Señal que llega al actuador (para luego realizar la dinámica del robot)

Kp: Constante de proporcionalidad. Es una constante

e: señal de error

Control PID

Integral: Ejecutar una acción de control así

$$u = Ki. \int_0^t e \, dt$$

Siendo:

u: Acción de control. Señal que llega al actuador (para luego realizar la dinámica del robot)

Ki: Constante de integración.

e: señal de error

Control PID

Derivativa: Ejecutar una acción de control así

$$u = Kd.\ de/dt$$

Siendo:

u: Acción de control. Señal que llega al actuador (para luego realizar la dinámica del robot)

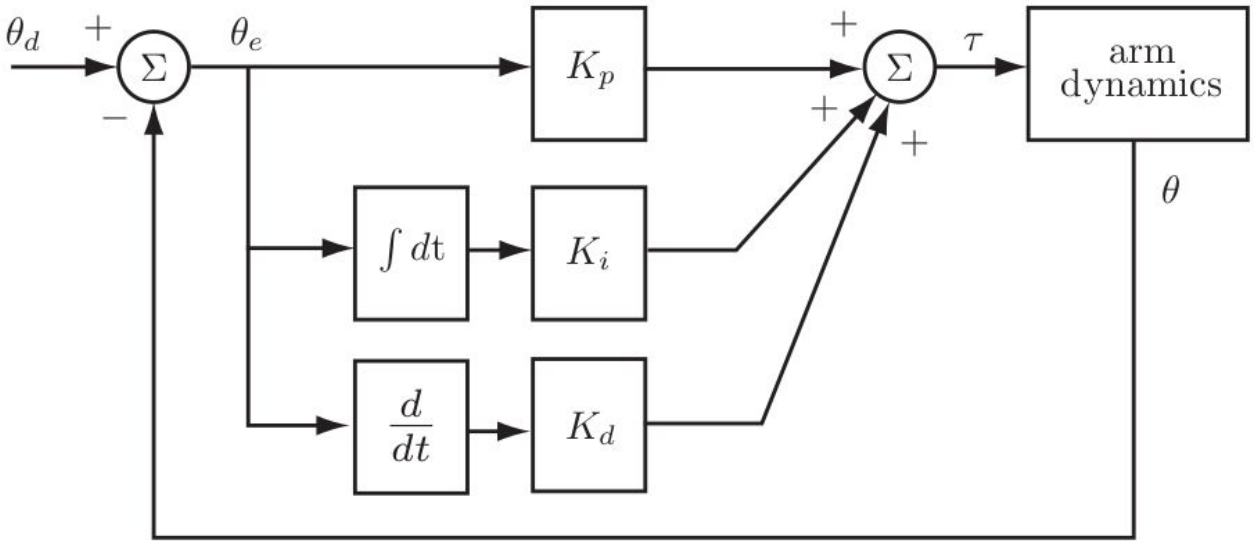
Ki: Constante de derivación.

e: señal de error

Control PID

Si se combinan estas **tres acciones**, el **Controlador** puede ejecutar **distintas dinámicas sobre la articulación**, logrando así, que la misma **pueda seguir la trayectoria deseada por el usuario**. Esto lo podemos ver en la siguiente figura:

Control PID



[14]

Control PID

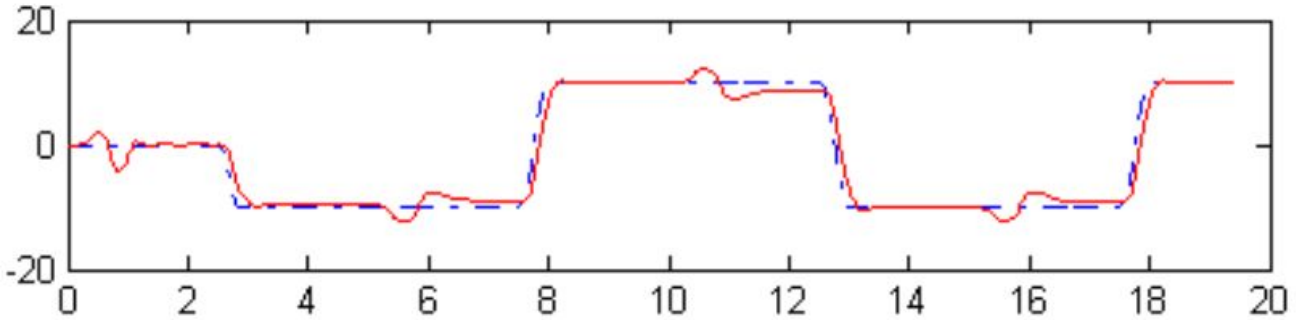
Matemáticamente queda:

$$\tau = K_p\theta_e + K_i \int \theta_e(\mathrm{t})d\mathrm{t} + K_d\dot{\theta}_e,$$

[14]

Control PID

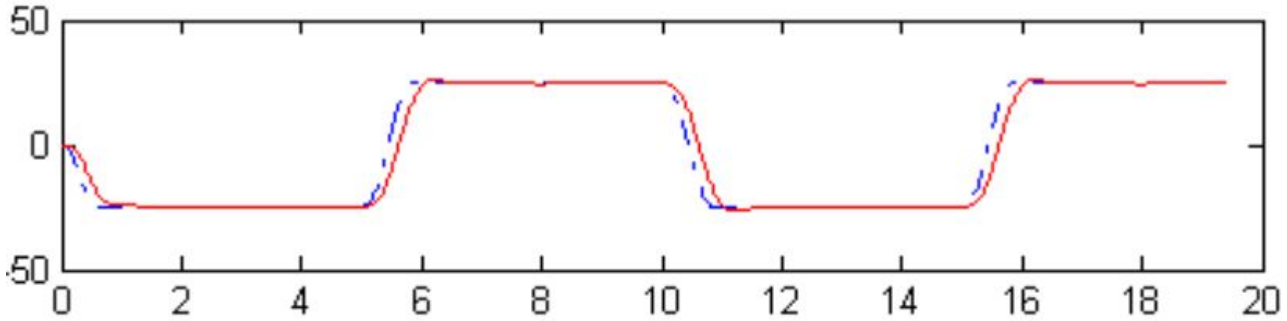
Por ejemplo, si tuviéramos un sistema con un **PID** que no está ajustado correctamente, podríamos tener una respuesta así:



La señal **azul** es la **deseada** y la **roja** la de la **dinámica** del Robot.

Control PID

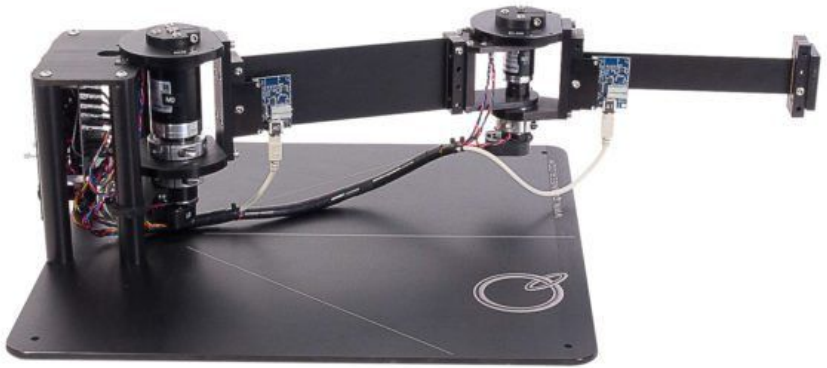
Si se ajusta mejor el **PID**, podríamos tener una respuesta así:



La señal **azul** es la **deseada** y la **roja** la de la **dinámica** del Robot.

Control PID

Las señales anteriores son tomadas de las hojas de datos del Robot 2 DOF Serial Flexible Link, el cual usa otro controlador pero, a los fines prácticos, nos sirve para comprender el funcionamiento de un PID



Control Dinámico

- Control mono-articular: Control PID.
- **Control Multiarticular: Control PID.**
- Control adaptativo: Con planificación de ganancia, y con modelo de referencia. Modelado e identificación del control de movimiento del robot.

Control Multiarticular

La consideración de que ***cada articulación no se ve afectada más que por sus propias características dinámicas***, siendo el ***efecto de las demás poco significativo, no siempre resulta aceptable.***

Tener en cuenta que ahora, la posición deseada de cada articulación, pasa a ser un **vector**. Por ejemplo, si tenemos cinco articulaciones, tendremos que hacer un control para cada articulación y por ende nos quedaría un vector de cinco articulaciones. Para cada una de ellas, tendremos los Kp, Ki y Kd correspondientes.

Control Dinámico

- Control mono-articular: Control PID.
- Control Multiarticular: Control PID.
- **Control adaptativo: Con planificación de ganancia, y con modelo de referencia. Modelado e identificación del control de movimiento del robot.**

Control Adaptativo

Por lo visto anteriormente, ***si se conoce perfectamente el modelo dinámico del robot***, es posible ***conseguir que la trayectoria real, coincida exactamente con la deseada***.

Sin embargo, ***en la práctica***, esta técnica tiene una ***aplicación limitada***, bien porque los ***parámetros*** que definen al modelo (inercias, rozamientos, etc), ***no son conocidos con precisión***, o bien porque el ***cálculo continuo en tiempo real*** del modelo inverso es excesivamente ***lento***.

La técnica de ***Control Adaptativo*** se puede aplicar ***en ocasiones donde el modelo del robot cambia continuamente por modificar las condiciones de funcionamiento***.

Control Adaptativo

La idea básica del **Control Adaptativo** es ***modificar en tiempo real, los parámetros que definen al controlador PID.***

En el caso de un robot, ***el comportamiento del sistema y su modelo cambia con los valores de sus variables articulares y con la carga que transporta***. De esta forma, ***suponiendo que se hubiese ajustado perfectamente un controlador PID*** para determinadas condiciones, este ajuste no sería válido cuando algunas de estas condiciones cambien.

Para este autoajuste, existen técnicas, que se describe a continuación.

Control Adaptativo

Control Adaptativo por planificación de ganancias:

Resulta ser el caso ***más simple*** de control adaptativo. ***Un número determinado de controladores*** están ***precalculados para diferentes condicione***s de funcionamiento. ***Según se detecta*** que éstas se modifican, ***se conmuta de un controlador a otro***.

Control Adaptativo con modelo de referencia:

Realiza una ***comparación en línea*** entre el ***comportamiento del sistema real y el deseado***. ***El error entre ambos comportamientos*** se utiliza para modificar ***los parámetros del Controlador.***

Control Adaptativo autoajustable:

Identifica ***continuamente el modelo del sistema*** y utiliza algún ***algoritmo de diseño para autoajustar los parámetros del Controlador.***

Planificación

- Introducción a la Inteligencia Artificial.
- Inteligencia Artificial (IA)
- Introducción a la Robótica.
- Herramientas Matemáticas para la localización espacial y Simulación en Matlab.
- Dinámica del robot.
- Control cinemático.
- Control Dinámico.
- ***Robótica Embedded, Compiladores y Lenguajes de programación.***
- Diseño y Aplicaciones de Robots.
- Cobots.
- Impresoras 3D y 4D

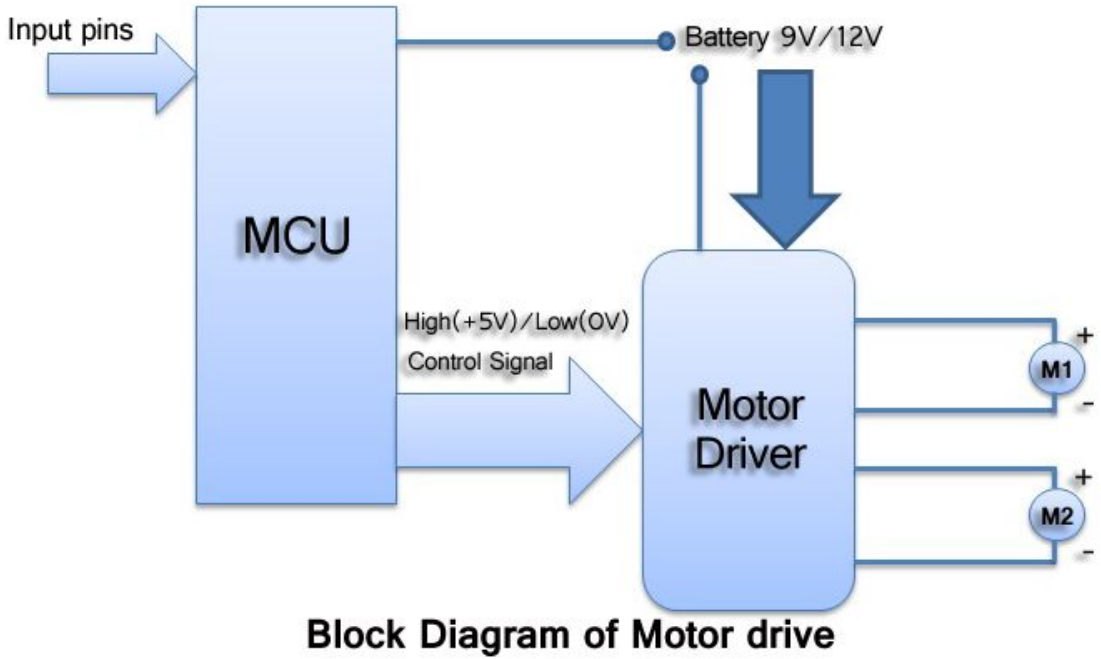
Robótica Embedded

- **Generalidades en Drivers de Potencia DC, Brushless y Direct Drive.**
- Generación de PWM Conmutación electrónica de fases.
- Puente H monofásico y Trifásico: Modelo Spice. Control del puente electrónico: Por par complementario.
- Controladores Embedded y Sistemas Operativos.
- Compiladores y Lenguajes estándar de programación: Abb, Adept, Microsoft.

Drivers de Potencia

Un Driver de motor resulta ser la ***interfaz*** entre el ***motor y la unidad computacional***.

[40]



Block Diagram of Motor drive

Drivers de Potencia

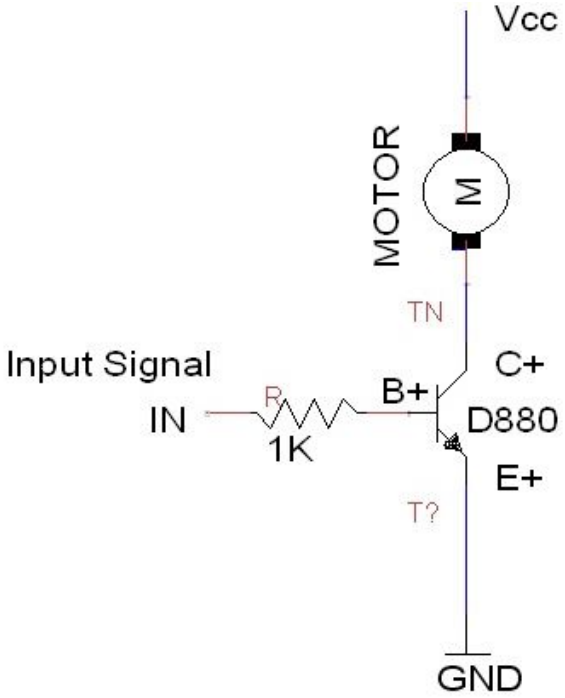
Se puede decir que esta interfaz es como un ***"puente"*** entre la ***computadora de control*** y el ***motor***.

La ***necesidad*** de esta interfaz radica en que el ***nivel de potencia que pueden entregar las computadoras***, no es ***suficiente*** para activar a un motor.

En el caso de un ***motor de DC***, la configuración más simple como interfaz, es con un transistor, tal como se muestra a continuación:

Drivers de Potencia

Input (IN)	Motor Action
0 volt	Motor Stops
5V	Motor is in Action



Transistor como llave electrónica

[40]

Drivers de Potencia

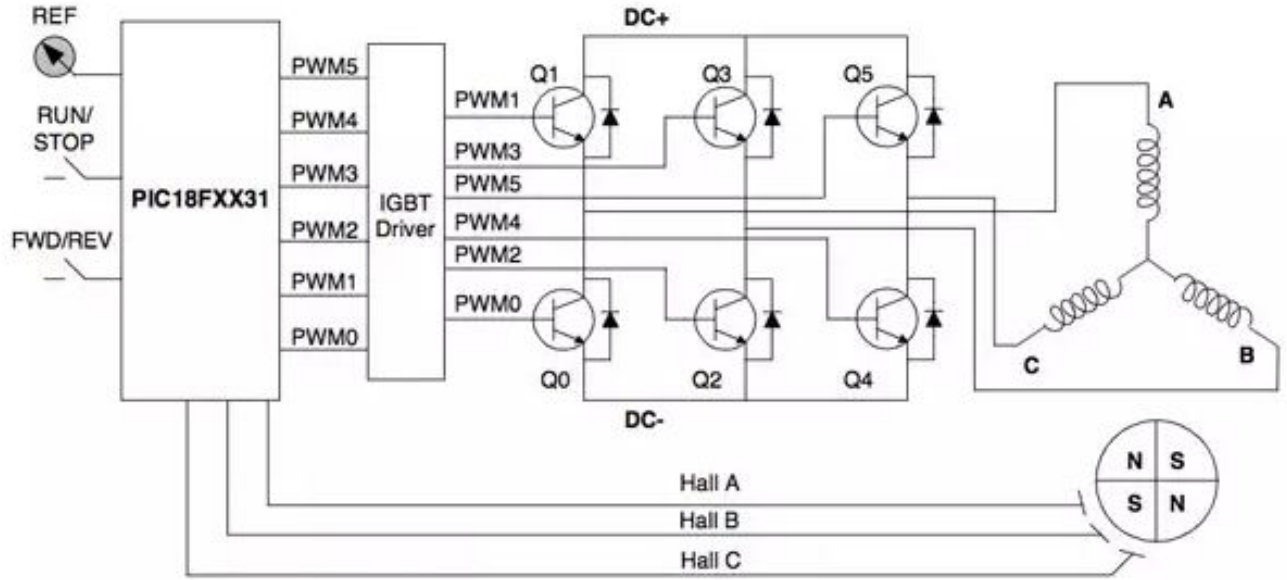
En el caso de un motor **Brushless** (sin escobillas), el driver es diferente al anterior.

En este caso, el sistema muestra las ***tres bobinas del motor dispuestas en forma de «Y»***.

La salida del microcontrolador incluye ***señales moduladas de ancho de pulso (PWM)*** que determinan el ***promedio de la tensión*** y el ***promedio de corriente*** en las bobinas (y por lo tanto, la velocidad y par de torsión del motor).

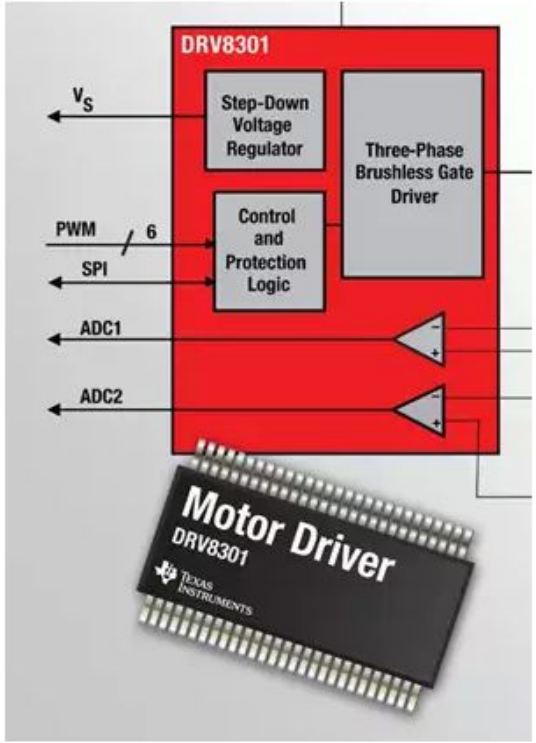
El motor utiliza ***tres sensores de efecto Hall*** (A, B y C) para ***indicar la posición del rotor***. El rotor utiliza dos pares de imanes permanentes para generar el flujo magnético.

Drivers de Potencia



[41]

Drivers de Potencia



[41]

Drivers de Potencia

Direct drive:

Este término se puede aplicar a cualquier motor que ***impulse directamente una carga o rotor sin elementos de transmisión como engranajes, poleas o cadenas***. **[42].** Esto garantiza mayor durabilidad.

Robótica Embedded

- Generalidades en Drivers de Potencia DC, Brushless y Direct Drive.
- **Generación de PWM Conmutación electrónica de fases.**
- Puente H monofásico y Trifásico: Modelo Spice. Control del punte electrónico: Por par complementario.
- Controladores Embedded y Sistemas Operativos.
- Compiladores y Lenguajes estándar de programación: Abb, Adept, Microsoft.

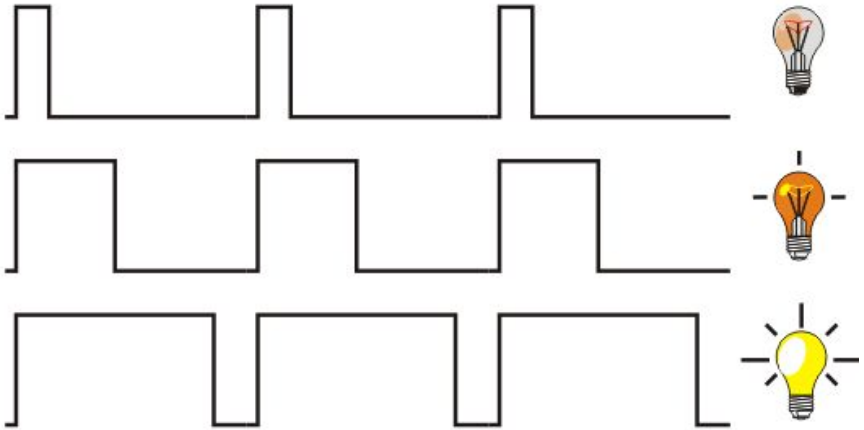
Generación Señal PWM

Pulse Width Modulation, o PWM, es una técnica para obtener ***resultados analógicos con medios digitales***.

Para esto, se utiliza ***onda cuadrada***, es decir, una señal conmutada entre encendido y apagado.

Este patrón de ***encendido-apagado*** puede simular tensiones entre el encendido total (5 voltios) y el apagado (0 voltios). La duración de "encendido" se llama ***ancho de pulso***.

Para obtener valores análogicos variables, se ***modula*** el ancho de pulso. **[43]**



Generación Señal PWM

Si repite este patrón de ***encendido-apagado*** lo suficientemente rápido con un motor de DC, el resultado es que el motor podrá cambiar su velocidad.

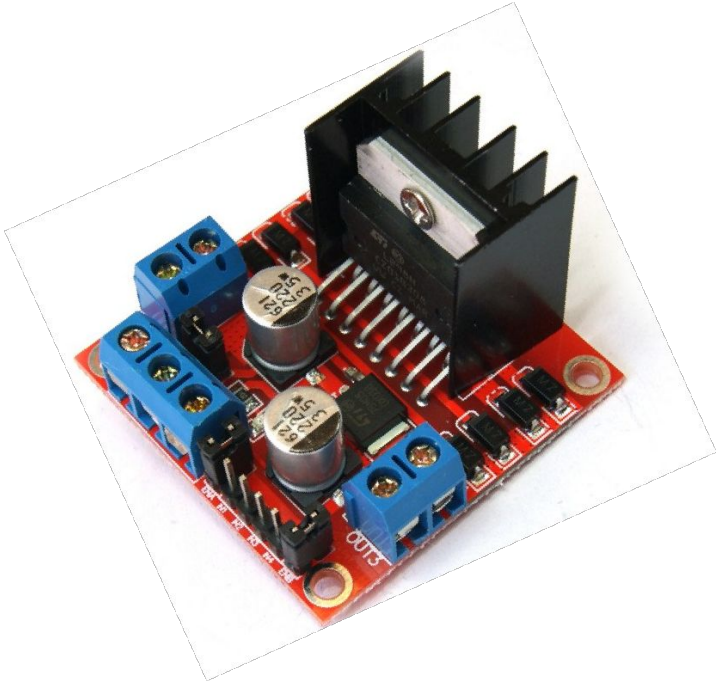
La relación entre la duración del ***ancho del pulso*** y el ***período*** de la señal, se la llama ***ciclo de trabajo.***

Robótica Embedded

- Generalidades en Drivers de Potencia DC, Brushless y Direct Drive.
- Generación de PWM Conmutación electrónica de fases.
- **Puente H monofásico y Trifásico: Modelo Spice. Control del puente electrónico: Por par complementario.**
- Controladores Embedded y Sistemas Operativos.
- Compiladores y Lenguajes estándar de programación: Abb, Adept, Microsoft.

Puente H

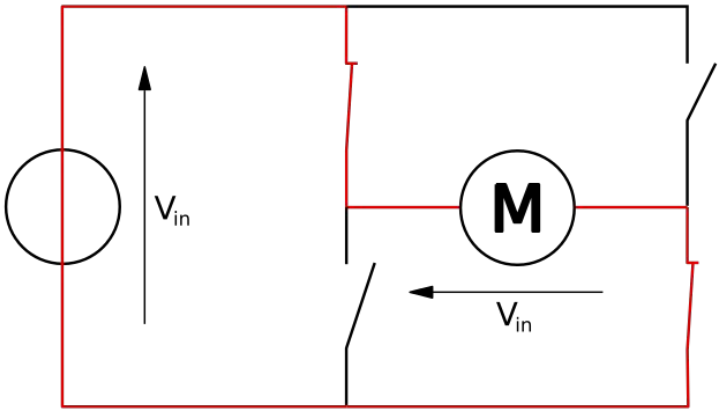
El puente H ***modifica el sentido de giro*** de un motor DC.
En particular, el módulo con circuito integrado L298, puede controlar hasta dos motores de DC, o un motor Paso a Paso



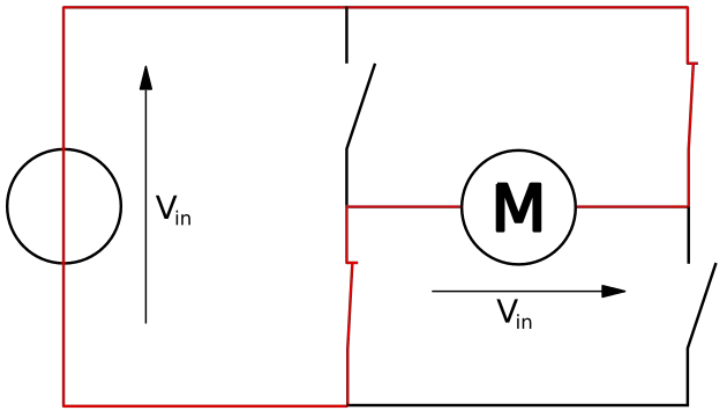
Puente H

¿Cómo funciona el puente H?

Gira en un determinado sentido



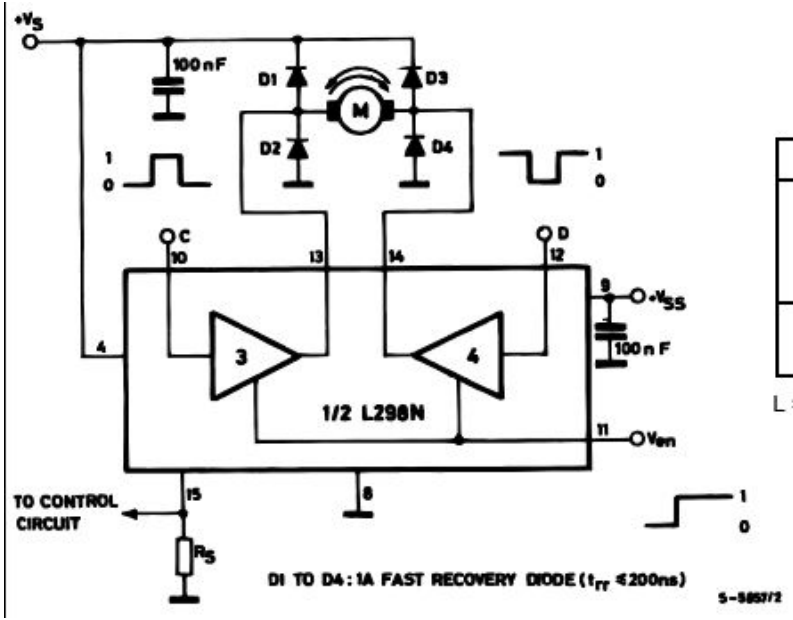
Gira en sentido inverso



[45]

Puente H

El fabricante provee una "Tabla de verdad" para manejar el sentido del motor



Inputs		Function
V_{en} = H	C = H ; D = L	Forward
	C = L ; D = H	Reverse
	C = D	Fast Motor Stop
V_{en} = L	C = X ; D = X	Free Running Motor Stop

L = Low H = High X = Don't care

Actividad

En base a los controladores estudiados, analizar el **Driver TB6612**

1) Estudiar su funcionamiento teniendo en cuenta el visto para el L298.

Sugerencia: https://www.luisllamas.es/arduino-motor-dc-tb6612fng/

2) De la hoja de datos, estudiar cómo se produce el cambio de giro del motor (modos de operación).

Sugerencia: https://www.sparkfun.com/datasheets/Robotics/TB6612FNG.pdf

Robótica Embedded

- Generalidades en Drivers de Potencia DC, Brushless y Direct Drive.
- Generación de PWM Conmutación electrónica de fases.
- Puente H monofásico y Trifásico: Modelo Spice. Control del puente electrónico: Por par complementario.
- **Controladores Embedded y Sistemas Operativos.**
- Compiladores y Lenguajes estándar de programación: Abb, Adept, Microsoft.

Controladores

Se conoce como **sistema embebido** a un sistema está diseñado **específicamente para una función en particular pero**, desarrollado bajo **Hardware, Firmware y Software**.

A diferencia de los sistemas computacionales de oficina y laptops, estos sistemas solucionan un problema específico y están dispersos en todos los ambientes posibles de la vida cotidiana. **[16]**

Controladores

Por ejemplo, un termostato digital tiene un una **función dedicada al monitoreo y control** de la **temperatura circundante**. El usuario tiene **opciones** para **configurar las temperaturas** bajas y altas deseadas, **pero no puede cambiar su funcionalidad para que funcione de otra manera.**

Una de las ventajas de los Sistemas Embebidos es que, para actualizarlos, no es necesario modificar todo el sistema, simplemente **actualizando el Firmware**, se logra el objetivo. Esto se da actualmente en los **TVs, osciloscopios y modems**, dado que necesitan actualizar la información que intercambian, entre otra cosas.

Controladores

Entonces.....**¿qué diferencias existe entre una PC y un Sistema Embebido?**

Controladores

Podemos empezar pensando en las **características de una PC**:

- Tiene Microprocesador.
- Una gran memoria secundaria como disco rígido o disco de estado sólido.
- Unidades de E / S como pantalla, teclado, mouse y otros.
- Sistemas Operativos: Linux, Mac, Windows, etc.

[16]

Controladores

Características de un Sistema Embebido:

- Incorpora hardware que incluye el núcleo y las E / S necesarias para una función específica.
- Incorpora el software de la aplicación principal en Flash incorporado.
- Incorpora un **Sistema Operativo en Tiempo Real** (RTOS) que **supervisa las tareas del software de la aplicación** que se ejecutan en hardware. **[15]**

Además hay que tener en cuenta que los sistemas de **memoria y de procesamiento son mucho más limitados que los de una PC.**

Controladores

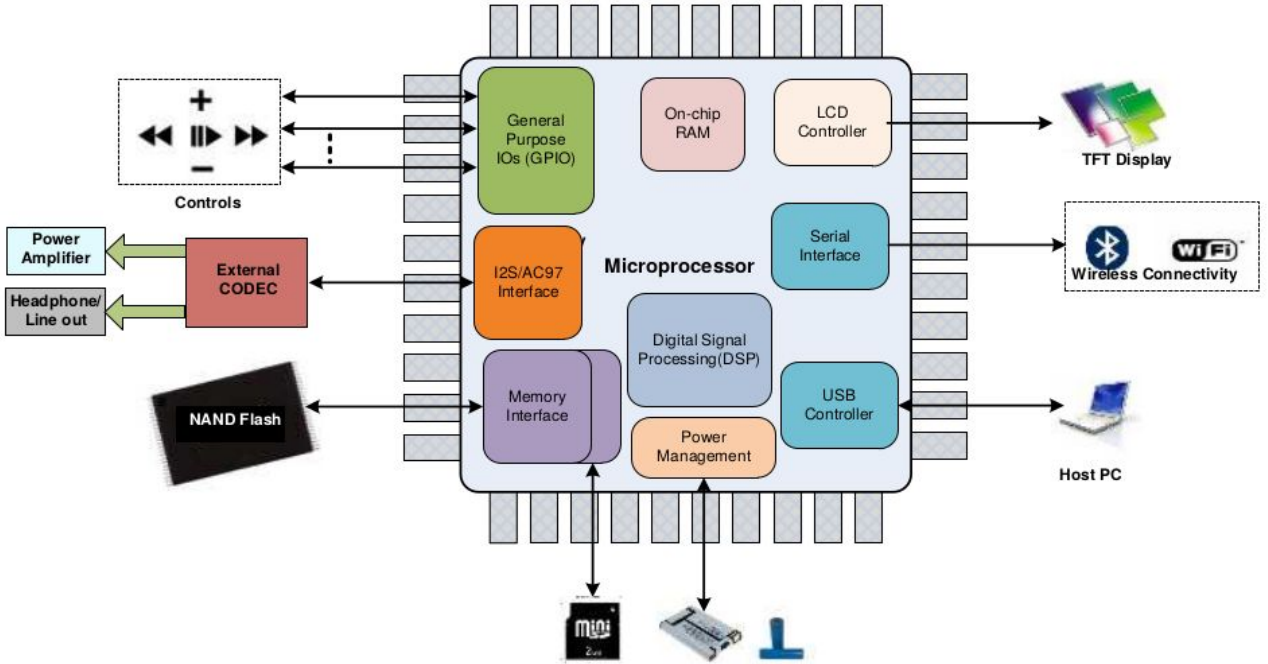
1) Ejemplo: Sistema de Audio

El trabajo principal es **reproducir el audio** a partir de un archivo teniendo en cuenta un menú accesible por el usuario.

Para llevar a cabo esta tarea, estos sistemas incluyen en su interior, un **microcontrolador**, el cual **actúa como "cerebro"**, en el sentido de que puede **detectar las distintas opciones que puede seleccionar el usuario** (a través de una interfaz) y, en base a esto, **seleccionar una canción, su volumen, su ecualización**, etc. Además, permite mostrar la información en algún display y guardar la información en algún sistema de memoria. En la siguiente figura se observa esto:

Controladores

Ejemplo: Sistema de Audio



[16]

Controladores

De la imagen anterior se pueden ver varias cosas:

1) Reciben información de un control.
2) Envían información a un display (TFT).
3) Protocolos de comunicación para audio.
4) Activan interfaces de potencia (transistores), para activar la ventilación.

Para todo esto, es esencial usar módulos que ya vienen integrados en el microcontrolador, tales como: **I2C**, **LCD Controller** , **USB controller**, etc.

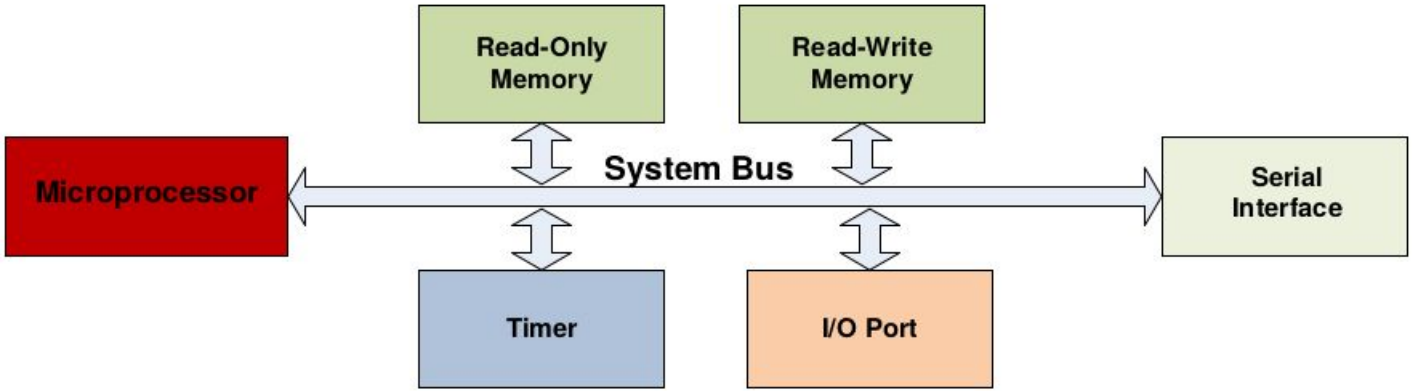
En un **microcontrolador**, aparte de los módulos mencionados, generalmente se encuentra un **microprocesador** o un **procesador embebido**.

Controladores

Entonces.....**¿qué diferencias existe entre un Microcontrolador y un Microprocesador?**

Controladores

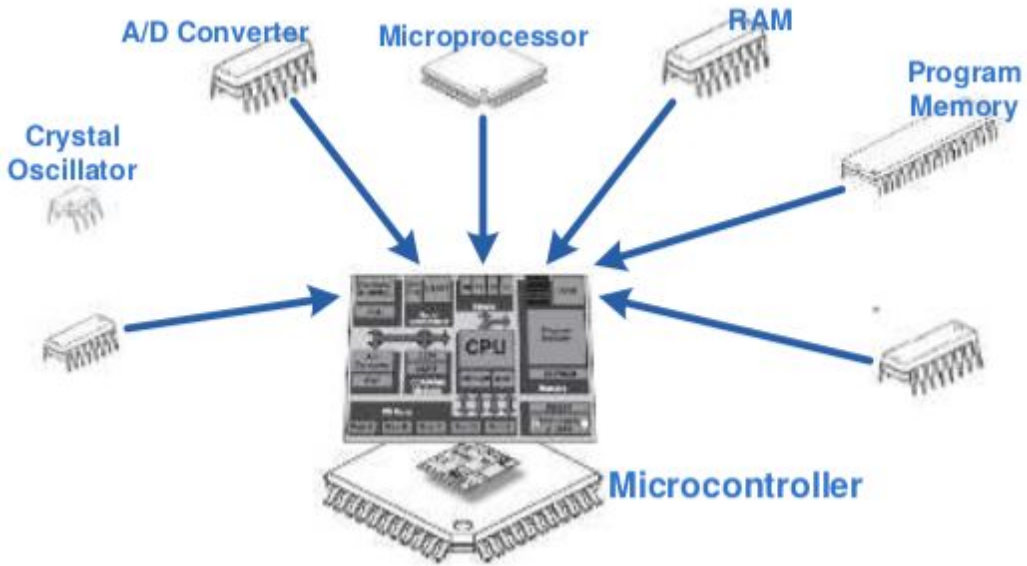
Un **microprocesador** es una **Unidad Central de Procesamiento** de computadora digital de propósito general. Lo que muchas veces se llama, **CPU**. Para que el sistema sea una ***microcomputadora***, además se le debe agregar memoria, como se muestra a continuación:



[16]

Controladores

A un **microcontrolador** lo podemos pensar de la siguiente forma:



[16]

Controladores

Algunos **microcontroladores** actuales, pueden ser: PIC, AVR (8 bits), LPC4337, STM32F103 (32 bits). Estos últimos contienen un procesador **ARM**, siendo uno de los más importantes actualmente. Dentro de **ARM** se pueden encontrar modelos Cortex M3, Cortex M4, etc.

A continuación veremos la arquitectura de un **microcontrolador PIC16F886**

http://ww1.microchip.com/downloads/en/devicedoc/41291d.pdf

Controladores

Robótica Embedded

- Generalidades en Drivers de Potencia DC, Brushless y Direct Drive.
- Generación de PWM Conmutación electrónica de fases.
- Puente H monofásico y Trifásico: Modelo Spice. Control del punte electrónico: Por par complementario.
- Controladores Embedded y Sistemas Operativos.
- **Compiladores y Lenguajes estándar de programación: Abb, Adept, Microsoft.**

Sistemas operativos

El ***sistema operativo del robot (ROS) es un marco flexible para escribir software de robot***. Es una ***colección de herramientas, bibliotecas y convenciones*** que tienen como ***objetivo simplificar la tarea de crear un comportamiento*** complejo y robusto en una amplia variedad de plataformas robóticas

https://www.ros.org/about-ros/

https://www.ros.org/

Compiladores

Lenguaje de programación ABB:

RAPID es un ***lenguaje de programación de alto nivel*** utilizado para ***controlar robots industriales ABB***. RAPID fue introducido junto con S4 Control System en 1994 por ABB, reemplazando el lenguaje de programación ARLA. **[46]**

Robotics **A**pplication **P**rogramming **I**nteractive **D**ialogue

Compiladores

Gazebo permite ***construir escenarios 3D*** en su computadora con robots, utilizando ***obstáculos y otros objetos***. Esto permite probar robots en escenarios complejos o peligrosos sin dañar al robot real.

Rviz es un ***visualizador 3D*** de código abierto que ***utiliza datos de sensores y marcadores de visualización personalizados para desarrollar capacidades de robot en un entorno virtual*** **[47]**

Planificación

- Introducción a la Inteligencia Artificial.
- Inteligencia Artificial (IA)
- Introducción a la Robótica.
- Herramientas Matemáticas para la localización espacial y Simulación en Matlab.
- Dinámica del robot.
- Control cinemático.
- Control Dinámico.
- Robótica Embedded, Compiladores y Lenguajes de programación.
- **Diseño y Aplicaciones de Robots.**
- Cobots
- Impresoras 3D y 4D

Aplicaciones de Robots

ABB:

Aplicaciones de Robots

Las empresas dedicadas a componentes oleodinámicos suelen usar robots de ABB

https://new.abb.com/news/detail/72216/the-first-yumir-in-italy-celebrates-its-fifth-birthday

Aplicaciones de Robots

EPSON:

https://epson.com.ar/robots-industriales-automatizacion-produccion

Aplicaciones de Robots

https://mediaserver.goepson.com/ImConvServlet/imconv/f552772775d79ce22cb36366d9490bd3e9896502/original?assetDescr=Robotics_Specifications_2020_Catalog_CPD-54833R2.pdf

Aplicaciones de Robots

https://mediaserver.goepson.com/ImConvServlet/imconv/f552772775d79ce22cb36366d9490bd3e9896502/original?assetDescr=Robotics_Specifications_2020_Catalog_CPD-54833R2.pdf

Aplicaciones de Robots

KUKA:

https://www.kuka.com/es-es

Aplicaciones de Robots

KUKA:

Sistemas de robots

KUKA ofrece robots industriales en una gran variedad de versiones con distintas capacidades y alcances de carga útil. Descubra más^.

Máquinas de producción

Nuestras máquinas de producción le permiten integrar sistemas y componentes estandarizados en procesos de producción customizados. Obtenga más información aquí.

Sistemas de producción

Desde células de producción individuales a un sistema de producción completo: nuestras divisiones de Industries y Systems son especialistas en la integración de sistemas.

https://www.kuka.com/es-es

Aplicaciones de Robots

KUKA:

https://www.kuka.com/es-es/productos-servicios/sistemas-de-robot/robot-industrial

Aplicaciones de Robots

Kawasaki:

https://robotics.kawasaki.com/en1/index.html?language_id=1

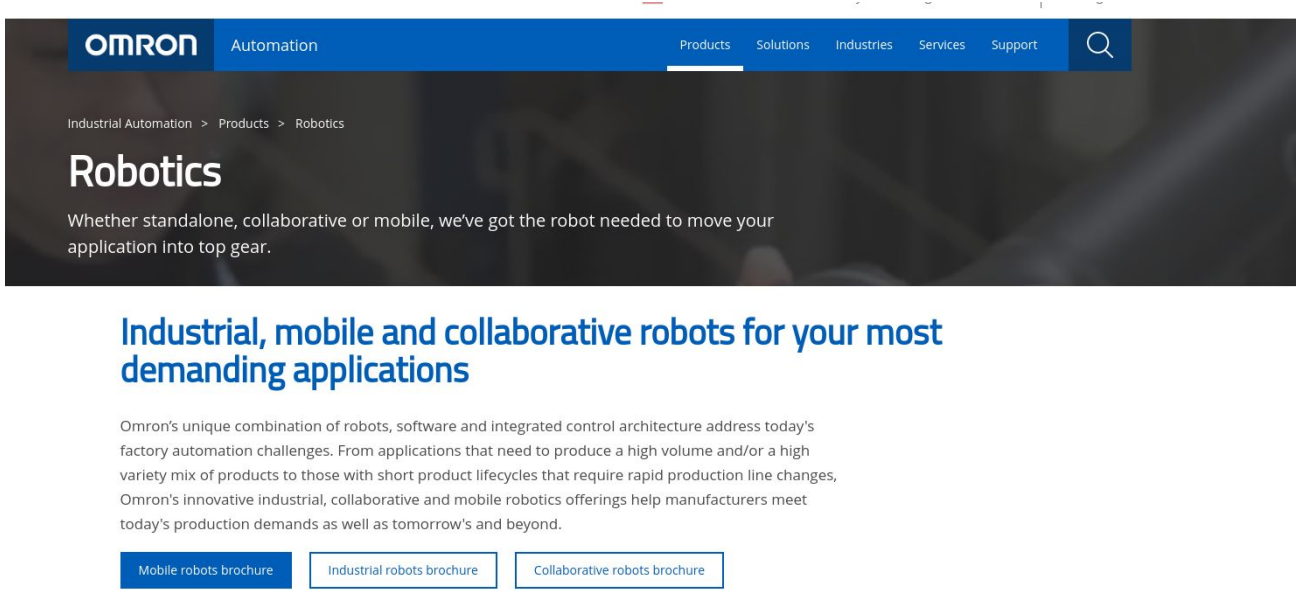
Aplicaciones de Robots

Kawasaki:

https://robotics.kawasaki.com/en1/products/robots/medical/

Aplicaciones de Robots

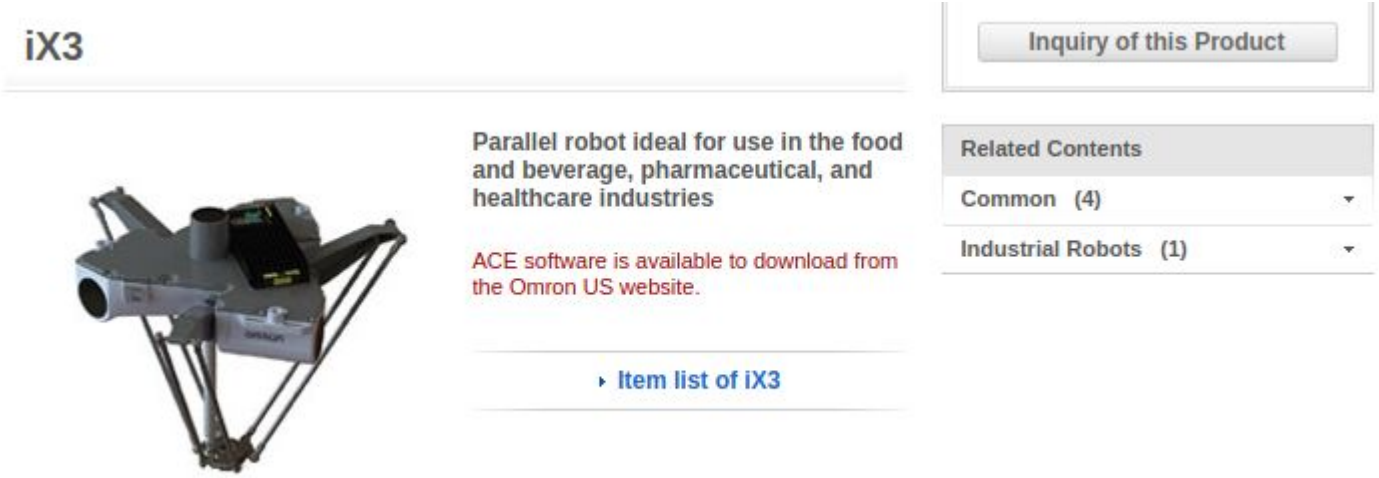
Omron:



https://automation.omron.com/en/us/products/category/robotics

Aplicaciones de Robots

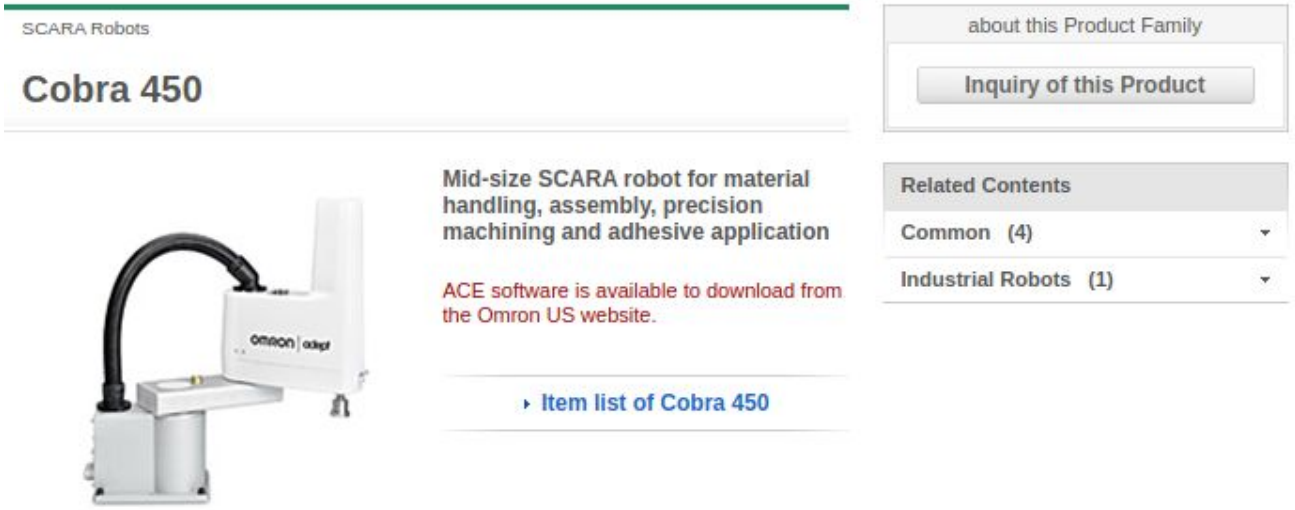
Omron:



http://www.ia.omron.com/products/family/3823/

Aplicaciones de Robots

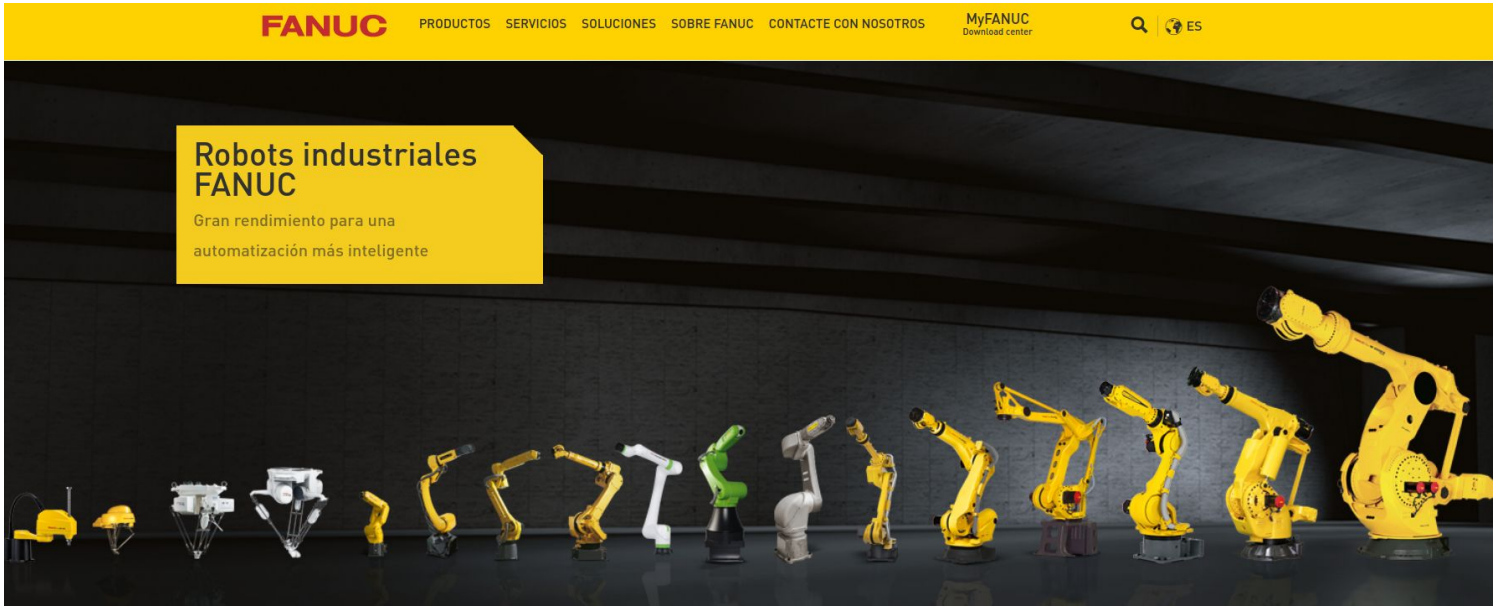
Omron:



http://www.ia.omron.com/products/family/3717/

Aplicaciones de Robots

Fanuc:



https://www.fanuc.eu/es/es/robots

Aplicaciones de Robots

Fanuc:

https://www.fanuc.eu/es/es/robots