

*Premio AADECA para proyectos estudiantiles en las áreas de medición industrial,
control automatización y robótica*

CONTROL DE UN ROBOT DIDÁCTICO

Por Javier Alejandro Jorge

javierbrk@gmail.com www.geocities.com/javierbrk/index.html

Para la cátedra de “Robótica y Animatrónica” de la Universidad Nacional de Córdoba a cargo del profesor Hugo Nicolás Pailos

Objetivos del proyecto

Si bien las prácticas de posicionamientos de robot se pueden simular con programas especiales que manejen 3 (tres) dimensiones o Herramientas matemáticas para la representación espacial (puesto que en la asignatura mencionada para la cual se hizo el proyecto se utiliza el método de Denavit Hartenberg) el autor creyó conveniente realizar un control de un pequeño robot educativo que permita, dada la información en forma matricial, realizar los movimientos espaciales respetando la posición inicial y final dadas por el usuario del sistema

1. Brindar un conjunto de herramientas para comprender la teoría de modelado de robots. (documentación de herramientas, Simulaciones, animaciones 3d...)
2. Facilitar la puesta en practica del proceso completo de modelado de un robot real para que pueda ser repetido como experiencia y ampliado en sucesivas iteraciones.

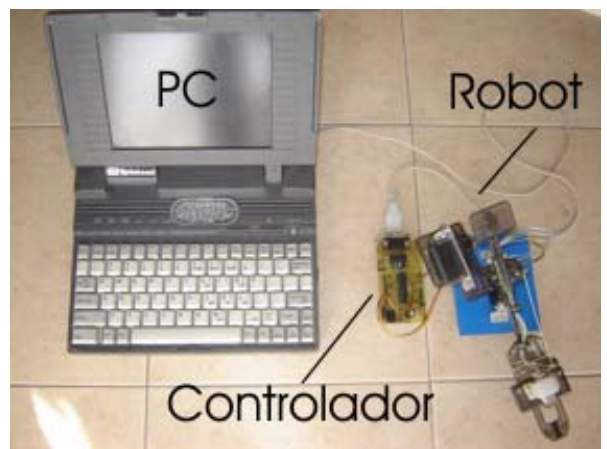
Solución adoptada

Se creó documentación detallada que cubre los siguientes aspectos: repaso de MatLab, matrices de transformación, teoría de cinemática directa e inversa, y modelado de robots con el uso del toolbox de robótica.

Una vez cubierto el aspecto teórico, el paso siguiente fue la creación de un robot y su controlador. Primero se modelo el robot con la ayuda de la documentación anteriormente citada y luego se lo controlo haciendo rutinas de prueba y secuencias.

Contenido del proyecto

- Documento de introducción a MatLab
- Documento de explicación de conceptos de robótica y descripción detallada del proyecto.
- Robot “Natalí”
- Interfaz robot PC (controlador)
- Archivos de modelado del robot
- Archivos de simulaciones del robot
- Archivos de rutinas de movimiento del robot
- Animaciones 3D Studio



Características

- Permite practicar con un robot de bajo costo, que además es totalmente transportable por su pequeño tamaño pero que respeta las reglas de posicionamiento de cualquier robot industrial.
- Facilita la comprensión del movimiento de los robots en el espacio al mismo tiempo que se puede experimentar

diferentes rutinas que pueden crear los alumnos.

- Ayuda a afianzar el conocimiento en las áreas de cinemática inversa y directa
- Permite modelar un robot real partiendo de un conocimiento mínimo y superficial de la robótica
- Permite generar rutinas de movimientos tan complejas como se desee, simularlas y luego hacer que el robot las ejecute.
- Esta totalmente documentado lo que facilitará las futuras expansiones como visión artificial, control fuzzy entre otras.

Descripción

Documentación

El documento creado para la asignatura robótica y animatrónica esta dirigido a estudiantes de ingeniería de los cuales solo se requieren conocimientos básicos de álgebra lineal y robótica.

Aquí se explicaran comandos básicos de MatLab, conceptos importantes de robótica, el uso del toolbox de robótica para MatLab y uso del puerto serie para la comunicación con el robot.

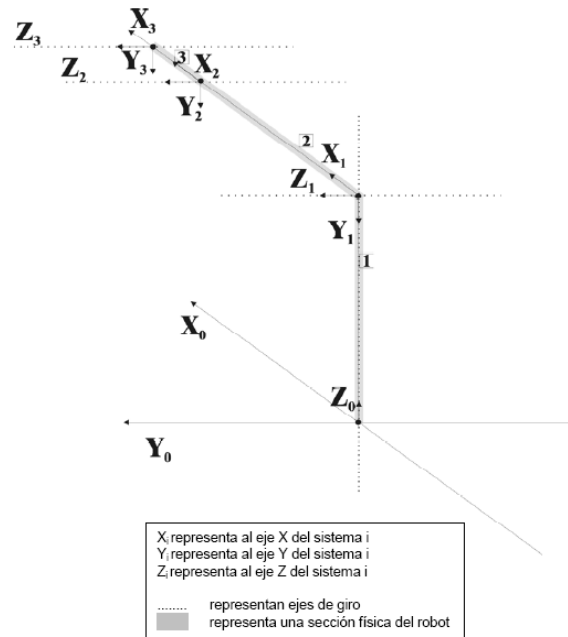
A lo largo del documento son citados una colección de archivos “.m” que muestran implementaciones de lo explicado.

Durante el desarrollo del texto se incluye información sobre como utilizar MatLab para las operaciones básicas con matrices y como se implementan los conceptos de robótica en esta maravillosa herramienta para ingeniería. Se comienza con una explicación del uso de matrices y funciones de MatLab, a continuación se introducen conceptos de robótica y se muestran aplicaciones de lo aprendido anteriormente. El mayor salto se da en la sección del toolbox donde se podrá aprovechar el potencial de MatLab mediante un amplio conjunto de instrucciones fáciles de utilizar.

En todo momento se acompaña al texto con código que puede ser ejecutado por el lector, este es destacado con una fuente diferente como se muestra a continuación.

>> código ejecutable en MatLab

Además, el documento describe las prácticas con el robot real (modelado, cinemática directa e inversa del robot Natali).



Modelo del robot Natali según el algoritmo de Denavit & Hartenberg

Por último, en el apéndice podrá encontrar las principales funciones de MatLab para comunicaciones serie RS232 y algunas de las funciones mas importantes del toolbox de robótica explicadas.

Software

El software utilizado es MatLab ya que es un poderosísimo entorno de trabajo para desarrollos que involucren álgebra lineal y representación grafica de resultados. Además su funcionalidad puede ser extendida mediante toolboxes específicos. Uno de ellos es el de robótica. Este no solo provee funciones para modelado de robots sino también para simular y analizar resultados con robots experimentales como el realizado en el trabajo. Aprovechando todo lo mencionado anteriormente se crearon los siguientes complementos para acompañar a la documentación y controlar el robot.

Rutinas utilizadas en la documentación

Archivo	Función
Cinemática-directa.m	en este archivo se resuelven problemas sencillos de

	cinemática directa
Cinematica-inversa.m	Resolución problemas sencillos de cinemática inversa
creando_rl.m	en este archivo se genera un objeto de tipo "robot" llamado rl, que es un robot de tipo cilíndrico.
robot1.m	crea natali y luego lo grafica en diferentes posiciones
Secuencia.m	Simula las secuencias Q1 y Q2 generadas en el archivo noname y realiza los movimientos usando el robot

longitud de las extremidades, límites de giro por soft, entre otras.

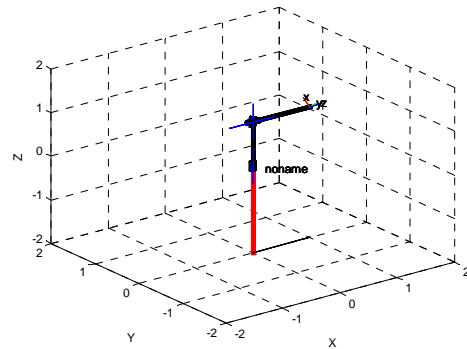


Imagen del modelo de robot Natali en matlab

Como se puede observar en el diagrama las funciones de mayor importancia son mover moversec y acond.

Funciones de control del robot

Archivo	Función
Accond.m	Permite adecuar los datos para poder enviarlos al driver
moversec.m	Mover para secuencias de movimientos
Secuencia.m	secuencias Q1 y Q2 generadas en el archivo noname y realiza los movimientos usando el robot

Mover permite enviar al robot los valores de referencia para el movimiento de cada motor a través del puerto serie.

Toma como parámetro a "q" que puede ser: la matriz de movimientos individuales de cada eje o la matriz de transformación homogénea que representa la posición del efector final del robot, Para que todo esto sea posible es necesario acondicionar los resultados a valores convenientes mediante la función acond, La importancia de la función acond se observa cuando q es una matriz de transformación y es necesario hacer la cinemática inversa, y algunas veces los valores de movimientos individuales que lo llevan a un lugar pueden ser mas de uno y algunos de ellos pueden no ser realizables, allí es donde interviene esta función dando como resultado solo movimientos realizables por el robot.

Mover sec toma una secuencia de puntos y utiliza la función mover para ejecutar cada punto de la secuencia MOVERSEC(SEC) SEC es una matriz de tres columnas por n filas

Arquitectura del software realizado:

Rutinas de Movimientos		
Mover		Moversec
API serial MatLab	Robotics toolbox	Acond
Sistema operativo		

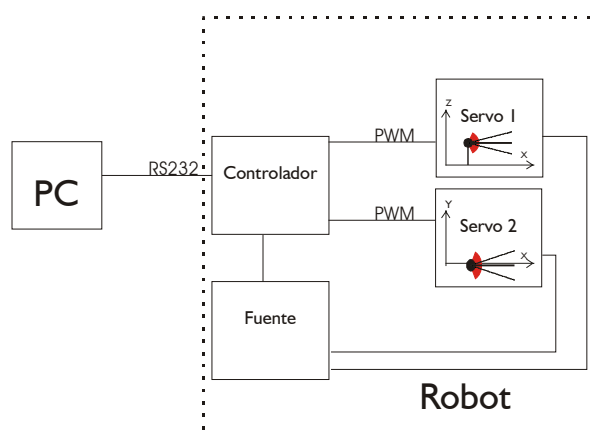
Todas las rutinas de movimientos utilizaran las funciones mover o moversec, que para funcionar utilizan tanto las funciones de la api de comunicaciones con el puerto serie como funciones del toolbox para graficas y otros calculos complejos como la función acond, que permite cambiar los parámetros del robot como

Además de los programas funciones para MatLab también se programo el microcontrolador pic del robot.

La programación del microcontrolador tiene un conjunto de funciones básicas de comunicación y configuración genéricas, que podrían considerarse como un conjunto de API para

PIC, en base a ellas se construyo el software de aplicación que envía a los servos el pwm que determinara la posición de cada articulación. Lo interesante da la programación no son los programas sino el método, al pic se le coloca este conjunto de instrucciones de manera tradicional, luego toda programación futura será transferida por el usart como en cualquier comunicación serial, solo que estando en modo programación, lo que se transfiere pasara a la memoria de programa para luego ser utilizada, el sistema del controlador podría considerarse como un mini ordenador, pues tiene su BIOS (el conjunto de APIs genéricas de entrada salida) y además vía puerto serie se le puede transferir programas y de a cuerdo a los programas que se transfieran podrá ejecutar comandos recibidos por el puerto.

Hardware



1 La Computadora

Esta es una computadora IBM. compatible que tiene MatLab y 3D Studio Max, el primer software, servirá para programar las trayectorias del robot "Natalí" tanto en forma virtual como en forma real, en la PC se generarán también los códigos para enviar a la Placa.

2 Controlador

Es el Hardware que permite interpretar los códigos que vienen de la PC (en RS232) de cada movimiento que se obtienen mediante MatLab y traducirlos en movimientos prácticos del Robot. Esta es la placa que convertirá el código en rutinas de movimiento para los actuadotes del robot "Natalí", o sea que tiene el

firmware desarrollado que permite este propósito.

Microcontrolador

Para el control de los servomotores se opto por el microcontrolador PIC16F628A, que presenta una gran ventaja costo-beneficio con respecto a los otros microcontroladores, aunque fácilmente se podría implementar en cualquiera de la gama media de microchip.

A su vez consta de dos importantes características, que es la de no necesitar un cristal externo ya que posee uno interno, lo que reduce la circuitería y además puede cambiar el contenido de la memoria de programa mediante código que se encuentre en dicha memoria, (puede escribir sus propios programas o recibirlos vía puerto serie como en este caso)

Comunicación a la PC

Se utilizo un circuito integrado MAX232 que tiene la función de convertir niveles de tensión TTL a RS-232. y viceversa. Esto se debe a que TTL utiliza de 0V a 5v y el protocolo RS-232 12V y -12V.

Interfase

La interfase se implemento mediante el protocolo RS-232, esto es muy simple utilizando el microcontrolador PIC16F628A, ya que el mismo tiene implementado en el silicio un USART que simplifica mucho la tarea del programador.

Las configuraciones del puerto de la PC deben ser una tasa de transferencia de 19200bps, sin bits de paridad, un BIT de stop y sin control de flujo.

La posiciones que se mencionan en la siguiente sección se refieren a la de los servo motores, estas pueden ir de 0 a 100, donde 0 es 180° y 100 es 0°.

Comandos que recibe el controlador

El robot puede ser comandado opcionalmente por teclado, esto fue utilizado en un primer momento para las pruebas del hard. Los comandos aceptados desde el teclado para el posicionamiento son:

'1' – Mueve el primer servo a la máxima posición izquierda.

'2' – Mueve el primer servo a la posición central.

'3' – Mueve el primer servo a la máxima posición derecha.

'q' – Mueve el segundo servo a la máxima posición derecha.

'w' – Mueve el segundo servo a la posición central.

'e' – Mueve el segundo servo a la máxima posición derecha.

'5' – Lo mueve ligeramente hacia la derecha.

'r' – Es la misma función que la '4' solamente que es para el servo 2.

't' – Es la misma función que la '5' solamente que es para el servo 2.

'8' – Esta función, útil para calibrar, devuelve la posición actual de los dos servos. La posición puede variar de 0 a 100, donde 0 es la posición máxima a la izquierda y 100 la máxima a la derecha.

':' – Este carácter se utiliza cuando se quiere mover cualquiera de los servos a una posición deseada.

Para utilizar esta función se debe utilizar el siguiente formato:

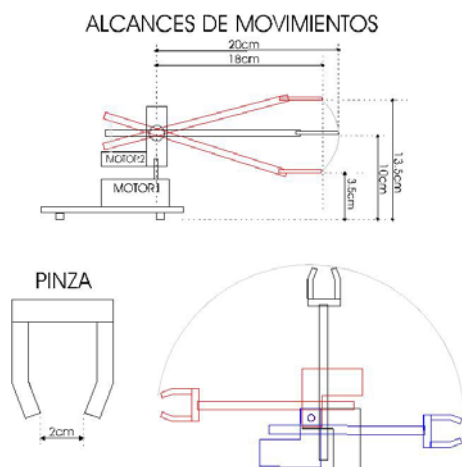
```
<STX> = ':'
```

```
<STX><ServoN><VAL_H><VAL_L>
```

donde <ServoN> es el numero de servo al cual se quiere enviar la nueva posición. <VAL_H> <VAL_L> es la posición a la cual se desea que se ubique el servo, este valor no puede ser menor a 0 ni mayor a 100 ya que son las posiciones límite. Si el valor recibido esta fuera de estos límites no será aceptado.

Este ejemplo posiciona al servo 2 en la posición 25: <:><2><2><5>

3 Robot "Natalí"



Es el robot elegido para el desarrollo de este proyecto, el robot es un producto de la firma

OWI (Robotic arm Y-01) y cuenta con dos grados de libertad (uno que rota el brazo y el otro que lo levanta) y una pinza como efector final. Cabe acotar que se han reemplazado los actuadores originales por servomotores tipo "futaba" para darle mas robustez al sistema mecánico

Servo motores

Para el control de los servos se utilizo un buffer tipo cmos cd40106, este sirve para proteger las salidas del microcontrolador de ruidos generados por los motores que las podrían dañar permanentemente.

Conclusiones:

Reviendo los objetivos planteados al proponer el presente proyecto, podemos llegar a concluir que:

- La selección de los software ha sido acertada puesto que tanto MatLab como 3D Studio Max han permitido realizar los objetivos.
- Se han estudiado y solucionado varios tipos de robots desde el punto de vista cinemático y aplicado los algoritmos de Denavit Hartenberg para cada caso
- Se han solucionado los problemas básicos de posicionamiento de un robot, en general y particularmente de "Natalí".
- Se han desarrollado y explicado rutinas en MatLab para diseño y posicionamiento de robots y graficado el efector final.
- Se ha desarrollado el hardware que comunica a MatLab con los actuadores del robot.
- Se han desarrollado rutinas básicas de movimiento del robot que respetan la posición del efector final tanto en el software realizado como en el robot real utilizado.
- Es evidente que el sistema realizado es útil para el aprendizaje de la robótica y la animatrónica en general y particularmente para la iniciación de la programación de robots y animatrónicos.
- Las animaciones realizadas facilitan el aprendizaje de la robótica.