



WIKI

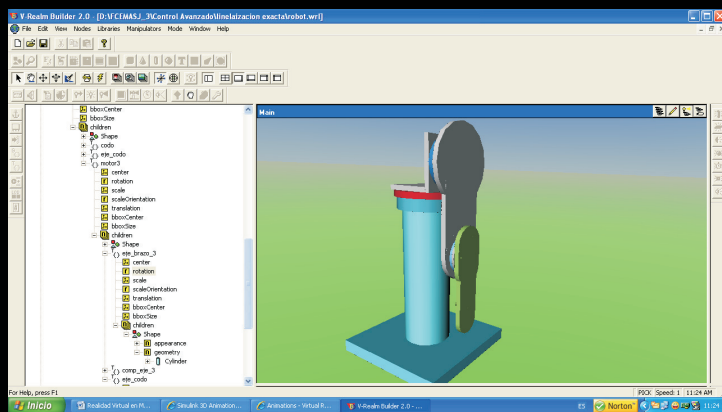
By Blasted Robotics FCE. BUAP.



Programador de PICs Brenner. USBurn.



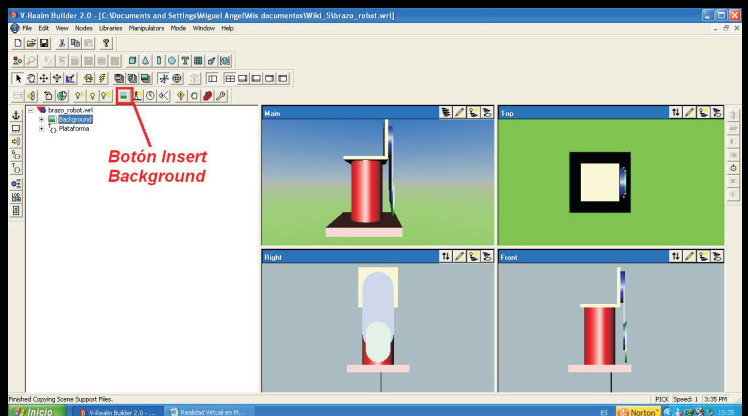
DTMF con
Microcontrolador PIC
Soldadura SMD.



Realidad Virtual en Matlab.

Configura el ADSL
en Linux.

Control no lineal de un
Levitador Magnético.



Índice.

El quemador de PIC Brenner.

USBurn.

3-6.

DTMF con microcontrolador.

(PIC ó AVR)

7-11.

Realidad Virtual en Matlab.

12-24.

Soldadura de componentes SMD

(*Surface Mount Device* –

Dispositivo de montaje

***superficial*) tipo “J” o “Gullwing”.**

25-26.

ADSL en Linux

27-32.

**Control no lineal de un levitador
magnético.**

33-47.

Trivia.

48-50.

Mensaje del editor.

La presente revista digital tiene como finalidad la divulgación y difusión de temas con información sobre proyectos electrónicos educativos. Los proyectos son escritos por alumnos de la Facultad de Ciencias de la Electrónica de la BUAP. De manera gratuita es como publicamos esta revista. El lector es libre de modificar los diseños presentados a su antojo, siempre y cuando sea con fines educativos y sin fines de lucro. Todos los proyectos que son publicados en esta revista fueron probados mínimo una vez y funcionaron correctamente al 100%. Se utilizan hardwares y softwares libres así como patentados con su respectiva licencia.

Los archivos y demos contenidos de esta revista, así como los del sitio web están debidamente permitidos para su reproducción.

En este número 5 de la revista electrónica Wiki, presentamos proyectos y prácticas que van desde la soldadura de componentes superficiales, circuitos para armar como un programador de PICs, probado y armado, un DTMF para uso en telefonía; también artículos como la realidad virtual con Simulink, muy bueno por cierto, la configuración del ADSL en Linux, y la teoría de un modelo no lineal de un levitador magnético. Todo esto para crecer más en conocimiento. Si gustas compartir un proyecto con la comunidad, mándalo, para que lo leamos. ;)

El quemador de PIC Brenner. USBurn.

Hace algún tiempo navegando en la red, me encontré con un foro creo que de HobbyPIC, y ahí mostraban los programadores de PICs mas sobresalientes, entre ellos la evolución del GTP_USB, que ahora ya no es libre, el Eclipse, que es muy bueno y programa también AVR's, y finalmente uno que se llama Brenner, creado por unos alemanes, me llamó la atención, abrí el link de la pagina de internet, y comencé a ver que programa una gran cantidad de microcontroladores de la marca Microchip, la mayoría, en este link pueden checar todo lo referente a software, hardware y firmware todo libre para armar su propio programador: <http://www.sprut.de/electronic/soft/usburn/usburn.htm>

El Brenner es un quemador de PIC que utiliza el software US-Burn, para programar códigos hexadecimales a familias 12, 16, 18, 24, 30 y 33, sorprendente! Además el

programador es interfaz USB, funciona en XP, Vista e incluso en Linux. Personalmente no lo he probado pero en la página de los autores hay un cromo demostrando que funciona en Windows 7. Programa también en PCs con micros AMD.

Existen 2 versiones de quemador, la 8 y la 9, en esta ocasión probamos la 8, ya que es la versión que programa los más utilizados, de 8, 14, 18, 28 y 40 pines, las versiones de 5V; la otra versión está dedicada solo a los micros que se alimentan a 3.3V.

El Brenner 8 programa la mayoría de los PICs comunes, dsPICs y familias de 16 bits.

Sin más, mostramos el circuito del programador en la figura 2, cabe señalar que en la página web del autor se encuentran más versiones como la SMD, o versiones para maquinado, y si de plano los PCB no son lo tuyo, viene una versión para placa universal.



Figura 1. Brenner 8 probado por los Blasted Robotics.

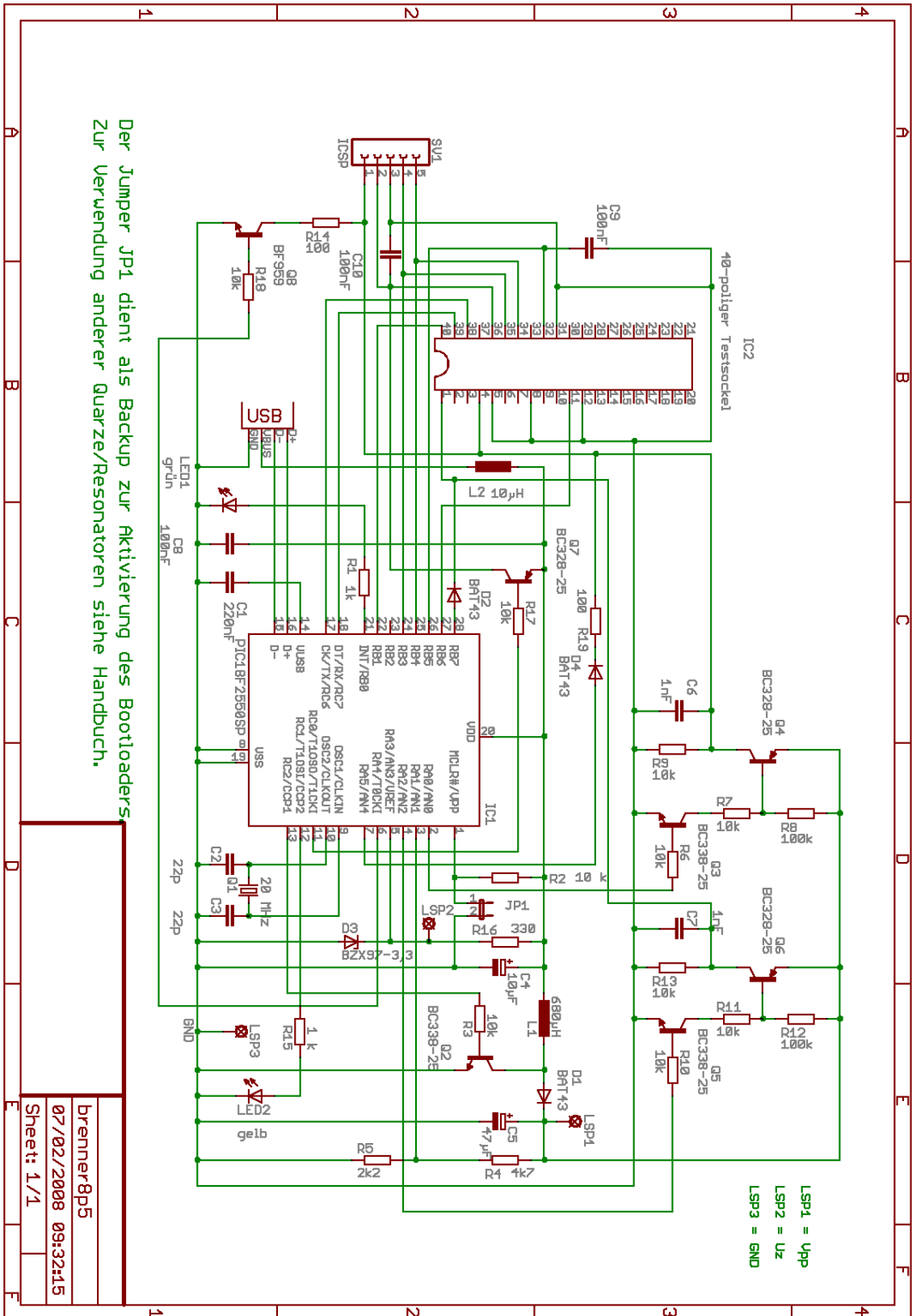


Figura 2. Esquemático del Brenner 8.

La mayoría de los componentes se consiguen fácilmente, excepto por la bobina de 680uH, pero esa se puede conseguir en placas de TV, fuentes de computadora o en tiendas nacionales.

La lista de componentes es la siguiente:

Lista de componentes para el Brenner8P R5:

Parte	Valor	Generico
C1	220nF	Capacitor 220n
C2, C3	22p	Cap ceramico 22p
C4	10µF	Cap electrolitico rad10/100
C5	47µF	Cap electrolitico rad47/35
C6, C7	1nF	X7R-2,5 1,0n
C8	100nF	Z5U-5 100n
C9	100nF	Z5U-5 100n
C10	100nF	Z5U-2,5 100n
D1, D2	BAT43	BAT 43 de ST
D3	BZX97-3,3	Zener 3,3V
D4	BAT43	BAT 43 de ST
IC1	PIC18F2550SP	PIC18F2550-I/SP
	28-pin base DIP	GS 28P-S
IC2	40-pin Zócalo ZIF	TEX 40
JP1	Jumper	Jumper 2,54 RT
L1	680µH	Bobina SMCC 680µ
L2	10µH	Bobina SMCC 10µ
LED1	Verde	LED5mm2MAGn
LED2	Amarillo	LED5mm2MAge
LSP1	Pin macho	Peine macho
LSP2	Pin macho	
LSP3	Pin macho	
Q1	20 MHz	Crystal cuarzo 20,0000-HC49U-S
Q2, Q3	BC338-25	Transistor BC338-25
Q4	BC328-25	BC328-25
Q5	BC338-25	BC338-25
Q6, Q7	BC328-25	BC328-25
Q8	BF959	Transistor BF959
R1	1k	1/4W 1K
R2, R3	10k	1/4W 10K
R4	4k7	1/4W 4,7K
R5	2k2	1/4W 2,2K
R6, R7	10k	1/4W 10K
R8	100k	1/4W 100K
R9..R11	10k	1/4W 10K
R12	100k	1/4W 100K
R13	10k	1/4W 10K
R14	100	1/4W 100
R15	1 k	1/4W 1K
R16	330	1/4W 330
R17, R18	10k	1/4W 10K
R19	100	1/4W 100
SV1	ICSP	Peine macho o conector BL 1X10G 2,54
X2	USB-B-H	Conector USB tipo B hembra USB BW

Una vez armado el circuito, se programa el PIC18F2550 ya sea con el “bootloader” (software que una vez programado dentro del microcontrolador, lo puedes programar cuantas veces quieras sin un quemador, sólo con el puerto USB o serie, que gran variedad de softwares ya disponen en su versión ICD) o directamente con el firmware.

Lo puedes probar conectando el cable USB a tu computadora, los dos LED harán un parpadeo, luego el ordenador reconocerá el dispositivo y pedirá la instalación del driver. En este caso se instaló y probó con éxito el driver 2k_xp_vista, de la carpeta del zip: usbur18a3.zip.

Instalado el driver y reconocido el dispositivo ya podemos usar el software

que viene en el zip, el software se llama `usb18a3.exe` y al ejecutarlo aparece el ambiente GUI.

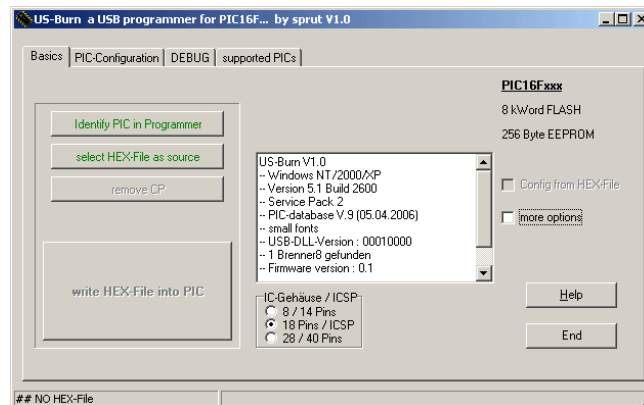


Figura 3. GUI del quemador USBurn 18a3.¹

Lo siguiente es mencionar que es muy fácil de usar, los controles vienen en inglés y trae pestañas de test de hardware para que puedas probar tu circuito, con tu propio multímetro por si no funcionase tu quemador.

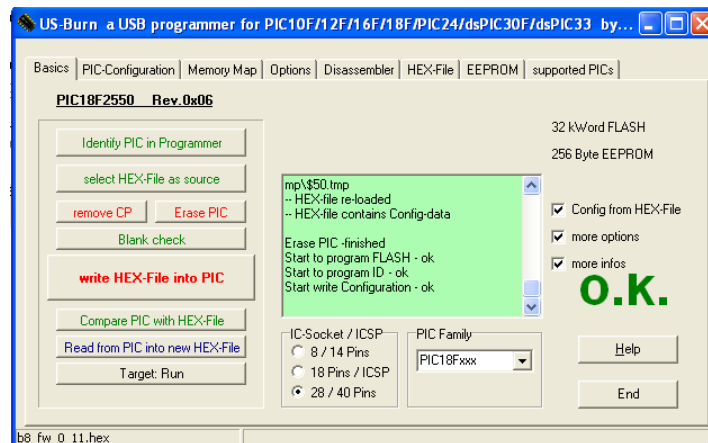


Figura 4. El software después de quemar un firmware en un PIC18F2550.

El firmware lo puedes descargar directamente de la página del autor o de la página de la revista, o pedirlo al correo, posteriormente publicaremos programadores de MCUs de 32 bits e híbridos para AVR's como el de microchip, cuyo código fuente y firmware son libres.

Conclusión: Este quemador es muy bueno, rápido, económico, confiable y poderoso. No dudes en armar el tuyo, es gratis para uso propio. En el próximo número publicaremos un quemador o grabador, comúnmente llamado por nosotros programador, de los clásicos y poderosos también AVR's diseñado por una persona de los foros de la web, probado por nosotros.

Editado por: Miguel Ángel San.

¹ Tomado de <http://www.sprut.de/electronic/soft/usburn/usburn.htm>

DTMF con microcontrolador. (PIC ó AVR)

Introducción.

El presente trabajo se presenta como práctica adicional a la materia Sistemas de Comunicación 2 del plan de estudios de electrónica de la FCE BUAP. Consiste en diseñar un sistema de marcado por cualquier medio tal que podamos enviar y recibir tonos DTMF, multifrecuencia de doble tono, por medio de un alambre, que puedan ser tecleados mediante un teclado matricial y visualizados a la salida con cualquier desplegador.

Marco teórico.

DTMF.

En telefonía, el sistema de **marcación por tonos**, también llamado sistema multifrecuencial o **DTMF** (Dual-Tone Multi-Frequency), consiste en lo siguiente: Cuando el usuario pulsa en el teclado de su teléfono la tecla correspondiente al dígito que quiere marcar, se envían dos tonos, de distinta frecuencia: Uno por columna y otro por fila en la que esté la tecla, que la central descodifica a través de filtros especiales, detectando instantáneamente que dígito se marcó.

La Marcación por tonos fue posible gracias al desarrollo de circuitos integrados que generan estos tonos desde el equipo terminal, consumiendo poca corriente de la red y sustituyendo el sistema mecánico de interrupción-conexión (el anticuado disco de marcar).

Este sistema supera al de marcación por pulsos por cuanto disminuye la posibilidad de errores de marcación, al no depender

de un dispositivo mecánico. Por otra parte es mucho más rápido ya que no hay que esperar tanto tiempo para que la central detecte las interrupciones, según el número marcado.

No obstante, las modernas centrales telefónicas de conmutación digital, controladas por ordenador, siguen admitiendo la conexión de terminales telefónicos con ambos tipos de marcación.

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Tabla 1. Relación entre frecuencias altas y bajas para el teclado telefónico.¹

Microcontrolador PIC.

Los '**PIC**' son una familia de microcontroladores tipo RISC fabricados por Microchip Technology Inc. y derivados del PIC1650, originalmente desarrollado por la división de microelectrónica de General Instruments.

El nombre actual no es un acrónimo. En realidad, el nombre completo es **PICmicro**, aunque generalmente se utiliza como *Peripheral Interface Controller* (Controlador de Interfaz Periférico).

CM8870PI.

El circuito integrado CM8870PI es un receptor integrado CMOS de tonos DTMF. Fabricado por California Micro Devices que usa un cristal de cuarzo o un resonador cerámico, el circuito integrado es capaz de trabajar en ambientes industriales, a

menos de 35mW de consumo de potencia con sus 18 pines en presentación DIP o 20 en presentación SMD. Este se utiliza en

aplicaciones de PABX, radio mobil, control remoto, sistemas de contestación telefónica, etc.

Desarrollo.

Para el diseño del sistema de transmisión, realizamos el siguiente diagrama a bloques que se muestra en la figura 1 y describe las partes que contendrá el sistema.

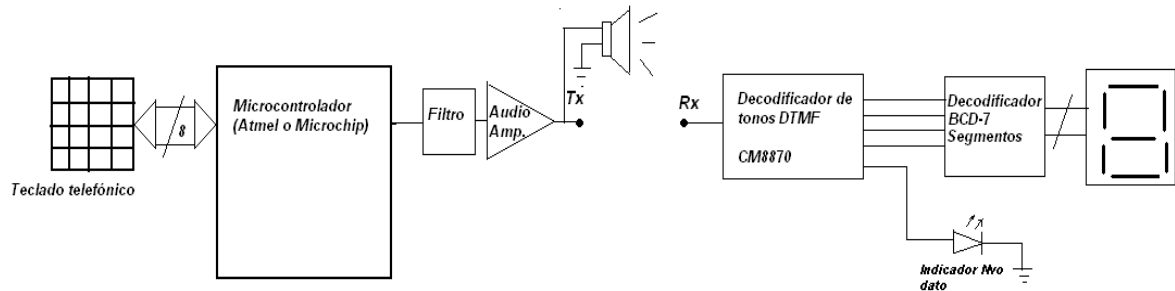


Figura 1. Diagrama a bloques del sistema.

Para la primera etapa, vamos a necesitar un microcontrolador PIC, cabe señalar que el primer diseño de este sistema se diseñó con uno de la marca Atmel, un atMega8, pero en general se puede desarrollar con cualquier otro microcontrolador, en especial con los de la marca Microchip, que son mucho más factibles de programar.

Transmisor.

Utilizamos un PIC16F877A programado en lenguaje Basic, con el compilador PicBasic Pro, para la lectura del teclado matricial y la generación de un PWM, "Pulse Width Modulation", que hará las veces del variador de voltaje en la salida A0 del microcontrolador, variando el ancho de pulso, variará la amplitud de voltaje con la respectiva señal correspondiente de la tecla que se pulso generando así la señal

DTMF a la salida de un filtro sencillo escalera que se conecto después del pin A0. Las frecuencias que se programaron para la generación de señal PWM no son de dominio público, por eso no se anotaron en este reporte, pero las podemos usar con fines educativos.

El diagrama eléctrico que se armó es el que se muestra en la figura 2, utilizamos un cristal de cuarzo de 20 MHz para un buen desempeño de la emisión del pulso, los valores para el filtro son de resistencias $R6=R7=1K\Omega$, y los capacitores C1 y C2 de 0.1uF. Las resistencias pull-up del teclado matricial son de 470 Ohms cada una, la polarización del PIC16F877A es la clásica mostrada en la hoja de datos del microcontrolador, que puede ser descargada gratuitamente.

Las rutinas DTMFout no son de dominio público por eso no son mostradas en este artículo, pero se pueden usar en el compilador de manera gratuita. Los retardos de duración del tono igualmente se pueden modificar al gusto. El programa se compiló con cero errores y se grabó en el PIC con uno de nuestros quemadores gratuitos Brenner de diseño alemán. La simulación se muestra en la figura 3, el simulador usado fue el Proteus profesional 7.2.

Detector.

Una vez quemado el PIC se armó el diagrama mostrado en la figura 2, y se probó con un simple sounder. El sonido que generó coincidió con los tonos deseados, tonos bastante similares a los

escuchados en un teléfono comercial, el ruido mostrado se puede filtrar cambiando los valores del filtro o diseñando un filtro activo de alto orden, también se puede agregar una etapa de potencia para conectar una bocinita o tener una mejor calidad de señal.

El detector se armó fácilmente con un circuito integrado CM8870PI que es bastante barato y de menor tamaño si se usa en su versión SMD. Otra ventaja de este integrado es su versatilidad, al ser CMOS, podemos trabajar con una mayor tolerancia en los niveles lógicos.

El diagrama se muestra en la figura 4, tomado de la hoja de especificaciones del circuito integrado.

Simulación y resultados experimentales.

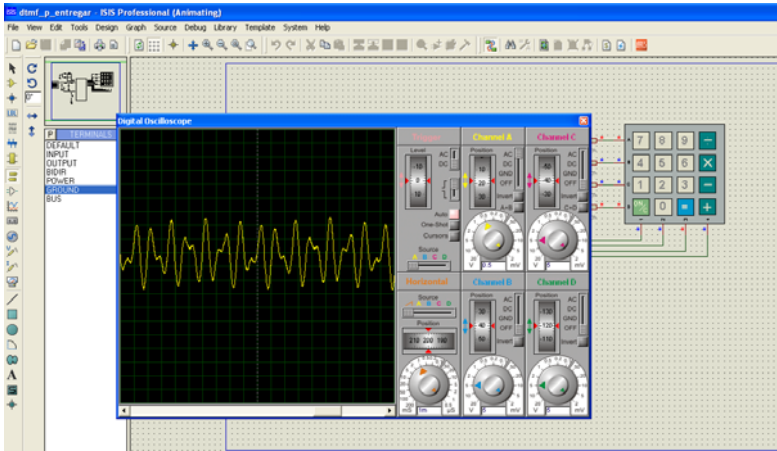


Figura 3. Simulación en Proteus.

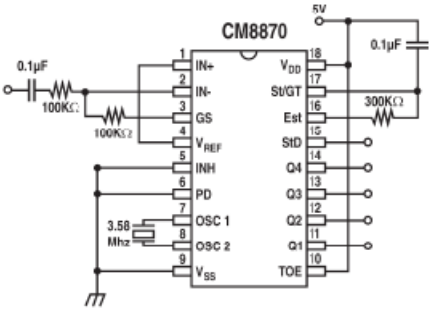


Figura 4. Conexión del CM8870PI.

Decodificador.

Una vez armado el circuito se conecta la salida de nuestro PIC al capacitor en la entrada del 8870 y ya se puede utilizar. En este diseño las salidas de Q1-Q4 entregan código BCD, y la terminal STD es el testigo que indica si ha llegado un nuevo dato, las salidas de código BCD fueron conectadas a otro PIC más pequeño usado como decodificador pero se pudo haber usado un simple 74HC47 o cualquier otro decodificador, simplemente para ser visualizado en un display de 7 segmentos.

Todos los bloques fueron probados uno por uno y armados en uno solo al final, como se muestra en la figura 1.

Conclusiones.

El circuito funcionó al 100% y correctamente al censar las teclas, emitir el sonido y decodificar el tono enviado. La figura 5 muestra una de las primeras pruebas hecha al circuito, aquí podemos notar que en versión SMD el tamaño se reduciría a una pequeña área del tamaño de una moneda de 10 pesos. La utilidad de este circuito puede ser de gran interés en el diseño de sistemas de marcación públicos más baratos, marcación autónoma en sistemas mínimos para emisión de mensajes de alerta en pozos de agua

Referencias.

Revistas Wiki publicadas en línea: <http://mx.geocities.com/blastedrobotics/> al 19 de julio de 2009. Publicada por el grupo estudiantil Blasted Robotics de la FCE. BUAP.

Pic microcontroller project book: for PICbasic and PICbasic por compilers. Iovine, John. McGraw-Hill. 2004.

Wikipedia the free encyclopedia. www.wikipedia.com

potable como algunos que ya hemos diseñado, por medio de telefonía celular, control de sistemas de domótica, espionaje de números marcados en líneas comerciales, que por cierto está prohibido en Inglaterra, en México posiblemente igual, en fin, este circuito puede echar a volar nuestra imaginación.

Se agradece al profesor encargado de la materia por proponer investigaciones de este tipo, por que pensamos que hace mucha falta explotar la creatividad y más la investigación de temas que en la actualidad son muy sencillos de hacer.

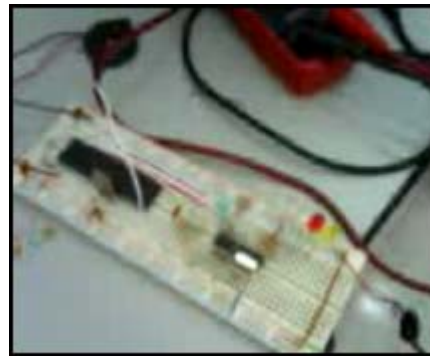


Figura 5. Primer prototipo probado (Aún sin la etapa del BCD-7Seg.)

Gracias por su comprensión y su fina atención.
;).

Autores: Miguel Ángel, Luis & Aarón.

ⁱ Obtenido de "http://es.wikipedia.org/wiki/Marcaci%C3%B3n_por_tonos"

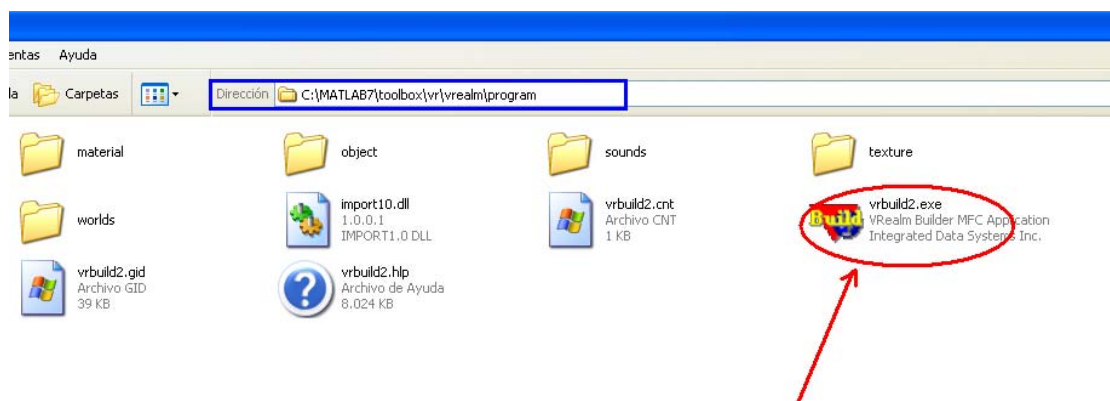
Realidad Virtual en Matlab.

Como lo define la enciclopedia libre, la **realidad virtual** es un sistema o interfaz informático que genera entornos sintéticos en tiempo real, representación de las cosas a través de medios electrónicos o representaciones de la realidad, una realidad ilusoria, pues se trata de una realidad perceptiva sin soporte objetivo, sin red extensa, ya que existe sólo dentro del ordenador. Por eso puede afirmarse que la realidad virtual es una pseudorrealidad alternativa, perceptivamente hablando.

En esta ocasión veremos como crear un ambiente de realidad virtual usando nuestro ordenador por medio de la poderosa herramienta Matlab, por medio de sus toolbox llamadas Virtual Reality y Simulink. Estas herramientas corren perfectamente en plataformas como Windows, Linux, Solaris y Mac. El formato que se utiliza para guardar estos ambientes es el .vrml, en Matlab se incluye un editor, llamado V-Realm Builder.

En las versiones más recientes de Matlab podemos importar modelos CAD desde SolidWorks y Pro/Engineer, o hacer links directos desde SimMechanics. Podemos editar en cualquier software que maneje el formato VRML97 standard como Cathia, Autodesk Inventor y varios más. [1]

Para poder simular la realidad virtual necesitamos tener instalado el Simulink, en este caso usamos la versión para Matlab 7, y tanto el visor como el editor de VRML tiene que ser instalado, los dibujos de CAD los vamos a hacer en V-Realm Builder, el ejecutable del builder lo encontramos en la carpeta raíz donde instalamos el Matlab, por ejemplo en C:\MATLAB7\toolbox\vr\vrealm\program donde veremos un acceso directo con el nombre de vrbuilder.exe, el cual podemos anclar al escritorio o a inicio para poder trabajar con él.



Creamos un acceso directo de este icono.

Figura 1. Localización del builder.

Ahora necesitamos instalar los plugins para Simulink, abrimos Matlab, y en la command window tecleamos las siguientes instrucciones.

- Para instalar el Plug-In para el viewer hay que seguir estos pasos:

1. Iniciamos Matlab.
2. En la Command Window, teclear

```
vrinstall -install viewer
```

Se despliega el siguiente mensaje

```
Do you want to use OpenGL or Direct3d
acceleration? (o/d)
```

3. Checamos el manual de nuestra tarjeta de graficos para determinar el metodo de aceleración. Si no estas seguro, selecciona Direct 3d tecleando

```
d
```

El instaler blaxxun comienza a correr y despliega el siguiente cuadro de dialogo:



Figura 2. Cuadro de diálogo del instalador blaxxun.

4. Sigue las instrucciones en los siguientes cuadros.
5. En la Command Window, escribe

```
vrinstall
```

```
-check
```

Si la instalación fue exitosa, se despliega el siguiente mensaje:

```
External VRML viewer: installed
```

Si no su exitosa se despliega el siguiente:

```
VRML
```

```
viewer: not installed
```

Para instalar el editor VRML hay que seguir estos pasos:

1. Iniciar Matlab.
2. En la Command Window, escribir
3. vrinstall

```
-install editor
```

O teclear

```
vrinstall('-install','editor')
```

Se despliega el siguiente mensaje:

```
Starting editor installation...
```

```
Done.
```

4. Tecleamos
5. vrinstall
-check

Si la instalación fue exitosa, se despliega el siguiente mensaje:

VRML editor: installed

Para más información de todas las toolbox y otras grandes aplicaciones es recomendable leer las application notes de Mathworks. [3]

Creando el dibujo.

Como ejemplo vamos a utilizar un brazo robot. Lo vamos a dibujar desde cero. Para ello hacemos doble clic en el acceso directo que creamos del builder, una vez dentro del programa veremos la interfaz donde dibujaremos. Hacemos clic en archivo/nuevo e inmediatamente nos percatamos de un espacio de trabajo CAD.

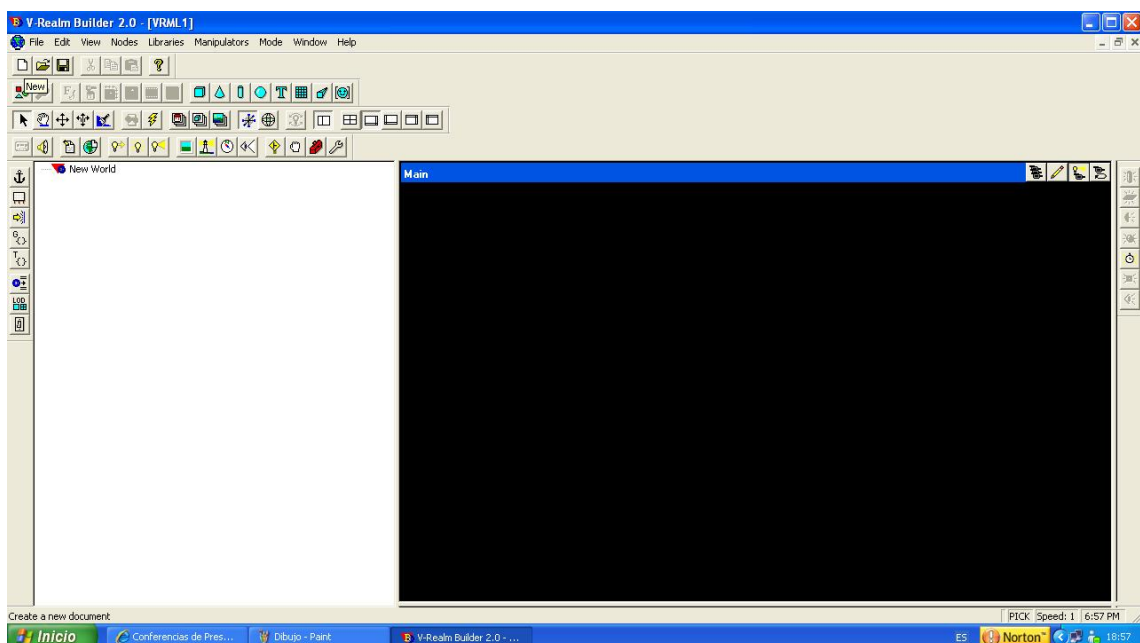


Figura 3. Interfaz de usuario tipo CAD del builder V Realm de Matlab.

Como primer paso guardamos nuestro dibujo con un nombre, para guardarlo hacemos clic en archivo/guardar como... ó en el ícono de guardar. Las funciones de menú aquí son las mismas que en cualquier otra interfaz ANSI.

Guardamos el archivo con el nombre brazo_robot.wrl, recordemos que la extensión en este programa es wrl.

Hacemos clic en el ícono de “insert box” que está situado en la barra de herramientas de sólidos en el espacio de trabajo.



Figura 4. Herramienta insert box.

Aparece un cubo y sus respectivos menús a la izquierda, en la figura 5 se muestran algunas descripciones de cada uno.

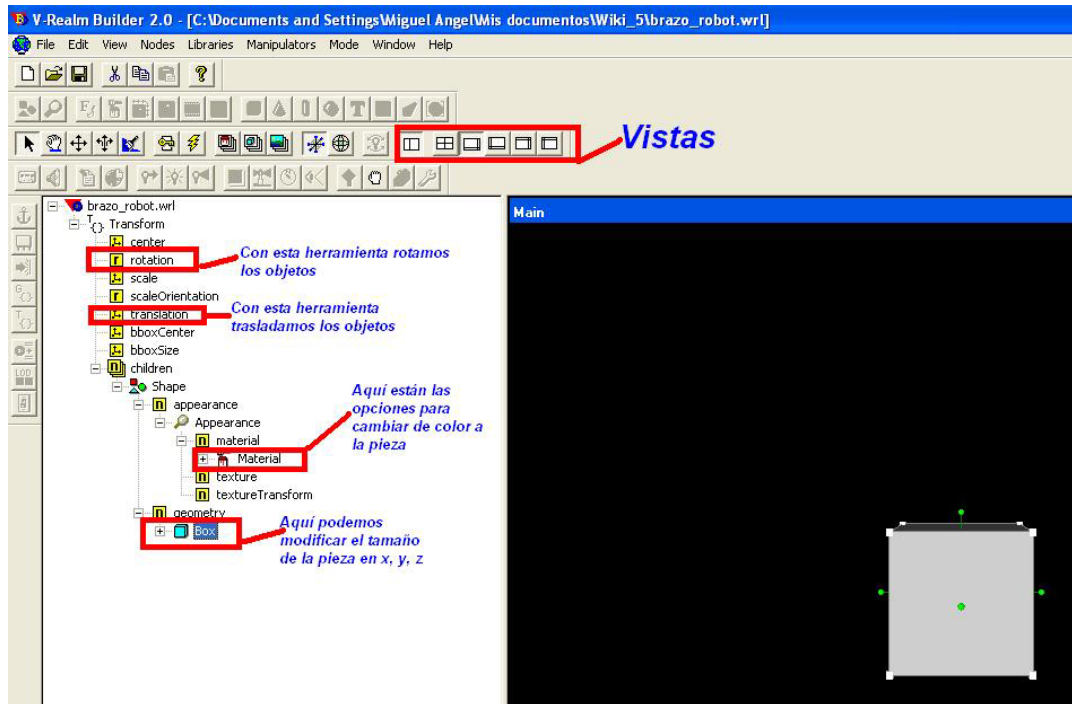


Figura 5. Inserción de una cubo.

Para dibujar por ejemplo una plataforma, hacemos doble clic en box, del menú contextual que aparece a la izquierda y permite cambiar de tamaño como lo dice la figura 5 y en "size" hacemos doble clic. Aparecerá un cuadro de diálogo con opciones, note que las unidades están en pulgadas y que seleccionando las check boxes podemos editar el tamaño de nuestra pieza.

La editamos al tamaño de 4 en x, 0.5 en y , 4 en z y hacemos clic en aceptar.

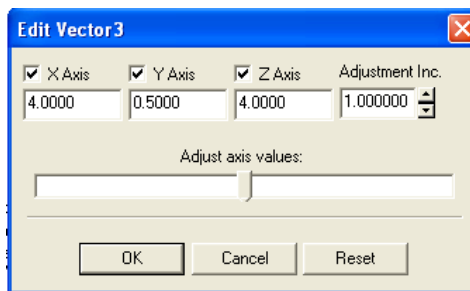


Figura 6. Cuadro de diálogo edición de tamaño.

Se dibujo ya nuestra plataforma base, ahora como dato importante la pieza que dibujamos queda en el centro de coordenadas de referencia, así que la podemos rotar o trasladar con base a este origen, como lo describe la figura 5.

La trasladaremos 4 unidades abajo haciendo doble clic a “translation” de los menús contextuales y seleccionando la coordenada “y” con el valor de -4.

Si se desaparece el dibujo es porque trasladamos demasiadas unidades, para verlo nuevamente hacemos clic en la opción “pan mode” de las herramientas de navegación. Después con un suave clic sostenido sobre el workspace negro vamos despacio hacia arriba o hacia abajo hasta encontrar la pieza, los movimientos bruscos pueden arrojar muy lejos las piezas, pero en el menú del workspace negro podemos hacer reset a la vista el en último ícono. También podemos hacer clic en el modo de 4 vistas y esto nos ayudará a navegar sobre el CAD fácilmente.

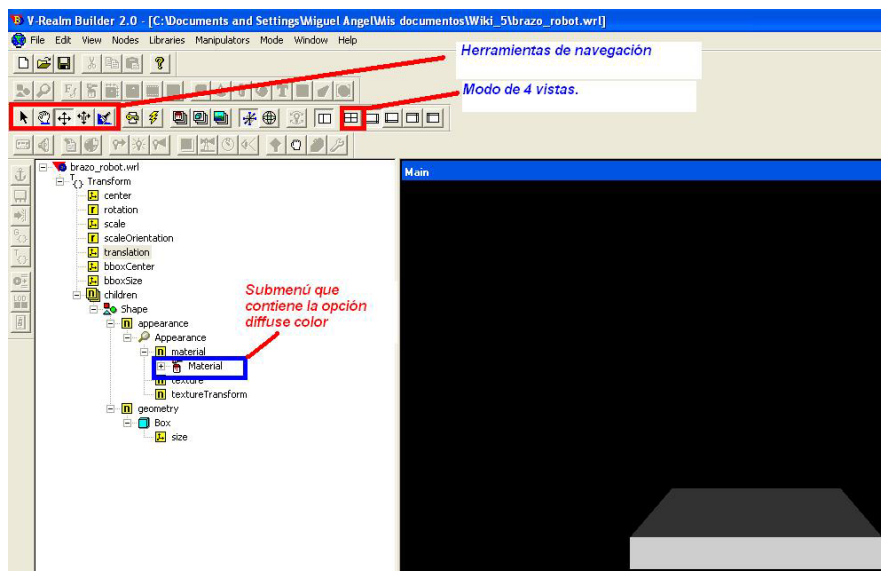


Figura 7. El cubo que dibujamos trasladado.

Para aplicar un color a la pieza hacemos clic a +Material y ahí encontraremos una serie de opciones para dar colores y texturas.

Por ejemplo doble clic en “diffuseColor” y elegir el color de nuestro antojo, o doble clic a material directamente y usar colores predefinidos en librerías.

Si queremos no perdernos podemos asignarle el nombre a nuestros dibujos, solo basta editar cada rama “transform” con un doble clic lento y cambiamos el nombre, por ejemplo a plataforma y el nombre de cada figura en su caso.

Ahora vamos a dibujar la base del robot, hacemos un clic a “children” y seleccionamos cualquier figura como “insert Cylinder” o cualquier otra, un children quiere decir que la figura que se inserte será parte de la misma estructura de la primera y más general, en

caso de dibujar otras estructuras sin relación con la primera el clic se hace en la rama más general.

Una vez dibujado el cilindro como children de la plataforma se dibuja con referencia de origen a ella, de no haber hecho clic en children el dibujo se habría referenciado al origen inicial.

Nuevamente podemos editar el radio, la altura y el color con las opciones de su menú contextual que aparece debajo de las del cubo. A esta figura le asignamos el nombre base.

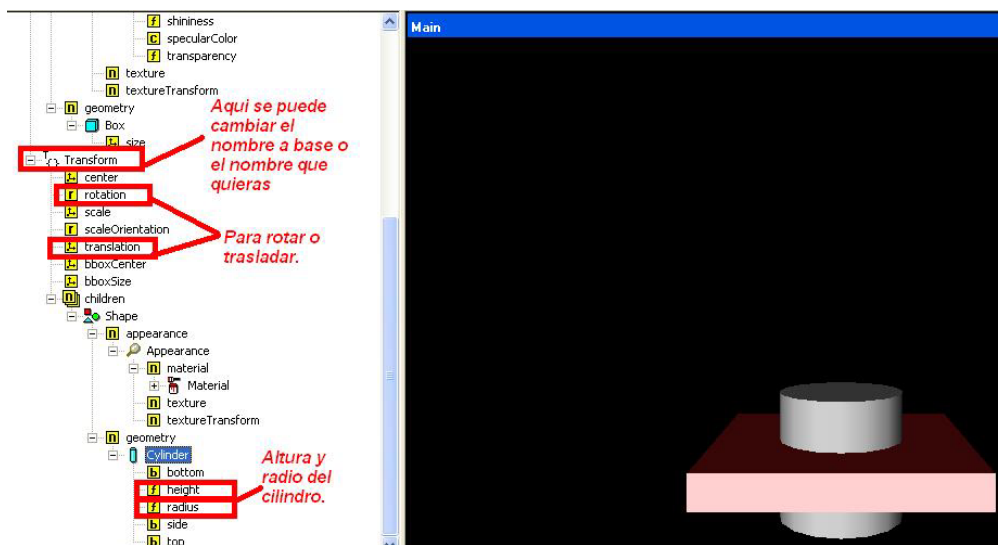


Figura 8. Herramientas de edición de cilindro, muy parecidas al de las otras piezas.

El radio de la base será de 1, y la altura de 4. Y con traslation lo trasladaremos 2 unidades en y.

Con un poco de práctica podremos construir el siguiente children, lo haremos en children de “base”, ya que esta parte es la que se moverá con la realidad virtual de Simulink. Será un cubo delgado para montarle un soporte plano de iguales medidas pero perpendicular es entre sí. Con medidas (2.5, 0.2, 2.5) en la opción “size” lo dibujamos y después lo trasladamos en y 2 unidades. Para una mejor referencia puede usar la opción de “Equalize view panes” o modo de cuatro vistas y las otras opciones de navegación suavemente para referencias como en cualquier otro editor CAD.

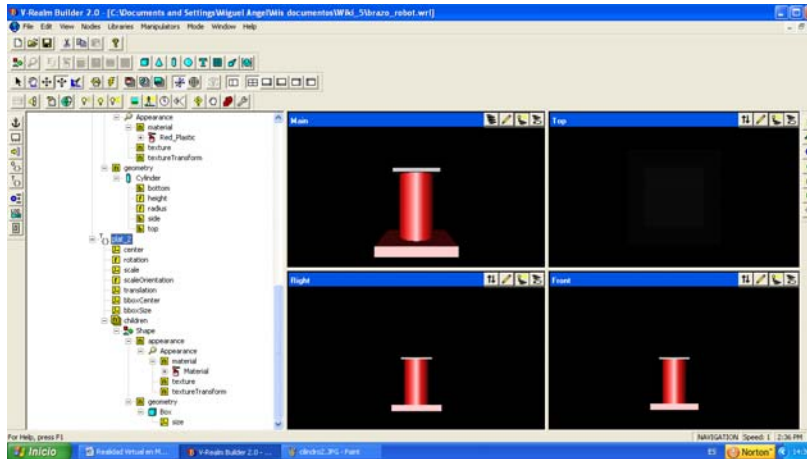


Figura 9. Cuatro vistas del dibujo, en esta parte deberían lucir así.

Ahora dibujamos la parte perpendicular a la pequeña base, hacemos exactamente el mismo dibujo, pero ahora en children de esta pieza que aquí le llamamos motor1 porque hará las veces de un motor, sólo que ahora usaremos la opción “rotation”, haciendo un doble clic y seleccionando la pestaña rotation, anotamos 90 unidades, después seleccionamos la rotación en z con 1 unidad. Lo trasladamos 1.2 en x, 1.2 en y, 0 en z y la nombramos base2. Guardamos el dibujo. Hasta este momento el dibujo deberá parecer al de la figura 10.

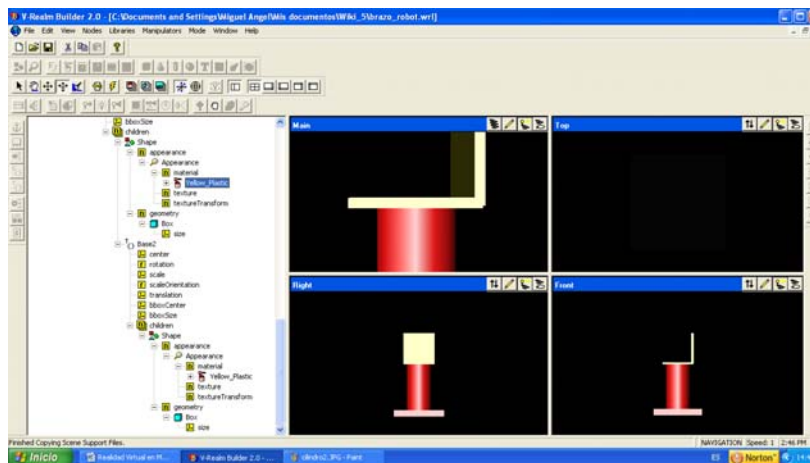


Figura 10. Primera parte terminada del dibujo.

Para los eslabones dibujamos un children sobre la última pieza que creamos, nosotros le llamamos base2, este children lo llamaremos motor2, porque hará las veces del segundo motor. Sus medidas serán radio 1 y altura 0.3. Lo trasladaremos en y -0.26 unidades.

Lo uniremos con un eslabón de forma de prisma como children de motor2. Con medidas (3, 0.1, 2) en (x, y, z) respectivamente, note que las referencias varían en relación a cada nueva referencia. Trasladamos -1.5, -0.1, 0 y colocamos otro children.

El nuevo children no importa si es de motor2 o de eslabon ya que solo es para que se vea bien el dibujo., tendrá las mismas medidas que motor2 pero trasladado en -3, -0.1, 0.

Tendremos ahora el primer eslabón terminado. A esta pieza la llamamos fin_eslabon.

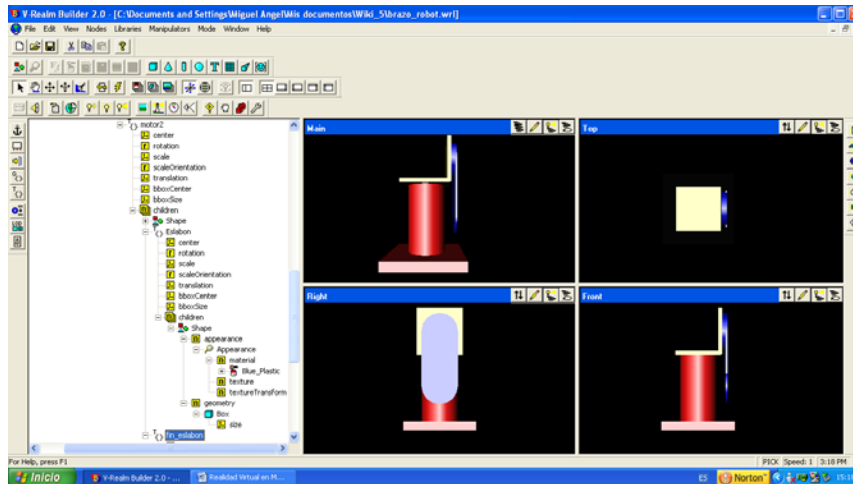


Figura 11. Dibujo del primer eslabón.

Por último haremos exactamente el mismo procedimiento para dibujar el segundo eslabón, sobre fin_eslabon dibujamos un children con la herramienta insert cylinder. Ahora de radio 0.8 y de altura 0.1. Este se llamará motor3.

Lo trasladamos en y 0.1 unidades, sobre él construimos un children de forma de prisma pequeño y le damos forma de eslabón con medidas (1, 0.1, 1.6) y trasladado -0.5 en x. A esta parte la llamamos eslabon2. Posteriormente veremos porqué la importancia de nombrar a cada parte del dibujo.

Por último para que nuestro robot se vea bien dibujamos el último cilindro como children de motor3, con las mismas medidas que este pero trasladado -1 unidad en x. Adicionalmente podemos insertar un background haciendo un clic en la rama principal brazo_robot y clic en el botón inset background.

Guardamos nuestro dibujo y cerramos el programa, estamos listos para la mejor parte, darle movimiento al robot. Tenemos ya este dibujo en formato .wrl que es un CAD diseñado para soportar realidad virtual.

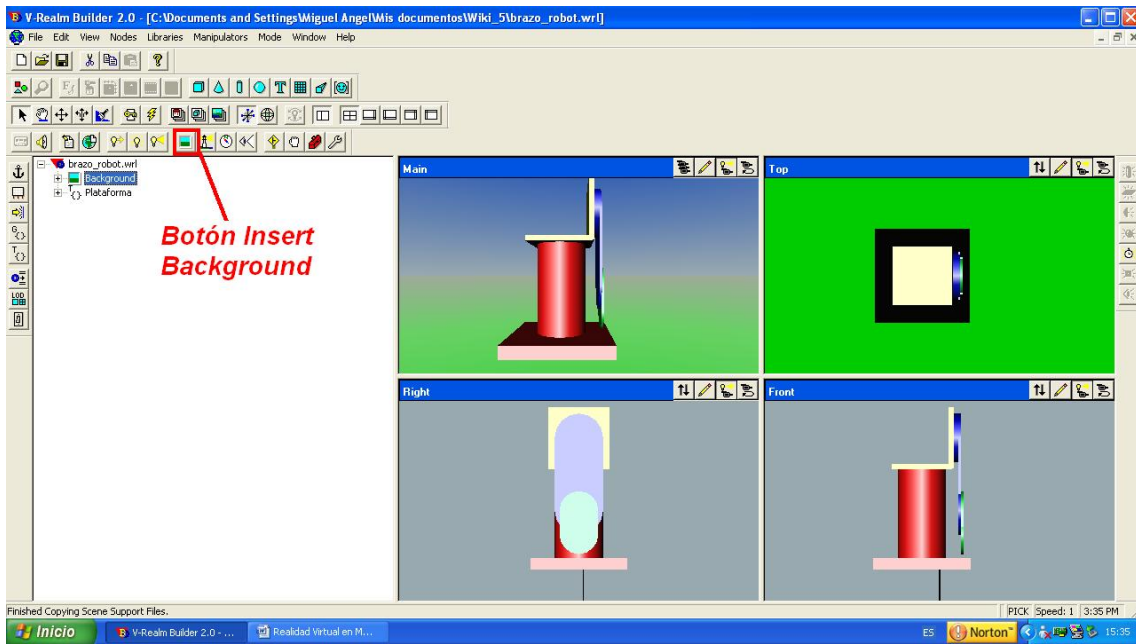


Figura 12. Dibujo CAD terminado.

Moviendo el diseño CAD WRL con Simulink Virtual Reality Toolbox.

Abrimos Matlab y después Simulink, para este diseño tenemos una ley de control ya hecha para un péndulo, dibujada a bloques en Simulink, ingresaremos el movimiento a los otros eslabones con entradas rampa o cosenos. También podemos utilizar archivos de Simulink demos del mismo Matlab. Puede dibujar sus propias leyes de control o si lo prefiere usar los demos de HIDs, si usted es usuario novato, puede ingresar solo valores constantes o entradas escalón.

Una vez en Simulink, dibujamos nuestros bloques o leyes de control para la salida que deseamos mover, por ejemplo la implementación del diagrama a bloques de la figura 13.

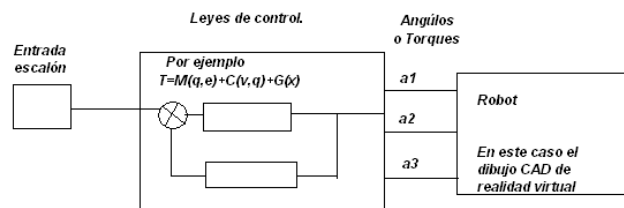


Figura 13. Sencillo sistema a bloques de un control de un robot de 3 grados de libertad, sólo para bosquejo del ejemplo.

En Simulink tomamos un control no lineal de un péndulo y 2 valores constantes. (Note que no se involucra ninguna ley de control de robótica.)

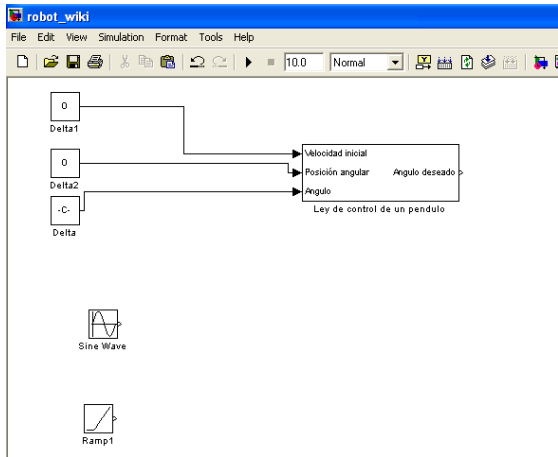


Figura 14. Bloques de Simulink para este ejemplo.

Ahora en la barra de herramientas Browser de Simulink hacemos clic en Virtual Realty Toolbox, (En este caso se utilizó una licencia original de una Universidad). Ahí arrastramos la herramienta VR Sink, hacemos doble clic, y en el cuadro de diálogo que aparece en browse elegimos la ruta de donde guardamos nuestro diseño CAD del robot. También para ver la simulación más lento podemos decrementar el tiempo de muestreo como se muestra en la figura 16, en este caso anotamos 0.01.

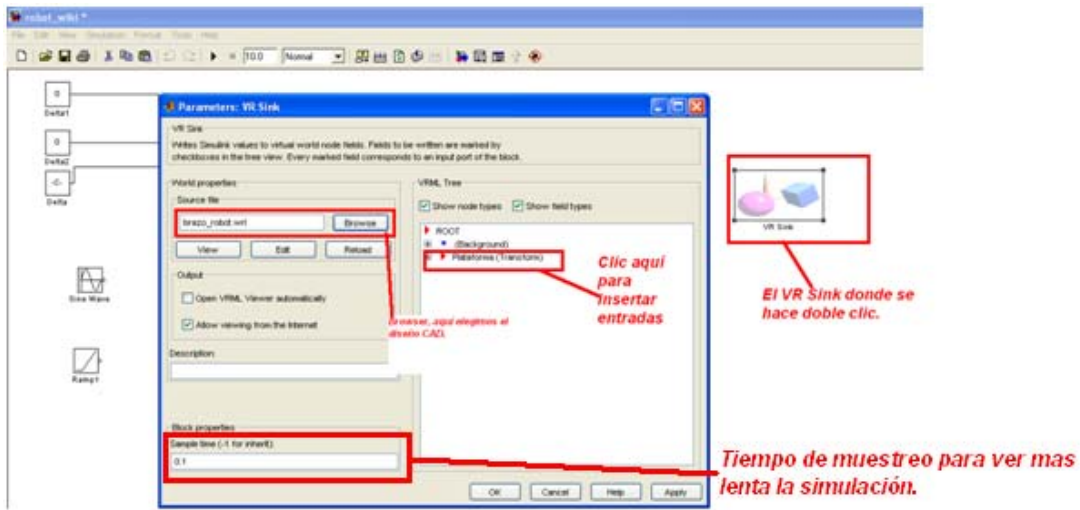


Figura 16. Seleccionando el diseño CAD para VR Sink.

Hacemos clic en la pestaña plataforma y a cada children, aquí viene la importancia de haber nombrado anteriormente cada pieza, los movimientos se van a dar a motor1, motor2 y motor 3, por ello seleccionamos con una palomita estas 3 casillas. Hacemos clic en ok y tendremos el VR Sink con 3 entradas.

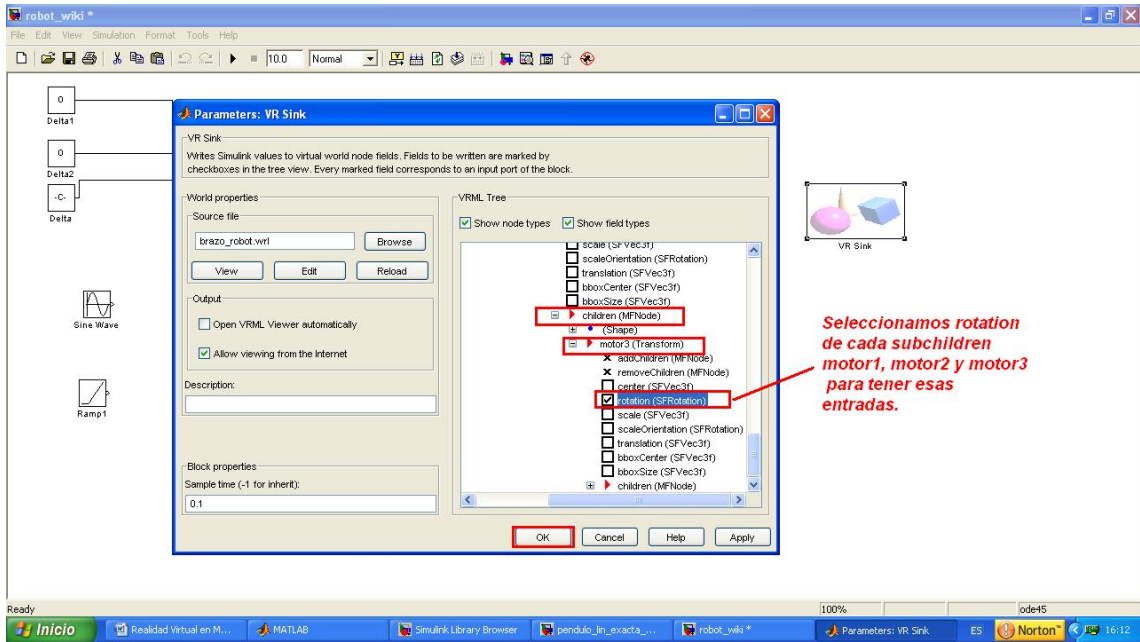


Figura 17. Seleccionando las entradas del VR Sink.

Como vamos a manejar ángulos, necesitamos dar entradas al VR Sink en coordenadas de ángulos pero en forma “CUATERNIÓN” si usted lee esa información en la red verá que podemos agregar un multiplexor de 4 entradas para unir la coordenada de giro en (x, y, z) y la magnitud de ángulo para conectar al VR Sink.

El diagrama abloques final para cada cuaternion queda como en la figura 18.

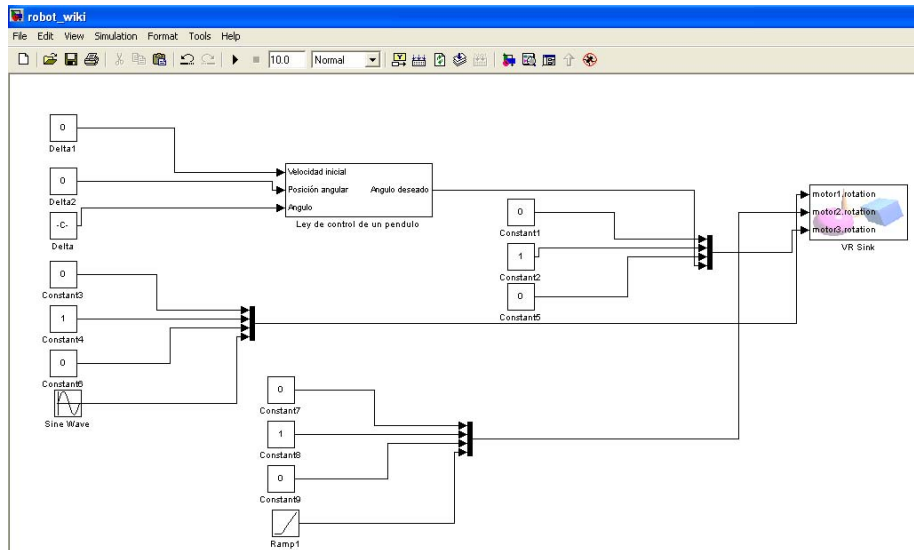


Figura 18. Diagrama a bloques de Simulink final con cuaterniones.

Ahora hacemos doble clic en el VR Sink. Wow! Aparece el ambiente de realidad virtual con nuestro robot. Simplemente basta hacer un clic en “play” y el robot responderá a los movimientos que Simulink le ordena.

Si usted cuenta con una ley de control fuerte puede simularla en realidad virtual. Puede generara videos y demás, y la s versiones de Matlab contienen demos fáciles que le ayudarán a aprender más.

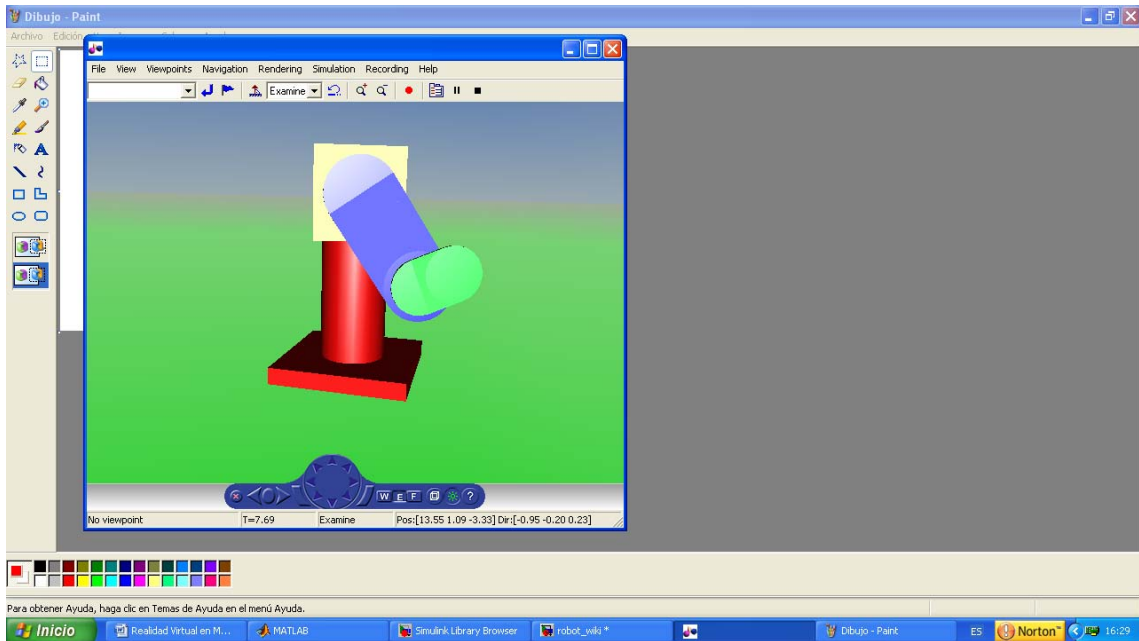


Figura 19. Simulación corriendo.

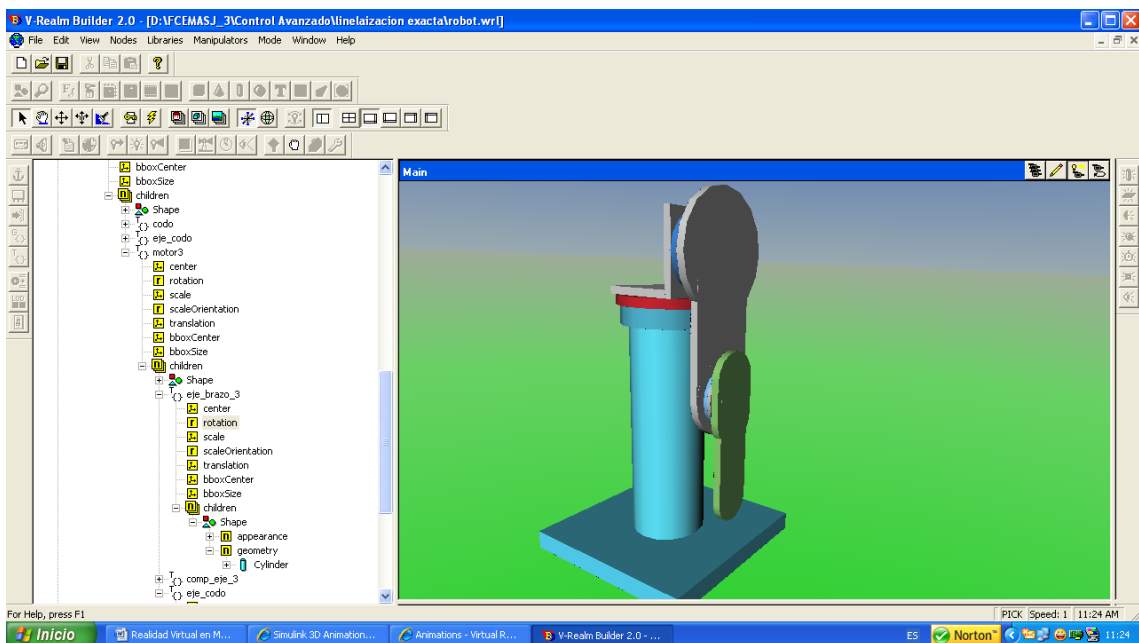
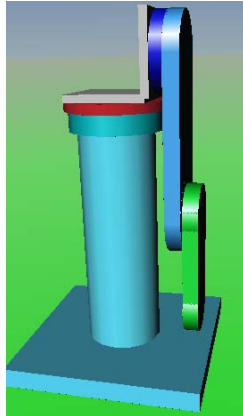


Figura 20. Robots más detallados.

Ejemplo hecho para la FCE. →

Autor: Miguel Ángel San Pablo J. blastedrobotics@yahoo.com.mx



Referencias:

[1]<http://www.mathworks.com/products/3d-animation/description1.html>

[2]<http://www.mathworks.com/access/helpdesk/help/toolbox/sl3d/index.html?access/helpdesk/help/toolbox/sl3d&http://www.mathworks.com/support/product/product.html?product=VR>

[3]<http://www.mathworks.com/access/helpdesk/help/helpdesk.html>

Soldadura de componentes SMD (*Surface Mount Device –Dispositivo de montaje superficial*) tipo “J” o “Gullwing”.

En la actualidad los circuitos integrados son cada vez más pequeños y ya sea por cuestiones de espacio o de tecnología alguna vez necesitaremos soldar componentes superficiales como algunos microcontroladores, TTL etc., en fin si tienes un dispositivo de este tipo escribimos unos tips para que lo puedas soldar sin problemas.

En la soldadura SMD debemos tomar en consideración que existen varias técnicas, una de las mas usadas es el montaje de un cautín de alambre de cobre sobre el cautín normal, el alambre de cobre se corta en forma de filo y se enreda para usarlo como

mini cautín, esto en caso de que la punta este demasiado deteriorada. Otra es la de usar un cautín pequeñito casi nuevo con punta afilada, o tener un cautín con punta de “ola”.

En cualquier caso necesitaremos comprar liquido “flux” para que el estaño fluya fácilmente sobre las terminales del dispositivo sin derramarse o pegarse; necesitamos un cautín de punta apta para este caso, como lo mencionamos anteriormente, y un poco de interés para practicarlo. A continuación escribimos unos sencillos pasos para soldar componentes SMD.

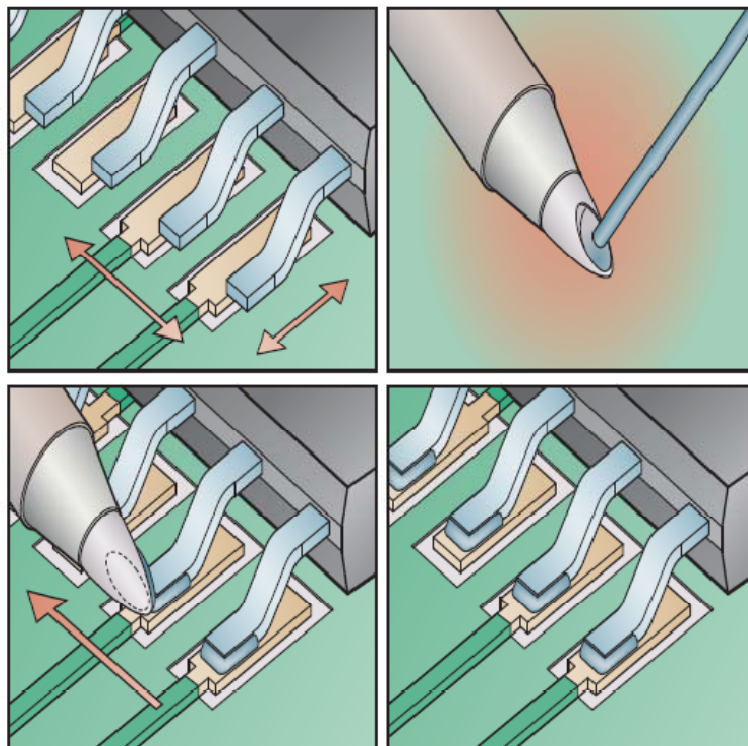


Figura 1. Soldado SMD.ⁱ

¿Qué hacer?ⁱⁱ

1. Aplique un poco de flux en las terminales de circuito impreso, en las esquinas de las huellas de los circuitos integrados.
2. Coloque el circuito integrado en su huella correspondiente, cuidando de alinear las terminales en las esquinas, en forma de diagonal, de acuerdo con los terminales colocados en la placa de circuito impreso.

3. Sitúe la punta de ola en la unión entre uno de los terminales que hace esquina del circuito integrado y su terminal correspondiente del circuito impreso. Si fuese necesario añada algo más de flux.

4. Retire la punta de soldadura. Esperaremos a que el estaño se solidifique y, a continuación, soldaremos la terminal correspondiente opuesta en la diagonal.

5. Coloque algo más de estaño en la punta de ola para crear una pequeña bañera.

6. Mantenga la punta de soldadura de manera que la bañera de la punta esté en contacto con la superficie superior de las terminales del circuito integrado. Lentamente, con un movimiento suave, desplace la punta del cautín a través de la fila de terminales, de manera que a cada terminal se aplique una cantidad uniforme de estaño.

7. Repita los pasos 5 y 6 sobre la(s) otra(s) cara(s) del circuito integrado.

8. Si fuese necesario, limpie e inspeccione la soldadura. El método que acabamos de describir también se puede aplicar a los circuitos integrados con terminales "J".

La ventaja del método propuesto es que no tenemos porqué añadir estaño a la punta de cada terminal del circuito integrado. El vertido del estaño que hay en la "bañera" a la punta del soldador está basado en el principio de la física denominado "acción capilar". Debido a que cada terminal de soldadura de la superficie del circuito impreso y cada terminal del circuito integrado tiene una relación de superficie idéntica, cada contacto atrae una cantidad idéntica de estaño, de manera que el estaño sobrante será rechazado hacia la punta del soldador. Además, el estaño también es atraído por una fuente caliente que, en nuestro caso, es la propia punta del soldador. El proceso que acabamos de describir

solamente funciona si existe la suficiente cantidad de flux en los terminales de soldadura. Después de todo, el estaño no hace nada excepto derretirse durante el calentamiento. Sin embargo, para obtener mejores resultados, el estaño también debería fluir correctamente.

El cobre, el estaño y el plomo son metales "porosos" que se protegen a sí mismos de la corrosión formando una fina capa de óxido. Agentes como el flux de soldadura producen un efecto químico que remueve las capas de óxido. Se recomienda utilizar fluxes de estaño basados en resina, ya que son más fáciles de retirar usando limpiadores de flux baratos basados en alcohol. Hay una gran cantidad de fabricantes de puntas de soldador para los métodos de soldadura de flujo continuo, como pueden ser JBC, Xytronic, Metcal, Sëller y Pace.

ⁱ Cortesía del Circuit Technology Center.

ⁱⁱ Elektor Electronics Worldwide.

Como configurar este tipo de conexión en banda ancha

ADSL en Linux >> (parte 1)

Trataremos un problema que aqueja a muchos usuarios de servicios de banda ancha. ¿No saben como configurar la conexión? En esta nota les contamos la manera de hacerlo.

Tal como todos esperábamos, el paso del tiempo y el aumento de los servicios disponibles en lo que respecta a Internet han producido un cambio importante en las conexiones típicas que tenemos en nuestra casa, oficina o empresa. La tecnología DSL (*Digital Subscriber Line*) es cada día más accesible, por lo que se volvió una sana costumbre entre los consumidores.

Todo viene de maravillas: lo pensaste, contrataste DSL, te lo configuraron e incluso anda. Pero ¿qué pasó con Tux? ¿No funciona DSL bajo GNU/Linux? ¡Claro que lo hace! Y muy bien.

Esta es la primera de dos notas sobre conexiones DSL en GNU/Linux. En esta entrega conoceremos algunos aspectos básicos acerca de cómo configurar el kernel para que soporte nuestra placa de red Ethernet, y cómo dejar andando la conexión **ADSL** usando las herramientas que nos ofrece GNU/Linux. En la próxima publicación

hablaremos sobre módems USB y su configuración.

MODEMS ETHERNET

Los módems Ethernet son aquellos que se conectan a la placa de red de la computadora. Antes de centrarnos en la configuración del servicio de ADSL, es imprescindible que configuremos la placa bajo GNU/Linux. Linux ofrece un buen soporte para todo lo que respecta a redes. Por ende, lo más probable es que la placa de red de nuestra máquina funcione correctamente en dicho sistema operativo. Para saber si está activa en el sistema y es perfectamente reconocida, veamos si existe para nuestro querido Tux. Este es el log que nos ofrece el kernel:

```
root@nixbox# cat /var/log/messages | grep eth
```

Y la salida será algo como:

```
May 24 22:03:08 nixbox kernel: forcedeth.c: Reverse Engineered nForce ethernet driver. Version 0.25.
```

```
May 24 22:03:08 nixbox kernel: eth0: forcedeth.c: subsystem: 01565:2301 bound to 00:04.0
```

```
May 24 22:03:08 nixbox logger: /etc/rc.d/rc.inet1: /sbin/dhccpd -d -t 10 -d eth0
```

Estas líneas del **log** indican que en el kernel hemos compilado el driver adecuado que nos brinda soporte para nuestra placa de red. Además, es bueno señalar que dicho driver se carga automáticamente en el sistema cuando éste se inicia.

Independientemente de la distribución que usemos y del modelo de placa de red que tengamos, la salida antes exhibida puede diferir simplemente en el nombre de la placa de red o en lo que se refiere a la carga automática del driver (línea correspondiente

al llamado de un script ubicado en el directorio `/etc/rc.d/`).

Noten que escribí `grep eth` y no `grep eth0` para consultar por la placa. Entiendan que `eth0` es el dispositivo asignado en `/dev/` a la primera placa de red, por lo que la segunda va a ser `eth1`, la tercera `eth2`, etc.). De esta manera, si tenemos más de una placa instalada en el sistema, podremos verlas a todas funcionando, o no.

Si todo salió como esperábamos, significa que la placa de red está funcionando correctamente; en este caso, podemos obviar las siguientes líneas. Si aún no es así, continuamos leyendo. Puede pasar que la placa de red no esté configurada en el kernel, algo altamente probable si veníamos usando una versión del kernel compilada por nosotros mismos, y nunca habíamos necesitado una placa de este tipo. También puede ser que hayamos instalado la placa de red en la máquina tiempo después de haber instalado GNU/Linux, o que la versión del kernel instalada en el sistema no ofrezca soporte para nuestra nueva placa de red. En cualquiera de estos casos, vamos a solucionarlo.

El primer paso es obtener el código fuente de la última versión del kernel en www.kernel.org. Hay que tratar de conseguir la versión más reciente de todas las disponibles, ya que seguramente nos será más provechosa si estamos buscando obtener soporte para una placa de red que la

versión del kernel instalado actualmente en nuestra máquina no ofrezca.

Aunque no necesitemos de un nuevo driver, siempre es mejor usar un kernel actualizado. Una vez que descargamos el código fuente, los pasos para compilar el kernel son los siguientes:

```
root@nixbox# gzip -d linux-2.6.x.tar.gz
root@nixbox# tar -xvf linux-2.6.x.tar
root@nixbox# cd linux-2.6.x
root@nixbox# make menuconfig
```

En este momento, obtendremos un menú basado en curses que nos permitirá configurar todo el kernel a nuestro gusto. Cabe notar que es imprescindible, y a la vez, obvio, seleccionar el soporte para la placa de red que tenemos en nuestra máquina, independientemente de que sea una placa onboard (muy común en los mothers más nuevos) o una placa de red PCI, sólo por citar un ejemplo.

Asegurémonos de conseguir con exactitud el modelo de la placa, y hagamos la configuración efectiva. Entre las opciones que nos da `menuconfig` (en este ejemplo usamos un kernel de la serie 2.6.x), vamos a **Device Drivers / Networking support / Ethernet (10 or 100 Mbit)**. Una vez allí, seleccionamos “[*] Ethernet (10 or 100 Mbit)” y la placa de red en cuestión. En mi caso:

```
[*] EISA, VLB, PCI and on board controllers
< > AMD PCnet32 PCI support
< > AMD 8111 (new PCI lance) support
< > Adaptec Starfire/DuraLAN support
< > Broadcom 4400 ethernet support (EXPERIMENTAL)
<*> Reverse Engineered nForce Ethernet support (EXPERIMENTAL)
```

Luego de seleccionar el driver necesario para nuestra placa de red, salimos, y dentro de **Device Drivers/Networking support**, habilitamos el soporte para PPP, en caso de que aún no lo hayamos hecho. Es casi imposible no disponer del soporte PPP incluido en el kernel default de nuestra distribución. Aun así, supongamos que no lo tenemos. Entonces, lo seleccionamos dentro del menú correspondiente y lo incluimos en el propio kernel o bien como un módulo.

Ahora compilamos la imagen correspondiente:

```
# make clean
# make dep
# make bzImage
# make modules
# make modules_install
```

Puede ocurrir que el **make clean** y **make dep** no sean necesarios; de todos modos, intenten y vean qué pasa.

Luego de realizar los pasos antes descriptos, obtendremos una imagen en **/arch/i386/boot/bzImage** dentro del directorio del código fuente del kernel. Si no hubo errores, el kernel estará listo. Ahora lo agregamos a nuestro gestor de arranque. Para lilo, vamos a **/etc/lilo.conf** y hacemos las modificaciones necesarias para agregarlo

```
May 24 22:04:41 arwen kernel: PPP generic driver version 2.4.2
```

```
May 24 22:04:41 arwen pppoe[1329]: PPP session is 430
```

Ya tenemos casi todo lo necesario para conectarnos a nuestro ISP DSL.

POINT-TO-POINT PROTOCOL OVER ETHERNET (PPPOE)

PPPOE es el protocolo usado por muchos ISP para brindar su servicio de ADSL.

a las opciones de boot. Luego, para que estos cambios se hagan efectivos, escribimos:

```
root@nixbox# lilo
```

Y obtendremos una salida del estilo:

```
root@nixbox:~$ lilo
Added 2.6ipv4
Added 2.6ipv6
Added win_2k
```

Ya tenemos el soporte necesario para la placa de red dentro de nuestro kernel.

El paso siguiente es reiniciar el sistema y bootear el nuevo kernel.

Una vez que GNU/Linux nos deja la opción de login, ingresamos nuestro usuario y pwd, y si todo salió bien, al escribir:

```
root@arwen# cat /var/log/messages | grep eth
```

deberíamos obtener la salida tan ansiada, que indique que la placa de red está funcionando. Tenemos que asegurarnos, también, de que PPP se encuentre en buen estado usando una línea similar:

```
root@arwen# cat /var/log/messages | grep PPP
```

Si todo está correcto, obtendremos una salida del tipo:

Claramente podemos ver que precisaremos un cliente PPPOE para GNU/Linux si queremos conectar nuestra PC a Internet.

Vamos a usar **Roaring Penguin** (www.roaringpenguin.com) para conectarnos a nuestro servidor de servicio ADSL.

Rp-pppoe es un cliente libre de pppoe, disponible en versiones **Linux**, **NetBSD** y **Solaris**. Para continuar, instalamos **rp-pppoe** en nuestro sistema GNU/Linux. Podemos hacerlo desde los archivos RPM, desde el código fuente disponible en la página o bien desde el gestor de paquetes preferido (portage, dpkg, slackpkg, etc.); elegimos la forma que más nos guste. No voy a explicar nada respecto de la instalación de rp-pppoe, ya que dicha aplicación viene por defecto en casi todas las distribuciones, además de que

es un paquete sumamente simple de instalar.

Antes de proceder a realizar este paso, conviene verificar si tenemos rp-pppoe instalado en el sistema:

```
root@nixbox# whereis adsl-setup
```

En caso de que esté instalado, obtendremos una salida de comando similar a la siguiente:

```
adsl-setup: /usr/sbin/adsl-setup /usr/man/man8/adsl-setup.8.gz
/usr/share/man/man8/adsl-setup.8.gz
```

Este comando (whereis) verifica si el nombre **adsl-setup** (que es el archivo de configuración de rp-pppoe) está en alguna parte.

Si la aplicación no está instalada, whereis no encontrará nada y dirá:

```
root@nixbox# whereis rp-pppoe
rp-pppoe:
```

Ese comando nos lleva hasta un asistente sumamente intuitivo y claro al que debemos ofrecerle los datos pertinentes sobre nuestra cuenta. La salida del comando **adsl-setup** será similar a la siguiente:

```
Welcome to the Roaring Penguin ADSL client setup. First, I will run some
checks on your system to make sure the PPPoE client is installed properly...
Looks good! Now, please enter some information:
USER NAME
>>> Enter your PPPoE user name (default oceansoul@loneliness):
```

Ingresamos nuestro nombre de usuario, por lo general escrito bajo la forma

```
nombre@servidor.
```

CONFIGURACION DE RP-PPPOE

Luego de compilar rp-ppoe, instalarlo del rpm o bien verificar que está en nuestro sistema, comenzaremos a configurar la conexión ADSL en cuestión.

Para hacerlo, escribimos:

```
root@nixbox# adsl-setup
```

Luego, **adsl-setup** nos preguntará por nuestra interfaz de red. Si tenemos una sola placa de red **10baseX** (lo más probable), dejamos la opción predefinida, que es "eth0".

**Enter the Ethernet interface connected to the ADSL modem
For Solaris, this is likely to be something like /dev/hme0.
For Linux, it will be ethn, where 'n' is a number.
(default eth0):**

Seleccionamos la opción por default presionando la tecla ENTER, para indicar que lo correcto es **(default eth0)**. Continuamos:

**Do you want the link to come up on demand, or stay up continuously?
If you want it to come up on demand, enter the idle time in seconds
after which the link should be dropped. If you want the link to
stay up permanently, enter 'no' (two letters, lower-case.)
NOTE: Demand-activated links do not interact well with dynamic IP
addresses. You may have some problems with demand-activated links.
>>> Enter the demand value (default no):**

En este paso, rp-pppoe nos pregunta si queremos estar conectados sólo cuando lo solicitamos o si preferimos hacerlo de manera permanente.

Ya que estamos pagando por un servicio, respondemos que no, para estar conectados durante el tiempo que deseemos.

Para seguir con la configuración de nuestra cuenta ADSL, vamos a responder sobre los DNS primarios y secundarios que nos da el proveedor del servicio.

Este paso es de fundamental importancia, ya que algunos ISP brindan direcciones de

resolución de nombres automáticamente (no es preciso indicar al programa dichas direcciones), pero algunas veces esto no funciona del todo bien bajo GNU/Linux.

Hay que prestar atención a este simple paso, y conseguir las direcciones correspondientes del servidor primario y del secundario, respectivamente.

Para obtener estos datos, consultamos a quien nos ofrezca el servicio de ADSL. Averiguamos ambas direcciones y las ingresamos en orden:

**Please enter the IP address of your ISP's primary DNS server.
If your ISP claims that 'the server will provide DNS addresses',
enter 'server' (all lower-case) here.
If you just press enter, I will assume you know what you are doing
and not modify your DNS setup.
>>> Enter the DNS information here:**

Ahora viene el momento de ingresar el password:

**PASSWORD
>>> Please enter your PPPoE password:
>>> Please re-enter your PPPoE password:**

La opción nos pide que ingresemos nuestro password dos veces, para confirmarlo. Seguimos con las preguntas (ya falta poco):

The firewall choices are:

0 - NONE: This script will not set any firewall rules. You are responsible for ensuring the security of your machine. You are STRONGLY recommended to use some kind of firewall rules.

1 - STANDALONE: Appropriate for a basic stand-alone web-surfing workstation

2 - MASQUERADE: Appropriate for a machine acting as an Internet gateway for a LAN

>>> Choose a type of firewall (0-2):

Esta pregunta nos ofrece configurar un firewall. Mi consejo es responder con la opción 0, que corresponde a ninguno. Sugiero que cuando configuren un firewall, lo

hagan ustedes mismos y, sobre todo, eviten las reglas automáticas, que pueden traerles tantos dolores de cabeza. Respondemos con "0" a esta opción.

**** Summary of what you entered ****

Ethernet Interface: eth0

User name: oceansoul@loneliness

Activate-on-demand: No

DNS: Do not adjust

Firewalling: NONE

>>> Accept these settings and adjust configuration files (y/n)?

Esta es la última pregunta, para saber si queremos hacer efectiva la configuración que fuimos completando. Respondemos que sí ("y") para que se generen los archivos de conexión que utiliza nuestro cliente pppoe. ¡Hemos finalizado! Ahora, para conectarnos a Internet escribimos `adsl-start`, y para desconectarnos, `adsl-stop`.

ubicado generalmente en `/etc/rc.X/rc.local` o similar.

Antes de conectarnos a Internet, lo mejor sería configurar un firewall con **iptables**, ya que lo más probable es que tengamos nuestro equipo conectado a Internet durante mucho tiempo. Conviene tomarnos un poco más de trabajo, para así usar nuestra flamante conexión con total tranquilidad.

FINALIZANDO...

Lo último que me resta decir es que podemos decidir agregar el llamado a **adsl-start** al archivo `/etc/rc.d/rc.local` en el caso de mi slackware, o al script de inicio local correspondiente a nuestro GNU/Linux,

Más sobre el tema:
raulinux@hotmail.com

Control no lineal de un levitador magnético.

Resumen

En este reporte presentamos las características principales de un sistema de levitación magnética, además vemos la forma de construcción de un control no lineal para un sistema no lineal, este control es un control tipo linealización exacta. Una simulación en MATLAB del control y una representación en realidad virtual para nuestro sistema.

I. Introducción

Los sistemas de levitación magnética son interesantes, interesantes en el aspecto de que al momento de diseñar un control, es difícil por la naturaleza no lineal de los sistemas, esta es una tarea importante para el control.

El profesor nos propuso utilizar el control por linealización exacta con el fin de aplicar un control no lineal al sistema y ver el comportamiento del mismo con el control.

En este documento mostramos que fue lo que nos motivó para trabajar en este proyecto, una breve descripción de nuestro sistema además de nuestro objetivo; El método para controlar nuestro sistema.

Una representación en SIMULINK de la planta, utilizamos la herramienta de VIRTUAL REALITY de MATLAB. Dentro de esta representación tenemos dos animaciones, una para el sistema de levitación magnética de una esfera y otra para un automóvil.

II. Motivación

Es interesante ver que un objeto flota o se mantiene despegado del suelo sin necesidad de tener cables que lo mantengan suspendido en el aire, la levitación magnética es un fenómeno que cautiva y bueno algunas de esas razones ya se han

mencionado, además ver sistemas que funcionen a través de este fenómeno es aun mucho mas interesante. Sabemos que estos sistemas ya se están aplicando o están en fase de aplicación, y por mencionar algunos ejemplos:

En primer lugar tenemos; al tren de levitación magnética que es mejor conocido como "MAGLEV" en Japón y recientemente en Alemania, el cuál debido a que no ejerce fricción sobre el suelo, llega a alcanzar velocidades enormes. Aunque debemos saber que la fricción es ejercida ahora por el aire, también recordamos la película de volver al futuro en la cuál viajaban a través del tiempo en un carro que levitaba para evitar el roce con el suelo y de esa manera alcanzar una velocidad apropiada para lograr pasar la barrera del tiempo.

En el caso numero dos tenemos que se pretende generar a futuro energía eléctrica mediante aerogeneradores con levitación magnética, esto significa que los aerogeneradores estarán suspendidos en el aire consecuentemente no tendrán soportes y la fricción con el suelo no estará presente por lo tanto al pasar el aire por el aerogenerador de manera casi inmediata se empezara a generar energía eléctrica debemos saber que normalmente los aerogeneradores no proporcionan la energía suficiente debido al roce con sus soportes o con el suelo mismo

En nuestro caso el sistema de levitación magnética se logra energizando un electromagneto, la fuerza del campo magnético esta controlada por la inductancia en el electromagneto

III. Objetivo

El objetivo de nuestro proyecto es controlar un sistema de levitación magnética con un control mediante linealización exacta este es un control para un sistema no lineal, el problema que se presenta con la linealización exacta es encontrar si es posible una realimentación

$$u=a(x)+b(x)n$$

y una transformación de coordenadas

$$x=F(x),$$

tal que el correspondiente sistema en lazo cerrado, usando la realimentación, en las nuevas coordenadas es lineal y controlable, como lo describe la siguiente ecuación

$$x'=F(x),$$

IV. Sobre el proyecto

Nuestro proyecto consiste en hacer que una esfera de metal levite o este suspendida en el aire mediante la fuerza del campo de un electromagneto. La corriente en el electromagneto debe ser incrementada hasta que la fuerza magnética producida sea igual o mas grande que la fuerza gravitatoria que actúa en la esfera.

Desde el punto de vista del control, un sistema de levitación magnética es un sistema complejo debido a su no linealidad; de manera análoga su inestabilidad y en el caso de nuestro proyecto, en un sentido de limitadas entradas y salidas.

Sabemos que basados en el modelo no lineal del sistema, encontrar un control es complicado, y no existen métodos precisos en la literatura para diseño de un control para un sistema de levitación magnética



Figura 1. Sistema de levitación magnética.

V. Marco Teórico

V.1. Modelo dinámico

El sistema de levitación magnética que modelamos se muestra en la figura 2, aquí una bobina hace las veces de un electromagneto, el cual ejerce una fuerza sobre la esferita de metal. El diagrama eléctrico que se muestra en la figura 2 tiene como fin agregar un control de posición de la esfera. Se trata de un circuito RL en serie con una fuente de voltaje como entrada. Aplicando la ley de voltajes de Kirchoff sobre la malla encontramos la relación mostrada en la ecuación 1.

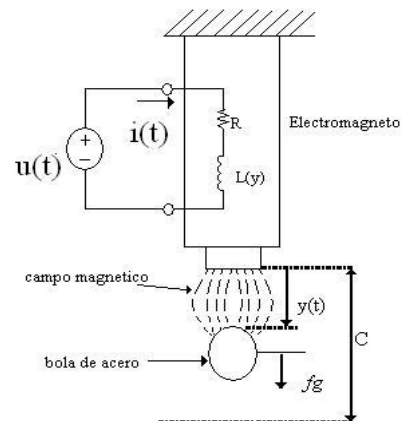


Figura 2. Diagrama del sistema de levitación magnética.

Por ley de Kirchhoff obtenemos la siguiente ecuación

$$u(t) = v_R + v_L$$

Aplicando la ley de ohm y como la inductancia es variable tenemos que:

$$u(t) = i(t)R + \frac{dL(y)i(t)}{dt}$$

Simplificando y tomando en cuenta que todas las variables dependen de t podemos representar la ecuación como:

$$u = Ri + \frac{di}{dt} [L(y)i] \quad (1)$$

Ley de Biot Savart:

$$d\vec{B} = \frac{\mu_0}{4} \frac{id\vec{S} \times \vec{r}}{r^3} \quad (3)$$

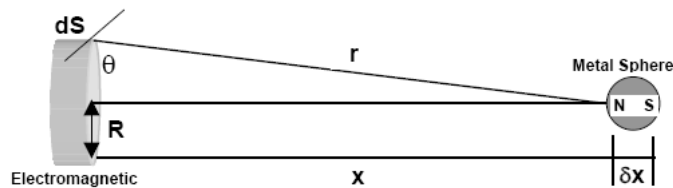


Figura 3. Esquema para el análisis magnético.

$$d\vec{S} = dS\hat{j} = ds[0 \ 1 \ 0]$$

Sabemos además que

$$\vec{r} = r[\sin\theta \ 0 \ \cos\theta]$$

Haciendo el producto de $d\vec{S} \times \vec{r}$ tenemos que

$$d\vec{S} \times \vec{r} = [dsr \cos\theta \ 0 \ dsr \ \text{sen}\theta]$$

Cabe señalar que $L(y)$ es una inductancia variable en el circuito RL, pero esta inductancia genera un campo electromagnético que actúa directamente atrayendo o repeliendo la esfera metálica la cual podemos reconocer como un dipolo magnético

Para el caso de la dinámica de la esfera utilizamos una relación física que relaciona la fuerza electromagnética generada por la inductancia variable en la bobina creando un electroimán y la posición de la esfera por medio de la Ley de Newton (segunda ley de Newton) .

Entonces la $\sum \vec{f}_i(x, y, z) = 0$

$$m\ddot{y} = mg + F_B$$

(2)

Dónde F_B es la fuerza magnética generada por la inductancia variable.

La relación necesaria es la ley conocida como ley de Biot Savart

$$= ds[r \cos \theta \quad 0 \quad r \sin \theta]$$

$$= dsr[\cos \theta \quad 0 \quad \sin \theta]$$

$$r \cos \theta = R$$

Entonces la ecuación de Biot Savart queda como:

$$dB_x = \frac{\mu_0 IdSR}{4\pi r^3}$$

Donde R^3 queda como :

$$r^3 = (R^2 + x^2)^{\frac{3}{2}} = (\sqrt{R^2 + x^2})^3 \quad (4)$$

Sustituyendo la ecuación 4 en 3 tenemos que:

$$dB_x = \frac{\mu_0 IdSR}{4\pi (\sqrt{R^2 + x^2})^3} = \frac{\mu_0 IdSR}{4 (R^2 + x^2)^{\frac{3}{2}}}$$

Integrando B_x tenemos que:

$$\int_0^{2\pi} dB_x = \frac{\mu_0 IdSR}{4\pi (R^2 + x^2)^{\frac{3}{2}}}$$

$$B_x = \frac{\mu_0 IR}{4\pi (R^2 + x^2)^{\frac{3}{2}}} \int dS \rightarrow \pi R^2$$

$$B_x = \frac{\mu_0 IR}{4\pi (R^2 + x^2)^{\frac{3}{2}}} S|_0^{2\pi} \rightarrow 2\pi R$$

$$B_x = \frac{\mu_0 IR^2 2\pi}{4\pi (R^2 + x^2)^{\frac{3}{2}}}$$

Haciendo los despejes respectivos tenemos

$$B_x = \frac{\mu_0 IR^2}{2 (R^2 + x^2)^{\frac{3}{2}}}$$

$$B_T = \frac{\mu_0 n I R^2}{2} \int_x^{x+L} \frac{dx}{(R^2 + x^2)^{\frac{3}{2}}}$$

Donde n es el número de vueltas por metro

$$\begin{aligned} \int_x^{x+L} \frac{dx}{(R^2 + x^2)^{\frac{3}{2}}} &= \frac{1}{(R^2 + x^2)^{\frac{3}{2}}} = \frac{\partial L(m)}{\partial x} \\ &= \frac{\mu_0 n I R^2}{2} \frac{x}{R^2 (R^2 + x^2)^{\frac{1}{2}}} \Big|_x^{x+L} \\ &= \frac{\mu_0 n I R^2}{2} \left(\frac{x+L}{R^2 (R^2 + x^2)^{\frac{1}{2}}} - \frac{x}{R^2 (R^2 + x^2)^{\frac{1}{2}}} \right) \end{aligned}$$

$$B_x = k_1 I G(x)$$

$$F_B = k_1 k_2 I^2 (G(x) - G(x + \delta x))$$

$$F_B = -k_2 I^2 \delta x G'(x)$$

Si δx es pequeño comparado con L y R

$$F_B \approx k I^2 G'(x) \tag{5}$$

Ahora podemos sustituir (5) en (2) para obtener la relación (6)

$$m\ddot{x} = mg - k I^2 G'(x) \tag{6}$$

La ecuación (6) es una generalización en coordenadas de movimiento x , y , z . en este caso solo utilizamos la coordenada en y . entonces sustituyendo solo los valores para el movimiento en y en las ecuaciones (4) y (5) podemos llegar al caso particular siguiente.

$$m\ddot{y} = mg + \frac{1}{2} I^2 \frac{\partial L(y)}{\partial y} \tag{7}$$

Donde:

u : es el voltaje de alimentación del electromagneto.

i : denota la corriente en el electromagneto.

g : es la constante de gravedad.

m : es la masa de la esfera.

R : la resistencia del alambrado.

y : la distancia medida de la cara superior de la esfera con respecto a la cara inferior del electromagneto.
 c : es el rango máximo de operación del sistema.
 $L(y)$ es la inductancia variable del sistema.

Las expresiones para la energía almacenada en el campo magnético del análisis de circuitos y la teoría de campos proveen otra expresión definitoria para la inductancia.

$$W = \frac{1}{2} LI^2$$

$$W = \frac{1}{2} \int_{\mathcal{V}} (\mathbf{B} \cdot \mathbf{H}) d\mathcal{V}$$

Igualando las ecuaciones tenemos:

$$L = \int_{\mathcal{V}} \frac{(\mathbf{B} \cdot \mathbf{H}) d\mathcal{V}}{I^2} \quad (8)$$

Resolviendo la integral para el caso de la bobina de una capa con núcleo de aire como es el caso de la bobina utilizada en este proyecto como electroimán:

$$L(y) = \frac{k}{y_D + y} + L_L \quad (9)$$

Donde k , y_D y L_L son constantes de magnitud positiva y forman parte de la solución de la integral (8).

$$u = Rl + \frac{d(L(y)l)}{dt}$$

$$m\ddot{y} = mg + \frac{1}{2} l^2 \frac{\partial L(y)}{\partial y}$$

$$u = Rl + L(y) \frac{dl}{dt} + \frac{dL(y)}{dy} l \quad (10)$$

$$\ddot{y} = g + \frac{1}{2m} l^2 \frac{\partial L(y)}{\partial y} \quad (11)$$

Sustituimos (9) en (10) y en (11):

$$u = Rl + \left(\frac{k}{y_D + y} + L_L \right) \frac{dl}{dt} + \frac{d\left(\frac{k}{y_D + y} + L_L \right)}{dy} l \quad (12)$$

$$\ddot{y} = g + \frac{1}{2m} l^2 \frac{\partial \left(\frac{k}{y_D + y} + L_L \right)}{\partial y} \quad (13)$$

Resolvemos (12) y (13):

$$u = Rt + \left(\frac{k}{y_D + y} + L_L \right) \frac{dt}{dt} - \frac{k}{(y_D + y)^2} t \quad (14)$$

$$\dot{y} = g - \frac{1}{2m} t^2 \frac{k}{(y_D + y)^2} \quad (15)$$

Despejamos la ecuación (14)

$$\frac{dt}{dt} = \frac{(y_D + y)(u - Rt + \frac{k}{(y_D + y)^2} t)}{k + L_L} \quad (16)$$

$$= \frac{u}{\left(\frac{k}{y_D + y} \right) + L_L} - \frac{t \left(R - \frac{k y}{(y_D + y)^2} \right)}{\left(\frac{k}{y_D + y} \right) + L_L}$$

Multiplicamos la ecuación (16) por $1 = y_D + y / y_D + y = \frac{(y_D + y)^2}{(y_D + y)^2}$

$$\frac{dt}{dt} = \frac{y_D + y}{y_D + y} \frac{u}{\left(\frac{k}{y_D + y} \right) + L_L} - \frac{(y_D + y)^2 t \left(R - \frac{k y}{(y_D + y)^2} \right)}{(y_D + y)^2 \left(\frac{k}{y_D + y} \right) + L_L}$$

$$= \frac{(y_D + y)u}{k + L_L(y_D + y)} + \frac{t(R(y_D + y)^2) - tk\dot{y}}{(y_D + y)k + L_L(y_D + y)^2}$$

$$\frac{dt}{dt} = t \left[\frac{(R(y_D + y)^2 - ky)}{(y_D + y)k + L_L(y_D + y)^2} \right] + u \left[\frac{(y_D + y)}{k + L_L(y_D + y)} \right] \quad (17)$$

Representando el sistema en el espacio de estados:

$$x = \begin{bmatrix} y \\ \dot{y} \\ t \end{bmatrix} \begin{array}{l} \leftarrow \text{posicion} \\ \leftarrow \text{velocidad} \\ \leftarrow \text{corriente} \end{array}$$

Ahora modelamos el sistema en la forma:

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \quad (18)$$

Derivamos entonces x:

$$\dot{x} = \begin{bmatrix} \dot{y} \\ \ddot{y} \\ \frac{dt}{dt} \end{bmatrix} \quad (20)$$

Sustituyendo nuevamente los valores de cada componente:

$$\dot{x} = \frac{d}{dt} \begin{bmatrix} y \\ \dot{y} \\ i \end{bmatrix} = \begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dot{i} \end{bmatrix} = \begin{bmatrix} \dot{y} \\ -\frac{k}{2m} \frac{i^2}{[y_0 + y]^2} + g \\ i \left[k \dot{y} - R[y_0 + y]^2 \right] \\ \frac{k[y_0 + y] + L_\infty[y_0 + y]^2}{k[y_0 + y] + L_\infty[y_0 + y]^2} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{y_0 + y}{L_\infty[y_0 + y] + k} \end{bmatrix} u \quad (21)$$

Después de que despejamos las variables, procedemos a escribir nuestra matriz de estados con nuestras variables y' e i'

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} \dot{y} \\ g - \frac{kt^2}{2m(y_0+y)^2} \\ \frac{i(-R(y_0+y)^2 + ky)}{(y_0+y)k + L_\infty(y_0+y)^2} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{(y_0+y)}{k + L_\infty(y_0+y)} \end{bmatrix} u \quad (22)$$

En este paso tenemos ya la ecuación de estados, podemos observar que $f(x)$ y $g(x)$ son campos vectoriales suaves, y ya podemos pensar en un control por linealización exacta, entonces, hacemos una transformación llamada *difeomorfismo*, ya que el sistema contiene una parte no lineal en su segunda componente. Como el grado relativo del sistema es 3, definimos la transformada:

$$T(x) = z = \begin{bmatrix} z_1(x) \\ z_2(x) \\ z_3(x) \end{bmatrix} = \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} h(x) \\ L_f h(x) \\ L_f^2 h(x) \end{bmatrix} \quad (23)$$

L_f^i son las primeras (n-1) derivadas de Lie del sistema.

$$T(x) = z = \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} y \\ \dot{y} \\ g - \frac{kt^2}{2m(y_0+y)^2} \end{bmatrix} \quad (24)$$

Una vez con esta transformación, aplicamos la teoría de difeomorfismos, o en particular las n derivadas de Lie que se representan como:

$$y^{(n)} = L_f^n h(x) + L_g L_f^{n-1} h(x) u \quad (25)$$

Que son la n -ésima derivada de y en función de las derivadas de Lie o derivadas del difeomorfismo. Así realizamos otro cambio para un nuevo espacio de estados transformado.

Para obtener \dot{z}

$$\dot{z} = \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} L_f h(x) \\ L_f^2 h(x) \\ L_f^3 h(x) + L_g L_f^2 h(x)u \end{bmatrix} \quad (26)$$

Ya que por tratarse ahora de las n derivadas de y contamos con la n-ésima componente definida con un término $L_g L_f^{n-1} h(x)u$ adicional.

Necesitamos linealizar tal que tengamos el sistema

$$\dot{z} = Az + Bv \quad (27)$$

Ahora aplicamos (24) y la siguiente ley de control (28) a (26):

$$u = \frac{(-L_f^3 h(x) + v)}{L_g L_f^2 h(x)} \quad (28)$$

Para tener finalmente el sistema linealizado.

$$\dot{z} = \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} z_2 \\ z_3 \\ v \end{bmatrix} = Az + Bv \quad (29)$$

Proponemos el cambio de variable: $v = -kz$

$$\dot{z} = Az \quad (30)$$

$$\Rightarrow \dot{z} = Az + B(-kz) = (A - Bk)z$$

$$\Rightarrow \dot{z} = (A - Bk)z \quad (31)$$

$$Bk = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} [k_1 \quad k_2 \quad k_3] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ k_1 & k_2 & k_3 \end{bmatrix}$$

$$A - Bk = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ k_1 & k_2 & k_3 \end{bmatrix}$$

$$A - Bk = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -k_1 & -k_2 & -k_3 \end{bmatrix} \quad (32)$$

Aquí bastará con probar que $(A - Bk)$ es Hurwitz para demostrar estabilidad del sistema, como k es un vector de ganancias, podemos sintonizar el sistema de tal forma que sea estable, por ello encontramos los valores propios de $(A - Bk)$.

Tenemos que:

$$\begin{aligned} \text{Det}(\lambda I - (A - Bk)) &= \begin{vmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{vmatrix} - \begin{vmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -k_1 & -k_2 & -k_3 \end{vmatrix} = \begin{vmatrix} \lambda & -1 & 0 \\ 0 & \lambda & -1 \\ k_1 & k_2 & \lambda + k_3 \end{vmatrix} \\ &= |\lambda(\lambda)(\lambda + k_3) - (-k_2)| - (-1)(k_1) = \lambda(\lambda^2 + k_3\lambda) + \lambda k_2 + k_1 \\ &= \lambda^3 + \lambda^2 k_3 + \lambda k_2 + k_1 \end{aligned} \quad (33)$$

Para encontrar la solución de λ y propuesta de k , recurrimos a la ecuación de Cardano la cual nos lleva a la expresión siguiente

$$\lambda^3 + k_3 \lambda^2 + k_2 \lambda + k_1 = 0$$

$$y = x - \frac{a}{3}$$

$$x^3 + px + q = 0$$

$$p = k_2 - \frac{k_3^2}{3}$$

$$q = \frac{k_3}{3} \left(-\frac{k_3}{3} + \frac{k_2^2}{3} - k_2 \right) + k_1$$

$$\begin{aligned} x = & \sqrt[3]{-\frac{[\frac{k_3}{3}(\frac{k_2^2}{3} - \frac{k_3}{3} - k_2) + k_1]}{2}} + \sqrt[3]{\frac{[\frac{k_3}{3}(\frac{k_2^2}{3} - \frac{k_3}{3} - k_2) + k_1]^2}{4} + \frac{(-\frac{k_3}{3} + k_2)^3}{27}} + \\ & + \sqrt[3]{-\frac{[\frac{k_3}{3}(\frac{k_2^2}{3} - \frac{k_3}{3} - k_2) + k_1]}{2}} - \sqrt[3]{\frac{[\frac{k_3}{3}(\frac{k_2^2}{3} - \frac{k_3}{3} - k_2) + k_1]^2}{4} + \frac{(-\frac{k_3}{3} + k_2)^3}{27}} \end{aligned}$$

(34)

La ecuación (34) nos sirve para graficar el lugar geométrico de las raíces y así poder tener una idea de los valores aptos para las k .

Ahora para la parte de la ley de control retomamos (28) y evaluamos las derivadas.

$$u = \frac{(-L_f^3 h(x) + v)}{L_g L_f^2 h(x)} = \frac{\frac{d}{dt} \left(-\frac{kt^2}{2m(y_p + y)^2} + g \right) f(x) + v}{\frac{d}{dt} \left(g - \frac{kt^2}{2m(y_0 + y)^2} \right) g(x)}$$

$$\begin{aligned}
u &= \frac{\frac{-2kt}{2m(y_D+y)^2}f(x) + v}{\frac{-2k}{2m(y_D+y)^2}g(x)} = \frac{\frac{-kt}{m(y_D+y)^2}f(x) + v}{\frac{-2kt}{2m(y_D+y)^2} \left[\frac{(y_D+y)}{k+L_L(y_D+y)} \right]} = \frac{\frac{-kt}{m(y_D+y)^2}f(x) + v}{\frac{-kt}{m(y_D+y)[k+L_L(y_D+y)]}} \\
u &= \frac{\frac{-kt}{m(y_D+y)^2} \left[\frac{l(R(y_D+y)^2 - k\dot{y})}{(y_D+y)k+L_L(y_D+y)^2} \right] + v}{\frac{-kt}{m(y_D+y)[k+L_L(y_D+y)]}} = \frac{\frac{-kt}{m} \left[\frac{(-lR + tk\dot{y}(y_D+y)^{-2})}{(y_D+y)k+L_L(y_D+y)^2} \right] + v}{\frac{-kt}{m(y_D+y)[k+L_L(y_D+y)]}} \\
u &= \frac{(m(y_D+y)[k+L_L(y_D+y)]) \left(\frac{-kt}{m} \left[\frac{(-lR + tk\dot{y}(y_D+y)^{-2})}{(y_D+y)k+L_L(y_D+y)^2} \right] + v \right)}{-kt} \\
u &= \frac{(m(y_D+y)[k+L_L(y_D+y)]) \left(\frac{-kt}{m(y_D+y)} \left[\frac{(-lR + tk\dot{y}(y_D+y)^{-2})}{k+L_L(y_D+y)} \right] + v \right)}{-kt} \\
u &= \frac{\left(\frac{-ktm(y_D+y)[k+L_L(y_D+y)]}{m(y_D+y)} \left[\frac{(-lR + tk\dot{y}(y_D+y)^{-2})}{k+L_L(y_D+y)} \right] + m(y_D+y)[k+L_L(y_D+y)]v \right)}{-kt} \\
u &= \frac{-kt(-lR + tk\dot{y}(y_D+y)^{-2}) + m(y_D+y)[k+L_L(y_D+y)]v}{-kt} \\
u &= \frac{-kt(lR - tk\dot{y}(y_D+y)^{-2})}{-kt} - \frac{-kt}{kt} \frac{m(y_D+y)[k+L_L(y_D+y)]v}{m(y_D+y)[k+L_L(y_D+y)]} \\
u &= lR - \frac{tk\dot{y}}{(y_D+y)^2} + \frac{m(y_D+y)[k+L_L(y_D+y)]v}{-kt} \tag{35}
\end{aligned}$$

Aquí a la ecuación (9) la multiplicamos por $\frac{1}{y_D+y}$

$$\begin{aligned}
\frac{L}{y_D+y} &= \frac{k}{(y_D+y)^2} + \frac{L_L}{y_D+y} \\
\frac{k}{(y_D+y)^2} &= \frac{L}{y_D+y} - \frac{L_L}{y_D+y} \approx -\frac{L_L}{y_D+y} \tag{36}
\end{aligned}$$

Sustituimos (36) en (35) para tener una ley de control finalmente:

$$u = lR + \frac{ky\dot{L}_L}{y_D+y} - \frac{m(y_D+y)[k+L_L(y_D+y)]v}{kt} \tag{37}$$

Donde v es la realimentación de estados $v = -kx$
Vectorialmente:

$$v = [-k_1 -k_2 -k_3] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = -k_1 x_1 - k_2 x_2 - k_3 x_3 \quad (38)$$

Tal que $x_1 = y - y_d$; $x_2 = \dot{y}$; $x_3 = g - \frac{kt^2}{2m(y_0 + y)^2}$

Y y_d es la posición deseada.

Como para valores de k la matriz es Hurwitz, procedemos a sintonizar y proponer ganancias en un software como Mathematica o Maple y en su defecto Matlab.

VI. Marco experimental

Para realizar nuestras simulaciones ocupamos el programa Virtual Reality de MALAB y además la herramienta de SIMULINK

1. Control del sistema de levitación magnética para la planta original

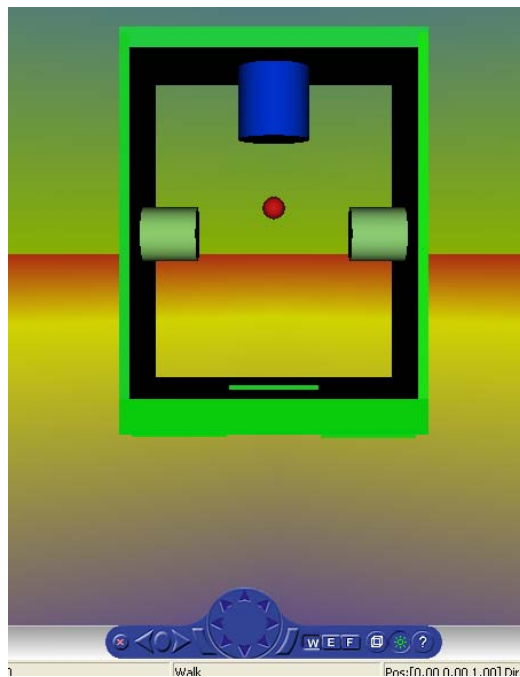


Figura 4. Representación del sistema en realidad virtual.

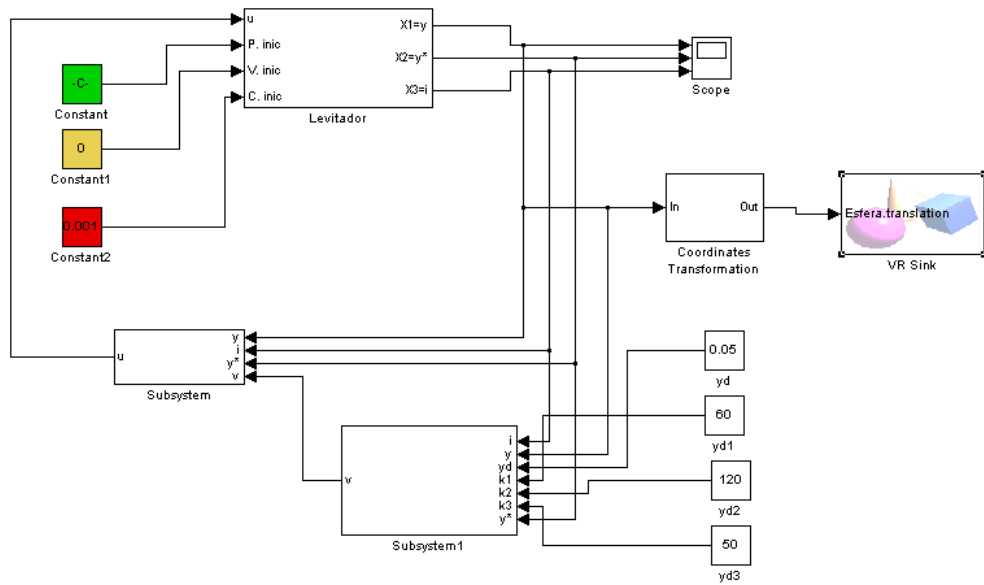


Figura 5. Esquema del control del levitador en Simulink.

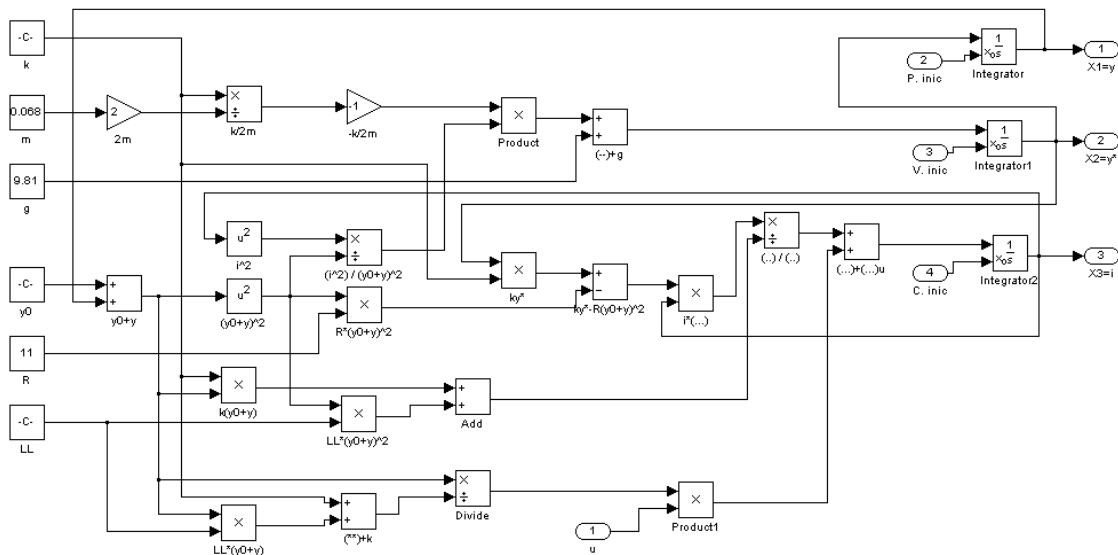


Figura 6. Bloque de control del sistema de levitación magnética

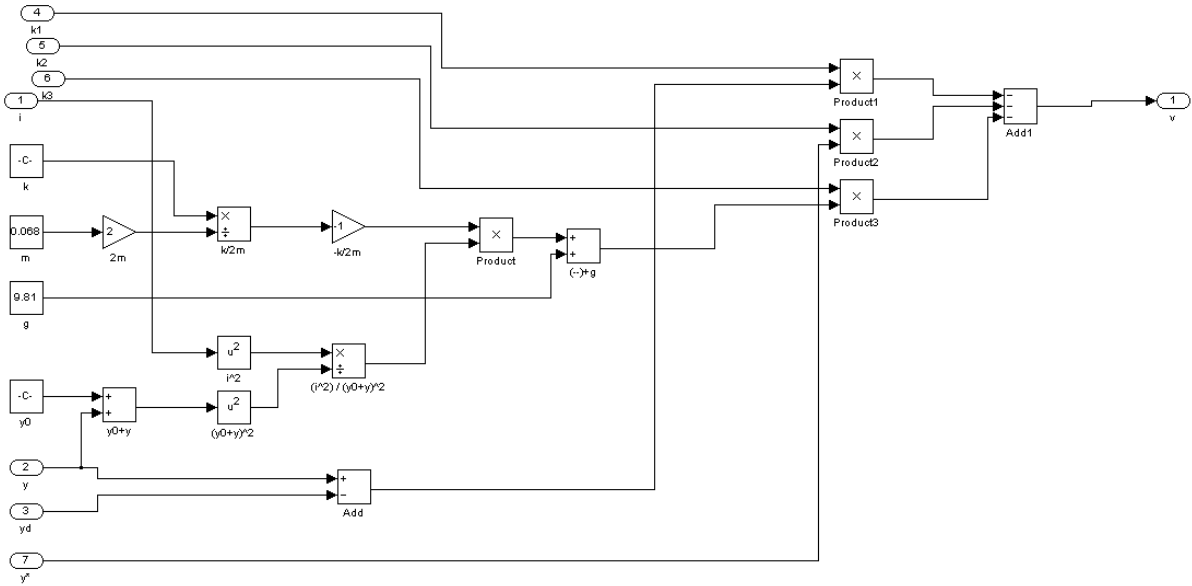


Figura 7. Bloque de la realimentación de estados v.

2. Aplicación del sistema de levitación magnética

Queremos que nuestro control se aplique en un sistema de levitación magnética mas común aunque los carros que levitan aun no se han hecho presentes, no tardaran en salir al mercado es por eso que este control lo hemos aplicado a un automóvil de levitación magnética.

Este es el diagrama del sistema de levitación magnética

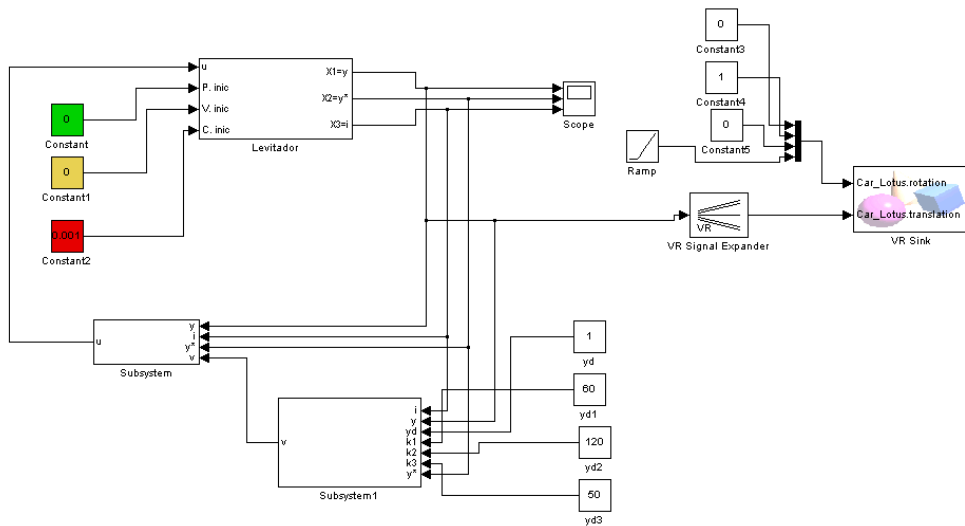


Figura 8. Sistema de control con intarfaz a realidad virtual.

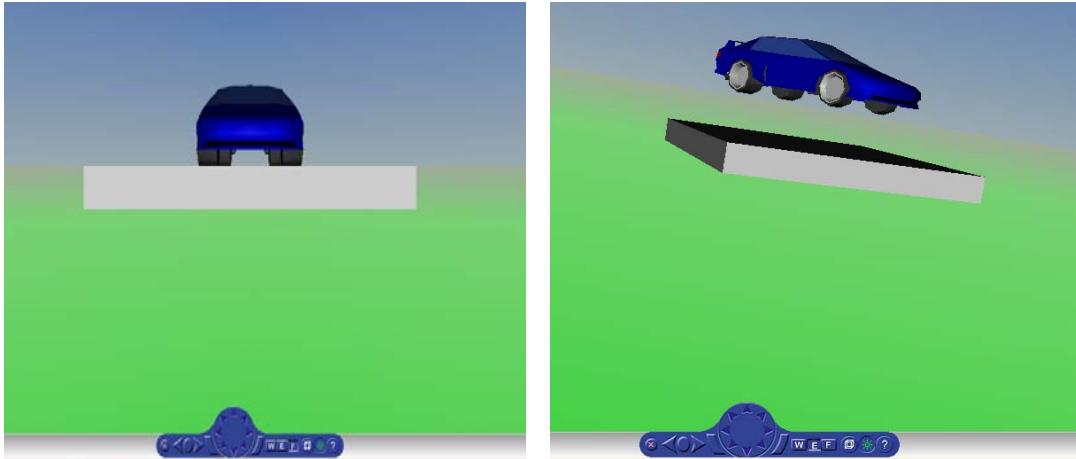


Figura 9. Un ejemplo de aplicación del levitador en el builder de realidad virtual de Matlab.ⁱ

VII. Conclusiones

Es importante saber diseñar controles para sistemas no lineales, en esta práctica observamos que la linealización exacta es un método interesante para controlar sistemas pero hay que realizar muchos pasos, decimos esto por que para encontrar nuestras ganancias "K" del sistema teníamos una ecuación de 3 orden y pues esto nos llevo tiempo, aunque por una parte estuvo bien ya que adquirimos nuevos conocimientos y corroboramos lo que aprendimos en clase.

Al final logramos el objetivo del proyecto, controlar la levitación de la esfera, y con la aplicación del control que obtuvimos quedó aun mas claro; y es que, el hecho de recurrir a diferentes referencias bibliográficas, ayudó a ampliar nuestra visión acerca de las aplicaciones que tiene el control en la vida actual por ejemplo la de hacer levitar un carro, sobre este tema ya se esta trabajando, pero nos pareció interesante poder aplicarle nuestro control a este sistema no lineal, esta vez, lo hicimos levitar hasta una altura "y" y ya en esta posición hicimos que girara hasta posicionarse en otra pero siempre a esa altura.

Bibliografía

- [1] A. Kuroschi. Curso de Algebra Superior. Ed. MIR, Moscú, 1975.
- [2] A. Isidori, Nonlinear Control Systems, Springer- Verlag, Berlin, Third Edition, 1995.
- [3] H. Khalil, Nonlinear Systems, Prentice Hall, Second Edition, 1996.
- [4] Camarillo, Alfredo et al. Evaluación experimental de controles no lineales, Instituto Tecnológico de la Laguna, Torreón Coahuila
- [5] A. El hajjaji M. Outladsine Modeling and Nonlinear Control of Magnetic Levitation, IEEE transactions on industrial electronics, Vol 48., August 2003

ⁱ Imagen del auto tomada de los archivos .wrl incluidos en el software demo de Matlab.

Trivia.

¡Hola lectores de WIKI, en esta ocasión toca al área de la Física, así que te las debes de saber!



Una linterna de mano es un ejemplo simple de un circuito eléctrico.

Las baterías de la linterna suministran la energía potencial eléctrica a una corriente

de electrones, los cuales fluyen a través de un filamento de foco de la linterna donde la energía potencial se transforma en luz y calor. En seguida los electrones regresan a las baterías para repetir el ciclo.

¿La cantidad de corriente que sale del filamento incandescente del foco de la lámpara es menor que, mayor que o igual a la cantidad de corriente que entra en el filamento?

Las partículas con carga expulsadas desde el sol son conducidas por el campo magnético terrestre hacia los polos norte y sur de nuestro planeta. Cuando estas partículas chocan con moléculas gaseosas en la atmósfera superior, las moléculas emiten luz.

El resultado es el brillante espectáculo luminoso llamado aurora astral (en latitudes situadas muy al sur) o aurora boreal (en latitudes muy septentrionales).

¿Por qué las partículas solares con carga se desplazan principalmente hacia los polos de la tierra y no hacia el ecuador?





El número, fecha de caducidad y nombre del titular de una tarjeta de crédito están

codificados en un patrón magnetizado en la banda del reverso de la tarjeta. Cuando se hace pasar la tarjeta a través del lector de tarjetas, la banda en movimiento baña los circuitos del lector con un campo magnético variable que induce corrientes en los circuitos. Estas corrientes transmiten la información de la banda al banco del titular de la tarjeta.

¿Qué ocurriría sino se hiciese pasar la tarjeta por la ranura del lector, sino sólo se dejase inmóvil en ella?

La ley de Gauss es otra forma de describir el comportamiento de las cargas y los campos eléctricos. Una consecuencia de esta ley es que las cargas eléctricas de un conductor se encuentran en la superficie de éste, no en su interior. Es por eso que ésta niña adquiere una carga eléctrica al tocar la esfera metálica con carga. Los cabellos con carga de la cabeza de la niña se repelen y se erizan.

Si la niña se pone de pie dentro de una esfera metálica grande con carga, ¿se le erizará el cabello?





Estas dos inmensas galaxias –cada una de las cuales contiene miles de millones de estrellas, situadas a 114 millones de años

luz de la tierra- han estado chocando en cámara lenta durante decenas de millones de años. Dicho “choque” en realidad es una interacción gravitacional entre las dos galaxias. (Las estrellas individuales están tan separadas que es poco probable que choquen unas con otras).

La masa de estas dos galaxias juntas es más de 10^{11} veces mayor que la del sol. ¿Por qué, entonces, no arrancan a la Tierra de su órbita en torno al sol y la atraen hacia sí?

Tal vez el material a más alta temperatura que jamás verá el lector en la atmósfera exterior del sol, llamada corona. La corona, que está a una temperatura aproximada de 2,000,000°C, emite una luz que literalmente está fuera de este mundo.

Sin embargo, por ser delgada la corona, su luz es más bien tenue y sólo podemos verla durante un eclipse solar total.

¿Es correcto decir que la corona contiene calor?



Envía tus respuestas a blastedrobotics@yahoo.com.mx

Y si eres de los primeros en responder a todas las preguntas correctamente, recibirás un fabuloso premio.