

UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

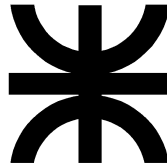
Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

Grupo N°:		Año y División:	2010 - R 6055
Integrantes:	1 – Bobbio Matias 115548-9 2 – Gonzalez Maximiliano 116943-9		
		Fecha:	
Título del Proyecto:	Análisis de la dinámica del Robot e implementación en FPGA		

Índice:

- **Introducción sobre dinámica del robot.**
- **Análisis dinámico**
- **Implementación en código VHDL**
- **Resultados de la simulación**
- **Conclusiones finales.**

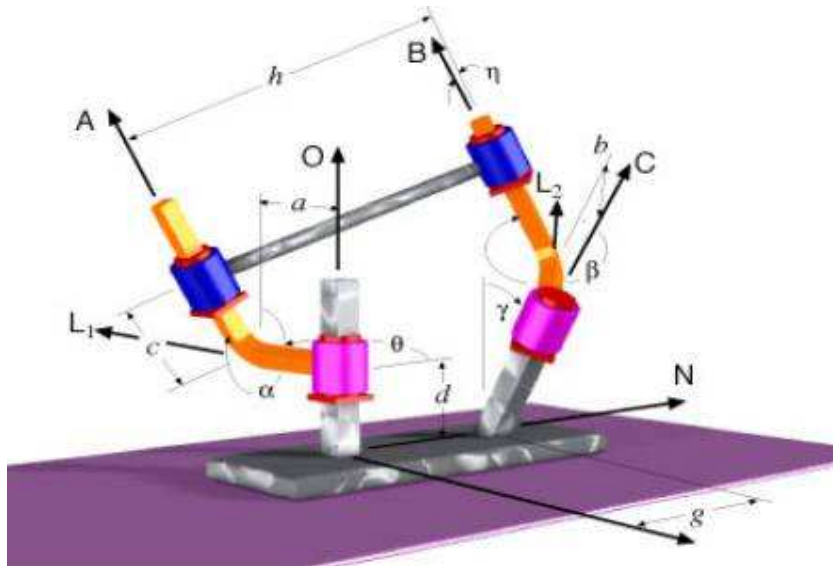


UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL Buenos Aires
Departamento de Electrónica
Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

Introducción Dinámica del Robot

La dinámica se ocupa de la relación entre las fuerzas que actúan sobre un cuerpo y el movimiento que en él se origina. Por lo tanto, el modelo dinámico de un robot tiene por objeto conocer la relación entre el movimiento del robot y las fuerzas implicadas en el mismo.

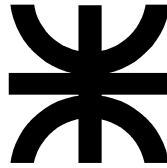


Esta relación se obtiene mediante el denominado modelo dinámico, que relaciona matemáticamente:

1. La localización del robot definida por sus variables articulares o por las coordenadas de localización de su extremo, y sus derivadas: velocidad y aceleración.
2. Las fuerzas pares aplicados en las articulaciones (o en el extremo del robot).
3. Los parámetros dimensionales del robot, como longitud, masa e inercias de sus elementos.

La obtención de este modelo para mecanismos de uno o dos grados de libertad no es excesivamente compleja, pero a medida que el número de grados de libertad aumenta, el planteamiento y obtención del modelo se complica enormemente. Por este motivo no siempre es posible obtener un modelo dinámico expresado de una forma cerrada, esto es, mediante una serie de ecuaciones, normalmente del tipo diferencial de segundo orden, cuya integración permita conocer que el movimiento surge al aplicar unas fuerzas o que fuerzas hay que aplicar para obtener un movimiento determinado.

El modelo dinámico debe ser resuelto entonces de manera iterativa mediante la utilización de un procedimiento numérico.



UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

El problema de la obtención del modelo dinámico de un robot es, por lo tanto, uno de los aspectos más complejos de la robótica, lo que ha llevado a ser obviado en numerosas ocasiones. Sin embargo, el modelo dinámico es imprescindible para conseguir los siguientes fines:

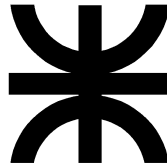
1. Simulación del movimiento del robot.
2. Diseño y evaluación de la estructura mecánica del robot.
3. Dimensionamiento de los actuadores.
4. Diseño y evaluación del control dinámico del robot.

Este último fin es evidentemente de gran importancia, pues de la calidad del control dinámico del robot depende la precisión y velocidad de sus movimientos. La gran complejidad ya comentada existente en la obtención del modelo dinámico del robot, ha motivado que se realicen ciertas simplificaciones, de manera que así pueda ser utilizado en el diseño del controlador.

Es importante hacer notar que el modelo dinámico completo de un robot debe incluir no solo la dinámica de sus elementos (barras o eslabones) sino también la propia de sus sistemas de transmisión, de los actuadores y sus equipos electrónicos de mando. Estos elementos incorporan al modelo dinámico nuevas inercias, rozamientos, saturaciones de los circuitos electrónicos, etc. aumentando aun más su complejidad.

Por último, es preciso señalar que si bien en la mayor parte de las aplicaciones reales de robótica, las cargas e inercias manejadas no son suficientes como para originar deformaciones en los eslabones del robot, en determinadas ocasiones no ocurre así, siendo preciso considerar al robot como un conjunto de eslabones no rígidos. Aplicaciones de este tipo pueden encontrarse en la robótica espacial o en robots de grandes dimensiones.

FUENTE: Fundamentos_de_robotica_-_Barrientos_Penin_Balaguer_y_Aracil



Informe Trabajo Practico N°2

Análisis Dinámico

One-legged Robot

2.2 Robot model

Figure 3 shows the dynamic model of the experimental robot. In this study, only movement on x-z plane is considered. Table 1 shows parameters of each links and robot parameters.

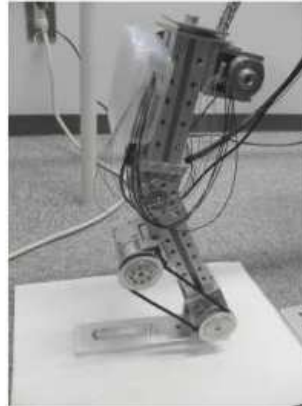


Figure 1: Test One-legged Robot.

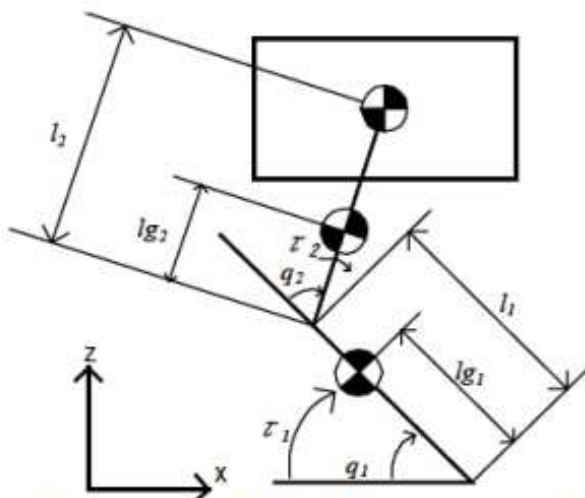
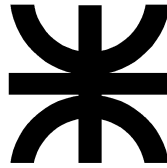


Figure 3: Dynamic Model of One-legged Robot.

Table 1: One-legged robot specifications

Mass(kg)	1.336(kg)
Width	0.18(m)
Depth	0.04(m)
Height	0.36(m)
l_0 (Body link)	0.18(m)
l_1 (2nd thigh length)	0.18(m)
l_2 (thigh length)	0.13(m)
m_0 (body mass)	0.235(kg)
m_1 (2nd thigh mass)	0.465(kg)
m_2 (thigh mass)	0.450(kg)
M_p (frame mass)	0.185(kg)
I_1	2nd thigh inertia
I_2	thigh inertia
lg_1	gravity point of 2nd thigh
lg_2	gravity point of thigh



UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

Utilizando el análisis Dinámico según el enfoque energético de Lagrange-Euler obtenemos la siguiente ecuación:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \mathbf{J}(\mathbf{q})^T \mathbf{F}_e = \boldsymbol{\tau}$$

Donde M es la matriz de inercias, C es la matriz columna de fuerzas de Coriolis, centrífuga, centrípeta, G la matriz columna de fuerzas de gravedad y la matriz J es el jacobiano del punto de trabajo de la fuerza de empuje.

$$\mathbf{M}(\mathbf{q}) = \begin{pmatrix} m_1 l_{g1}^2 + I_1 + m_2 (l_1^2 + l_{g2}^2 + 2l_1 l_{g2} c_2) + I_2 & m_2 (l_{g2}^2 + l_1 l_{g2} c_2) + I_2 \\ m_2 (l_{g2}^2 + l_1 l_{g2} c_2) + I_2 & m_2 l_{g2}^2 + I_2 \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix}$$

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -2m_2 l_1 l_{g2} s_2 \dot{q}_1 \dot{q}_2 - m_2 l_1 l_{g2} s_2 \dot{q}_2^2 \\ m_2 (l_{g2}^2 + l_1 l_{g2} c_2) \dot{q}_1 \dot{q}_2 \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \quad \ddot{\mathbf{q}} = \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix}$$

$$\mathbf{G}(\mathbf{q}) = \begin{bmatrix} m_1 g l_{g1} c_1 + m_2 g (l_1 c_1 + l_{g2} c_{12}) \\ m_2 g l_{g2} c_{12} \end{bmatrix} = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}$$

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \Rightarrow \dot{\mathbf{J}}(\mathbf{q}) = \begin{bmatrix} -l_1 c_1 \dot{q}_1 - l_2 c_{12} (\dot{q}_1 + \dot{q}_2) & -l_2 c_{12} (\dot{q}_1 + \dot{q}_2) \\ -l_1 s_1 \dot{q}_1 - l_2 s_{12} (\dot{q}_1 + \dot{q}_2) & -l_2 s_{12} (\dot{q}_1 + \dot{q}_2) \end{bmatrix} = \begin{bmatrix} \dot{J}_{11} & \dot{J}_{12} \\ \dot{J}_{21} & \dot{J}_{22} \end{bmatrix}$$

$$\mathbf{J}(\mathbf{q})^T = \begin{bmatrix} l_2 c_{12} & l_2 s_{12} \\ -(l_1 c_1 + l_2 c_{12}) & -(l_1 s_1 + l_2 s_{12}) \end{bmatrix} = \begin{bmatrix} J_{11}^T & J_{12}^T \\ J_{21}^T & J_{22}^T \end{bmatrix}$$

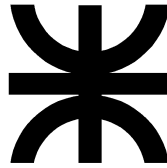
Siendo:

$$c_1 = \cos(q_1), c_{12} = \cos(q_1 + q_2), s_1 = \sin(q_1) \text{ y } s_{12} = \sin(q_1 + q_2)$$

Obteniendo la siguiente matriz:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} M_{11} \ddot{q}_1 + M_{12} \ddot{q}_2 \\ M_{21} \ddot{q}_1 + M_{22} \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} + \begin{bmatrix} J_{11}^T F_{e1} + J_{12}^T F_{e2} \\ J_{21}^T F_{e1} + J_{22}^T F_{e2} \end{bmatrix}$$

Donde $F_e = M_p \ddot{x} + M_p \mathbf{g}$ y como $\dot{x} = J(\mathbf{q}) \dot{q} \rightarrow \ddot{x} = \dot{J}(\mathbf{q}) \dot{q} + J(\mathbf{q}) \ddot{q}$ queda $F_e = M_p \dot{J}(\mathbf{q}) \dot{q} + M_p \mathbf{g}$



UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

Por lo tanto:

$$\begin{bmatrix} Fe1 \\ Fe2 \end{bmatrix} = M_p \begin{bmatrix} \dot{J}_{11}(q) & \dot{J}_{12}(q) \\ \dot{J}_{21}(q) & \dot{J}_{22}(q) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + M_p \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} M_p [\dot{J}_{11}(q)\dot{q}_1 + \dot{J}_{12}(q)\dot{q}_2 + g_1] \\ M_p [\dot{J}_{21}(q)\dot{q}_1 + \dot{J}_{22}(q)\dot{q}_2 + g_2] \end{bmatrix}$$

Y reemplazando cada termino:

$$\begin{aligned} \tau_1 = & [m_1 l_{g1}^2 + I_1 + m_2 (l_1^2 + l_{g2}^2 + 2l_1 l_{g2} c_2) + I_2] \ddot{q}_1 + [m_2 (l_{g2}^2 + l_1 l_{g2} c_2) + I_2] \ddot{q}_2 - (2m_2 l_1 l_{g2} s_2) \dot{q}_1 \dot{q}_2 - \\ & -(m_2 l_1 l_{g2} s_2) \dot{q}_2^2 + m_1 g l_{g1} c_1 + m_2 g (l_1 c_1 + l_{g2} c_{12}) + M_p l_2 c_{12} [-l_1 c_1 \dot{q}_1 \dot{q}_1 - l_2 c_{12} (\dot{q}_1 + \dot{q}_2) \dot{q}_1 - \\ & - l_2 c_{12} (\dot{q}_1 \dot{q}_2 + \dot{q}_2 \dot{q}_2) g_1] + M_p l_2 s_{12} [-l_1 s_1 \dot{q}_1 \dot{q}_1 - l_2 s_{12} (\dot{q}_1 \dot{q}_1 + \dot{q}_2 \dot{q}_1) - l_2 s_{12} (\dot{q}_1 \dot{q}_2 + \dot{q}_2 \dot{q}_2) + g_2] \end{aligned}$$

$$\begin{aligned} \tau_2 = & [m_2 (l_{g2}^2 + l_1 l_{g2} c_2) + I_2] \ddot{q}_1 + (m_2 l_{g2}^2 + I_2) \ddot{q}_2 + m_2 (l_{g2}^2 + l_1 l_{g2} \dot{q}_1 \dot{q}_2) + m_2 g l_{g2} c_{12} - \\ & - M_p [l_1 c_1 + l_2 c_{12}] [-l_1 c_1 \dot{q}_1 \dot{q}_1 - l_2 c_{12} (\dot{q}_1 \dot{q}_1 + \dot{q}_2 \dot{q}_1) - l_2 c_{12} (\dot{q}_1 \dot{q}_2 + \dot{q}_2 \dot{q}_2) + g_1] - \\ & - M_p [l_1 s_1 + l_2 s_{12}] [-l_1 s_1 \dot{q}_1 \dot{q}_1 - l_2 s_{12} (\dot{q}_1 \dot{q}_1 + \dot{q}_2 \dot{q}_1) - l_2 s_{12} (\dot{q}_1 \dot{q}_2 + \dot{q}_2 \dot{q}_2) + g_2] \end{aligned}$$

Donde τ_1 y τ_2 son los pares de los motores efectivo (incluyendo rozamiento y otras perturbaciones) que actúa sobre la articulación 1 y 2 respectivamente.

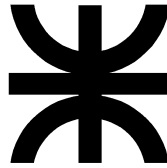
Como se ve, el modelo dinámico responde a un sistema de 2 ecuaciones diferenciales de 2° orden no lineales y acopladas de la forma:

$$\tau_1 = AS^2[Q_1(s) + Q_2(s) + Q_1(s)Q_2(s) + Q_2^2(s)] + BS[Q_1(s) + Q_2(s)] + C$$

$$\tau_2 = DS^2[Q_1(s) + Q_2(s) + Q_1(s)Q_2(s)] + ES[Q_1(s) + Q_2(s)] + F$$

Pero para trabajar con matlab vamos a expresar la ecuación matricial:

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) + J(q)^T [M_p \dot{J}(q)\dot{q} + M_p g] = \tau$$



UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

Teniendo en cuenta la tabla de valores del comienzo (tabla 1) y suponiendo los centros de masa como:

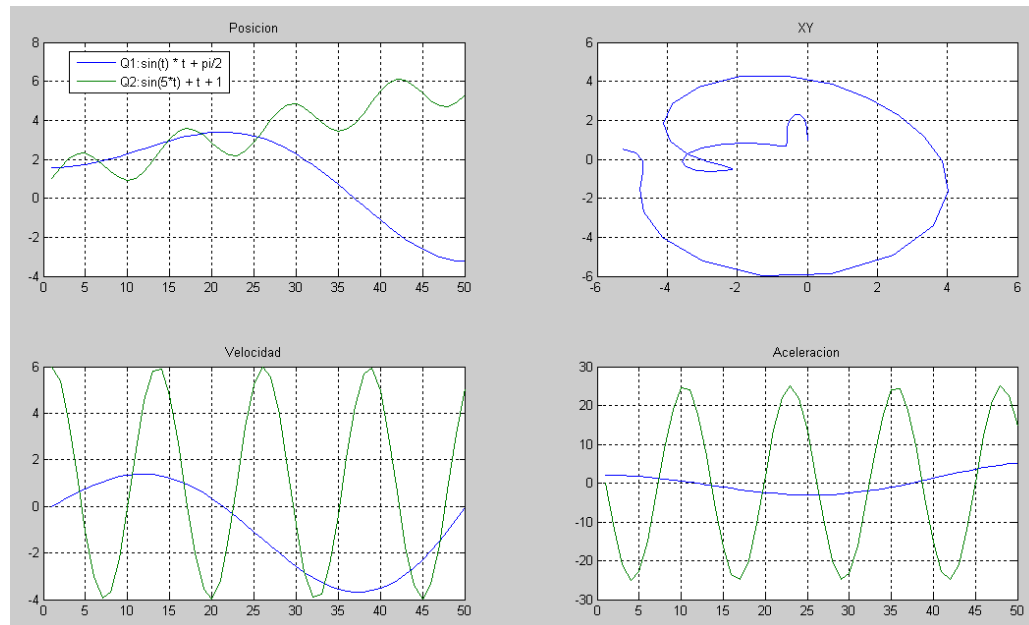
$l_{g1} = 0.7l_1 = 0.126m$ y $l_{g2} = 0.3l_2 = 0.039m$ y las inercias I_1 e $I_2 = 0$ para el movimiento inicial

Implementación con matlab:

Tras repetidos intentos de implementar un código propio que se encargue de realizar los cálculos antes expuestos, decidimos invertir tiempo en familiarizarnos con el "Robotic Toolbox" de Matlab.

Para esta practica utilizamos como base el ejemplo adjunto (Fuente.m), donde si bien la arquitectura es distinta, conceptualmente el análisis dinámico es perfectamente análogo al nuestro.

Este código genera los siguientes gráficos:



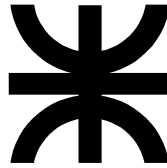
Donde vemos las funciones Posición

`sq1 = 'sin(t) * t + pi/2';`

`sq2 = 'sin(5*t) + t + 1';`

También se grafica la trayectoria realizada en el plano xy, la velocidad y la aceleración.

Se observa como el motor que controla a Q2 debe poder entregar aceleraciones muy abruptas y de mayor valor que el motor de Q1, pero el



UNIVERSIDAD TECNOLÓGICA NACIONAL

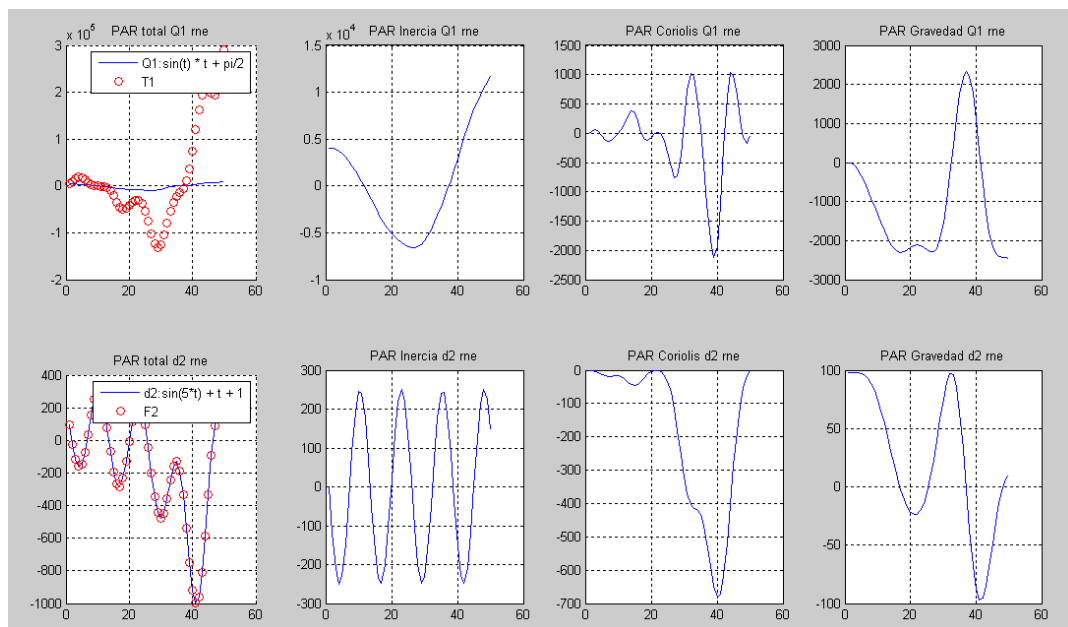
FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

mayor torque lo tiene esta ultima ya que debe soportar el peso del sistema. Esto puede observarse en las siguientes graficas por el efecto de todas las fuerzas en conjunto:

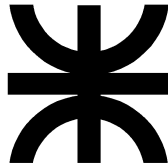


Se adjunta nuestro código MyLeg.m el cual por falta de tiempo no pudimos completar. Si bien se modificaron los parámetros dimensionales, se trabajó sobre la trayectoria y sobre la ecuación del torque.

Trascribo a continuación algunas de las modificaciones realizadas y el resultado de la simulación (el cual nos dice que estamos cometiendo algún error en la programación)

```
% Definición simbólica de la posición.
sq1 = '(sin(t)+1) * ( pi/2)'; %180<q1<pi/2
sq2 = '1'; %'(sin(t)+3) * ( pi/4)'; %5/4 pi<q2<1/4 pi

% Parametros dimensionales.
L1 = 0.18;
L2=0.18;
L0=0.13;
m1 = 0.45;
m2 = 0.456;
m0=0.235;
mp=0.185;
```

UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

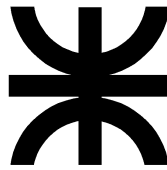
```
g = 9.8;
Ig1=g;
Ig2=g;
I1=0;
I2=0;

lg1=0.7*L1;
lg2=0.3*L2;

q1 = q(:,1);
qd1 = qd(:,1);
qdd1 = qdd(:,1);

q2 = q(:,2);
qd2 = qd(:,2);
qdd2 = qdd(:,2);
% ejemplo
% T1 = (m1*L1^2*q2.*q2).*qdd1 + 2*q2*m2.*qd1.*qd2 + m1*g*L1*cos(q1) +
m2*g*q2.*cos(q1);
%implemento la funcion x partes a fin de debugear
T1a= (m1*lg1^2 + I1 + m2*(L1^2+lg2^2+2*L1*lg2*cos(q2)) + I2).*qdd1;
T1b=(m2*(lg2^2+L1*lg2*cos(q2))+I2).*qdd2;
T1c=-(2*m2*L1*lg2*sin(q2).*qd1.*qd2)-(m2*L1*lg2*sin(q2).*qd2);
T1d=+m1*g*lg1*cos(q1)+m2*g*(L1*cos(q1)+lg2*cos(q1+q2));
T1e=+mp*L2*cos(q1+q2);
T1f=((-L1*cos(q1)-L2*cos(q1+q2)).*qd1 + (-L2*cos(q1+q2)).*qd2 +g);
T1g=mp*L2*sin(q1+q2);
T1h=((L1*cos(q1)+L2*cos(q1+q2)).*qd1 + (L2*cos(q1+q2)).*qd2 +g);
%sumo o multiplico segun corresponda los miembros de la funcion
T1=T1a + T1b +T1c +T1d +T1e.*T1f + T1g.*T1h;

%esta F2 tambien es del ejemplo
F2 = m2*qdd2 - m2*q2.*qd1.*qd1 + m2*g*sin(q1);
```



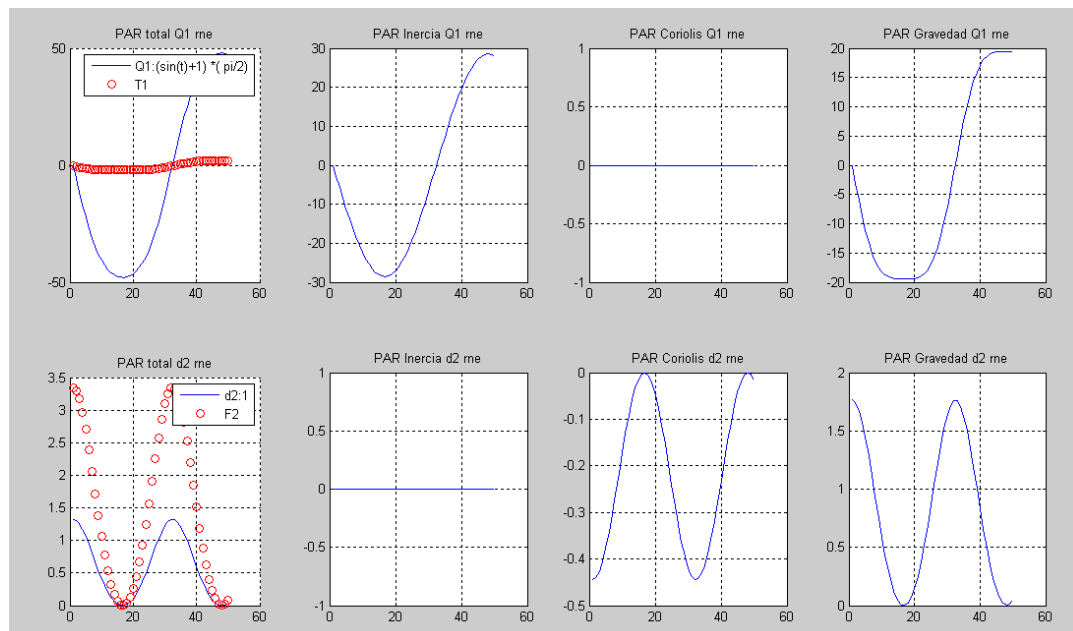
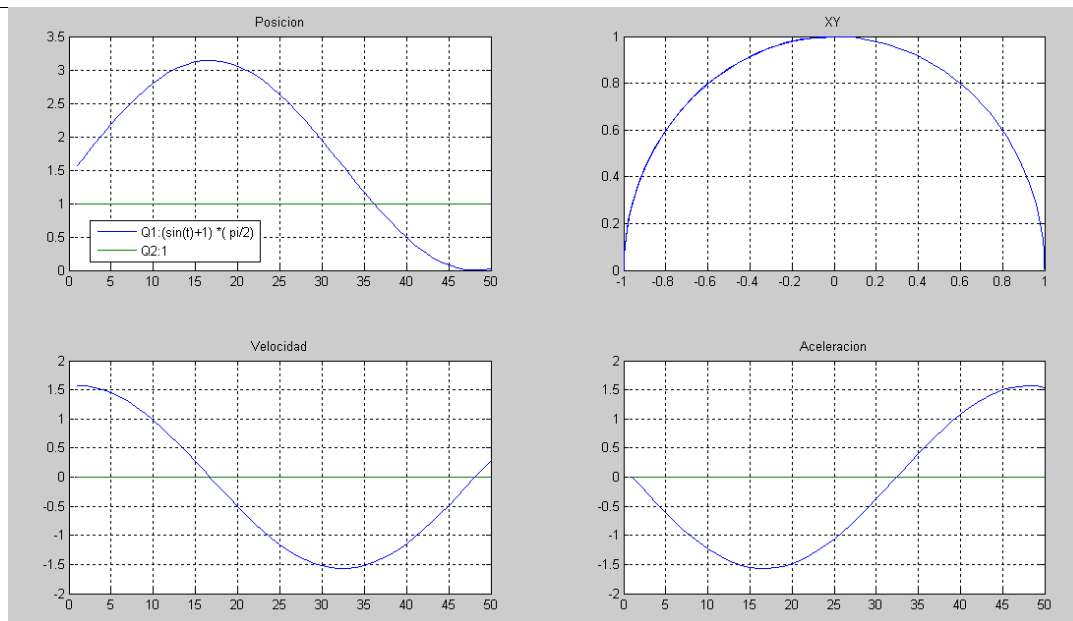
UNIVERSIDAD TECNOLÓGICA NACIONAL

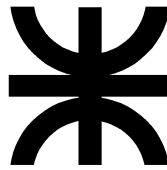
FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2





Informe Trabajo Practico N°2

Implementación en VHDL

Conceptos teóricos:

FIGURE 9: CONTROL BLOCK DIAGRAM

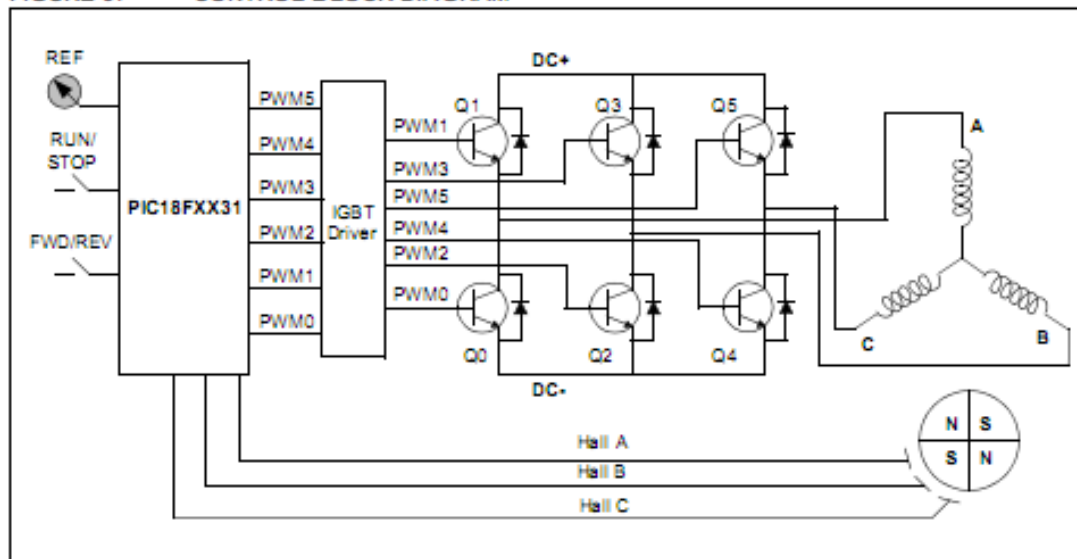
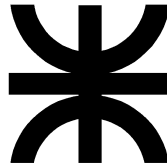


TABLE 3: SEQUENCE FOR ROTATING THE MOTOR IN CLOCKWISE DIRECTION WHEN VIEWED FROM NON-DRIVING END

Sequence #	Hall Sensor Input			Active PWMs		Phase Current		
	A	B	C			A	B	C
1	0	0	1	PWM1(Q1)	PWM4(Q4)	DC+	Off	DC-
2	0	0	0	PWM1(Q1)	PWM2(Q2)	DC+	DC-	Off
3	1	0	0	PWM5(Q5)	PWM2(Q2)	Off	DC-	DC+
4	1	1	0	PWM5(Q5)	PWM0(Q0)	DC-	Off	DC+
5	1	1	1	PWM3(Q3)	PWM0(Q0)	DC-	DC+	Off
6	0	1	1	PWM3(Q3)	PWM4(Q4)	Off	DC+	DC-

De acuerdo a este diagrama tendremos una secuencia de trabajo de 6 estados, los cuales cambiarán según lo que censen los hall.

Implementaremos el control de velocidad con PWM, aplicando esta señal solo a los Q 0, 2 y 4, y los otros Q serán activados con un pulso de continua.



UNIVERSIDAD TECNOLÓGICA NACIONAL

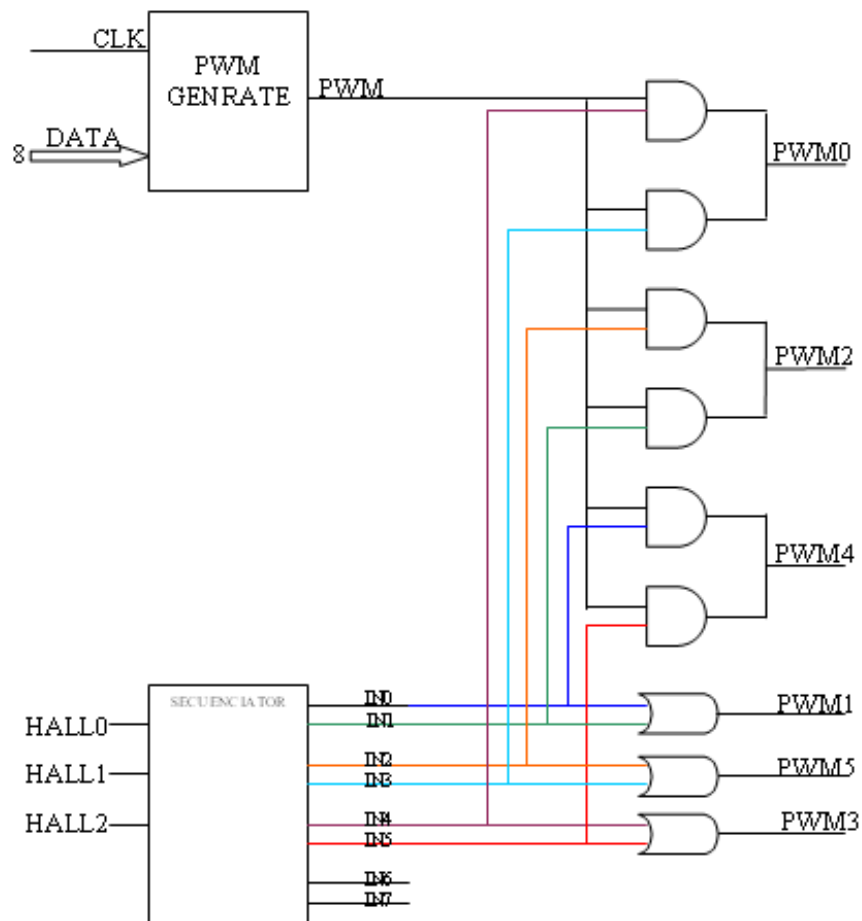
FACULTAD REGIONAL Buenos Aires

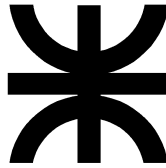
Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

Diagrama en bloques del diseño implementado:





UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

Código fuente:

```
--*****
--File: Logica.vhd
--Authors: M&M company. Matias Bobbio y Maximiliano A. Gonzalez
--author UTN FRBA - Robotica
--Date 2010.06.21
--Version 1.0.0
--Description: en base a la secuencia correspondiente, envia pwm
-- o un pulso continuo al Q que corresponda
--*****

library IEEE;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all ;
USE work.user_pkg.all;

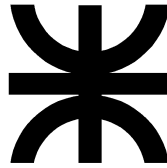
ENTITY Logica IS
PORT ( INI1,INI2,INI3,INI4,INI5,INI0,INPWM :in STD_LOGIC;
      OUT1,OUT2,OUT3,OUT4,OUT5,OUT0 :out STD_LOGIC
    );
END Logica;

ARCHITECTURE arch_logica OF Logica IS

BEGIN
OUT1<= INI0;
OUT1<= INI1;
OUT5<= INI2;
OUT5<= INI5;
OUT3<= INI4;
OUT3<= INI0;

OUT4<= INI0 and INPWM;
OUT2<= INI1 and INPWM;
OUT2<= INI2 and INPWM;
OUT0<= INI3 and INPWM;
OUT0<= INI4 and INPWM;
OUT4<= INI5 and INPWM;

END arch_logica;
```



UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

--*****

--File: Secuenciador.vhd
--brief Funcion generadora de señal senoidal
--Authors: M&M company. Matias Bobbio y Maximiliano A. Gonzalez
--author UTN FRBA - Robotica
--Date 2010.06.21
--Version 1.0.0
--Description: Este codigo toma la señales de los sensores hall
--y genera la secuencia de activacion de los Q
--*****

library IEEE;

USE ieee.std_logic_1164.all;

USE ieee.std_logic_arith.all ;

USE work.user_pkg.all;

ENTITY Secuenciador IS

PORT (Hall0, Hall1, Hall2 :in STD_LOGIC;

SecOut0, SecOut1, SecOut2, SecOut3, SecOut4, SecOut5 :out STD_LOGIC

);

END Secuenciador;

ARCHITECTURE ArchSecuenciador of Secuenciador IS

BEGIN

SecOut0 <= '1' when Hall0='0' and Hall1='0' and Hall2='1' else '0';

SecOut1 <= '1' when Hall0='0' and Hall1='0' and Hall2='0' else '0';

SecOut2 <= '1' when Hall0='1' and Hall1='0' and Hall2='0' else '0';

SecOut3 <= '1' when Hall0='1' and Hall1='1' and Hall2='0' else '0';

SecOut4 <= '1' when Hall0='1' and Hall1='1' and Hall2='1' else '0';

SecOut5 <= '1' when Hall0='0' and Hall1='1' and Hall2='1' else '0';

END ArchSecuenciador;

-- Teoria-----

--S H H H Active PWMs

--0 0 0 1 PWM1(Q1) PWM4(Q4)

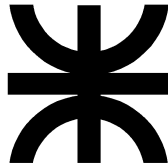
--1 0 0 0 PWM1(Q1) PWM2(Q2)

--2 1 0 0 PWM5(Q5) PWM2(Q2)

--3 1 1 0 PWM5(Q5) PWM0(Q0)

--4 1 1 1 PWM3(Q3) PWM0(Q0)

--5 0 1 1 PWM3(Q3) PWM4(Q4)



UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

--*****

--File: pwm_fpga.vhd
--Authors: M&M company. Matias Bobbio y Maximiliano A. Gonzalez
--author UTN FRBA - Robotica
--Date 2010.06.21
--Version 1.0.0
--Description: genera señal PWM
--*****

library IEEE;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all ;
USE work.user_pkg.all;

ENTITY pwm_fpga IS
PORT (clock,reset :in STD_LOGIC;
Data_value :in std_logic_vector(7 downto 0);
pwm :out STD_LOGIC
);

END pwm_fpga;

ARCHITECTURE arch_pwm OF pwm_fpga IS

SIGNAL reg_out : std_logic_vector(7 downto 0);
SIGNAL cnt_out_int : std_logic_vector(7 downto 0);
SIGNAL pwm_int, rco_int : STD_LOGIC;

BEGIN

-- 8 BIT DATA REGISTER TO STORE THE MARKING VALUES .
-- THE MARKING VALUES WILL DETERMINE THE DUTY CYCLE OF PWM OUTPUT

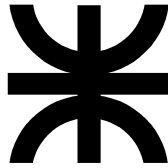
PROCESS(clock,reg_out,reset)

BEGIN

IF (reset ='1') THEN
reg_out <="00000000";
ELSIF (rising_edge(clock)) THEN
reg_out <= data_value;
END IF;

END PROCESS;

-- 8 BIT UPDN COUNTER. COUNTS UP OR DOWN BASED ON THE PWM_INT SIGNAL AND
GENERATES



UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

-- TERMINAL COUNT WHENEVER COUNTER REACHES THE MAXIMUM VALUE OR WHEN IT TRANSISTS
-- THROUGH ZERO. THE TERMINAL COUNT WILL BE USED AS INTERRUPT TO AVR FOR GENERATING
-- THE LOAD SIGNAL.
-- INC and DEC are the two functions which are used for up and down counting. They are defined in sepearate user_pakge library

PROCESS (clock,cnt_out_int,rco_int,reg_out)

BEGIN

```
IF (rco_int = '1') THEN
    cnt_out_int <= reg_out;
ELSIF rising_edge(clock) THEN
    IF (rco_int = '0' and pwm_int ='1' and cnt_out_int <"11111111")
THEN
        cnt_out_int <= INC(cnt_out_int);
        ELSE
            IF (rco_int ='0' and pwm_int ='0' and cnt_out_int >
"00000000") THEN
                cnt_out_int <= DEC(cnt_out_int);
                END IF;
            END IF;
        END IF;
    END PROCESS;
```

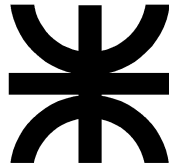
-- Logic to generate RCO signal

PROCESS(cnt_out_int, rco_int, clock,reset)

BEGIN

```
IF (reset ='1') THEN
    rco_int <='1';
ELSIF rising_edge(clock) THEN
    IF ((cnt_out_int = "11111111") or (cnt_out_int ="00000000"))
THEN
        rco_int <= '1';
        ELSE
            rco_int <='0';
            END IF;
        END IF;
```

END PROCESS;



UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

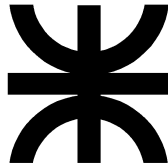
Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

-- TOGGLE FLIP FLOP TO GENERATE THE PWM OUTPUT.

```
PROCESS (clock,rco_int,reset)
BEGIN
    IF (reset = '1') THEN
        pwm_int <='0';
    ELSIF rising_edge(rco_int) THEN
        pwm_int <= NOT(pwm_int);
    ELSE
        pwm_int <= pwm_int;
    END IF;
END PROCESS;
pwm <= pwm_int;
END arch_pwm;
```



UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

```
--*****
--File: MotorControl.vhd
--Authors: M&M company. Matias Bobbio y Maximiliano A. Gonzalez
--author UTN FRBA - Robotica
--Date 2010.06.21
--Version 1.0.0
--Description: En esta ENTITY se interconectan todos los modulos
--*****
```

```
library IEEE;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all ;
USE work.user_pkg.all;
```

```
--*****
-- Declaro la entidad de Mayor jerarquia
--*****
```

```
ENTITY MotorControl Is
  PORT (DutyControl :in STD_LOGIC_VECTOR (7 downto 0);
        SHall0, SHall1, SHall2 :in STD_LOGIC;
        ClockPwm, ResetPwm :in STD_LOGIC;
        Pwm1,Pwm2,Pwm3,Pwm4,Pwm5,Pwm0 :out STD_LOGIC
        );
```

```
END MotorControl;
```

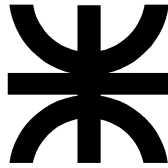
```
--*****
-- Declaro el comportamiento de la entidad
--*****
```

```
ARCHITECTURE ArchMotorControl of MotorControl IS
```

```
--*****
-- Defino que componentes voy a utilizar
--*****
```

```
COMPONENT Secuenciador Is
  PORT (Hall0, Hall1, Hall2 :in STD_LOGIC;
        SecOut0, SecOut1, SecOut2, SecOut3, SecOut4, SecOut5 :out STD_LOGIC
        );
END COMPONENT;
```

```
COMPONENT Logica IS
PORT ( INI1,INI2,INI3,INI4,INI5,INI0,INPWM :in STD_LOGIC;
```



UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

```
OUT1,OUT2,OUT3,OUT4,OUT5,OUT0 :out STD_LOGIC
);
END COMPONENT;
```

```
COMPONENT pwm_fpga
PORT (
  clock: in std_logic;
  reset: in std_logic;
  data_value: in std_logic_vector(7 downto 0);
  pwm: out std_logic
);
END COMPONENT;
```

```
--*****
-- Declaracion de Señales Internas
--*****
```

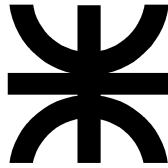
```
SIGNAL LinePwm, Secuencia0, Secuencia1, Secuencia2, Secuencia3, Secuencia4, Secuencia5:
STD_LOGIC;
SIGNAL SDutyControl : STD_LOGIC_VECTOR (7 downto 0);
SIGNAL SSecuencia : STD_LOGIC_VECTOR (1 downto 0);
SIGNAL SClockPwm, SResetPwm : STD_LOGIC;
SIGNAL SPwm1,SPwm2,SPwm3,SPwm4,SPwm5,SPwm0 : STD_LOGIC;
```

BEGIN

```
--*****
-- Describo interconexion de los bloques
--*****
```

```
BloquePwm : pwm_fpga
PORT MAP
(
  clock => ClockPwm,
  reset => ResetPwm,
  data_value => DutyControl,
  pwm => LinePwm
);
```

```
BloqueSecuenciador : Secuenciador
PORT MAP
(
  Hall0 => SHall0,
  Hall1 => SHall1,
  Hall2 => SHall2,
  SecOut0 => Secuencia0,
  SecOut1 => Secuencia1,
```



UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

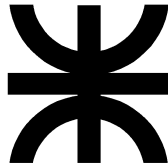
```
SecOut2 => Sequence2,  
SecOut3 => Sequence3,  
SecOut4 => Sequence4,  
SecOut5 => Sequence5  
);
```

BloqueLogica : Logica

PORT MAP

```
(  
INI0 => Sequence0,  
INI1 => Sequence1,  
INI2 => Sequence2,  
INI3 => Sequence3,  
INI4 => Sequence4,  
INI5 => Sequence5,  
INPWM => LinePwm,  
OUT1 => Pwm1,  
OUT2 => Pwm2,  
OUT3 => Pwm3,  
OUT4 => Pwm4,  
OUT5 => Pwm5,  
OUT0 => Pwm0  
);
```

END ArchMotorControl;



UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

```
--*****
--File: Test Bench Secuenciador
--brief Funcion generadora de señal senoidal
--Authors: M&M company. Matias Bobbio y Maximiliano A. Gonzalez
--author UTN FRBA - Robotica
--Date 2010.06.21
--Version 1.0.0
--Description: Este codigo se encarga de simular la generacion
--de las señales de los hall, delays y cambios de velocidad
--*****
```

LIBRARY ieee;

use ieee.std_logic_1164.all;

```
--*****
-- Declaracion de la entidad de Test Bench
--*****
```

ENTITY MotorControl_test_bench IS

END MotorControl_test_bench;

```
--*****
-- Descripcion del comportamiento del test bench
--*****
```

ARCHITECTURE arch_test_bench OF MotorControl_test_bench IS

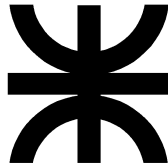
COMPONENT MotorControl Is

```
  PORT (DutyControl :in STD_LOGIC_VECTOR (7 downto 0);
        SHall0, SHall1, SHall2 :in STD_LOGIC;
        ClockPwm, ResetPwm :in STD_LOGIC;
        Pwm1,Pwm2,Pwm3,Pwm4,Pwm5,Pwm0 :out STD_LOGIC
        );
```

END COMPONENT;

```
--*****
-- Declaracion de señales internas
--*****
```

```
signal sig_reset: STD_LOGIC;
signal sig_Data_value: STD_LOGIC_VECTOR(7 downto 0) := "00011001";
```



UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

```
signal sig_clock: STD_LOGIC;  
signal one : STD_LOGIC := '1';  
signal zero : STD_LOGIC := '0';
```

```
signal SigSec0, SigSec1, SigSec2: STD_LOGIC;  
signal Sig_Pwm1, Sig_Pwm2, Sig_Pwm3, Sig_Pwm4, Sig_Pwm5, Sig_Pwm0 : STD_LOGIC;
```

```
shared variable ENDSIM: boolean:=false;  
constant clk_period:TIME:=200 ns;
```

BEGIN

```
__*****  
-- Descripcion de Ports  
__*****
```

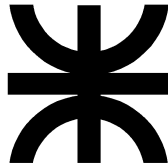
```
Bloque : MotorControl  
PORT MAP  
(  
  DutyControl => sig_Data_value,  
  SHall0 => SigSec0,  
  SHall1 => SigSec1,  
  SHall2 => SigSec2,  
  ClockPwm => sig_clock,  
  ResetPwm => sig_reset,  
  Pwm1 => Sig_Pwm1,  
  Pwm2 => Sig_Pwm2,  
  Pwm3 => Sig_Pwm3,  
  Pwm4 => Sig_Pwm4,  
  Pwm5 => Sig_Pwm5,  
  Pwm0 => Sig_Pwm0  
);
```

```
__*****  
-- Generacion de la señal de clock  
__*****
```

```
clk_gen: PROCESS
```

BEGIN

```
  IF ENDSIM = FALSE THEN  
    sig_clock <= '1';  
    wait for clk_period/2;  
    sig_clock <= '0';  
    wait for clk_period/2;  
  ELSE
```



UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

```
wait;  
END IF;
```

```
END PROCESS;
```

```
--*****  
-- Generacion de Secuencia de conmutacion  
--*****
```

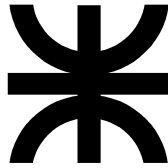
SecuenceGenerator: PROCESS

BEGIN

```
sig_reset <= '1';  
SigSec0 <= '0';  
SigSec1 <= '0';  
SigSec2 <= '0';  
wait for 20 us;  
sig_reset <= '0';
```

Bucle: LOOP -- aca emulo el sensado de los hall

```
SigSec0 <= '0';  
SigSec1 <= '0';  
SigSec2 <= '1';  
wait for 150 us;  
SigSec0 <= '0';  
SigSec1 <= '0';  
SigSec2 <= '0';  
wait for 150 us;  
SigSec0 <= '1';  
SigSec1 <= '0';  
SigSec2 <= '0';  
wait for 150 us;  
SigSec0 <= '1';  
SigSec1 <= '1';  
SigSec2 <= '0';  
wait for 150 us;  
SigSec0 <= '1';  
SigSec1 <= '1';  
SigSec2 <= '1';  
wait for 150 us;  
SigSec0 <= '0';  
SigSec1 <= '1';  
SigSec2 <= '1';  
wait for 150 us;  
-- Hall Sequence Hall Sensor Input Active PWM  
--1 0 0 1 PWM1(Q1) PWM4(Q4)  
--2 0 0 0 PWM1(Q1) PWM2(Q2)
```



UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

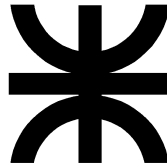
Informe Trabajo Practico N°2

```
--3 1 0 0 PWM5(Q5) PWM2(Q2)
--4 1 1 0 PWM5(Q5) PWM0(Q0)
--5 1 1 1 PWM3(Q3) PWM0(Q0)
--6 0 1 1 PWM3(Q3) PWM4(Q4)
```

CASE sig_Data_value IS -- de esta manera pega toda una vuelta con cada velocidad
(ancho del pwm)

```
    WHEN "00011001" => sig_Data_value<="01000000";
    WHEN "01000000" => sig_Data_value<="10000000";
    WHEN "10000000" => sig_Data_value<="11000000";
    WHEN "11000000" => sig_Data_value<="11100110";
    WHEN "11100110" => sig_Data_value<="00011001";
    WHEN OTHERS    => sig_Data_value<="00011001";
END CASE;
```

```
--IF (sig_Data_value = "01000000") THEN
    --sig_Data_value <= "11000000";
--ELSE
    --sig_Data_value <= "01000000";
--END IF;
END LOOP Bucle;
wait;
END PROCESS SequenceGenerator;
END arch_test_bench;
```

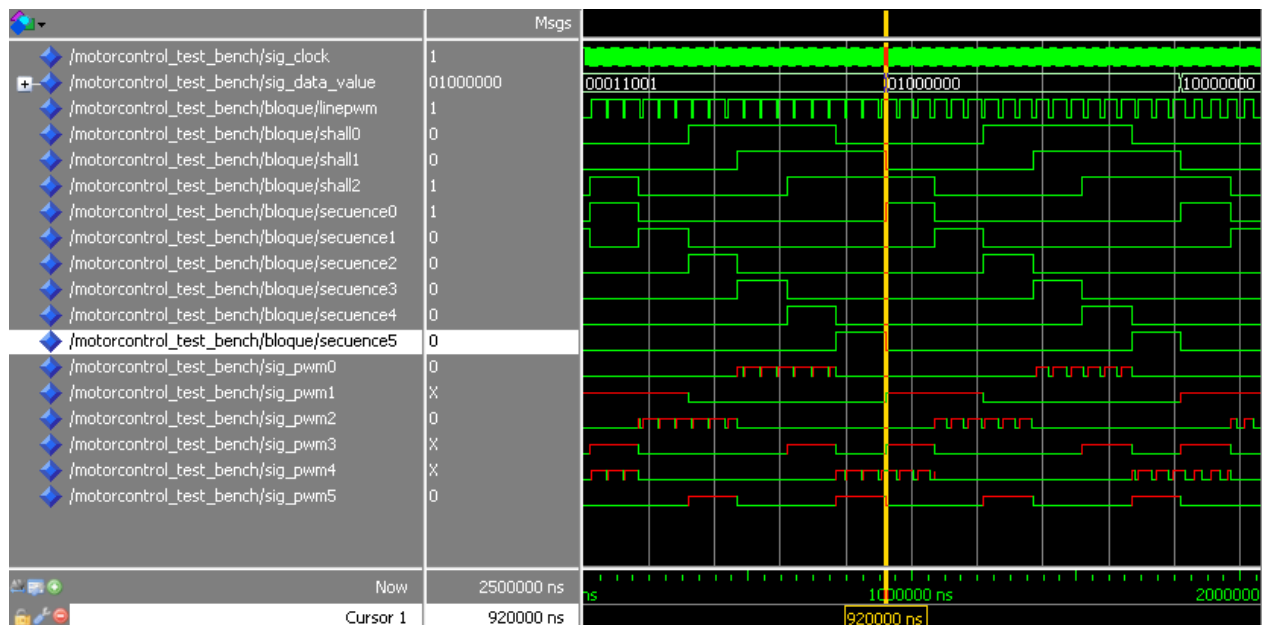
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL Buenos Aires
Departamento de Electrónica
Cátedra: Robótica - Plan 1995

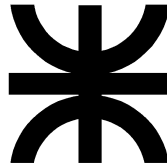
Informe Trabajo Practico N°2

Resultado de la simulación

Para demostrar el correcto funcionamiento hemos plotado las siguientes señales:

- clock
- sig_data_value: el cual varia la velocidad sig_hall: censado de los hall
- sequence: indica la secuencia de activación (6 estados)
- sig_pwm: señal pwm entregada a los Q, o cont. Según corresponda





UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

Conclusiones

Referente al estudio del modelo dinámico:

Nos dimos cuenta que el mismo es imprescindible para realizar la simulación del movimiento del robot, diseño de la estructura mecánica del robot, especificación de los motores.

Para ello se tienen en cuenta factores como masa, centros de gravedad de las partes, inercias, velocidades, fuerzas centrífugas, fuerza de coriolis, etc.

En el estudio dinámico : "La trayectoria depende de la aplicación", por lo que debe optimizarse la trayectoria ya sea por método de aproximación polinomial de 3er orden, 5to, cicloidal, trapezoidal, doble S, etc. Con el objetivo de obtener las curvas deseadas (comportamiento deseado), de velocidades máximas, discontinuidades de aceleración, Jerk, etc.

Respecto al desarrollo de la practica en si, aprendimos a utilizar una nueva herramienta dentro de el entorno de Matlab, el toolbox de robótica, el cual es una herramienta poderosa para realizar este tipo de análisis.

Referente a la programación en VHDL:

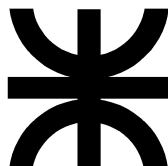
El realizar este TP nos permitió familiarizarnos con un nuevo lenguaje de programación VHDL. El mismo se caracteriza por ser un lenguaje de descripción de hardware.

Se utilizo para programar una FPGA, cuyo objetivo era controlar un motor de continua tipo brush less (ya que trabajamos energizando el estator y el rotor es del tipo imán permanente para evitar el desgaste)

Se realizo la simulación a través de nuestro Test Bench, con el objetivo de predecir el comportamiento del diseño. Analizando la forma de onda de las señales en puntos precisos se pudo comprobar el correcto funcionamiento.

El paso siguiente a lo realizado en este TP, seria generar la síntesis del circuito y bajarla finalmente a la FPGA.

Cabe aclarar que esta forma de trabajo permite implementar circuitos integrados de muy alta velocidad.



UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL Buenos Aires

Departamento de Electrónica

Cátedra: Robótica - Plan 1995

Informe Trabajo Practico N°2

.....
Firma y aclaración Representante