

# UNIVERSIDAD TECNOLÓGICA NACIONAL

## FACULTAD REGIONAL BUENOS AIRES

**Materia:** Robótica

**División:** R6055

**Tema:** Análisis dinámico de un robot e implementación en FPGA

**Profesor:** Ing. Hernán Gianetta

**Revisión del documento:** V1.00 (7-7-09)

**Alumnos:**

- Bole, Nicolás
- Ríos, Leandro
- Zbucki, Hernán

### Objetivos

- Simular mediante MATLAB el análisis dinámico de un robot *Five Bar Linkage*
- Incorporar la variabilidad y dependencia de los parámetros dinámicos del robot, cómo el centro de masa, las longitud, la gravedad, la inercia, etc en función de la cinemática del robot.
- Asimilar lenguaje y manejo VHDL.
- Utilizar el Quartus II y el ModelSim, y familiarizarse con estos programas.

### Enunciado

- Hacer una introducción sobre la dinámica del robot *Five Bar Linkage*.
- Hacer el análisis dinámico de dicho robot.
- Generar un modelo VHDL y FPGA de un control trifásico PWM.
- Realizar la simulación de dicho controlador.
- Escribir las conclusiones.

### Introducción

El robot a analizar es muy peculiar. Se llama *Five Bar Linkage* lo que en inglés significa *robot de cinco barras interconectadas*. La ventaja de este robot, muy comúnmente usado en las plantas de ensamblaje para soldar, manejar piezas, etc., es que los dos motores que permiten su movimiento están en la base del robot, a comparación de un brazo de 2 GDL común, los cuáles en cada *joint (juntura, unión)* necesita un motor. Esta característica permite reducir el momento de inercia mejorando el rendimiento de potencia utilizada, achicando los motores y mejorando el control del brazo.

Para que esta característica suceda se debe cumplir con la siguiente característica, dibujada en la figura 1:

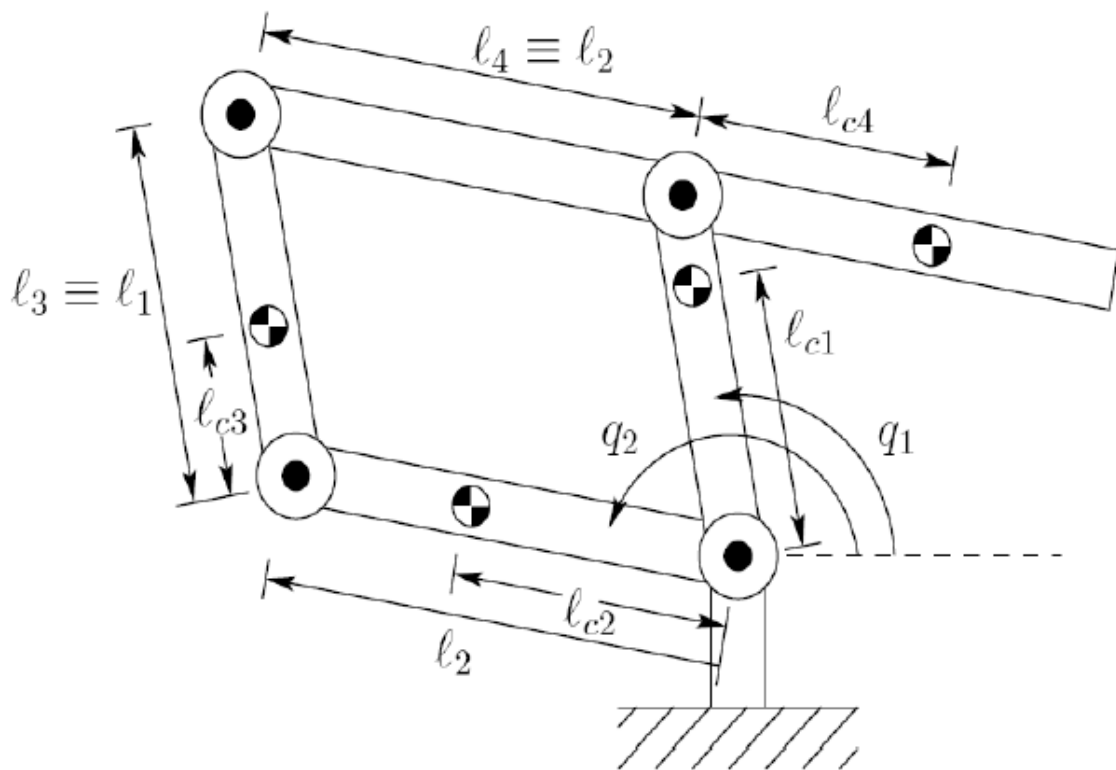


Figura 1

- **L4 debe ser igual a L2**
- **L3 debe ser igual a L1**

Si se cumplen estas características, los dos motores pasan a tener movimientos independientes.

El análisis dinámico del robot es el siguiente:

- Escribimos las matrices de los centros de masa de cada uno de los miembros.

Brazo 1:

$$\begin{bmatrix} x_{c1} \\ y_{c1} \end{bmatrix} = \begin{bmatrix} l_{c1} \cos q_1 \\ l_{c1} \sin q_1 \end{bmatrix}$$

Brazo 2:

$$\begin{bmatrix} x_{c2} \\ y_{c2} \end{bmatrix} = \begin{bmatrix} l_{c2} \cos q_2 \\ l_{c2} \sin q_2 \end{bmatrix}$$

Brazo 3:

$$\begin{bmatrix} x_{c3} \\ y_{c3} \end{bmatrix} = \begin{bmatrix} \ell_{c2} \cos q_1 \\ \ell_{c2} \sin q_2 \end{bmatrix} + \begin{bmatrix} \ell_{c3} \cos q_1 \\ \ell_{c3} \sin q_1 \end{bmatrix}$$

Brazo 4

$$\begin{bmatrix} x_{c4} \\ y_{c4} \end{bmatrix} = \begin{bmatrix} \ell_1 \cos q_1 \\ \ell_1 \sin q_1 \end{bmatrix} + \begin{bmatrix} \ell_{c4} \cos(q_2 - \pi) \\ \ell_{c4} \sin(q_2 - \pi) \end{bmatrix}$$

$$\begin{bmatrix} x_{c4} \\ y_{c4} \end{bmatrix} = \begin{bmatrix} \ell_1 \cos q_1 \\ \ell_1 \sin q_1 \end{bmatrix} - \begin{bmatrix} \ell_{c4} \cos q_2 \\ \ell_{c4} \sin q_2 \end{bmatrix}$$

- Calculamos las velocidades de los centros de masa

$$\begin{aligned} v_{c1} &= \begin{bmatrix} -\ell_{c1} \sin q_1 & 0 \\ \ell_{c1} \cos q_1 & 0 \end{bmatrix} \dot{q} \\ v_{c2} &= \begin{bmatrix} 0 & -\ell_{c2} \sin q_2 \\ 0 & \ell_{c2} \cos q_2 \end{bmatrix} \dot{q} \\ v_{c3} &= \begin{bmatrix} -\ell_{c3} \sin q_1 & -\ell_2 \sin q_2 \\ \ell_{c3} \cos q_1 & \ell_2 \cos q_2 \end{bmatrix} \dot{q} \\ v_{c4} &= \begin{bmatrix} -\ell_1 \sin q_1 & \ell_{c4} \sin q_2 \\ \ell_1 \cos q_1 & \ell_{c4} \cos q_2 \end{bmatrix} \dot{q} \end{aligned}$$

- Calculamos las velocidades angulares

$$\omega_1 = \omega_3 = \dot{q}_1 k, \omega_2 = \omega_4 = \dot{q}_2 k.$$

- Calculamos la matriz de inercia

$$D(q) = \sum_{i=1}^4 m_i J_{vc}^T J_{vc} + \begin{bmatrix} I_1 + I_3 & 0 \\ 0 & I_2 + I_4 \end{bmatrix}$$

- Sustituyendo por las matrices de la página anterior y usando identidades trigonométricas,

$$\begin{aligned} d_{11}(q) &= m_1 \ell_{c1}^2 + m_3 \ell_{c3}^2 + m_4 \ell_1^2 + I_1 + I_3 \\ d_{12}(q) &= d_{21}(q) = (m_3 \ell_2 \ell_{c3} - m_4 \ell_1 \ell_{c4}) \cos(q_2 - q_1) \\ d_{22}(q) &= m_2 \ell_{c2}^2 + m_3 \ell_2^2 + m_4 \ell_{c4}^2 + I_2 + I_4 \end{aligned}$$

- Definimos como condición de diseño

$$m_3 \ell_2 \ell_{c3} = m_4 \ell_1 \ell_{c4}$$

- La energía potencial queda como,

$$\begin{aligned} P &= g \sum_{i=1}^4 y_{ci} \\ &= g \sin q_1 (m_1 \ell_{c1} + m_3 \ell_{c3} + m_4 \ell_1) \\ &+ g \sin q_2 (m_2 \ell_{c2} + m_3 \ell_2 - m_4 \ell_{c4}) \end{aligned}$$

- Rescribiendo las ecuaciones anteriores

$$\begin{aligned} \phi_1 &= g \cos q_1 (m_1 \ell_{c1} + m_3 \ell_{c3} + m_4 \ell_1) \\ \phi_2 &= g \cos q_2 (m_2 \ell_{c2} + m_3 \ell_2 - m_4 \ell_{c4}) \end{aligned}$$

- Las ecuaciones dinámicas quedan como

$$d_{11} \ddot{q}_1 + \phi_1(q_1) = \tau_1, \quad d_{22} \ddot{q}_2 + \phi_2(q_2) = \tau_2$$

De esta forma T1 y T2 simbolizan al torque del motor 1 y 2 respectivamente. Vemos que dicho parámetro depende entonces de la energía potencial (que a su vez también depende del desplazamiento angular), de los elementos de la matriz de inercia y de la aceleración angular.

Ahora simularemos algunos casos, imponiéndole valores a la física del robot y a sus centros de masa; luego supondremos un valor a conseguir de torque del robot y allí veremos que cuales serán los desplazamientos angulares necesarios para lograrlo.

Variables (primer caso):

- 1) Suponemos velocidades angulares de rampa. Las longitudes de las barras, sus centros de masa y pesos son elegidos aleatoriamente para ver cómo responde el sistema, luego se acotaran a valores razonables según lo que pretendamos alcanzar. Los valores, entonces son los siguientes:

- $w1=w2=1.2 \cdot t$  [rad/seg]
- $M1=10\text{kg}$ ,  $M2=10\text{kg}$ ,  $M3=10\text{kg}$ ,  $M4=10\text{kg}$
- $L1=L3=1\text{m}$ ,  $L2=L4=1\text{m}$
- $G=9.8\text{m/s}^2$
- $Lc1=0.8$ ,  $Lc2=0.15\text{m}$ ,  $Lc3=0.4\text{m}$ ,  $Lc4=0.125\text{m}$ .

Analizamos  $q1$  y  $q2$  de 0 a 90 grados

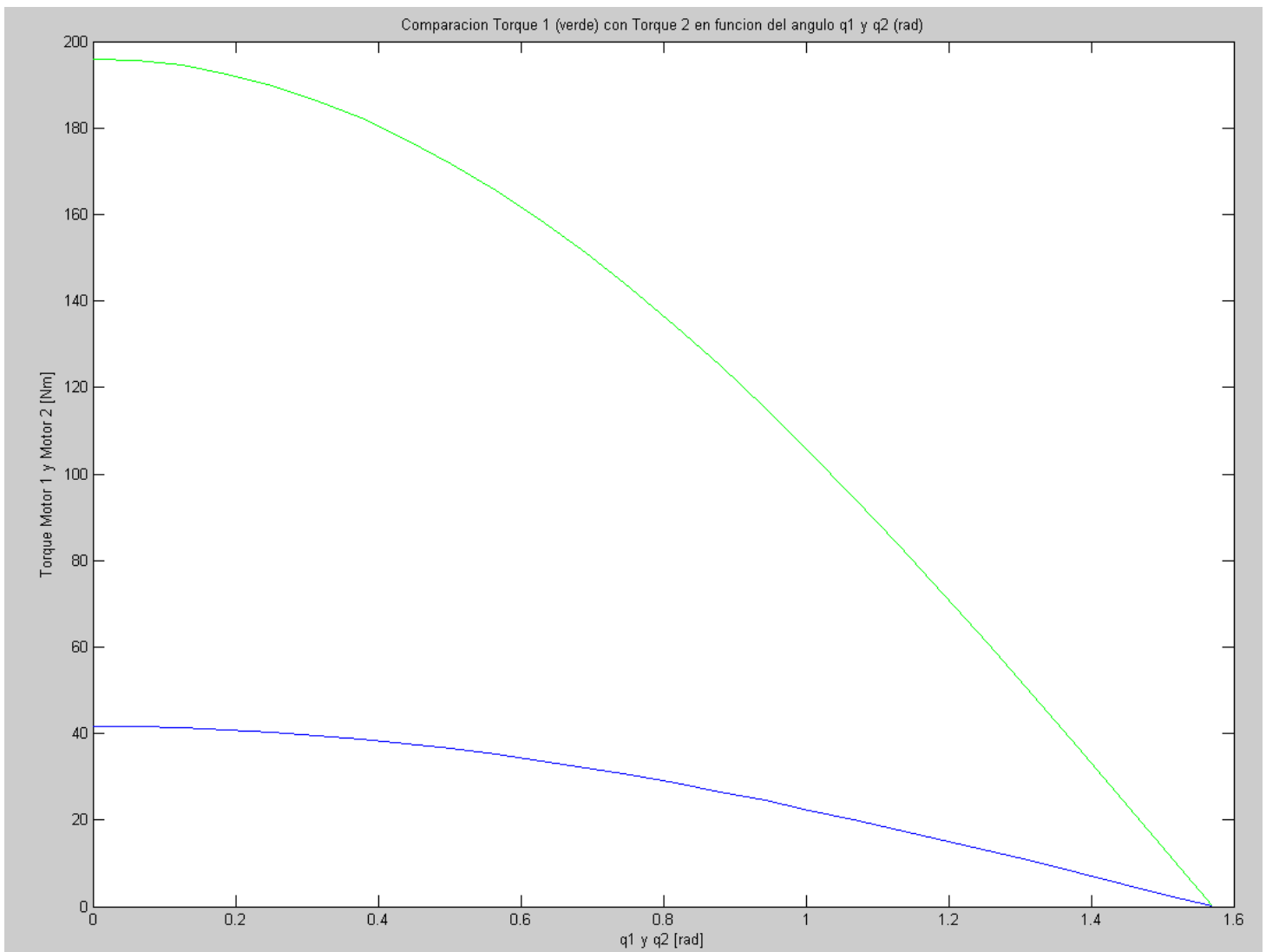


Figura 2

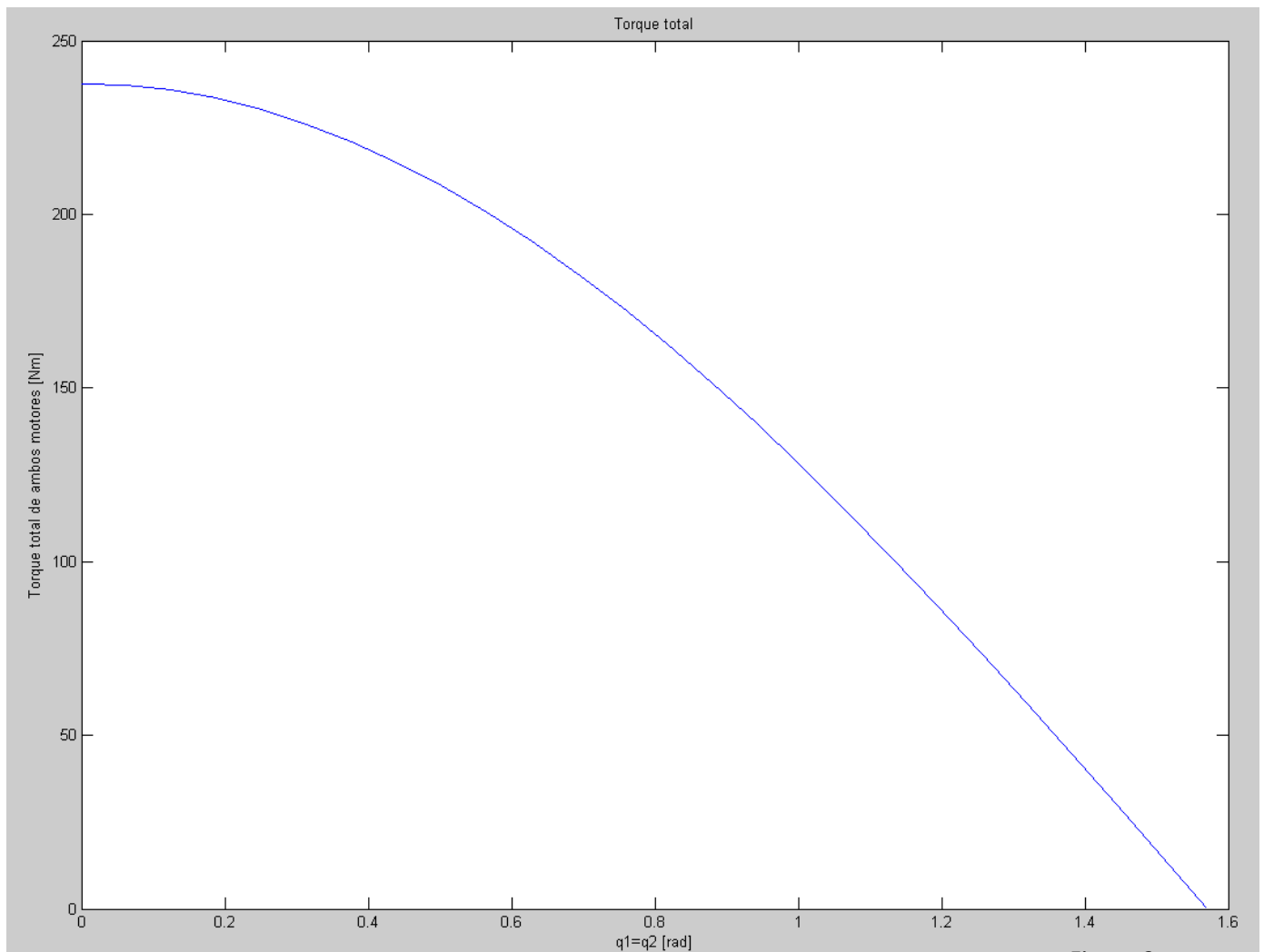


Figura 3

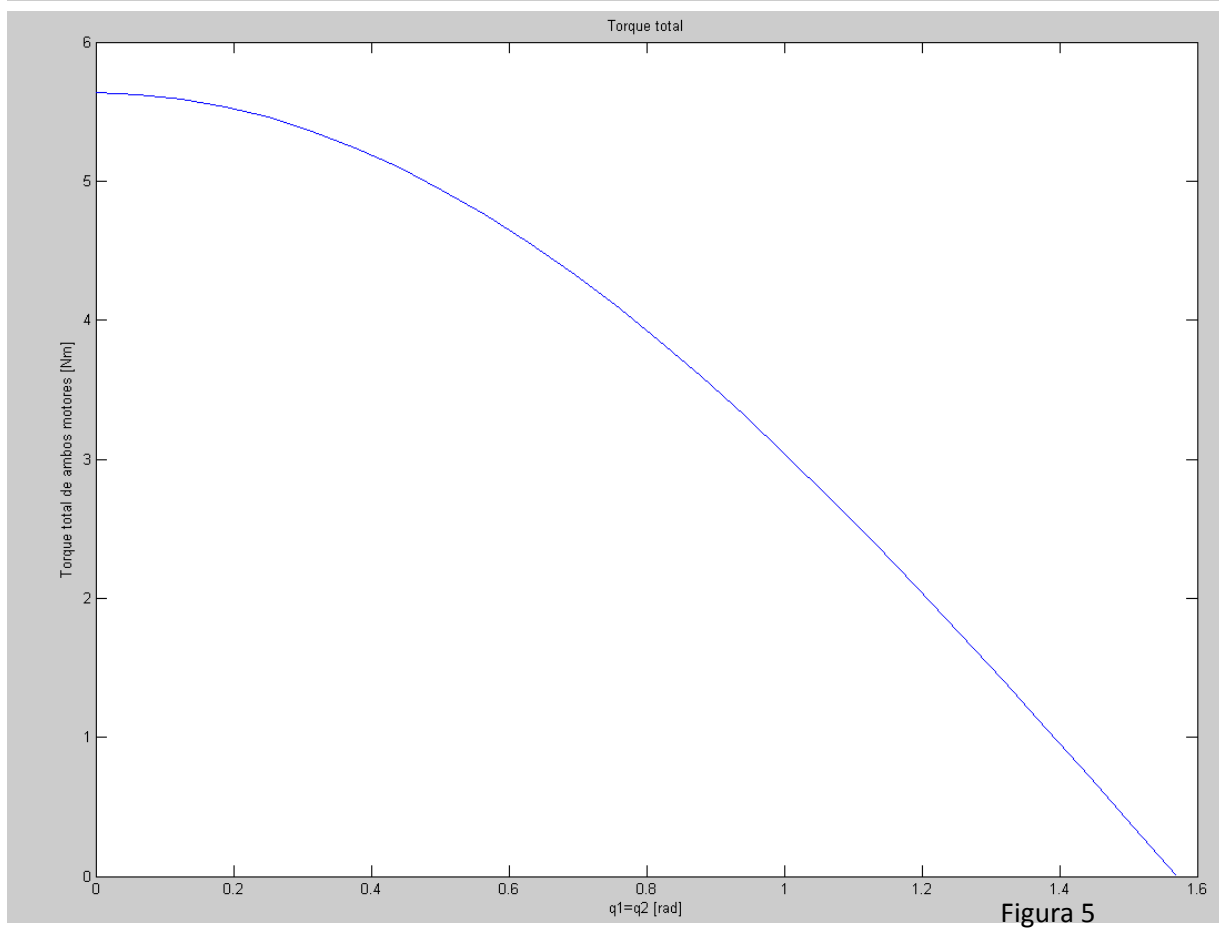
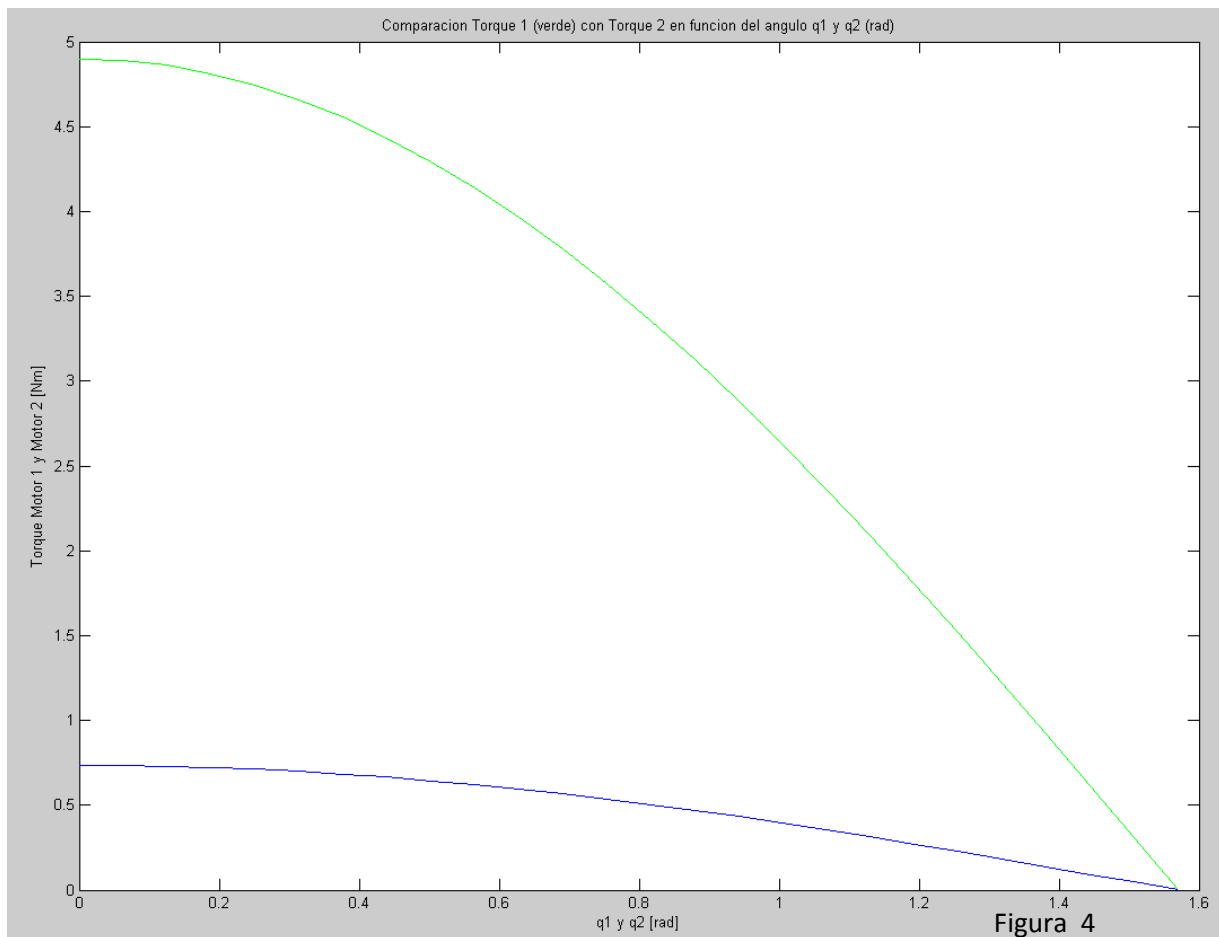
#### Conclusiones:

- El torque del primer motor es mucho mayor que el del segundo, pero este último se mantiene con una menor variación según las posiciones angulares.
- Pesos tales como 10kg por barra genera un gran torque para el motor 1; en los próximos cálculos intentaremos analizar la influencia de los factores centro de masa, longitud de barra y peso sobre el torque.
- El torque pico es menor que el torque pico de un BLDC (no es ni el doble)

2) Suponemos que alivianamos la física del robot y lo hacemos de un material más ligero y resistente. Los valores son los siguientes:

- $w1=w2=1,2 \text{ *t rad/seg}$
- $M1=0.5\text{kg}$ ,  $M2=0.5\text{kg}$ ,  $M3=0.5\text{kg}$ ,  $M4=0.5\text{kg}$
- $L1=L3=0.5\text{m}$ ,  $L2=L4=0.25\text{m}$
- $G=9.8\text{m/s}^2$
- $Lc1=0.4\text{m}$ ,  $Lc2=0.075\text{m}$ ,  $Lc3=0.2\text{m}$ ,  $Lc4=0.125\text{m}$ .

Analizamos q1 y q2 de 0 a 90 grados



### Conclusiones:

- La relación entre torques de los motores 1 y 2 se mantiene constante en función que mantengamos constante la relación de pesos de las barras que usamos en la primera simulación.
- Se mantiene la característica de poca variabilidad del torque del motor 2.
- El torque total e individual se reduce notablemente haciéndolo dependiente del peso de la barra. Suponiendo que la tecnología y los materiales usados para la construcción del robot son las requeridas, trataremos de incrementar los torques del motor variando los parámetros longitud de barra y momento de inercia.

3) Ahora intentamos reducir el momento de inercia al máximo, por lo que construimos al robot de modo de conseguir los centros de masa lo más cerca posible de los motores. Mantenemos los demás valores. Por lo tanto, los valores son los siguientes:

- $w_1=w_2=1,2 \text{ *t rad/seg}$
- $M_1=0.5\text{kg}$  ,  $M_2=0.5\text{kg}$ ,  $M_3=0.5\text{kg}$ ,  $M_4=0.5\text{kg}$
- $L_1=L_3=0.5\text{m}$ ,  $L_2=L_4=0.25\text{m}$
- $G=9.8\text{m/s}^2$
- $L_{c1}=0.1\text{m}$ ,  $L_{c2}=0.05\text{m}$ ,  $L_{c3}=0.1\text{m}$ ,  $L_{c4}=0.05\text{m}$ .

### Analizamos q1 y q2 de 0 a 90 grados

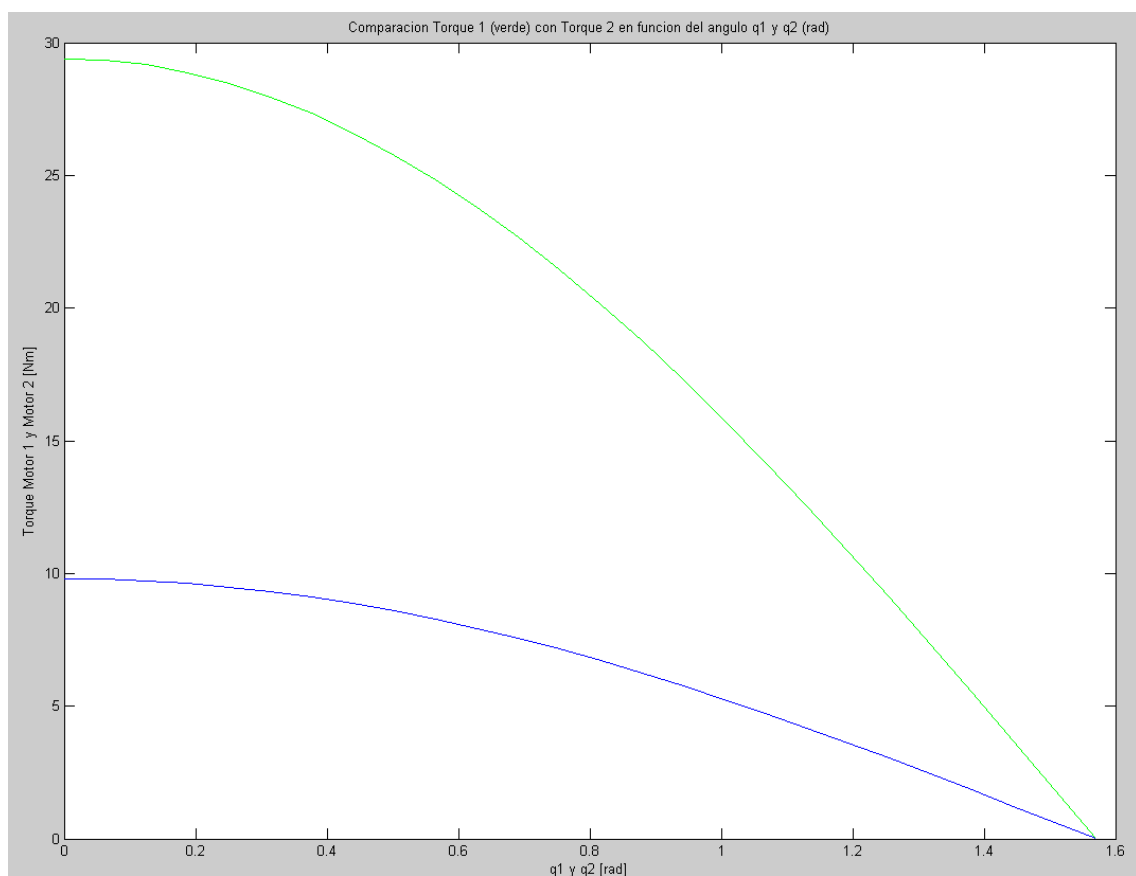


Figura 6



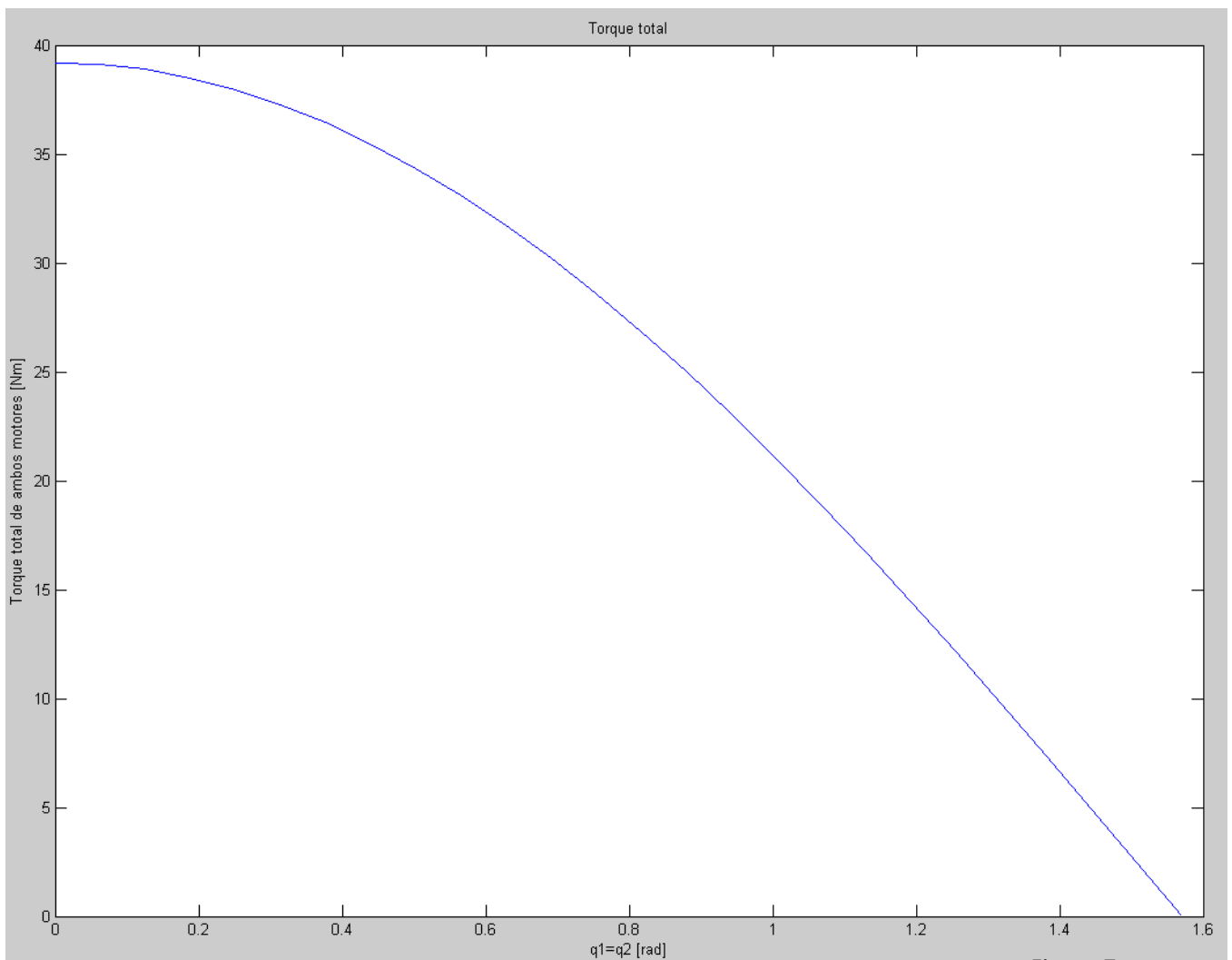


Figura 7

#### Conclusiones:

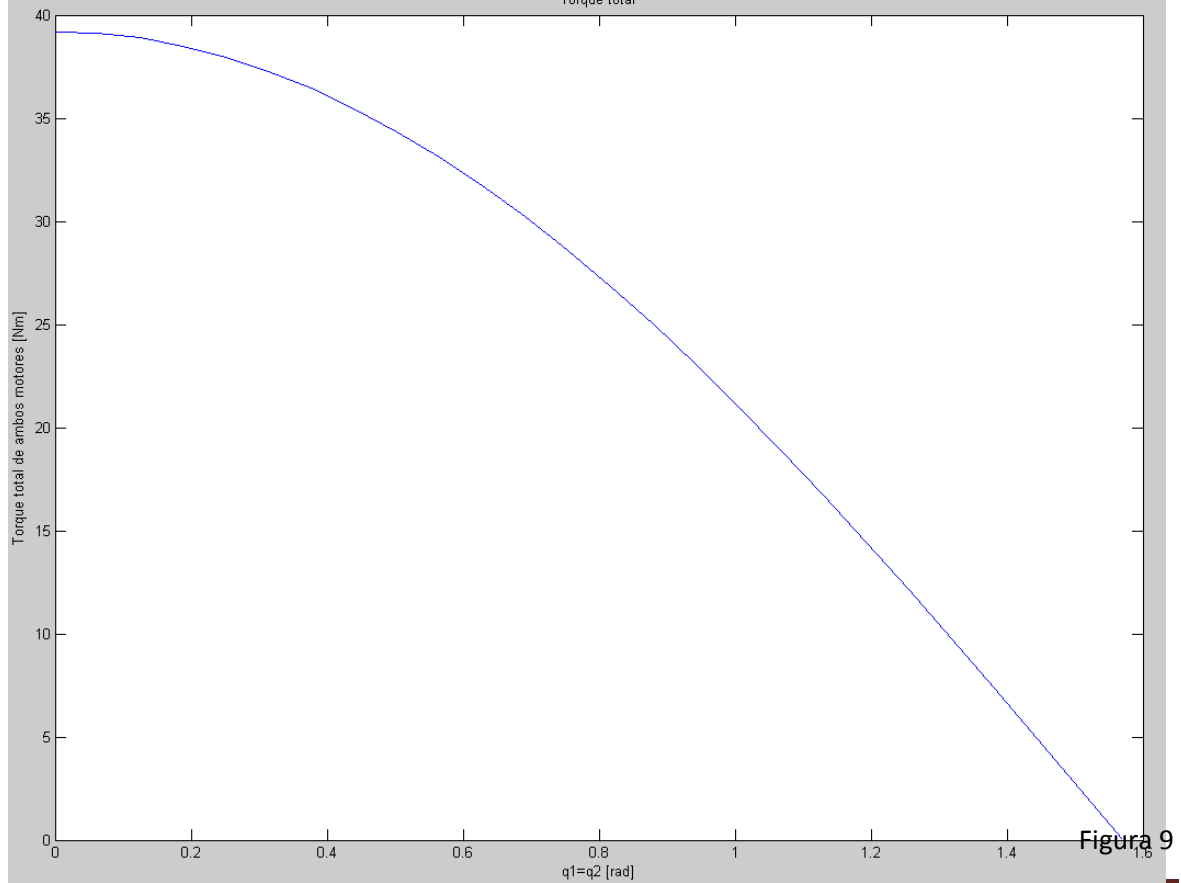
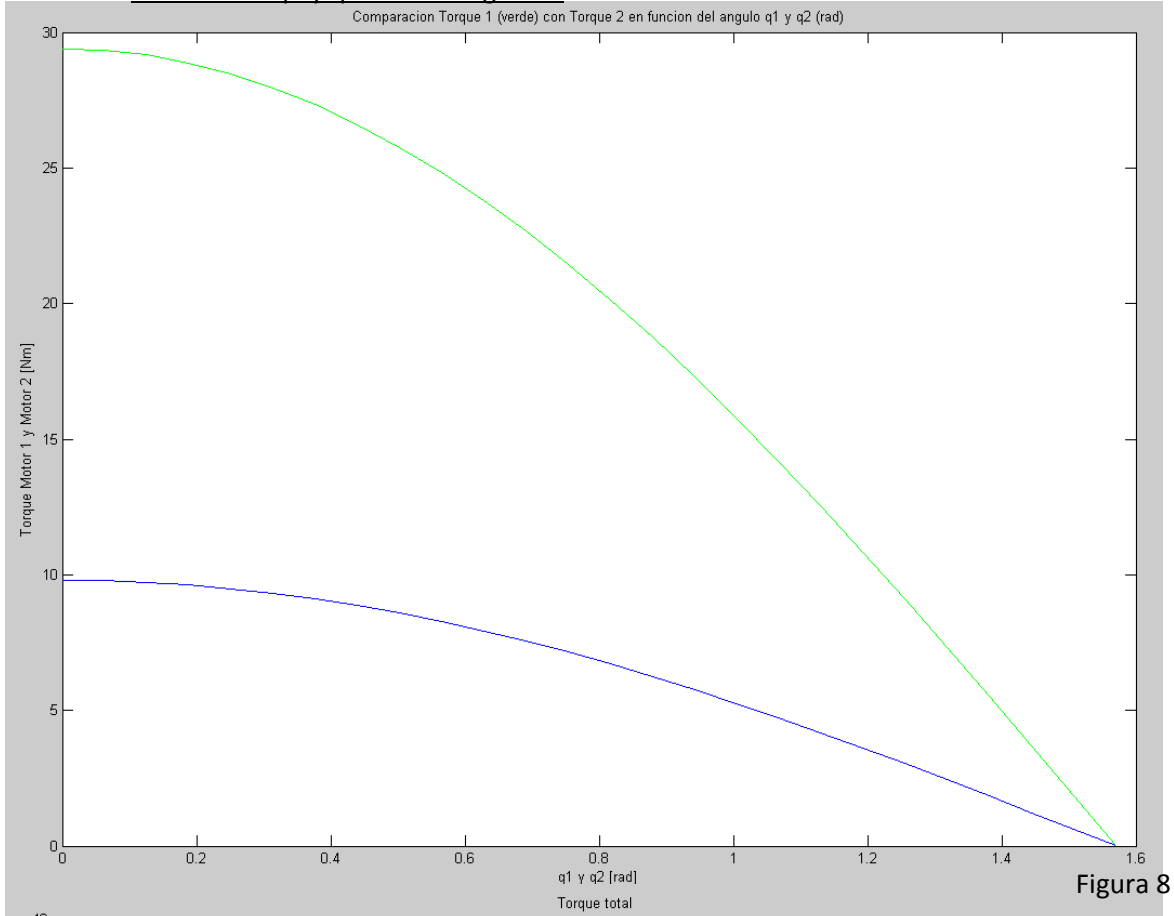
- Aumenta considerablemente el torque cuando los centros de masa se acercan al eje de los motores; este cálculo demuestra matemáticamente la ventaja en torque obtenido del five bar linkage con respecto a motores de 2 o más GDL comunes.
- Tomando como ejemplo un motor de ABB de su catalogo,(cuyo link es [http://library.abb.com/GLOBAL/SCOT/scot201.nsf/VerityDisplay/BD0499BCC7D1135DC1257354004A1D2C/\\$File/MS\\_flyer\\_EN\\_revB\\_LR.pdf](http://library.abb.com/GLOBAL/SCOT/scot201.nsf/VerityDisplay/BD0499BCC7D1135DC1257354004A1D2C/$File/MS_flyer_EN_revB_LR.pdf)), específicamente el MS4836Mx, su torque máximo es 31.1Nm y el nominal es 10.5Nm. Veremos que habría que modificar para alcanzar ese torque máximo con nuestro motor. Ya vimos que acercando los centros de masa vemos incrementos de torque: ahora intentaremos agrandar las dimensiones del robot manteniendo sus materiales de construcción livianos y resistentes.

4)

Entonces, hay conseguir más torque y con poco peso, como el que tenemos ahora. Hacemos el cálculo de agrandar el tamaño de las barras dejando cercanos a los centros de masa de los motores con sus respectivos ejes. Los valores son los siguientes:

- $M1=0.5\text{kg}$  ,  $M2=0.5\text{kg}$ ,  $M3=0.5\text{kg}$ ,  $M4=0.5\text{kg}$
- $L1=L3=2\text{m}$ ,  $L2=L4=2\text{m}$
- $G=9.8\text{m/s}^2$
- $Lc1=0.1\text{m}$ ,  $Lc2=0.05\text{m}$ ,  $Lc3=0.1\text{m}$ ,  $Lc4=0.05\text{m}$ .
- $w1=w2=1,2 \cdot t \text{ rad/seg}$

### Analizamos q1 y q2 de 0 a 90 grados



### Conclusiones:

- Los torques máximos permanecen inmutables a esa variación de dimensiones que se le aplicó al cálculo. De todas formas se dio esta situación solo para esta gama de valores: creemos que si el robot es pesado agrandar su tamaño puede producir variaciones en dichos torques.
- Los centros de masa se mantuvieron constantes entre el cálculo 3) y 4), por lo tanto si bien al agrandar las dimensiones la relación (dist. centro de masa/largo barra) se achicó, el torque permaneció inmutable. La longitud del centro de masa al eje es más relevante que el largo de la barra en si.

5)

Como no lo conseguimos en el análisis anterior, intentamos aumentando solo el peso de la barra 4. Mantenemos todos los valores anteriores excepto  $m_4$ . Los valores son los siguientes:

- $w_1=w_2=1,2 \text{ rad/seg}$
- $M_1=0.5\text{kg}$ ,  $M_2=0.5\text{kg}$ ,  $M_3=0.5\text{kg}$ ,  $M_4=8\text{kg}$
- $L_1=L_3=2\text{m}$ ,  $L_2=L_4=2\text{m}$
- $G=9.8\text{m/s}^2$
- $L_{c1}=0.1\text{m}$ ,  $L_{c2}=0.05\text{m}$ ,  $L_{c3}=0.1\text{m}$ ,  $L_{c4}=0.05\text{m}$ .

### Analizamos $q_1$ y $q_2$ de 0 a 90 grados

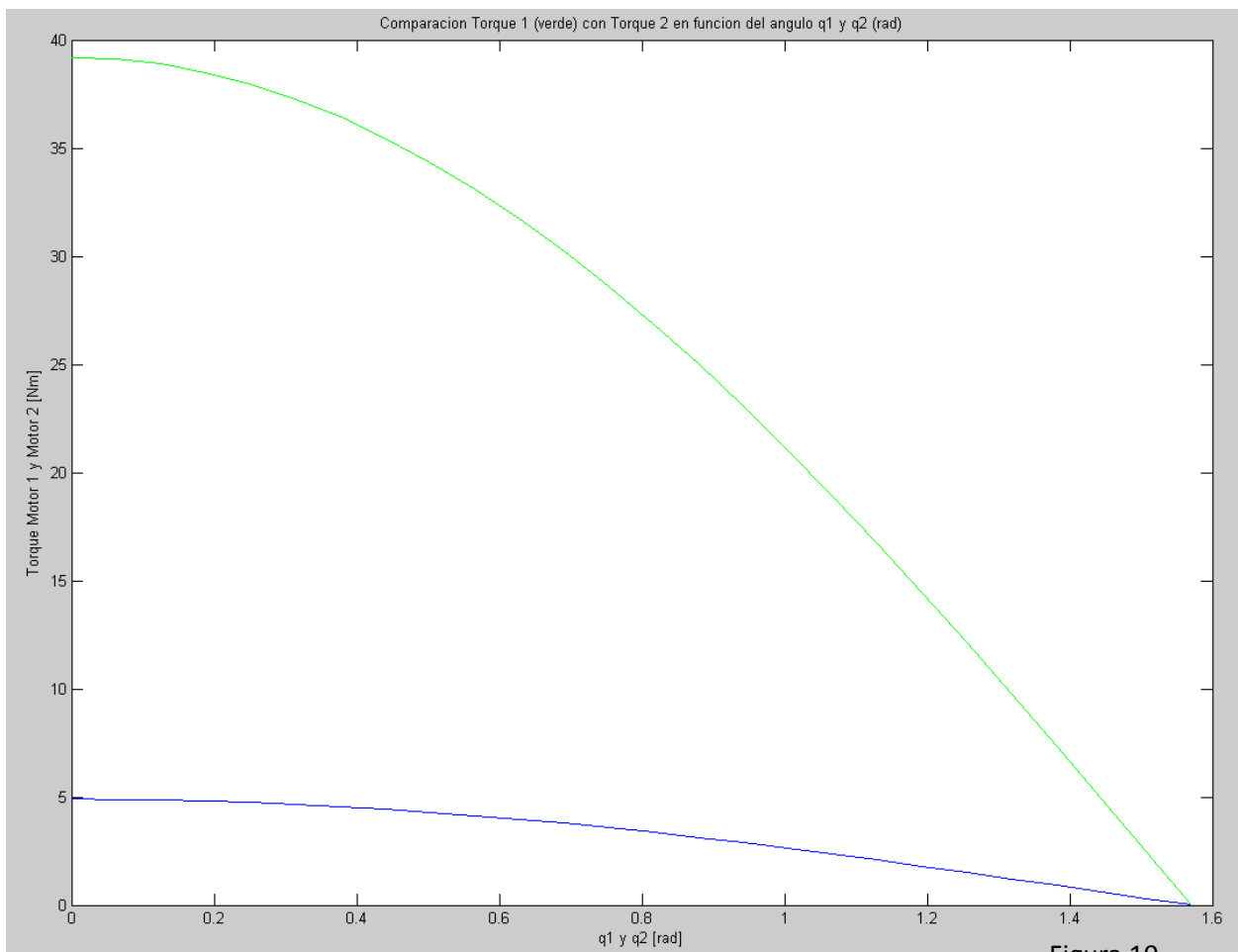


Figura 10

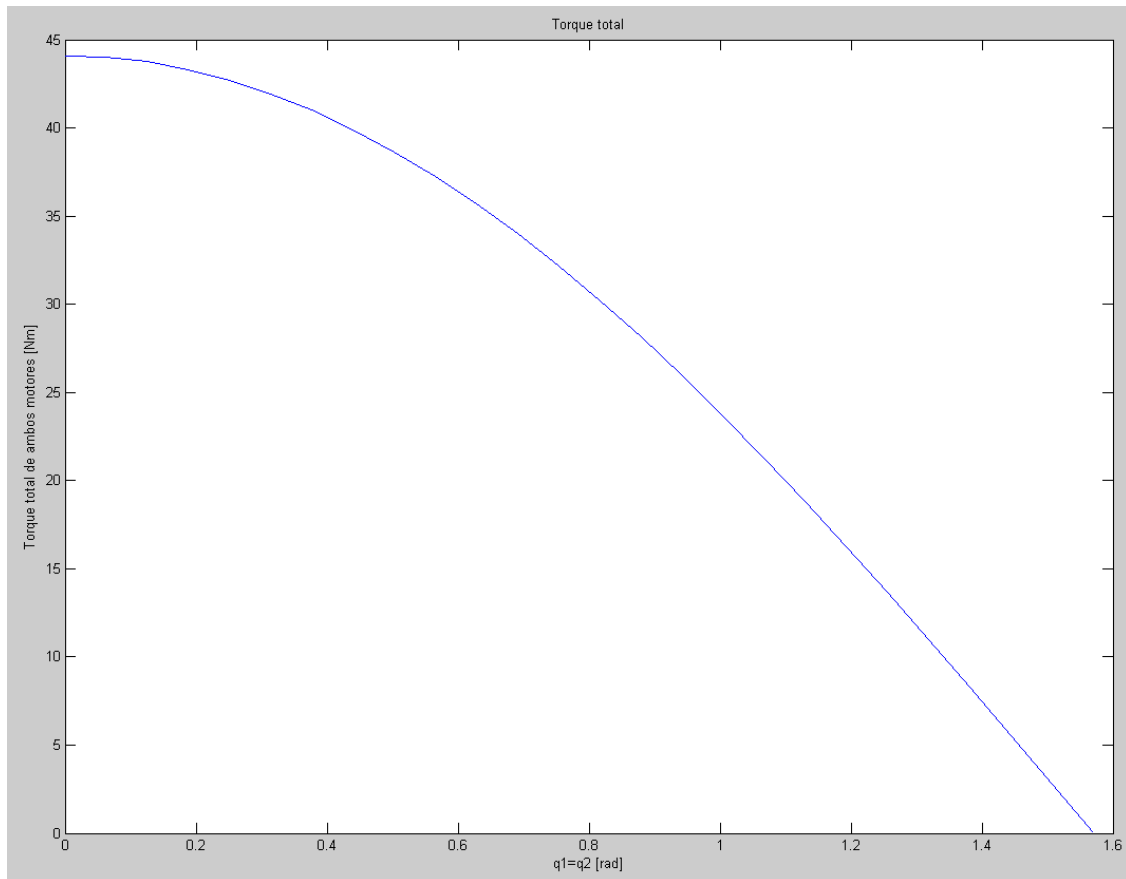


Figura 11

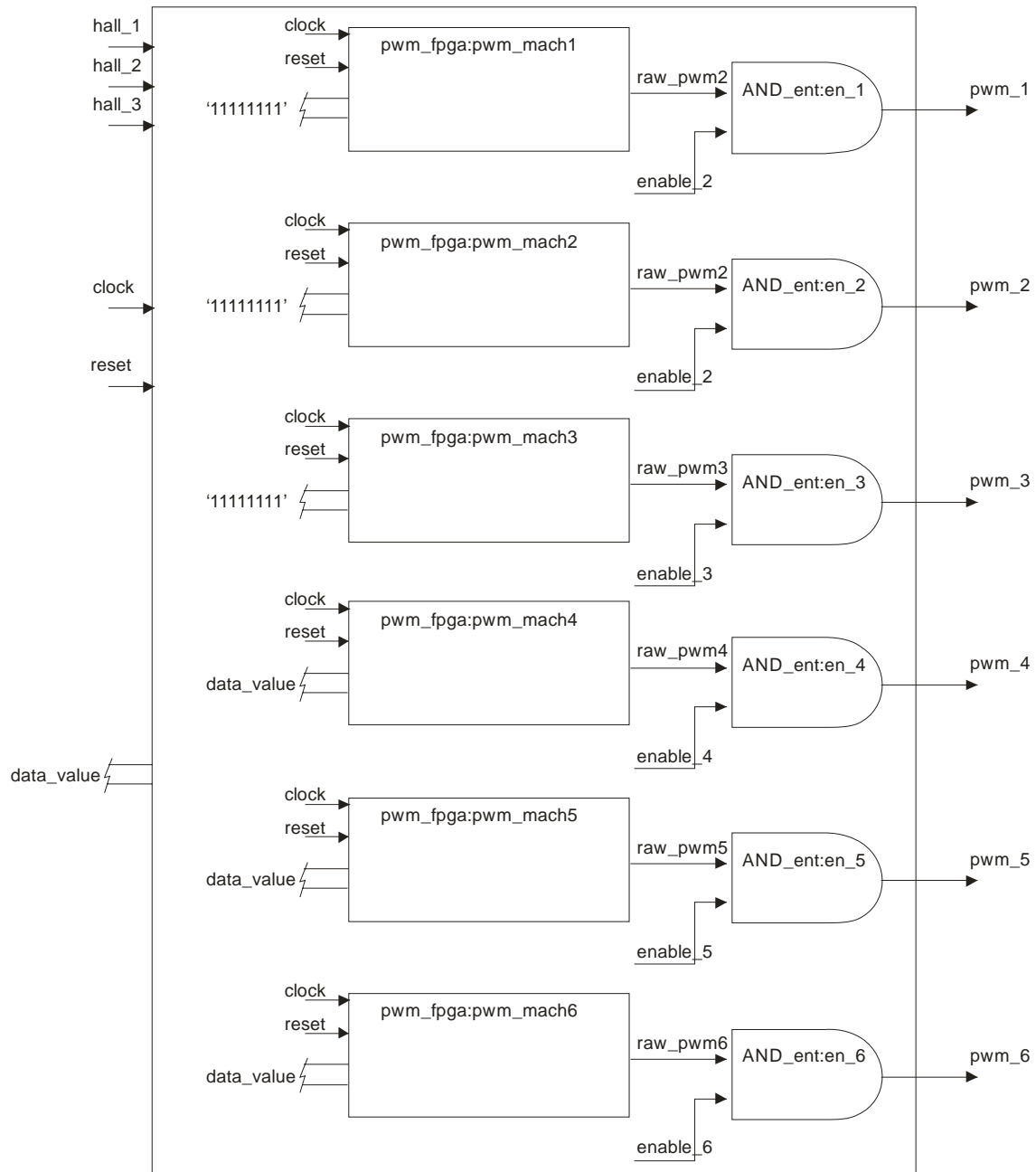
#### Conclusiones:

- Finalmente conseguimos la condición de torque requerida (Torque máximo superior a 31.1Nm), lo que nos permite seguir adelante en el diseño del robot. Si bien los valores en la realidad pueden ser inviables o inexistentes, esto es una primera aproximación a un análisis dinámico de un manipulador compuesto por teoría y cálculo computarizado
- Como predijimos antes de realizar este último cálculo, agrandar el peso de la última barra en la cadena cinemática aumento considerablemente los torques de los motores. Pero según nuestra percepción, este esquema esta acotado por la realidad física. Las demás barras deben ser extremadamente tolerantes a los movimientos y velocidades impuestas, contemplando las variables momento de inercia y peso. Pero por sobre todas las cosas la barra tiene que ser capaz de tener su centro de gravedad lo más cerca posible del eje y para realizarlo, suponemos que se deben hacer artilugios mecánicos que complicarían el movimiento o agregaría mas peso.
- Decidimos focalizar nuestro análisis a la variabilidad de los factores ya nombrados y modificados en los cálculos anteriores pero sin darle mucha influencia a la velocidad y aceleración, ya que intuitivamente nos podemos imaginar que a mayores velocidades, menores torques (siempre que se mantenga la potencia mecánica) y así también una relación semejante si la aceleración es variable. Pero no nos hubiésemos imaginado tan bien que pasaría si se variaba el peso, la longitud y los centros de masa. Por eso repetimos que si bien los valores pueden ser irrisorios y/o inválidos fue una buena oportunidad didáctica de ver las influencias dinámicas de los robots

### Simulación de un Driver PWM de motor trifásico.

Se procederá en el diseño de un driver trifásico basado en PWM de modo de controlar la velocidad y el torque de un motor-eje de un determinado robot. El diseño se basó en entidades prefabricadas: un modulo singular PWM (pwm\_fpga) y un modulo AND (AND\_ent). La entidad diseñada se llama TP2Robotica y se muestran a continuación el esquema, el código VHDL correspondiente y la simulación.

#### Diagrama de interconexión de la entidad diseñada:



Código VHDL de la entidad diseñada:

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

ENTITY TP2Robotica IS
PORT ( clock,reset           :in STD_LOGIC;
      hall_1                 :in STD_LOGIC;
      hall_2                 :in STD_LOGIC;
      hall_3                 :in STD_LOGIC;
      data_value             :in std_logic_vector(7 downto 0);
      pwm_1                  :out STD_LOGIC;
      pwm_2                  :out STD_LOGIC;
      pwm_3                  :out STD_LOGIC;
      pwm_4                  :out STD_LOGIC;
      pwm_5                  :out STD_LOGIC;
      pwm_6                  :out STD_LOGIC
    );

END TP2Robotica;

ARCHITECTURE behv1 OF TP2Robotica IS

    SIGNAL enable_1           : STD_LOGIC;
    SIGNAL enable_2           : STD_LOGIC;
    SIGNAL enable_3           : STD_LOGIC;
    SIGNAL enable_4           : STD_LOGIC;
    SIGNAL enable_5           : STD_LOGIC;
    SIGNAL enable_6           : STD_LOGIC;

    SIGNAL raw_pwm_1          : STD_LOGIC;
    SIGNAL raw_pwm_2          : STD_LOGIC;
    SIGNAL raw_pwm_3          : STD_LOGIC;
    SIGNAL raw_pwm_4          : STD_LOGIC;
    SIGNAL raw_pwm_5          : STD_LOGIC;
    SIGNAL raw_pwm_6          : STD_LOGIC;

    SIGNAL all_ones           :STD_LOGIC_VECTOR(7 downto 0);

    component pwm_fpga
    PORT ( clock,reset        :in STD_LOGIC;
          Data_value         :in std_logic_vector(7 downto 0);
          pwm                 :out STD_LOGIC
    );
end component;
```

```

component AND_ent
port(   x: in std_logic;
        y: in std_logic;
        F: out std_logic
);
end component;

```

**BEGIN**

```

en_1: AND_ent port map (enable_1,raw_pwm_1,pwm_1);
en_2: AND_ent port map (enable_2,raw_pwm_2,pwm_2);
en_3: AND_ent port map (enable_3,raw_pwm_3,pwm_3);
en_4: AND_ent port map (enable_4,raw_pwm_4,pwm_4);
en_5: AND_ent port map (enable_5,raw_pwm_5,pwm_5);
en_6: AND_ent port map (enable_6,raw_pwm_6,pwm_6);
pwm_1_mach: pwm_fpga port map (clock,reset,all_ones,raw_pwm_1);
pwm_2_mach: pwm_fpga port map (clock,reset,all_ones,raw_pwm_2);
pwm_3_mach: pwm_fpga port map (clock,reset,all_ones,raw_pwm_3);
pwm_4_mach: pwm_fpga port map (clock,reset,data_value,raw_pwm_4);
pwm_5_mach: pwm_fpga port map (clock,reset,data_value,raw_pwm_5);
pwm_6_mach: pwm_fpga port map (clock,reset,data_value,raw_pwm_6);

```

**init\_task: PROCESS (reset)**

**begin**

all\_ones <="00000001";

**END PROCESS init\_task;**

**task\_1: PROCESS (hall\_1,hall\_2,hall\_3)**

**BEGIN**

**if(hall\_1 = '1') THEN**

enable\_1 <= '1';

enable\_4 <= '1';

enable\_2 <= '0';

enable\_5 <= '0';

enable\_3 <= '0';

enable\_6 <= '0';

**END IF;**

**if(hall\_2 = '1') THEN**

enable\_1 <= '0';

enable\_4 <= '0';

enable\_2 <= '1';

enable\_5 <= '1';

enable\_3 <= '0';

enable\_6 <= '0';

**END IF;**

**if(hall\_3 = '1') THEN**

enable\_1 <= '0';

```

        enable_4 <= '0';
        enable_2 <= '0';
        enable_5 <= '0';
        enable_3 <= '1';
        enable_6 <= '1';
    END IF;

    END PROCESS task_1;

END behv1;

```

El banco de pruebas usado fue el siguiente:

```

--*****
--      Testbench File for design pwm_fpga produced by
--      Atmel System Designer on
--      May 25, 2001 1:54:17 pm
--*****

LIBRARY ieee;

use ieee.std_logic_1164.all;

ENTITY pwm_fpga_test_bench IS

END pwm_fpga_test_bench;

ARCHITECTURE arch_test_bench OF pwm_fpga_test_bench IS

COMPONENT TP2Robotica
PORT (
    clock: in std_logic;
    reset: in std_logic;
    hall_1           :in STD_LOGIC;
    hall_2           :in STD_LOGIC;
    hall_3           :in STD_LOGIC;
    data_value: in std_logic_vector(7 downto 0);
    pwm_1: out std_logic;
    pwm_2: out std_logic;
    pwm_3: out std_logic;
    pwm_4: out std_logic;
    pwm_5: out std_logic;
    pwm_6: out std_logic
);
END COMPONENT;

-- Internal signal declaration

SIGNAL sig_clock: std_logic;
SIGNAL sig_reset: std_logic;
SIGNAL sig_data_value  : std_logic_vector(7 downto 0);
SIGNAL sig_pwm_1 :std_logic;
SIGNAL sig_pwm_2 :std_logic;

```



```

SIGNAL sig_pwm_3 :std_logic;
SIGNAL sig_pwm_4 :std_logic;
SIGNAL sig_pwm_5 :std_logic;
SIGNAL sig_pwm_6 :std_logic;
SIGNAL sig_hall_1 :std_logic;
SIGNAL sig_hall_2 :std_logic;
SIGNAL sig_hall_3 :std_logic;

shared variable ENDSIM: boolean:=false;
constant clk_period:TIME:=100 ns;

BEGIN
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
            wait;
        end if;
    end process;

hall_gen: process

        BEGIN

        If ENDSIM = FALSE THEN
            sig_hall_1<= '1';
            sig_hall_2<= '0';
            sig_hall_3<= '0';
            wait for 338 us;
            sig_hall_1<= '0';
            sig_hall_2<= '1';
            sig_hall_3<= '0';
            wait for 338 us;
            sig_hall_1<= '0';
            sig_hall_2<= '0';
            sig_hall_3<= '1';
            wait for 338 us;
        else
            wait;
        end if;
    end process;

-- Instantiating top level design Component pwm_fpga

inst_pwm_fpga : TP2Robotica
PORT MAP(
        clock    => sig_clock,

```

```

        reset    => sig_reset,
        hall_1   => sig_hall_1,
        hall_2   => sig_hall_2,
        hall_3   => sig_hall_3,
        data_value    => sig_data_value,
        pwm_1    => sig_pwm_1,
        pwm_2    => sig_pwm_2,
        pwm_3    => sig_pwm_3,
        pwm_4    => sig_pwm_4,
        pwm_5    => sig_pwm_5,
        pwm_6    => sig_pwm_6

    );

stimulus_process: PROCESS

BEGIN

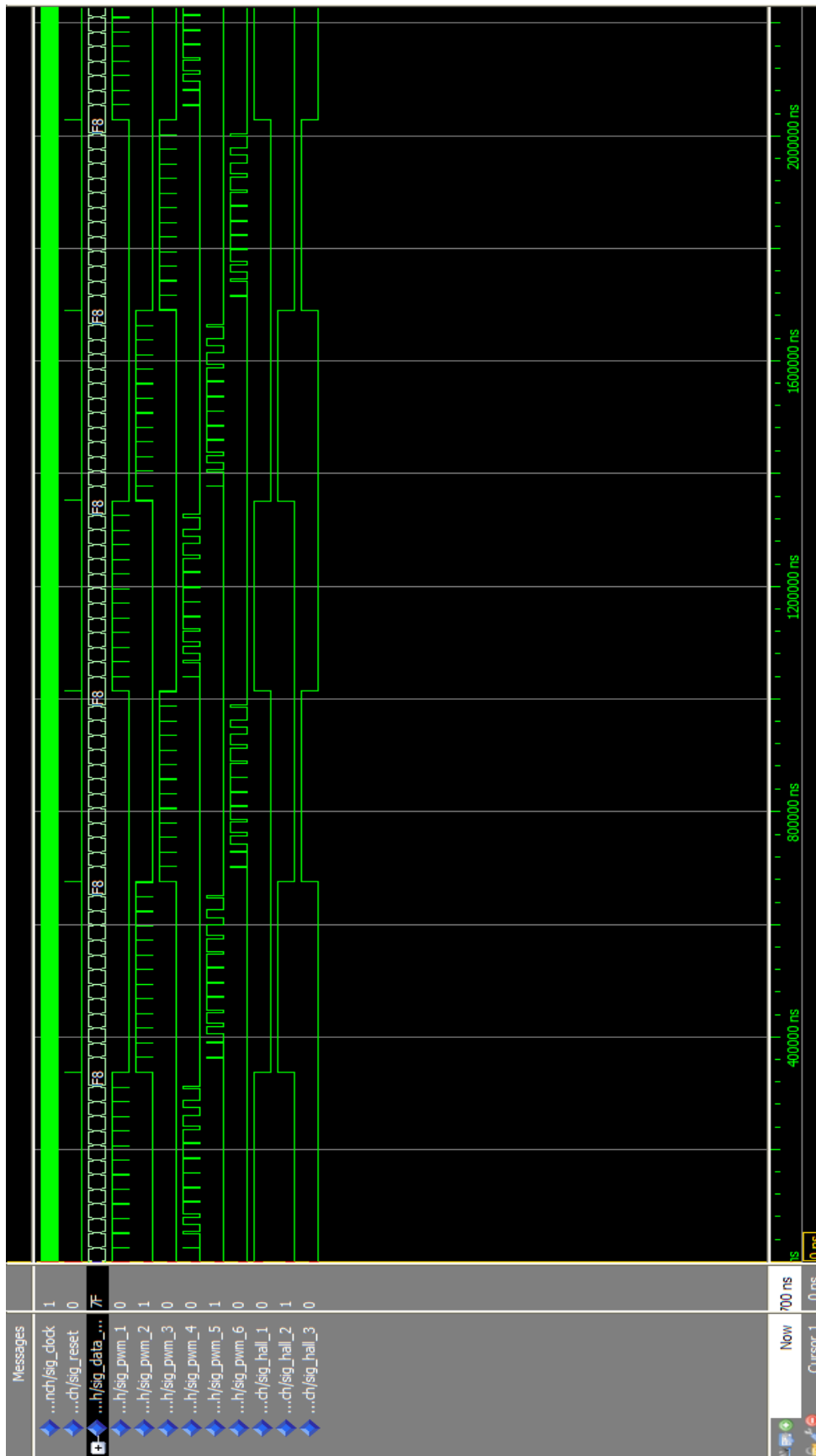
    If ENDSIM = FALSE THEN
        sig_reset <= '1';
        wait for 100 ns;
        sig_reset <= '0';
        sig_data_value <= "11111000";
        wait for 26 us;
        sig_data_value <= "11011111";
        wait for 26 us;
        sig_data_value <= "01111111";
        wait for 26 us;
        sig_data_value <= "01000000";
        wait for 26 us;
        sig_data_value <= "00100000";
        wait for 26 us;
        sig_data_value <= "00010000";
        wait for 26 us;
        sig_data_value <= "00001000";
        wait for 26 us;
        sig_data_value <= "00010000";
        wait for 26 us;
        sig_data_value <= "00100000";
        wait for 26 us;
        sig_data_value <= "01000000";
        wait for 26 us;
        sig_data_value <= "01111111";
        wait for 26 us;
        sig_data_value <= "11001111";
        wait for 26 us;
        sig_data_value <= "11111000";
        wait for 26 us;

    else
        wait;
    end if;

END PROCESS stimulus_process;
END arch_test_bench;

```

El resultado de la simulación fue la siguiente:



- sig\_clock = Clock de excitación.
- sig\_reset=Reset del driver.
- sig\_data= 8 bits indicando la velocidad propuesta para el sistema.
- sig\_pwm\_x= Salida individual de cada pwm del driver, listo para conectarse con el motor trifásico.
- sig\_hall\_x= Entrada individual de los sensores de efecto de hall para detectar la posición del rotor.

Figura 12

### Conclusiones finales:

- Fue muy interesante al principio llevar a la realidad simulada (virtualidad) los análisis dinámicos de los robots vistos en teoría y nuevamente ratificamos la facilidad de simular situaciones debido a MATLAB, Quartus II y ModelSim. También es una gran oportunidad de aprender el manejo de estos dos últimos software debido a que nunca los habíamos tenido en contacto directo, especialmente el ModelSim que nos sorprendió con la simulación gráfica.
- Vimos reflejado en los cálculos la variabilidad de factores físicos de un robot con respecto al torque: antes de realizar los cálculos teníamos una impresión deductiva de cuáles eran los factores de mayor influencia y cuánto influían pero vimos que en ciertas cosas estábamos equivocados a priori, como por ejemplo pensar que el largo de las barras era de vital influencia mientras que lo que realmente influye es el centro gravitatorio de las mismas.
- Aprovechamos también para aprender un lenguaje de programación, al principio complicado, pero que nos puede servir desde ahora en adelante en cualquier tarea de R&D de ASIC's o FPGA's. El VHDL nos acercó a lo que es la tecnología de hoy en día y también nos moldeó, de cierta forma, a no quedarnos conformes con saber C/C++. Es importante complementar todos los conocimientos de desarrollo, ya sea de bajo nivel o de alto nivel.