Description: http://www.siga.frba.utn.edu.ar/imag/utnba.png

**ROBOTICA**

**Proyecto robot SCARA**

Curso: R6055

Año: 2013

**Docente:**

Ing. Hernan Giannetta

**Ayudante:**

Ing. Damian Granzella

**Alumnos:**

Ignacio Tamayo

Javier Garbini

**ÍNDICE**

[Revision 3](#_Toc361009389)

[1 Introducción 4](#_Toc361009390)

[2 Memoria descriptiva 4](#_Toc361009391)

[2.1 Funcionamiento 4](#_Toc361009392)

[2.2 Modelo adoptado 4](#_Toc361009393)

[2.3 Medidas del robot 5](#_Toc361009394)

[3 Cinematica del robot 5](#_Toc361009395)

[3.1 Cinemática directa 5](#_Toc361009396)

[Algoritmo Denavit-Hartenberg 5](#_Toc361009397)

[Desplazamiento de los ejes 5](#_Toc361009398)

[3.2 Matriz DH del robot 6](#_Toc361009399)

[3.3 Cinemática Inversa 7](#_Toc361009400)

[Selección de método 7](#_Toc361009401)

[Espacios de trabajo 7](#_Toc361009402)

[Generación de trayectorias articulares 8](#_Toc361009403)

[4 Conclusiones de la etapa de diseño y cinemática 8](#_Toc361009404)

[5 Análisis dinámico 8](#_Toc361009405)

[6 PWM 13](#_Toc361009406)

[6.1 Estator 13](#_Toc361009407)

[6.2 Rotor 13](#_Toc361009408)

[6.3 Sensores de efecto Hall 14](#_Toc361009409)

[6.4 Conmutación de fases 15](#_Toc361009410)

[6.5 Torque y velocidad del motor 16](#_Toc361009411)

[7 VHDL 17](#_Toc361009412)

[7.1 Diagrama en bloques 17](#_Toc361009413)

[7.2 Tabla de estados 18](#_Toc361009414)

[7.3 Sensores efecto Hall 19](#_Toc361009415)

[7.4 FPGA 20](#_Toc361009416)

[8 Conclusiones de la etapa de dinamica 22](#_Toc361009417)

# Revision

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Nombre | Descripción | Versión |
| 03/06/2013 | TP1 | Cinemática Directa e Inversa | 1.0 |
| 22/06/2013 | TP2 | Análisis dinámico | 2.0 |
| 08/07/2013 | TP2 | Simulación en VHDL | 2.1 |

# Introducción

A fin de cumplimentar los requerimientos de cursada para para cátedra de la asignatura electiva de 6° nivel Robótica se realiza la presentación del Proyecto un robot Scara con 5 grados de libertad.

Se toma como base el trabajo realizado en el año 2012 con esta misma arquitectura, agregando un grado más de libertad.

# Memoria descriptiva

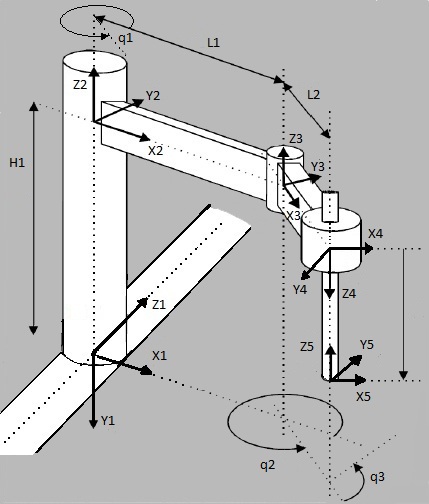
## Funcionamiento

SCARA es un tipo de configuración de robot manipulador las siglas en inglés significan Selective Compliance Assembly Robot Arm. Este robot fue creado por un grupo de industrias electrónicas japonesas, en colaboración con dos universidades, para insertar los componentes de forma vertical.

La configuración seleccionada para realizar el proyecto está formada por dos articulaciones de rotación con respecto a dos ejes paralelos entre sí y perpendiculares al plano de trabajo, dos de desplazamiento una dirección paralela a la de los ejes de rotación y la otra dirección en sentido perpendicular a los ejes de rotación.

## Modelo adoptado

Para el proyecto de la cátedra se eligió un SCARA con el agregado de un grado más de libertad, permitiendo que el robot se desplace linealmente sobre el eje Y del sistema de referencia inicial, como se muestra en la siguiente figura:



## Medidas del robot

H1 = 25 cm Altura de la barra.

L1 = 30 cm Longitud de la primera articulación.

L2 = 20 cm Longitud de la segunda articulación.

# Cinematica del robot

## Cinemática directa

### Algoritmo Denavit-Hartenberg

Para realizar el cálculo de la cinemática directa utilizaremos el algoritmo de Denavit-Hartenberg, el cual define los movimientos de las articulaciones de la siguiente manera:

* Rotación alrededor del eje zi-1 un ángulo θi
* Traslación a lo largo de zi-1 una distancia di ; vector di (0,0,di).
* Traslación a lo largo de xi una distancia ai ; vector ai (0,0,ai).
* Rotación alrededor del eje xi un ángulo αi.

### Desplazamiento de los ejes

Se muestra el gráfico se la secuencia de sistemas para el algoritmo

Art1 d1

Art2 q2

Art3 q3

Art4 d2

Art5 q3

S0

X0

Z0

Y0

S1

Y1

X1

Z1

S2

X2

Y2

Z2

S3

X3

Y3

Z3

S4

X4

Z4

Y4

S5

X5

Y5

Z5

S6

X6

Z6

Y6

Para el caso de nuestro robot los parámetros de la matriz son los siguientes:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **θ** | **d** | **a** | **α** |
| **T0** | -90 ° | 0 | 0 | 0 |
| **T1** | 90 ° | 0 | 0 | **d1** |
| **T2** | 0 | L1 | **q1** | H1 |
| **T3** | 180 ° | L2 | **q2** | 0 |
| **T4** | -180 ° | 0 | 0 | **d2** |
| **T5** | 0 | 0 | **q3** | 0 |

Se tiene una aparente articulación extra (6 filas en lugar de 5) pero esta articulación sirve solo para orientar el eje de coordenadas en el sentido del desplazamiento de la primera articulación.

Por medio de Matlab se obtiene la siguiente matriz de transformación homogénea:

## Matriz DH del robot

[cos(q1+q2), -sin(q1+q2), 0, L2\*cos(q1+q2)+L1\*cos(q1) ]

[sin(q1+q2), cos(q1+q2), 0, L2\*sin(q1+q2)+L1\*sin(q1)+d1 ]

[0, 0, 1, H1-d2 ]

[0, 0, 0, 1 ]

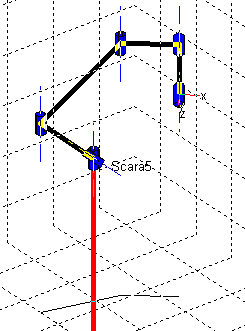
También usando MatLab se obtienes su matriz Jacobiana correspondiente

[ 0, -L2\*sin(q1+q2)-L1\*sin(q1), -L2\*sin(q1+q2), 0 ]

[ 1, L2\*cos(q1+q2)+L1\*cos(q1), L2\*cos(q1+q2), 0 ]

[ 0, 0, 0, -1 ]

Se valida el diseño y las matrices obtenidas usando el Toolbox Corke, del cual se valida el modelo obtenido



L0 = link([-pi/2 0 0 0]) ;

L1 = link([pi/2 0 0 0 1]);

L2 = link ([0 l1 0 h1]);

L3 = link([pi l2 0 0]);

L4 = link([-pi 0 0 0 1]);

L5 = link([0 0 0 0]);

Scara5 = robot({L0,L1,L2,L3,L4,L5},'Scara5');

drivebot(Scara5);

## Cinemática Inversa

### Selección de método

Para el cálculo de la cinemática inversa se utilizó el método geométrico, ya que resulta ser más sencillo de implementar. Como el robot posee un grado de libertad sobre el plano XY es muy tedioso utilizar la matriz de transformación inversa incluyendo esta articulación y a los fines didácticos el método geométrico es mucho más académico al momento de analizarlo.

### Espacios de trabajo

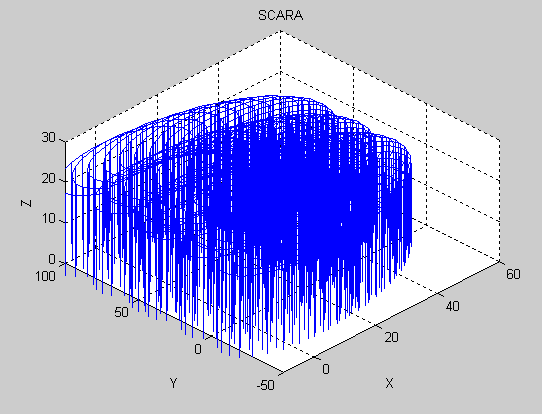
Considerando que los giros de las articulaciones están físicamente limitados por los motores que los van a realizar, además de evitar las coordenadas articulares inválidas antes halladas, se le definen excursiones máximas a las articulaciones y se itera para tener el espacio de trabajo del robot SCARA

Articulación d1: [0 a D1max] cm, Valor de D1max = 50 cm

Articulación d2: [0 a H1] cm, Valor de H1 = 25 cm

Articulaciones q1,q2,q3: [-145° a 145°]

A continuación se muestra el espacio de trabajo para 3 valores de la articulación d1, sobre el eje Y



### Generación de trayectorias articulares

Para poder controlar efectivamente el Robot, se necesita además del algoritmo cinemático una determinada forma de atravesar en el tiempo las coordenadas indicadas. Para nuestro robot SCARA se usa un generador de trayectorias polinómicas de 3er orden, de manera de evitar tener aceleraciones articulares bruscas.

Si usásemos una trayectoria lineal, efectivamente las aceleraciones tendrían valores infinitos en cada arranque de movimiento y esto exigiría demasiado a los motores.

Se implementa el siguiente flujo para la generación de trayectorias y las curvas temporales de velocidad y aceleraciones articulares que los motores deben proveer para conseguir el movimiento deseado

# Conclusiones de la etapa de diseño y cinemática

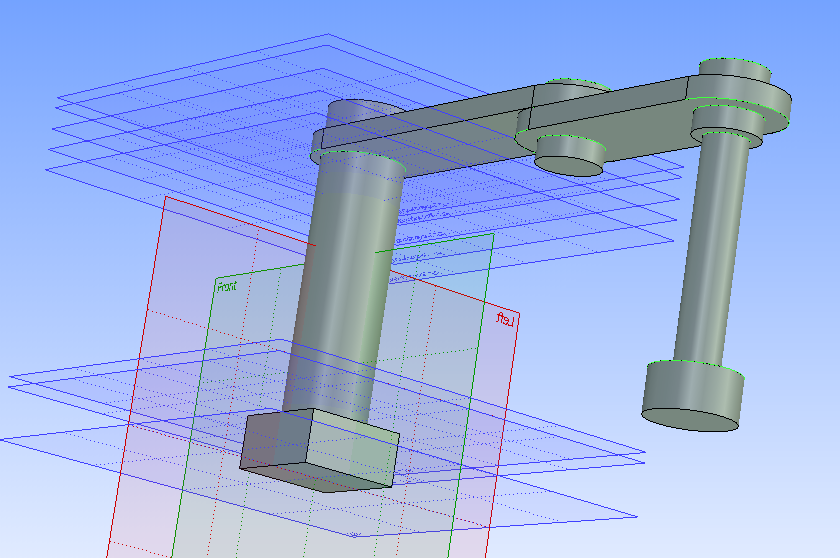
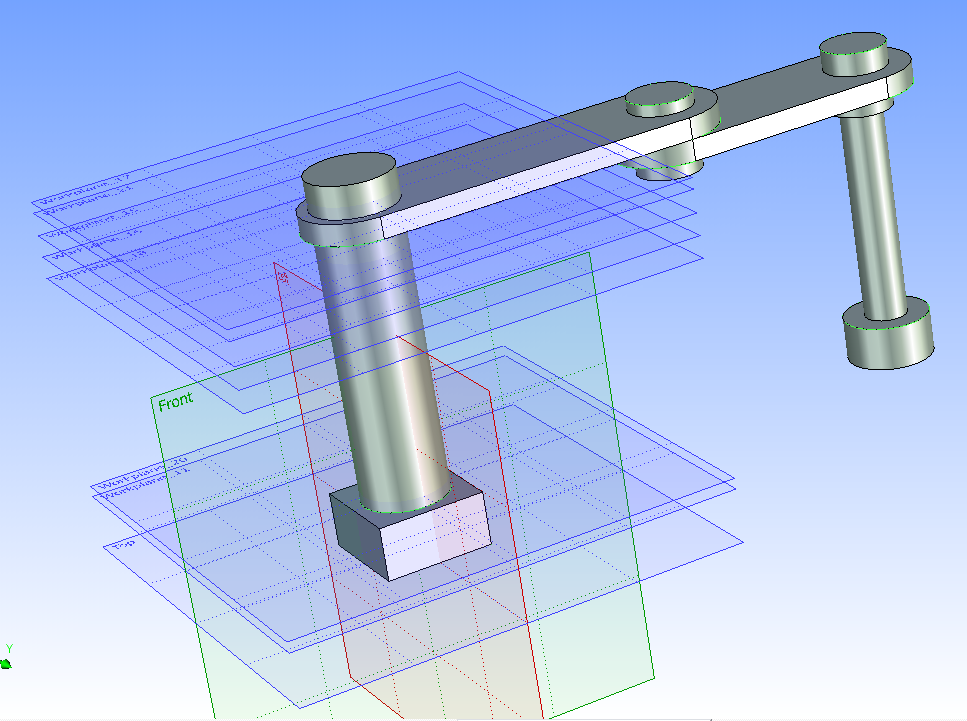
* Se ha desarrollado adecuadamente el modelo del Robot SCARA de 5 grados de libertad, usando el algoritmo de Denavit-Hartenberg
* Se verificó el modelo con el Toolbox Corke
* Se desarrolló un algoritmo cinemático inverso
* Se usa un generador de trayectorias articulares polinomial de 3 er order, para suavizar las curvas de aceleraciones
* Efectivamente, dada una trayectoria a seguir, el algoritmo cinemático inverso es capaz de seguirlo, siempre en los puntos dentro del área de trabajo válida
* Las velocidades y aceleraciones articulares no tiene discontinuidades por ser trayectorias polinomiales
* Las velocidades espaciales no tienen discontinuidades.

# Análisis dinámico

Una vez obtenido el generador de trayectorias polynomial de orden 3, basado en la técnica de interpolación segmentada por Splines[[1]](#footnote-1), se usa el Toolbox Corke para evaluar los parámetros dinámicos del movimiento del robot, tomando en cuenta sus formas y masas, hasta lograr determinar los parámetros de los actuadores necesarios

Primeramente, se modela el robot en T-Flex para obtener sus parámetros dinámicos

Con las dimensiones mencionadas anteriormente, y dándole un material metálico como construcción, se tiene este modelo en 3 dimensiones



Los parámetros dinámicos por cada eslabón son

Matriz Inercia Eslabón 1 = Centro de Masa Eslabón 1 = Masa Eslabón 1 =

[ 0.1394 0 0] [0 , 0 , 0.0975] mts 6.65675 Kg

[ 0 0.1394 0]

[ 0 0 0.0104] mts

Matriz Inercia Eslabón 2 = Centro de Masa Eslabón 2 = Masa Eslabón 2 =

[ 0.3654 0 0.0001] [0.2162 , 0 , 0] mts 3.88378 Kg

[ 0 0.5842 0]

[ 0.0001 0 0.2238] mts

Matriz Inercia Eslabón 3 = Centro de Masa Eslabón 3 = Masa Eslabón 3 =

[ 0.2729 0 0.0005] [0.1433, 0 , 0] mts 3.05477 Kg

[ 0 0.8889 0]

[ 0.0005 0 0.6198] mts

Matriz Inercia Eslabón 4 = Centro de Masa Eslabón 4 = Masa Eslabón 4 =

[ 0.0634 0 0] [0 , 0 , 0.1700] mts 1.96035 Kg

[ 0 0.5535 0]

[ 0 0 0.4905] mts

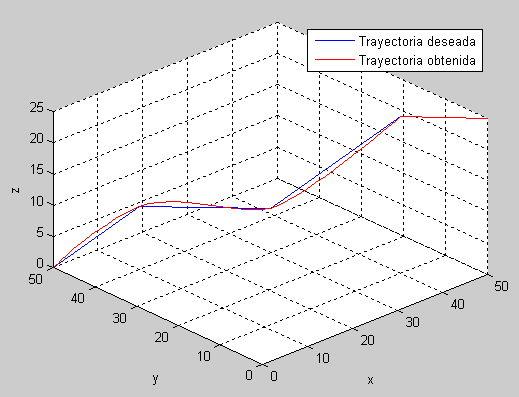
Matriz Inercia Eslabón 5 = Centro de Masa Eslabón 5 = Masa Eslabón 5 =

[ 0.0048 0 0] [0 , 0 , 0.0400] mts 1.56828 Kg

[ 0 0.3968 0]

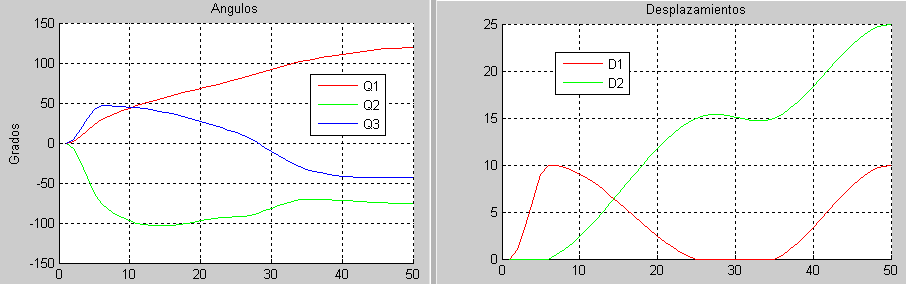
[ 0 0 0.3933] mts

Tomamos una trayectoria suave, una cuasi diagonal en el espacio, y se le da trayectoria con polinomio de 3er grado. Luego de usa esta trayectoria y el modelo del ToolBox Corke para tener los Torques/Fuerzas necesarias en las articulaciones. El ToolBox usa el modelo mecánico de Newton-Euler para sus cálculos



*Trayectoria realizada en 50 segundos*

Si bien las trayectorias son de curva suave, al ser el polinomio de 3er orden, las aceleraciones son lineales y los cambios de tramos generan cambios bruscos de aceleración y por ende de torques y fuerzas



Los valores de Fuera y Torque calculados son

Potencias medias:

D1=4.559028 [W]

Q1=6.290679 [W]

Q2=4.090474 [W]

D2=0.182755 [W]

Q3=0.000030 [W]

Torques/Fuerzas máx:

D1=2276.040633 [N]

Q1=621.742307 [Nm]

Q2=159.446726 [Nm]

D2=34.622761 [N]

Q3=0.002786 [Nm]

Torques/Fuerzas RMS:

D1=452.035790 [N]

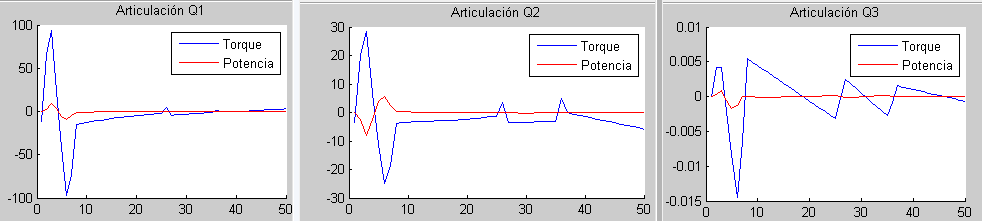
Q1=123.549475 [Nm]

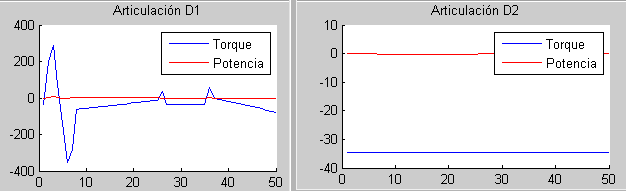
Q2=31.618331 [Nm]

D2=0.003761 [N]

Q3=0.000902 [Nm]

Sin embargo se tienen picos de fuerza y torque por las aceleraciones bruscas

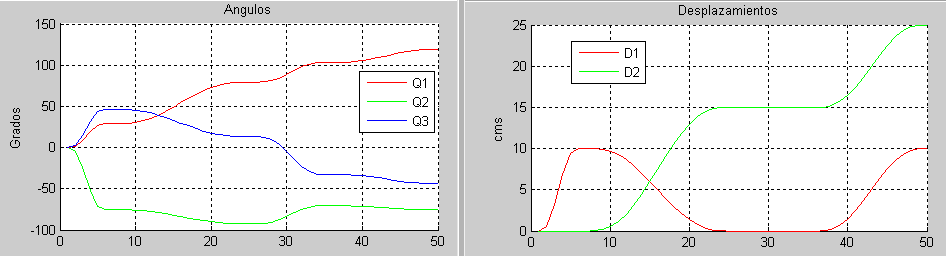




La articulación D2 tiene una componente de fuerza constante pues debe contrarrestar la fuerza peso de la última articulación, pues el peso se ejerce directamente en la dirección del eje.

Para reducir estos cambios violentos de torques se usa ahora una trayectoria polinomial de 5to orden. Si bien se complejiza el cálculo de la trayectoria, el comportamiento pedido a los motores es menos brusco y evita maltratos y el acortamiento de su vida útil

La trayectoria espacial es muy similar



Los valores de Fuerzas/Torques y Potencias tienen valores menores, siendo la misma trayectoria

Potencias medias:

D1=0.728995 [W]

Q1=1.068372 [W]

Q2=0.697858 [W]

D2=0.173075 [W]

Q3=0.000000 [W]

Torques/Fuerzas máx:

D1=483.187836 [N]

Q1=136.078460 [Nm]

Q2=40.847976 [Nm]

D2=34.626095 [N]

Q3=0.000000 [Nm]

Torques/Fuerzas RMS:

D1=109.833923 [N]

Q1=32.023366 [Nm]

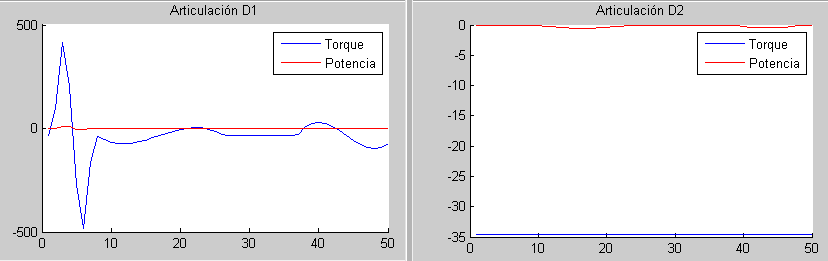
Q2=9.086058 [Nm]

D2=0.005641 [N]

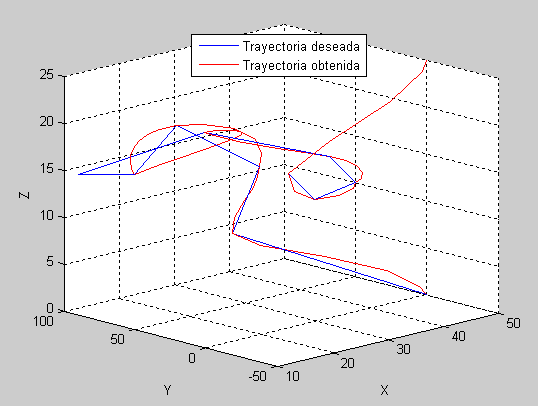
Q3=0.000000 [Nm]

Cambian los formas de las señales de Torque y Potencia de los motores, con transiciones más suaves





Finalmente, para completar el análisis, se propone una trayectoria mucho más exigente para obtener parámetros máximos para los motores



*Trayectoria ejecutada en 60 segundos, con puntos fuera del espacio de trabajo*

Los valores obtenidos para el dimensionamiento de los actuadores son

Potencias medias:

D1=8.315069 [W]

Q1=6.349446 [W]

Q2=3.394626 [W]

D2=0.324975 [W]

Q3=0.000891 [W]

Torques/Fuerzas máx:

D1=1461.339613 [N]

Q1=328.843058 [Nm]

Q2=133.946575 [Nm]

D2=34.669525 [N]

Q3=0.029818 [Nm]

Torques/Fuerzas RMS:

D1=375.162222 [N]

Q1=82.470954 [Nm]

Q2=29.242741 [Nm]

D2=0.016076 [N]

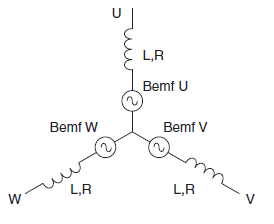
Q3=0.009119 [Nm]

# PWM

Para la implementación del robot se utilizaran motores BLDC, con sensores de efecto hall que determinaran la posición del rotor. El manejo de la velocidad se realizara por medio de un punte H que será controlado por medio de un PWM (pulse width modulation).

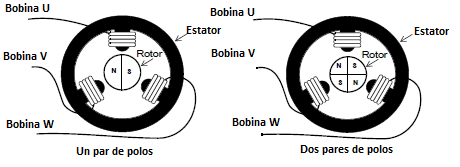
## Estator

Los motores básicos BLDC están compuestos por 3 fases las cuelas se forman por medio de 3 bobinas que se sitúan como se muestra a continuación:



## Rotor

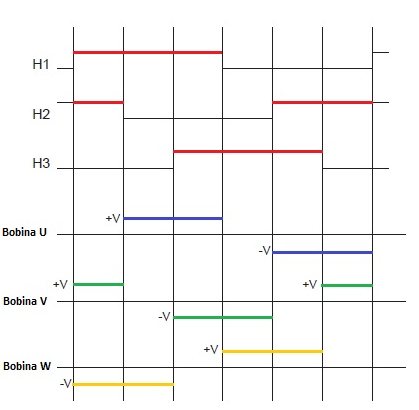
Estos tipos de motores poseen rotores con imán permanente, la cantidad de imanes determinaran los pasos del motor y así su torque a modo ilustrativo se muestra como se conforma un rotor



## Sensores de efecto Hall

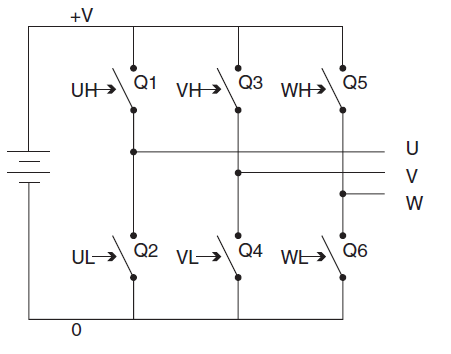
Un componente importante en este tipo de motores son los sensores de efecto Hall los cuales se encuentran ubicados a 120º cada uno, permitiendo así conocer la ubicación del rotor. De este modo se determinaran las secuencias para desplazar el rotor.

En la siguiente imagen se puede observar cómo se deben energizar las bobinas dependiendo del sensor que se encuentra activo.

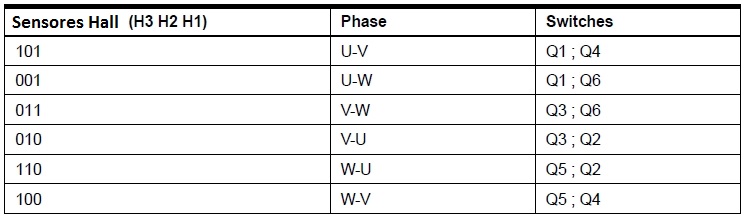


Conociendo esto se puede determinar la secuencia para energizar los transistores del puente H.

## Conmutación de fases

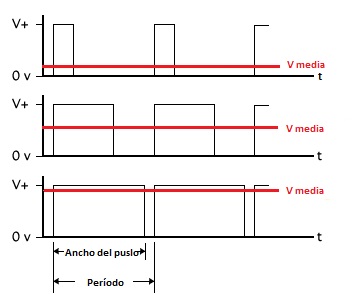


Como se vio anteriormente la conmutación de las fases depende de los valores de los sensores de efecto Hall. Cuando se energiza una bobina genera un campo magnético que permite que el rotor gire. El método consiste en energizar 2 bobinas al mismo tiempo y la tercera queda libre, de modo que 2 transistores conducen. En la siguiente tabla se puede observar la secuencia de conmutación dependiendo del valor de los sensores.

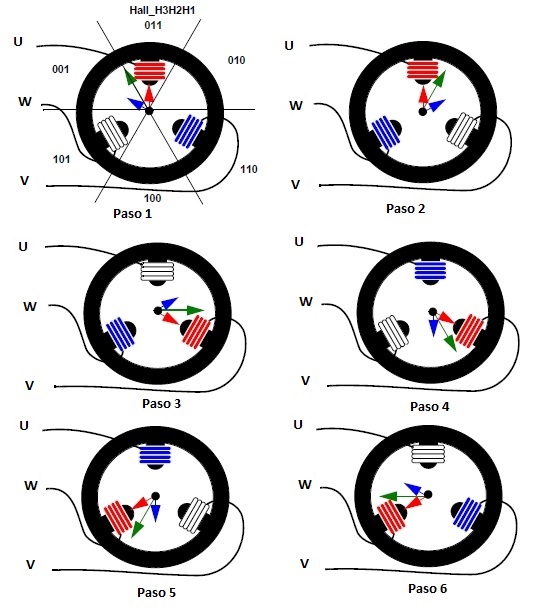


## Torque y velocidad del motor

Para generar la velocidad y el torque del motor se utiliza la modulación por ancho de pulso conocida como PWM. Con este método se varía el ciclo de actividad de la señal, así variando la tensión media sobre el motor logrando los cambios de velocidad necesarios.



En la siguiente imagen se muestra un ejemplo de una secuencia con los pasos del motor



En el paso 1 por ejemplo se encuentran energizadas las bobinas U y V, siendo que los transistores Q1 y Q4 conducen y el resto están apagados. Por lo tanto se observa que se forman dos flujos uno representado con el vector rojo correspondiente a la bobina U y el otro de color azul correspondiente a la bobina V, siendo el verde la suma de ambos.

El rotor intenta seguir la secuencia del flujo, en cuanto el sensor Hall cambia su valor se energizara la bobina W, apagándose el transistor Q4 y conduciendo el Q6. De esta forma se hace girar el rotor.

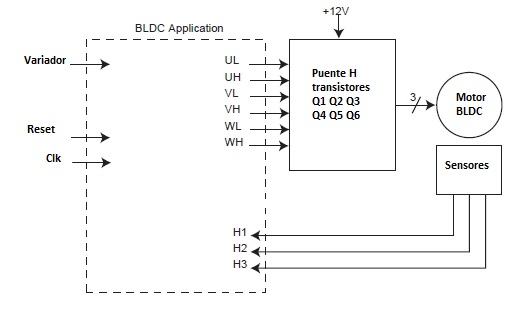
Para poder realizar variaciones de velocidad se utiliza la señal de PWM anteriormente mencionada.

Tabla de estados

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Estado 1** | **Estado 2** | **Estado 3** | **Estado 4** | **Estado 5** | **Estado 6** |
| **S1** | 1 | 1 | 0 | 0 | 0 | 0 |
| **S2** | 0 | 0 | 0 | pwm | pwm | 0 |
| **S3** | 0 | 0 | 1 | 1 | 0 | 0 |
| **S4** | pwm | 0 | 0 | 0 | 0 | pwm |
| **S5** | 0 | 0 | 0 | 0 | 1 | 1 |
| **S6** | 0 | pwm | pwm | 0 | 0 | 0 |

# VHDL

## Diagrama en bloques



## Tabla de estados

library IEEE;

USE ieee.std\_logic\_1164.all;

USE ieee.std\_logic\_arith.all ;

USE work.user\_pkg.all;

ENTITY driver\_mosfet IS

PORT (

S1 :out STD\_LOGIC;

S2 :out STD\_LOGIC;

S3 :out STD\_LOGIC;

S4 :out STD\_LOGIC;

S5 :out STD\_LOGIC;

S6 :out STD\_LOGIC;

hall\_sensor\_in :in integer range 0 to 5;

reset :in STD\_LOGIC;

clock :in STD\_LOGIC;

pwm\_in :in STD\_LOGIC

);

END driver\_mosfet;

ARCHITECTURE arch\_driver OF driver\_mosfet IS

-- Internal signal declaration

BEGIN

Cntrl\_PWM\_puenteH: PROCESS (reset, hall\_sensor\_in, pwm\_in) IS

BEGIN

IF (reset = '1') THEN

s1<= '0'; s2<= '0'; s3<= '0'; s4<= '0'; s5<= '0'; s6<= '0';

ELSE

CASE hall\_sensor\_in IS

WHEN 0 => --Estado 1

s1<= '1'; s2<= '0'; s3<= '0'; s4<= pwm\_in; s5<= '0'; s6<= '0';

WHEN 1 => --Estado 2

s1<= '1'; s2<= '0'; s3<= '0'; s4<= '0'; s5<= '0'; s6<= pwm\_in;

WHEN 2 => --Estado 3

s1<= '0'; s2<= '0'; S3<= '1'; s4<= '0'; s5<= '0'; s6<= pwm\_in;

WHEN 3 => --Estado 4

s1<= '0'; s2<= pwm\_in; s3<= '1'; s4<= '0'; s5<= '0'; s6<= '0';

WHEN 4 => --Estado 5

s1<= '0'; s2<= pwm\_in; s3<= '0'; s4<= '0'; s5<= '1'; s6<= '0';

WHEN 5 => --Estado 6

s1<= '0'; s2<= '0'; s3<= '0'; s4<= pwm\_in; s5<= '1'; s6<= '0';

WHEN OTHERS =>

s1<= '0'; s2<= '0'; s3<= '0'; s4<= '0'; s5<= '0'; s6<= '0';

END CASE;

END IF;

END PROCESS;

END ARCHITECTURE arch\_driver;

## Sensores efecto Hall

library IEEE;

USE ieee.std\_logic\_1164.all;

USE ieee.std\_logic\_arith.all ;

USE work.user\_pkg.all;

ENTITY Hall\_Sensor IS

PORT ( sig\_reset : in STD\_LOGIC;

clock : in STD\_LOGIC;

hall\_sensor : out integer range 0 to 5;

rco\_int : in STD\_LOGIC

);

END Hall\_Sensor;

ARCHITECTURE arch\_hall OF Hall\_Sensor IS

signal clock\_div10: std\_logic;

BEGIN

Gen\_clk10: PROCESS (sig\_reset, rco\_int) IS

VARIABLE hall: INTEGER range 0 to 20;

BEGIN

IF (sig\_reset = '1') THEN

hall:= 0;

clock\_div10 <= '1';

ELSE

IF rising\_edge(rco\_int) THEN

IF (hall = 20) THEN

hall:= 0;

clock\_div10 <= '1';

ELSE

hall:= hall + 1;

clock\_div10 <= '0';

END IF;

END IF;

END IF;

END PROCESS;

GenVarState: PROCESS (sig\_reset, clock\_div10) IS

VARIABLE hall: INTEGER range 0 to 5;

BEGIN

IF (sig\_reset = '1') THEN

hall:= 0;

ELSE

IF rising\_edge (clock\_div10) THEN

IF (hall = 5) THEN

hall:= 0;

ELSE

hall:= hall + 1;

END IF;

END IF;

END IF;

hall\_sensor <= hall;

END PROCESS;

END ARCHITECTURE arch\_hall;

## FPGA

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;

rco :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

-- 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;

-- 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;

rco <= rco\_int;

END arch\_pwm;

# Conclusiones de la etapa de dinamica

* Se ha desarrollado la etapa de dinámica logrando
* Se ha podido suavizar las curvas de velocidad y movimientos bruscos del robot.
* Se dimensionó el robot según la dinámica calculada.
* Se realizó el diseño físico del mimo.
* Fue posible verificar el proceso básico para el diseño robot dependiendo de las características dinámicas del mismo
* Se logró hacer las simulaciones necesarias para interpretar el comportamiento del robot.

1. <http://www.fi.unju.edu.ar/materias/materia/CN/document/Teoria_Calculo/INTERPOLACI%D3N__DE__SPLINES.pdf?cidReq=CN> [↑](#footnote-ref-1)