

Proyecto Final Integrador Robótica I

Elorga Eliseo, Leg.:10706
Cantaloube Adrian, Leg.:10695
Correa Elias, Leg.:10767

15 de noviembre de 2018

Índice

- Introducción
- Descripción y aspectos técnicos del robot
- Relevamiento/Dimensionamiento del espacio de trabajo
- Cinemática Directa
- Cinemática Inversa
- Matriz Jacobiana
- Matriz Jacobiana Inversa
- Singularidades
- Elipsoide de Manipulabilidad
- Espacio de trabajo
- Planificación de trayectorias
- Conclusión, mejoras y desafíos
- Referencias

ABB Robot Model IRB 7600-400/2.55

Introducción.

La idea principal de este trabajo es la de hacer un estudio de los aspectos cinemáticos del robot que hemos seleccionado que, en nuestro caso es un robot comercial de 6 grados de libertad modelo *IRB 7600* de la corporación multinacional ABB. La aplicación por la cual nos vimos inspirados en la selección del mismo, fue la de asistencia automática en el proceso de colada de acero fundido. El robot deberá realizar tanto la introducción de los crisoles al horno, como la extracción de los mismos y su posterior colada en los moldes. También se deberá tener en cuenta el cambio de herramienta para las diferentes operaciones.



Figura 1: ABB IRB 7600

Descripción y aspectos técnicos del robot.

1. Modelos:

El IRB 7600 está disponible en 5 modelos diferentes. La discriminación entre ellos se debe principalmente a la capacidad de carga y al alcance que provee:

Robot type	Handling capacity	Handling capacity for LeanID	Reach
IRB 7600	500 kg	-	2.55 m
IRB 7600	400 kg	390 kg	2.55 m
IRB 7600	340 kg	320 kg	2.8 m
IRB 7600	325 kg	290 kg	3.1 m
IRB 7600	150 kg	-	3.5 m

Figura 2: Standard robot versions

Para la aplicación que hemos seleccionado, encontramos que la mejor opción es utilizar el

modelo IRB 7600-400/2.55. Esta selección la vimos conveniente, ya que el modelo presenta mejores prestaciones de espacio de trabajo y mayor robustez.

2. Dimensiones:

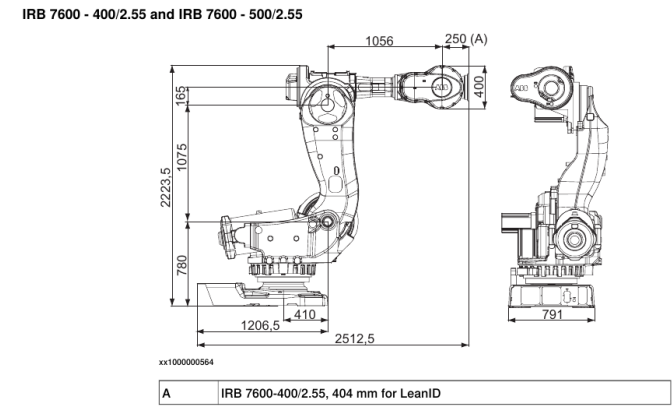


Figura 3: Dimensiones

3. Velocidades:

La siguiente tabla muestra la *velocidad máxima* soportada por cada articulación medida en [°/s]. A su vez las especifica para cada modelo. Es importante remarcar que la resolución de cada articulación esta comprendida entre los 0.001° a 0.005°.

IRB 7600	- 500/2.55	- 400/2.55	- 340/2.8	- 150/3.5	- 325/3.1
Axis No.	(°/s)	(°/s)	(°/s)	(°/s)	(°/s)
1	75	75	75	100	75
2	50	60	60	60	50
3	55	60	60	60	55
4	100	100	100	100	100
5	100	100	100	100	100
6	160	160	160	190	160

Figura 4: Velocidades

Descripción y relevamiento de las herramientas

A continuación agregamos un relevamiento de las dos herramientas que utilizamos en nuestra aplicación. La primera (presenta 7 filas) es utilizada para ingresar los crisoles con el acero crudo al horno y eventualmente si se desea, poder sacarlos pero NO colarlos. Por otra parte la segunda herramienta (1 fila) esta diseñada para un tipo específico de crisol que, mediante una configuración de diseño permite colar el acero fundido de su interior en un molde, sin volcarlo.

Cinemática Directa

Básicamente la finalidad de encontrar la solución al problema de cinemática directa de un robot consiste en establecer la posición y orientación del extremo del robot en función de los valores de los parámetros de las articulaciones.

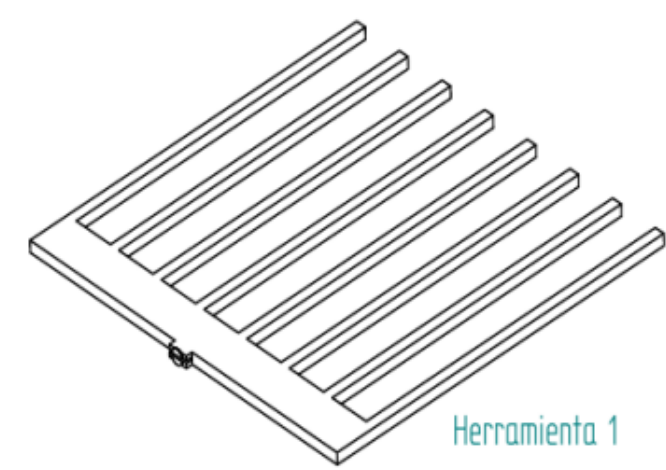


Figura 5: Herramienta 1

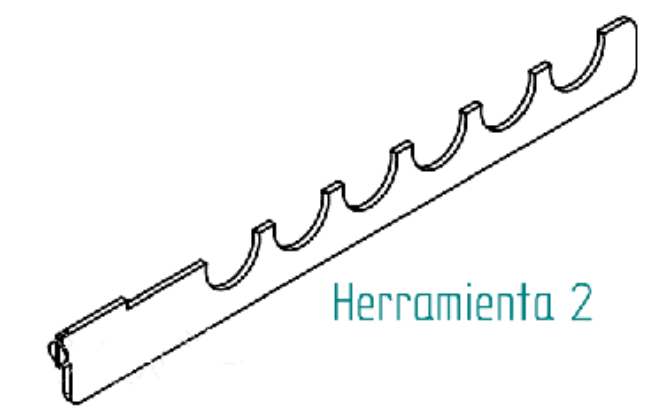


Figura 6: Herramienta 2

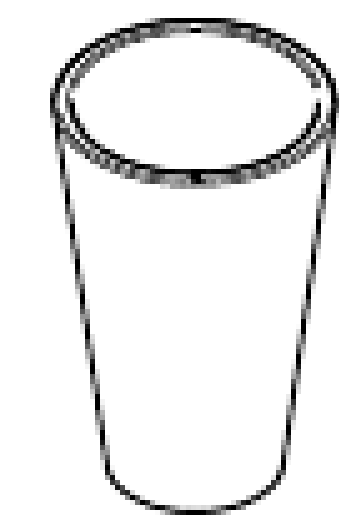


Figura 7: Crisol

Esto se logra gracias a la *asignación de sistemas de referencia* en cada una de las articulaciones y a la *lectura de los sensores internos* del robot. Ahora sí, mediante la composición de *transformaciones* básicas tanto de traslación como de rotación, será posible determinar la posición y orientación del extremo con respecto a un sistema de referencia fijo del mundo, situado por ejemplo, en la base del robot.

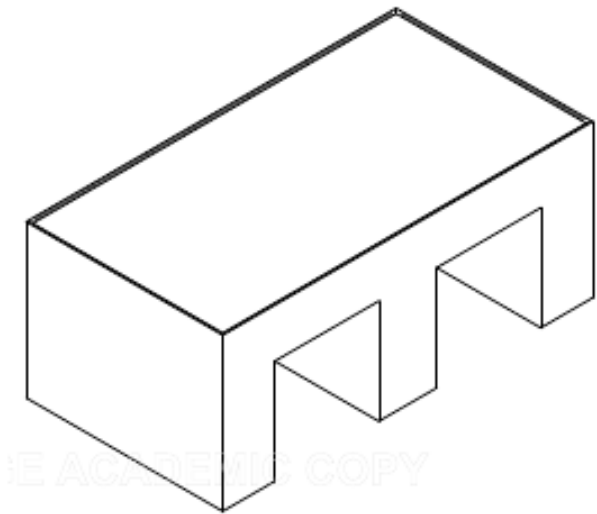


Figura 8: Mesa

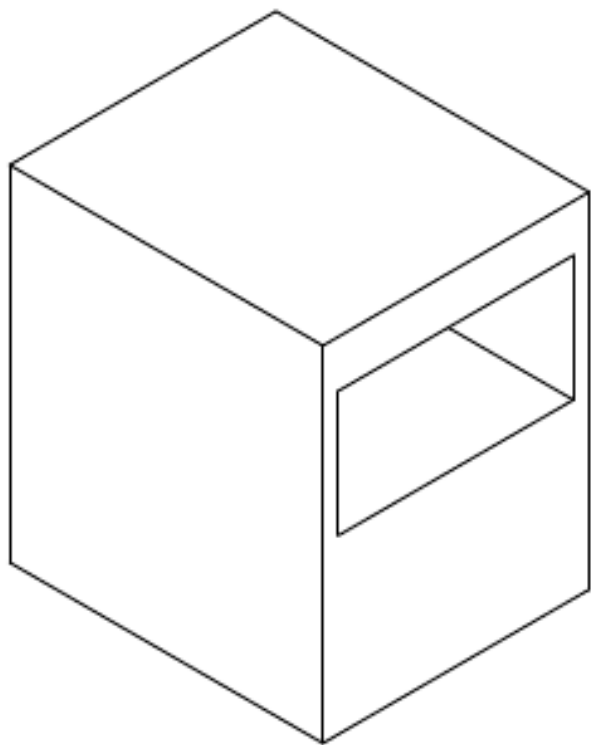


Figura 9: Horno

Parámetros Denavit - Hartenberg

Teniendo en cuenta que nuestro robot está compuesto por una concatenación de articulaciones de un grado de libertad y, que dos articulaciones consecutivas están conectadas por un eslabón (que no es más que un objeto rígido), se pueden establecer **cuatro parámetros**, dos relativos al tamaño y forma del eslabón y otros dos relacionados con la posición relativa entre dos eslabones consecutivos, parámetros relativos a la articulación que los enlaza.

A continuación se agrega una breve definición de los parámetros:

- a_i : Distancia entre los ejes i e $i+1$ de las articulaciones a los largo de la normal común. Este parámetro define en cierto modo el tamaño del eslabón, por lo que se le conoce como "Longitud del eslabón".

- α_i : Ángulo que existiría entre los ejes i e $i+1$ de las articulaciones si estos se cortasen en los puntos de corte de la línea normal común. Este parámetro en cierto modo mide "la forma del eslabón". También se lo conoce como "Ángulo de torsión del eslabón".
- d_i : Distancia entre las intersecciones de las normales comunes al eje de la articulación i , medida a lo largo de dicho eje. Esta medida en cierto modo expresa la distancia entre los dos eslabones, marcada por el tamaño y la forma de la articulación, por lo que se denomina "Longitud articular".
- θ_i : Ángulo que existiría entre las líneas normales comunes al eje de la articulación i si se cortasen en el mismo punto del eje de la articulación. De alguna forma expresa el ángulo que forman los dos eslabones, marcado nuevamente por la forma de la articulación, por lo que se denomina "ángulo articular".

Para la determinación de los parámetros se debe proceder siguiendo el paso a paso de la **Convención de Denavit - Hartenberg** (Ver Referencia), la cual propone tanto la asignación de ejes como la de sistemas de referencia.

Asignación de EJES

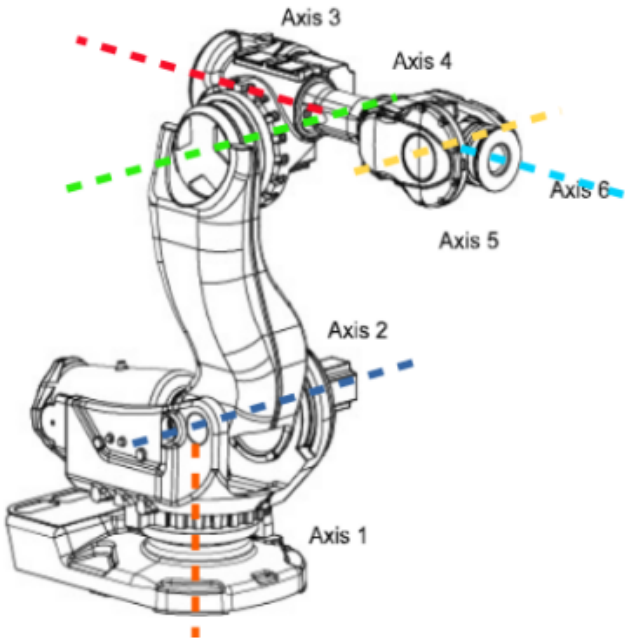


Figura 10: Asignación de ejes

Asignación de SISTEMAS DE REFERENCIA

Finalmente, una vez realizado los pasos previos, se llegó a las siguiente matriz de Denavit - Hartenberg (DH):

Habiendo determinado los parámetros DH, y teniendo en cuenta las cuatro transformaciones indi-

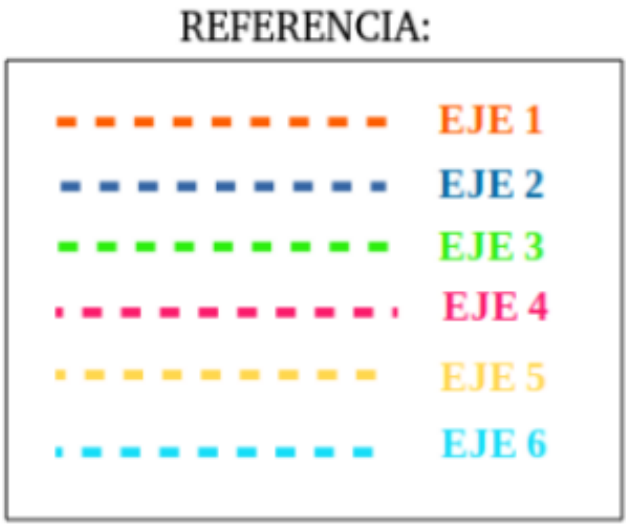


Figura 11: Referencias

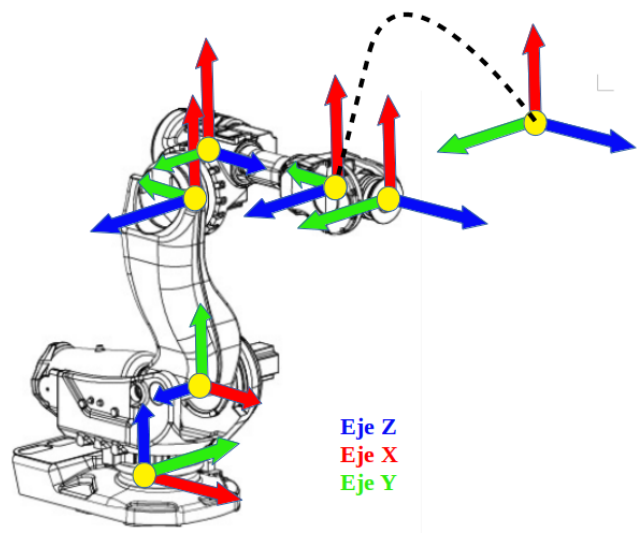


Figura 12: Asignación de sistemas de referencia

	θ_i	d_i	a_i	α_i	offset
1	$q_1(t)$	0.780	0.410	$\pi/2$	0
2	$q_2(t)$	0.000	1.075	0.000	0
3	$q_3(t)$	0.000	0.165	$-\pi/2$	0
4	$q_4(t)$	1.056	0.000	$\pi/2$	0
5	$q_5(t)$	0.000	0.000	$-\pi/2$	0
6	$q_6(t)$	0.250	0.000	0.000	0

Figura 13: Matriz DH

cadass en la Ec.(1), se obtiene la matriz de transformación homogénea para pasar del sistema $i - 1$ al i :

$$T_i^{i-1} = [Rot(z_{i-1}, \theta_i).Tras(z_{i-1}, d_i)].$$

$$Tras(x_i, a_i).Rot(x_i, \alpha_i)(1)$$

Implementación en MATLAB

Escogimos un vector de coordenadas articulares q_0 para el cual el robot se encontraría en su posición de reposo.

$q_0 = [0,330, 2,476, -1,189, 2,127, 0,563, -2,138];$
(2)

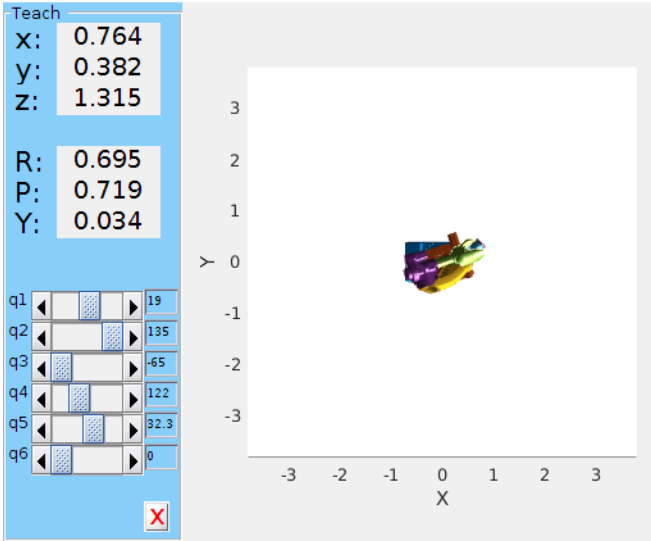


Figura 14: Vista superior

NOTA: Las transformaciones homogéneas totales correspondientes a ese arreglo de coordenadas articulares son (Ver Fig. 15): La de arriba corresponde al cálculo SIN considerar la herramienta (en vacío), y la de abajo es considerando la misma. Ésta tiene un largo equivalente de 1300[mm].

Tranformacion:

0.0114	0.7193	0.6946	0.7640
-0.0587	-0.6930	0.7186	0.3825
0.9982	-0.0489	0.0343	1.3150
0	0	0	1.0000

S =

0.0114	0.7193	0.6946	1.6670
-0.0587	-0.6930	0.7186	1.3166
0.9982	-0.0489	0.0343	1.3596
0	0	0	1.0000

Figura 15: T.H. Directa 0-6

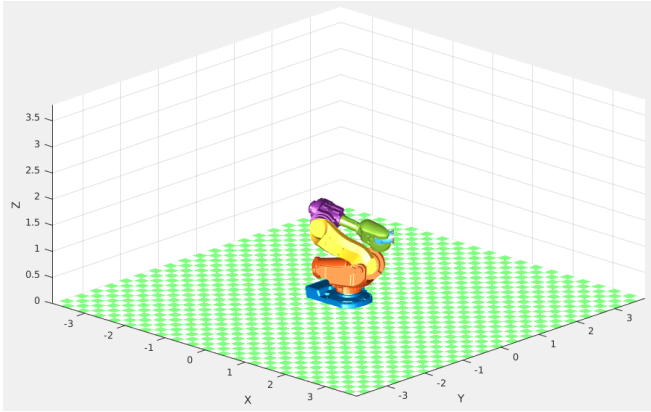


Figura 16: Vista de Reposo

Cinemática Inversa

En la cinemática directa resolvimos el problema de determinar la posición y orientación del extremo del robot según unos valores conocidos de las variables articulares. Sin embargo, desde el punto de vista práctico, resulta mas interesante el poder determinar los valores que tienen que tomar las variables articulares para que el extremo del robot se encuentre en una posición y orientación dada. Esto análisis se conoce como: **Problema cinemático inverso**. En donde tenemos como variables de entrada: La posición (x, y, z) , y la orientación (α, β, γ) . Y como variables de salida el valores de las variables articulares. $q_k = q_1, q_2, \dots, q_k$

$$q_k = f_k(x, y, z, \alpha, \beta, \gamma) \\ k = 1, 2, 3, \dots, n \therefore n : gdl$$

Para robots complejos (más de 3 grados de libertad) como es nuestro caso, para resolver este problema, nos basamos en la **Solución de Pieper** la cual nos propone dividir el problema en dos partes (también conocido como desacoplamiento cinemático), esto se debe a que una vez posicionada la muñeca en el espacio en la que los tres ejes se cortan, el movimiento de las misma en torno a sus ejes no altera la posición espacial del punto de corte. Dicho esto, en primer lugar se debe analizar la posición (con las tres primeras articulaciones), y luego la orientación (con las últimas tres articulaciones).

Nótese en la (Fig.17) que el punto de corte "muñeca", se define en el punto de intersección de los ejes de las últimas tres articulaciones. Para poder resolver la primera parte del problema cinemático inverso se debe referenciar las coordenadas cartesianas del extremo del robot al punto corte, y esto se logra mediante la siguiente expresión (en correspondencia con la (Fig.17)):

$$d_0^c = d_0^6 - d_6 \cdot \hat{z}_6 \quad (3)$$

Parte I - Posicionamiento

Una característica de la cinemática inversa es que la solución (en la mayoría de los casos y para $k > 1$) no es única. Para esta primera parte, la solución que propusimos fue basada en el método geométrico, haciendo uso de relaciones trigonométricas y de la propia geometría del robot. En ellas intervienen las coordenadas del extremo del robot y sus respectivas orientaciones, las coordenadas articulares y las dimensiones físicas de cada eslabón.

A continuación anexamos algunas figuras a modo de ilustrar las tres primeras articulaciones, luego de realizar el desacoplamiento:

Es importante mencionar que, las coordenadas (P_x, P_y, P_z) que allí aparecen son las que corresponden al punto de corte, respecto del sistema de

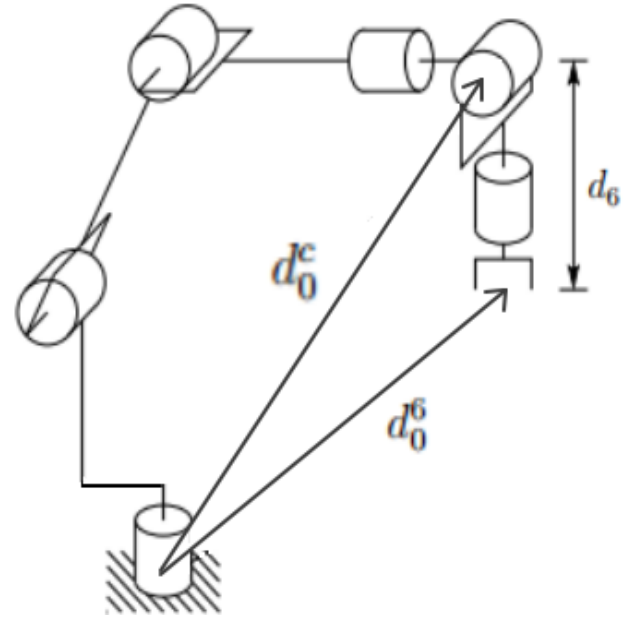


Figura 17: Punto de corte

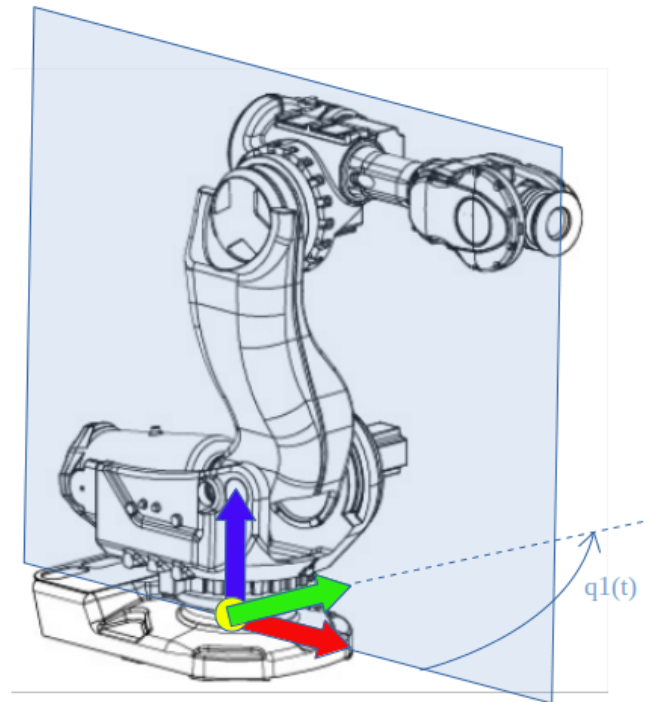


Figura 18: Robot desacoplado

referencia de la base S_0 En nuestra implementación las denotamos como (P_{xm}, P_{ym}, P_{zm})

$$\begin{aligned} P_{xm}^0 &= P_{x6}^0 - d_6 \cdot r_{13} \\ P_{ym}^0 &= P_{y6}^0 - d_6 \cdot r_{23} \\ P_{zm}^0 &= P_{z6}^0 - d_6 \cdot r_{33} \end{aligned}$$

A la hora de implementarlo en MATLAB, la matriz de transformación T_6^0 que utilizamos como dato entrada del problema, la obtuvimos previamente de aplicar cinemática directa. Expresándola en forma general:

$$T_6^0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_{x6} \\ r_{21} & r_{22} & r_{23} & P_{y6} \\ r_{31} & r_{32} & r_{33} & P_{z6} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ahora si, una vez referenciado el punto de corte, podemos expresar las coordenadas articulares en función de las coordenadas cartesianas:

$$q_1 = \arctg\left(\frac{P_{ym}}{P_{xm}}\right) \quad (4)$$

Esta última presenta 2 soluciones posibles:

$$q_1 = \theta_1 \therefore q_1 = \theta_1 + \pi$$

Habiendo definido un valor de q_1 , podemos observar que las siguientes dos articulaciones quedan contenidas en un mismo plano que, para hallar las relaciones trigonométricas que las vincule con las coordenadas cartesianas, trasladamos el sistema de referencia a la segunda articulación. Esto se logra, calculando la matriz de transformación T_1^0 y referenciando el punto de corte a este nuevo sistema S_1 :

$$\bar{p}_m^1 = (T_1^0)^{-1} \cdot \bar{P}_m \quad (5)$$

En la siguiente figura se puede observar como se redujo la cadena cinemática a una simplificada situación de un robot planar de dos grados de libertad.

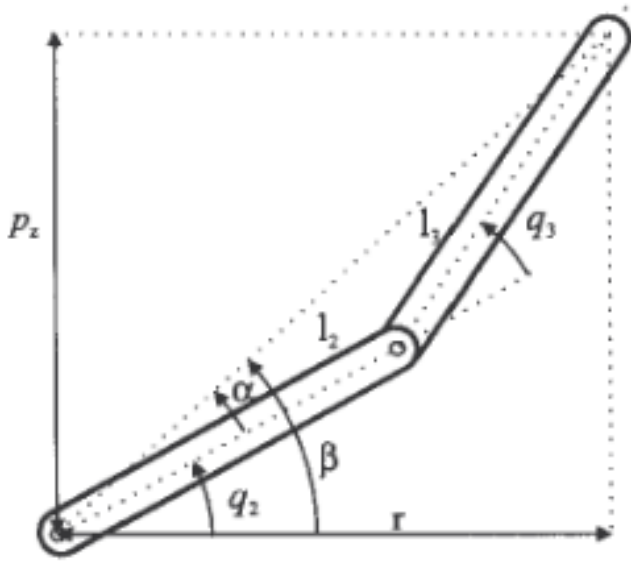


Figura 19: Simplificación-Robot planar 2gdl

En donde r corresponde:

$$r^2 = (p_{xm}^1)^2 + (p_{ym}^1)^2$$

Y por otro lado, si calculamos la hipotenusa formada por P_{zm} y r :

$$(d_m^1)^2 = r^2 + (P_{zm}^1)^2$$

Aplicando el teorema del coseno al triangulo formado por los dos eslabones y la hipotenusa del triangulo rectángulo, obtenemos:

$$r^2 + (P_{zm}^1)^2 = l_2 + l_3 - 2 * \cos(\pi - q_3)$$

En donde $\cos(\pi - q_3) = -\cos(q_3)$, entonces:

$$r^2 + (P_{zm}^1)^2 = l_2 + l_3 + 2 \cdot l_2 \cdot l_3 \cdot \cos(q_3) \quad (6)$$

Despejamos ahora el $\cos(q_3)$:

$$\cos(q_3) = \frac{r^2 + (P_{zm}^1)^2 - l_2 - l_3}{2 \cdot l_2 \cdot l_3}$$

Haciendo uso de la siguiente identidad:

$$\text{sen}^2(x) + \cos^2(x) = 1$$

Expresamos la solución en función de la tangente como:

$$q_3 = \arctg\left(\frac{\pm \sqrt{1 - \cos(q_3)^2}}{\cos(q_3)}\right) \quad (7)$$

Por otro lado vemos que la segunda articulación q_2 está definida por las orientaciones mediante:

$$q_2 = \beta - \alpha \quad (8)$$

El ángulo β corresponde a la orientación del eslabón 2 respecto del eje "r":

$$\beta = \text{atg}\left(\frac{P_{zm}^1}{r}\right) = \text{atg}\left(\frac{P_{zm}^1}{\pm \sqrt{(p_{xm}^0)^2 + (p_{ym}^0)^2}}\right) \quad (9)$$

El ángulo α corresponde a la orientación del eslabón 2 respecto de la hipotenusa " d_m^1 ":

$$\alpha = \text{atg}\left(\frac{l_3 \cdot \text{sen}(q_3)}{l_2 + l_3 \cdot \cos(q_3)}\right) \quad (10)$$

Habiendo definido estas dos orientaciones, la coordenada articular q_2 queda definida en función de las coordenadas cartesianas:

$$q_2 = \text{atg}\left(\frac{P_{zm}^1}{\pm \sqrt{(p_{xm}^0)^2 + (p_{ym}^0)^2}}\right) - \text{atg}\left(\frac{l_3 \cdot \text{sen}(q_3)}{l_2 + l_3 \cdot \cos(q_3)}\right) \quad (11)$$

Parte 2 - Orientación

Finalmente queda por determinar los valores de las tres últimas tres articulaciones que orientarán el extremo del robot para que el problema cinemático inverso quede totalmente definido. Para la solución de esta segunda parte, nos hemos basado en el método matricial, haciendo uso particularmente de las submatrices de rotación.

Si recordamos de cinemática directa :

$$\text{Rot}_{extremo}^0 = \text{Rot}_3^0 \cdot \text{Rot}_{extremo}^3 \quad (12)$$

En donde de esta última expresión la submatriz $\text{Rot}_{extremo}^0$ es dato de entrada, y habiendo resuelto el problema cinemático inverso para las primeras tres articulaciones resulta que Rot_3^0 también es conocida.

Multiplicando a ambos miembro por $(\text{Rot}_3^0)^{-1}$:

$$(Rot_3^0)^{-1} \cdot Rot_{extremo}^0 = (Rot_3^0)^{-1} \cdot Rot_3^0 \cdot Rot_{extremo}^3$$

Por último, expresamos de forma explícita la incógnita:

$$Rot_{extremo}^3 = (Rot_3^0)^{-1} \cdot Rot_{extremo}^0 \quad (13)$$

Las matrices de transformación para las articulaciones de la muñeca son :

$$T_4^3 = \begin{bmatrix} \text{sen}(q_4) & 0 & \text{cos}(q_4) & 0 \\ -\text{cos}(q_4) & 0 & \text{sen}(q_4) & 0 \\ 0 & -1 & 0 & l_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

$$T_5^4 = \begin{bmatrix} \text{cos}(q_5) & 0 & \text{sen}(q_5) & 0 \\ \text{sen}(q_5) & 0 & -\text{cos}(q_5) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

$$T_6^5 = \begin{bmatrix} \text{cos}(q_6) & -\text{sen}(q_6) & 0 & 0 \\ \text{sen}(q_6) & \text{cos}(q_6) & 0 & 0 \\ 0 & 0 & 1 & l_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

Para determinar los valores de q_4, q_5, q_6 se utiliza la siguiente igualdad:

$$Rot_{ext}^3 = (Rot_3^0)^T \cdot [x_{ext}^0, y_{ext}^0, z_{ext}^0] \quad (17)$$

Resolviendo las ecuaciones, obtenemos:

$$q_4 = \arccos\left(-\frac{z_{extremo}^0}{\text{sen}(q_5)}\right) \quad (18)$$

$$q_5 = \arccos(x_{extremo}^0 \cdot (-\text{sen}(q_3) \cdot \text{cos}(q_3) + \text{cos}(q_2) \cdot \text{sen}(q_3)) + \dots + y_{extremo}^0 \cdot (\text{cos}(q_2) \cdot \text{cos}(q_3) - \text{sen}(q_2) \cdot \text{sen}(q_3))) \quad (19)$$

Por simplicidad, se redujo la expresión de q_6 :

$$a = \frac{-x_{extremo}^0 \cdot (-\text{sen}(q_3) \cdot \text{cos}(q_3) + \text{cos}(q_2) \cdot \text{sen}(q_3))}{\text{sen}(q_5)}$$

$$b = \frac{-y_{extremo}^0 \cdot (\text{cos}(q_2) \cdot \text{cos}(q_3) - \text{sen}(q_2) \cdot \text{sen}(q_3))}{\text{sen}(q_5)}$$

q_6 resulta:

$$q_6 = \arccos(a + b) \quad (20)$$

Implementación en MALTAB

Para la implementación del problema cinemático inverso, partimos de una transformación homogénea calculada por cinemática directa, como dato de entrada. Para la misma escogimos un vector de coordenadas articulares (que es el que corresponde al posicionamiento del robot para ingresar al horno), para obtener la matriz de transformación T . A continuación de esquematiza el robot en dicha posición. Fig.(20)

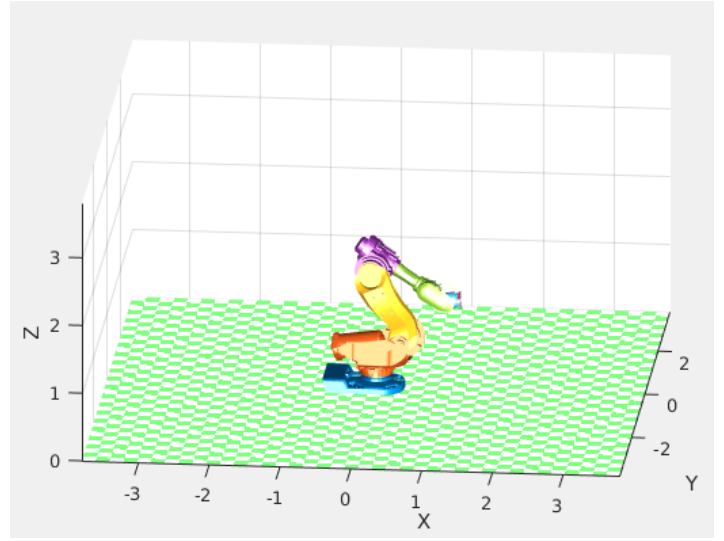


Figura 20: Cinemática inversa

$$q_{inv} = [-0,077, 1,986, -0,990, 2,975, 0,602, -3,053]$$

Se han evaluado las múltiples soluciones y en base a ello pudimos confeccionar un árbol con los valores de las coordenadas articulares. Fig(21)

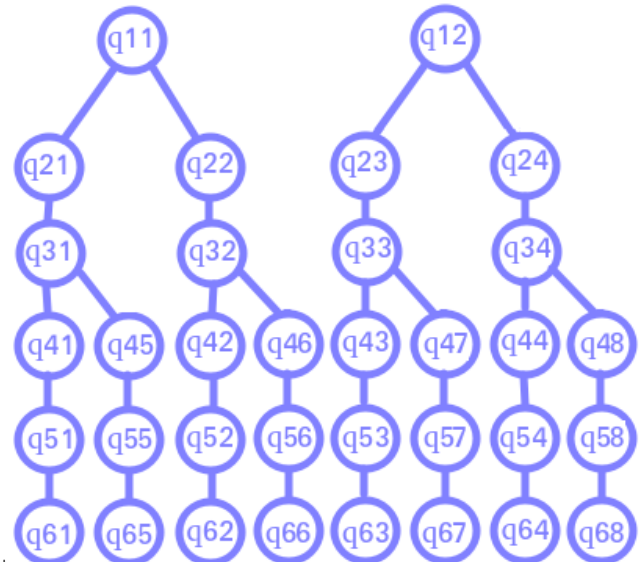


Figura 21: Árbol de soluciones

A continuación presentamos los valores obtenidos para la solución propuesta:

$$q_1 = [-0,0779, 3,0637]$$

$$q_2 = [1,9685, -0,3873, 1,9685, -0,3873]$$

$$q_3 = [-2,4017, 2,4017, -2,4017, 2,4017]$$

$$q_4 = [3,1416, 3,1416, 3,1416, 3,1416]$$

$$q_5 = [0,5972, 1,2030, 0,5972, 1,2030]$$

$$q_6 = [3,1416, 1,9664, 3,1416, 1,9664]$$

Otra alternativa para el cálculo de la cinemática inversa es haciendo uso de la función *objetoRobot.ikine* que dispone el *toolbox de Perter Corke*. La misma busca la solución mediante un método numérico iterativo.

Matriz Jacobiana

En muchas aplicaciones es de gran utilidad disponer de la relación entre las velocidades de las

coordenadas articulares y las posiciones y orientaciones del extremo del robot. La relación entre ambos vectores de velocidad se obtiene a través de la denominada **matriz Jacobiana**. Fig (22).

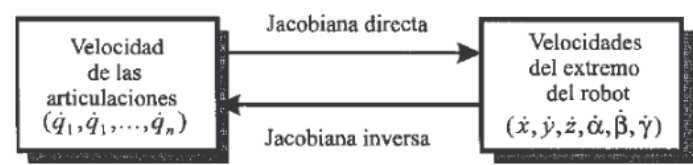


Figura 22: Relación espacio Cartesiano- Articular

Expresando la dependencia de las variable cartesianas en función de las coordenadas articulares:

$$\begin{aligned} x &= f_x(q_1, \dots, q_n) \\ y &= f_y(q_1, \dots, q_n) \\ z &= f_z(q_1, \dots, q_n) \\ \alpha &= f_\alpha(q_1, \dots, q_n) \\ \beta &= f_\beta(q_1, \dots, q_n) \\ \gamma &= f_\gamma(q_1, \dots, q_n) \end{aligned}$$

Si derivamos estas expresiones respecto al tiempo obtenemos:

$$\begin{aligned} \dot{x} &= \sum \frac{\partial f_x}{\partial q_i} \cdot \dot{q}_i \\ \dot{y} &= \sum \frac{\partial f_y}{\partial q_i} \cdot \dot{q}_i \\ \dot{z} &= \sum \frac{\partial f_z}{\partial q_i} \cdot \dot{q}_i \\ \dot{\alpha} &= \sum \frac{\partial f_\alpha}{\partial q_i} \cdot \dot{q}_i \\ \dot{\beta} &= \sum \frac{\partial f_\beta}{\partial q_i} \cdot \dot{q}_i \\ \dot{\gamma} &= \sum \frac{\partial f_\gamma}{\partial q_i} \cdot \dot{q}_i \end{aligned}$$

Por último lo expresamos en forma matricial:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = J \cdot \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} \tag{21}$$

Siendo **J** la matriz Jacobiana:

$$J = \begin{bmatrix} \frac{\partial f_x}{\partial q_1} & \dots & \frac{\partial f_x}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_\gamma}{\partial q_1} & \dots & \frac{\partial f_\gamma}{\partial q_n} \end{bmatrix}$$

Observaciones:

- La matriz $J(q)$ se denomina matriz Jacobiana y al ser función del tiempo, para cada instante de tiempo se tiene un valor distinto de la ella.
- El número de filas de la matriz igual al número de grados de libertad, en este caso .

- El número de columnas es igual al número de articulaciones del manipulador, que al igual es 6.

Implementación en MATLAB

Para la obtención de la matriz Jacobiana numérica hicimos uso del método objetoRobot.jacob0() el cual pide que se le pase como parámetro un arreglo de coordenadas articulares.

La matriz Jacobiana correspondiente al arreglo q_0 es:

La matriz Jacobiana Jq0 es:

-0.3825	-0.5060	0.1216	0.0530	-0.1501	0
0.7640	-0.1738	0.0418	-0.0564	0.1382	0
0.0000	0.4368	1.2826	0.1089	0.1445	0
0.0000	0.3249	0.3249	0.9079	0.3964	0.6946
0.0000	-0.9458	-0.9458	0.3119	-0.4221	0.7186
1.0000	-0.0000	-0.0000	-0.2799	0.8153	0.0343

Figura 23: Matriz Jacobiana directa - Numérica

Matriz Jacobiana Inversa

Con la matriz Jacobiana inversa buscamos determinar las velocidades articulares para que el extremo del robot se mueva a una velocidad, lineal y angular, determinada. Para obtenerla se proponen los siguiente métodos:

Primer alternativa: Supuesta conocida la relación directa se podría obtener la matriz inversa, invirtiendo simbólicamente la matriz. Es de difícil realización en la práctica ya que supone la inversión simbólica de una matriz de 6x6, cuyos elementos son funciones trigonométricas resulta complejo. Como **segunda alternativa** puede plantearse la evaluación numérica de la matriz J para una configuración q_i dada e invirtiendo numéricamente ésta. Sin embargo, se debe tener en cuenta que el valor numérico de la Jacobiana va cambiando constantemente (depende de t). La **tercera alternativa** es repetir el procedimiento seguido para la obtención de la Jacobiana directa, pero ahora partiendo de un modelo cinemático inverso.

Cabe destacar que existen puntos en donde la matriz Jacobiana no sea invertible por ser su determinante, denominado Jacobiano, nulo. Estas configuraciones en donde el Jacobiano se anula son llamadas configuraciones singulares.

Implementación en MATLAB

La matriz Jacobiana inversa correspondiente al arreglo q_0 es Fig.(24) :

Singularidades.

Hacemos mención de algunas características de una configuración singular:

La INVERSA de la matriz Jacobiana Jq0 es:

-0.5205	1.5151	0.0000	0.0130	0.0045	-0.3566
-1.7067	-0.5863	0.2335	-0.0470	0.0552	-0.2048
0.3449	0.1185	0.8962	-0.1600	0.1527	0.0414
1.2659	2.2666	-1.5172	1.4153	-1.3084	-1.2476
1.1433	-0.9883	-0.5965	0.4950	-0.5355	1.1923
-1.6702	-2.1801	1.7951	-0.5959	1.9186	1.0268

Figura 24: Matriz Jacobiana inversa - Numérica

- A velocidades finitas del extremo pueden corresponderles velocidades infinitas de las articulaciones.
- A fuerzas y pares finitos del extremo puede corresponderles fuerzas y pares infinitas de las articulaciones.
- Se pierde al menos un grado de libertad (alineación de ejes articulares).
- El movimiento en ciertas direcciones es imposible(fuera del espacio de trabajo).
- El rango del Jacobiano es menor al orden de la matriz Jacobiana.

En los puntos del límite del espacio de trabajo del robot existen singularidades, porque allí no es posible los movimientos en dirección axial del robot (por ejemplo que se encontrase totalmente estirado), lo que supone la pérdida de un grado de libertad en esa dirección del espacio, por alineación de 2 o mas de sus ejes articulares. En nuestro robot pasaría en las articulaciones q_4 y q_6 . Ver Fig (26)

Elipsoide de Manipulabilidad

Definimos como Elipsoide de Manipulabilidad a la capacidad de cambio en posición y orientación del efector final de un sistema robótico en una configuración dada. Esto quiere decir que, el efector final tiene mayor capacidad de movimiento en la dirección del eje mayor del elipsoide, por otra parte, en la dirección del eje menor la capacidad de desarrollar velocidad será menor.La elipsoide de manipulabilidad es una herramienta muy útil para ver gráficamente las posibilidades de velocidades a desarrollar en un punto de trabajo definido. Lo interesante de esta elipsoide es que también podemos observar gráficamente las singularidades, donde en vez de ver un elipsoide veremos una elipse en el plano que contiene a la singularidad. Es decir, desaparece uno de los ejes de la elipsoide.

Implementación en MATLAB
Haciendo uso del método `vellipse()` de la clase SerialLink del toolbox de Peter Corke se lograron obtener las elipsoides de manipulabilidad para algunos puntos de interés.

En la Fig.(25)se muestra el elipsoide de manipulabilidad para la posición de reposo. Se observa que

el efector final dispone de mayor capacidad de movimiento en la dirección del eje mayor del elipsoide. Por otra parte, en la dirección del eje menor la capacidad de desarrollar velocidad será menor. Este elipsoide simboliza la capacidad de manipulación.

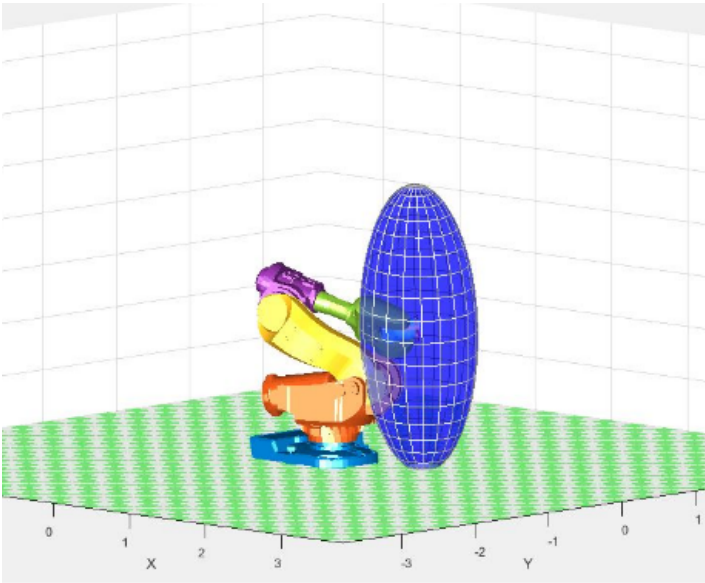


Figura 25: Elipsoide-Punto espacio de trabajo

Ahora tomamos un punto singular, por ejemplo la configuración:

$$q_{singular1} = [90, 1, 90, 173, 0, 0]$$

Realizando el cálculo de la matriz Jacobiana para este punto, vemos que 2 columnas (las correspondiente a la articulaciones q_4 y q_6), son coincidentes (linealmente dependientes). Ver Fig.(26). Al presentar esta dependencia lineal, el rango de la matriz resulta ser de 5 y por la tanto su determinante resulta ser nulo.

Js1 =

-2.7878	-0.0000	-0.0000	0	-0.0305	0
0.0000	-0.2065	-0.1878	0	-0.0043	0
-0.0000	2.3778	1.3029	0	0.2481	0
0.0000	1.0000	1.0000	0.0000	0.9925	0.0000
0	-0.0000	-0.0000	0.9998	-0.0021	0.9998
1.0000	0.0000	0.0000	0.0175	0.1219	0.0175

Figura 26: Matriz Jacobiana Singular

Como mencionamos mas arriba en las singularidades en lugar de observar el elipsoide, vemos una elipse.

Espacio de trabajo

En la Fig.(28) observamos el espacio de trabajo del manipulador. El mismo contempla todo los puntos en donde se podría situar el extremo de robot. Extraído del datasheet del fabricante.

Implementación en MATLAB

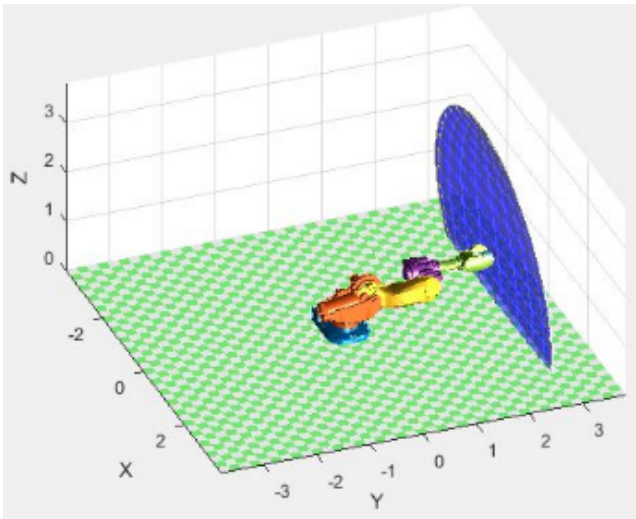


Figura 27: Configuración singular

Para su representación en MATLAB, usamos una función del lenguaje llamada *scatter()* la cual patea una dispersión de puntos en el espacio. Asignando adecuadamente la coordenada articular que se desee representar se obtuvieron los siguientes resultados:

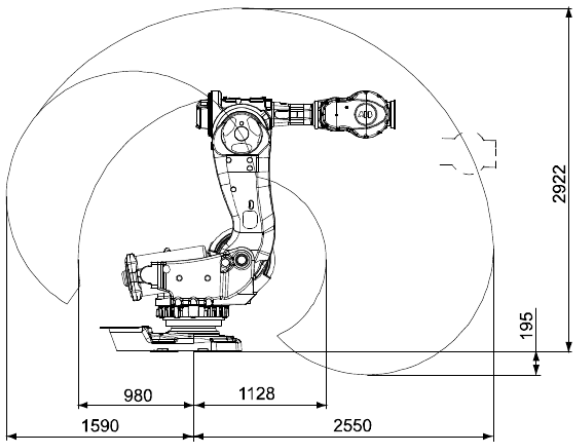


Figura 28: Espacio de trabajo

Si observamos al Fig.(29) podemos notar, que la altura máxima en z habiendo estirado por completo el robot, es de $z = 2,913$ frente a $z = 2,922$ que nos indica la Fig.(28)

En la Fig.(30) ubicamos el robot totalmente horizontal y mostramos el alcance máximo en esa posición.

Planificación de trayectorias

Para realizar la planificación de trayectoria se debe realizar un muestreo de los sucesivos y finitos puntos por los que debe posicionarse el robot para generar la trayectoria deseada. Cada uno de esos puntos vendrán dados por una 6-upla, típicamente $x, y, z, \alpha, \beta, \gamma$. Utilizando la cinemática inversa, convertimos cada uno de estos puntos en coordenadas articulares $(q_1, q_2, q_3, q_4, q_5, q_6)$, asegurando la continuidad en cada uno de estos puntos y asegurarse así la trayectoria. Es necesario también contar

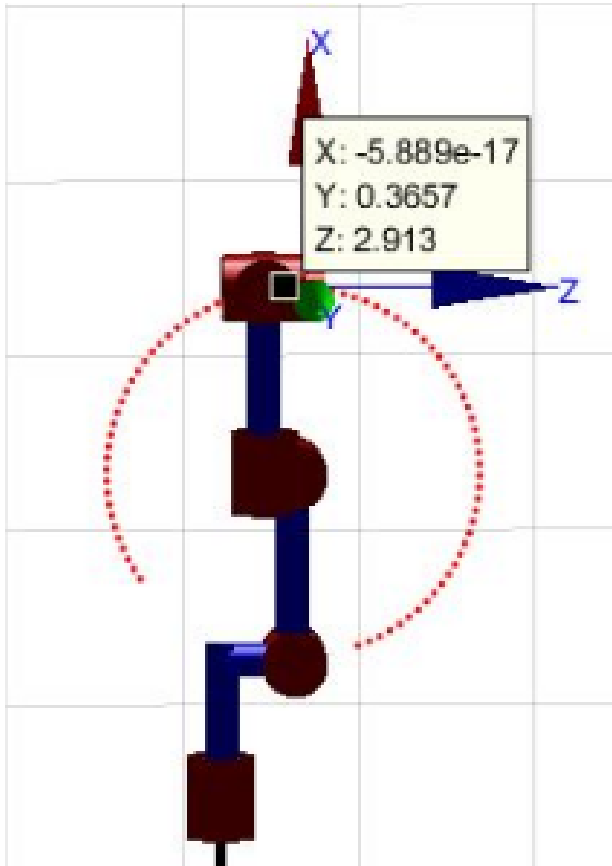


Figura 29: Espacio de trabajo-MATLAB

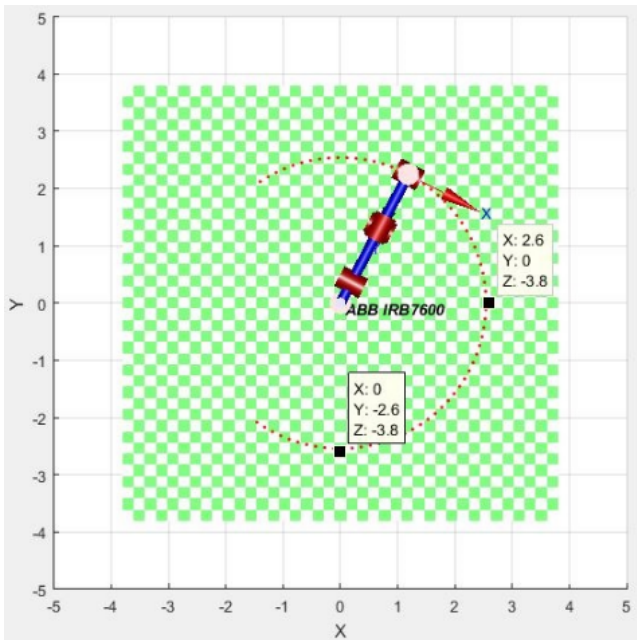


Figura 30: Espacio de trabajo-MATLAB

con métodos de interpolación para generar una expresión $q_i(t)$ para cada variable articular que pase o se aproxime a los sucesivos puntos de modo que, siendo una trayectoria realizable por los actuadores se transforme en una trayectoria cartesiana lo más próxima a la especificada por el programa del usuario (en cuanto a precisión, velocidad, etc.). La trayectoria articular debe ser muestreada para generar referencias al control dinámico. Para el propósito de este proyecto fue necesario localizar los puntos característicos de cada tarea según su trayectoria de trabajo. Comenzando y finalizando cada tarea en el punto q_0 Ec.(2) y analizando cada una de

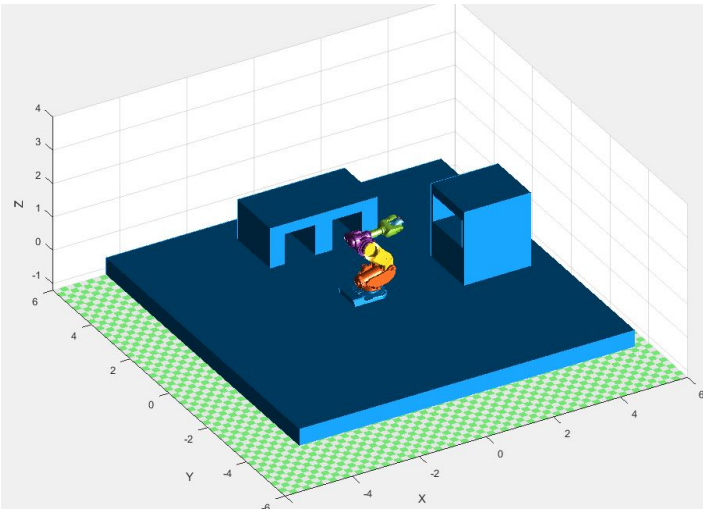


Figura 31: Espacio de trabajo-MATLAB

las orientaciones y puntos fundamentales para la realización del trabajo de colada, pudimos generar 12 tareas con su trayectoria específica, en las que mediante la selección de cualquiera de las 2 herramientas disponibles el robot estará en condiciones de:

- Tomar la herramienta 1 (Ver Fig. 5), posicionarse en la mesa para retirar las filas de crisoles, retirarlas e ingresarlas al horno para su fundición. Una vez fundidas, este será capaz de retirarlas del horno con el metal en estado líquido y las posicionará nuevamente en la mesa de donde las retiró, apoyándolas suavemente en la mesa y retirarse a la posición q_0 Ec.(2) para luego dejar la herramienta 1 en su posición específica.
- Tomar la herramienta 2 (Ver Fig. 6), posicionarse en la entrada del horno, retirar las sucesivas filas y para a cada una realizarle la colada en el molde correspondiente que se encuentra en la mesa.

Los crisoles que se usan para la tareas de colada serán siempre los mismos (normalizados), ya que las trayectorias del robot están configurados para ello. En el caso de querer fundir y que no haya que realizar colada (la fundición se solidifica en el crisol), podremos retirar las filas completas, pudiendo así utilizar diversos crisoles (iguales) en esta aplicación.

Implementación en MALTAB

Para generar las 11 diferentes tareas que debe realizar nuestro robot, lo que se hizo fue ubicar los puntos significativos para cada uno de ellas y generar la trayectoria a partir de `jtraj()` entre los sucesivos puntos iniciales y finales, y asegurarnos que realice la tarea correctamente. Lo más importante es que debe mantenerse el eje Z del efector final paralelo al plano $x_0 - y_0$ cuando se realicen

movimientos con los crisoles, ya que estos en su interior tienen acero fundido. A partir del método de cinemática inversa “`ikine()`” se lograron obtener con éxito y gran precisión todos los puntos para la generación de trayectorias. Cada trayectoria empieza en q_0 y termina en q_0 por lo que podemos hacer que cada una suceda a la otra en el orden adecuado según la aplicación. Los puntos más significativos son:

- Punto inicial q_0
- Entrada al horno, con un punto anterior para posicionarse antes de entrar al horno con la herramienta.
- Posición herramienta 1, con un punto anterior para que la herramienta baje lentamente para luego realizar el acople de la herramienta 1. Lo mismo con la herramienta 2.
- Posiciones de cada una de las 7 filas dentro del horno.
- Posiciones de la mesa de trabajo para cada una de las filas de crisoles, donde en un lugar se sitúan las 7 filas para retirar con la herramienta 1 y en otra zona de la mesa el espacio para realizar la colada de una fila con el metal líquido.

Como se mencionó anteriormente, para obtener la configuración articular según la coordenada cartesiana y la orientación del efector final deseada se utilizó el método “`ikine()`”, el cual pide una matriz de transformación homogénea como argumento. Estas matrices se obtuvieron a partir del análisis de la parte de rotación de la matriz de transformación homogénea.

Las matrices de orientación usadas fueron Fig.(32):

Las tareas que se pueden realizar son 11 en total. Se pueden observar en la Fig.(33):

Donde en el lugar de las X se coloca la posición deseada. Se logra observar que la parte de rotación de las matrices se ve modificada de una orientación a la otra. Estas matrices T son las que se les insertan como argumentos al método `ikine` para obtener todos los arreglos de $q = [q_1, q_2, q_3, q_4, q_5, q_6]$ para cada punto de interés, y con ellos generar las trayectorias.

También se realizó el análisis de las variaciones de posición, velocidad y aceleración con respecto al tiempo en cada una de las trayectorias, donde se logran diferenciar que las articulaciones se mueven de un punto a otro con su velocidad y aceleración. Algunos ejemplos son indicados a continuación: Por ejemplo para la tarea “1. Agarrar herramienta 1” tenemos 4 movimientos. Ir de q_0 a la posición de la herramienta 1 con un z mayor al z de la herramienta, bajar desde este z al z herramienta para realizar


```
%ORIENTACION HORNO
T=[ 0      0      1      X
    0     -1      0      X
    1      0      0      X
    0      0      0      1.0000];

% ORIENTACION MESA
T=[-1      0      0      X
    0      0      1      X
    0      1      0      X
    0      0      0      1.0000];

%ORIENTACION HERRAMIENTAS
T=[1      0      0      X
    0      0      0      X
    0      0      0      X
    0      0      0      1.0000];
-
```

Figura 32: Transformaciones

```
ELIJA RUTINA DEL IRB7600
1.Agarrar herramienta 1
2.Agarrar herramienta 2
3.Dejar herramienta 1
4.Dejar herramienta 2
5.Agarrar crisoles 7x6 en mesa
6.Demo filas de crisoles en HORNO
7.DEJAR en el Horno
8.SACAR del Horno crisoles 7x6
9.SACAR del Horno crisoles fila (elegir fila)
10.Hacer colada
11.DEJAR Crisoles 7x6 en mesa
0.Fin de operacion
```

Figura 33: Rutinas de operación.

el acople y retirar la herramienta para finalizar con el movimiento a q_0 nuevamente. Los movimientos más significativos son:

En esta Fig.(34) tenemos los perfiles de posición, velocidad y aceleración de cada una de las 6 articulaciones. Podemos observar que la máxima velocidad la obtuvo q_4 al igual que la aceleración, en este caso se mueven todas las articulaciones.

En cambio para la realización del movimiento en la Fig.(35) logramos ver que las articulaciones q_3 y q_5 sufren modificaciones en su posición, velocidad y aceleración para agarrar la herramienta 1 se debe mantener la posición de las coordenadas x e y y modificar el z. En este caso pasamos de un $z = 1,55$ a un $z = 1,5$, el movimiento debe ser lento para evitar la colisión del efector final con la herramienta

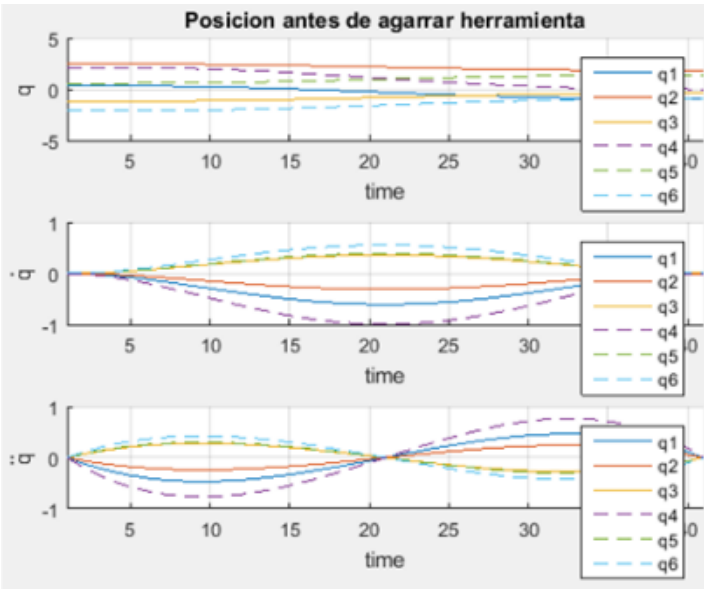


Figura 34: Perfil 1

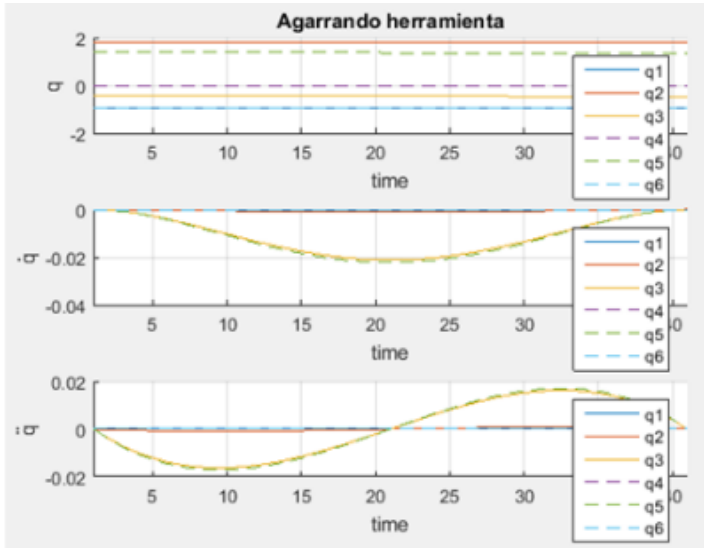


Figura 35: Perfil 2

y poder realizar el acople de forma suave y segura. Comparando las escalas de velocidades y aceleración de la Fig.(34) y la Fig.35 podemos ver que en el caso de la Fig (35) éstas son 100 veces más bajas con respecto a la Fig.(34). En el caso de la tarea “10.Hacer colada” se logra observar que en el momento de la realización de la colada solamente la articulación q_6 es la que realiza movimiento, en este caso gira 120 para que el metal líquido de los crisoles caiga sobre los moldes apoyados en la mesa.

Sensores

Un sensor es un dispositivo eléctrico y/o mecánico que convierte magnitudes físicas (como pueden ser luz, magnetismo, calor, etc), a valores medibles de dicha magnitud. Para nuestro robot el fabricante no provee información sobre sus sensores internos en el "Product specification

. De igual manera intuimos que como mínimo debe tener un encoder por cada articulación. Por otro lado nos pareció conviene proponer (llegado al caso de que no disponga del mismo), un sensor de

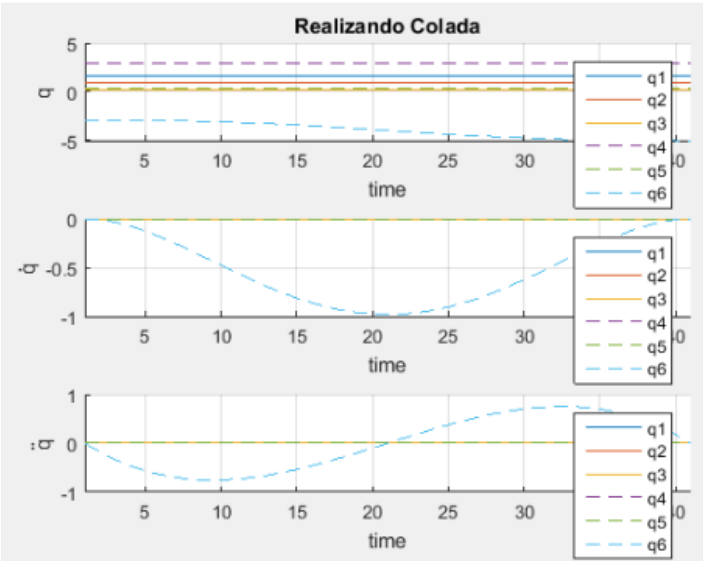


Figura 36: Perfil 3

temperatura, debido a que para nuestra aplicación el robot opera el las proximidades del horno. Asegurándonos de esta forma que no se excedan los limites requeridos de operación. Ver Fig.(37)

Description	Standard/Op-tion	Temperature
Manipulator during operation	Standard	+ 5 °C ! (+ 41 °F) to + 50 °C (+ 122 °F)
For the controller	Standard/Op-tion	See Product specification - Controller IRC5 with FlexPendant
For the spot welding cabinet	Standard	+5 °C (41 °F) to + 45 °C (+ 113 °F)
Complete robot during transportation and storage, for short periods (not exceeding 24 hours)	Standard	- 25 °C (- 13 °F) to + 55 °C (+ 131 °F) up to + 70 °C (+ 158 °F)

Figura 37: Requerimientos de operación

Conclusión, desafíos y mejoras.

La implementación de este robot en la industria trae consigo una evasión a cualquier riesgo que puede sufrir un ser humano al trabajar con metales fundidos y en un entorno hostil. Como beneficio secundario, el IRB7600 podrá cumplir las mismas tareas de forma precisa, rápida y completamente automática. Como primera instancia, una vez elegido el problema a solucionar y el robot que cumplirá esta tarea, demoramos varios días en poder encontrar la matriz de Denavit-Hartenberg, paso inicial para modelar el manipulador. Siempre al comenzar un análisis cinemático para un robot de esta magnitud, y nosotros recién adentrándonos en el mundo de la robótica, varios conceptos se vuelven confusos y es difícil iniciar dicho análisis. Una vez encontrada la matriz DH el estudio cinemático directo e inverso del robot se pudo resolver con la ayuda del software MATLAB y del toolbox “RVCtools” creado por Peter Corke el cual está específicamente diseñado para estudiar y/o los aspectos cinemáticos y dinámicos de una gran cantidad de manipuladores. El análisis del Jacobiano y de la relación de velocidades fue un tema interesante de tratar. Es importante

que los movimientos del robot sean fluidos y sin saltos importantes de velocidad que puedan, o forzar los motores, o bien en este caso, derramar metal fundido mientras el robot maniobra los crisoles. Finalmente, el estudio de la trayectoria no resultó ser complicado con la ayuda del software, aunque si es repetitivo y tedioso de confeccionar. Una vez dimensionado nuestro espacio de trabajo pudimos, luego de unas semanas, poder trazar todas las trayectorias que el efector final recorre, teniendo en cuenta que se deben evitar colisiones y que los posicionamientos deben coincidir con el entorno del robot. En el desarrollo de este proyecto se dejó a un lado el análisis dinámico del robot (ya que excede el alcance de la materia). Dicho análisis podría considerarse una mejora completamente necesaria si se quisiera materializar este robot. Como últimas palabras diremos que, en aspectos generales, no nos resultó difícil estudiar la cinemática de un robot. Sea más chico o más grande los pasos a seguir son los mismos siempre y cuando se utilice DH como convención (que fue en la que nos basamos).

Referencias.

- Fundamentos de Robótica - Antonio Barrientos
- Robótica (3^{era} Edición) - John J. Craig
- Robots y Sistemas sensoriales - Fernando Torres
- Video de aplicación URL:
<https://www.youtube.com/watch?v=78S0UdsZ994>
- Product specification IRB7600 - ABB
- Material de estudio por parte de la Cátedra (Trabajos Prácticos y apuntes) URL:
<http://ingenieria.uncuyo.edu.ar/estudios/materia/182>
- Documentación del lenguaje URL:
<https://la.mathworks.com/>
- Documentación del toolbox de Peter Corke URL:
<http://petercorke.com/wordpress/toolboxes/robotics-toolboxDocumentation>
- Para el relevamiento de las herramientas se utilizó el software de Siemens: Solid Edge Student Version 9. SolidWorks
- Modelos “.stl” fueron extraídos de la página oficial de ABB para el IRB7600-400-255
- Redacción del informe: Plataforma online Overleaf basada en LaTeX.