Proyecto Final Integrador Rob´otica I

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

15 de noviembre de 2018

## ´Indice

 Introducci´on

 Descripci´on y aspectos t´ecnicos del robot

 Relevamiento/Dimensionamiento del espacio de trabajo

 Cinem´atica Directa  Cinem´atica Inversa  Matriz Jacobiana

 Matriz Jacobiana Inversa  Singularidades

 Elipsoide de Manipulabilidad  Espacio de trabajo

 Planificaci´on de trayectorias

 Conclusi´on, mejoras y desaf´ıos  Referencias

**ABB** Robot Model IRB 7600-400/2.55

## Introducci´on.

La idea principal de este trabajo es la de ha- cer un estudio de los aspectos cinem´aticos del ro- bot que hemos seleccionado que, en nuestro caso es un robot comercial de 6 grados de libertad mode- lo *IRB 7600* de la corporaci´on multinacional ABB. La aplicaci´on por la cual nos vimos inspirados en la selecci´on del mismo, fue la de asistencia autom´atica en el proceso de colada de acero fundido. El robot deber´a realizar tanto la introducci´on de los crisoles al horno, como la extracci´on de los mismos y su posterior colada en los moldes. Tambi´en se deber´a tener en cuenta el cambio de herramienta para las diferentes operaciones.



Figura 1: ABB IRB 7600

## Descripci´on y aspectos t´ecnicos del robot.

1. *Modelos:*

El IRB 7600 est´a disponible en 5 modelos di- ferentes. La discriminaci´on entre ellos se debe principalmente a la capacidad de carga y al alcance que provee:

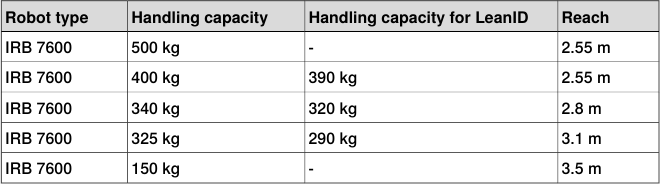


Figura 2: Standard robot versions

Para la aplicaci´on que hemos seleccionado, en- contramos que la mejor opci´on es utilizar el

1

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

1. *Dimensiones:*

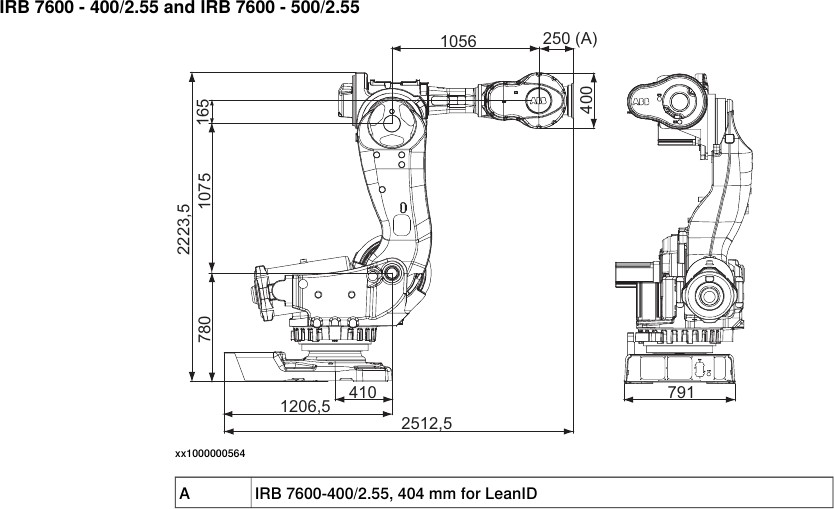


Figura 3: Dimensiones

1. *Velocidades:*

La siguiente tabla muestra la *velocidad m´axi- ma* soportada por cada articulaci´on medida en **[***◦***/s]**.A su vez las especifica para cada modelo. Es importante remarcar que la resoluci´on de cada articulaci´on esta comprendida entre los 0.001*◦* a 0.005*◦*.

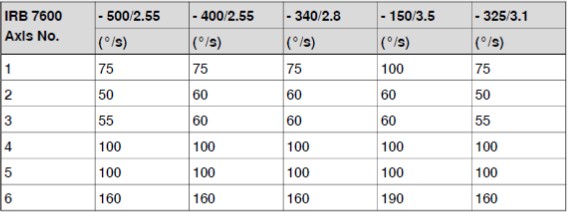


Figura 4: Velocidades

# Descripci´on y relevamiento de las he- rramientas

A continuaci´on agregamos un relevamiento de las dos herramientas que utilizamos en nuestra aplica- ci´on. 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 disen˜ada para un tipo especifico de crisol que, mediante una configuraci´on de disen˜o permite colar el acero fundido de su interior en un molde, sin volcarlo.

# Cinem´atica Directa

B´asicamente la finalidad de encontrar la soluci´on al problema de cinem´atica directa de un robot con- siste en establecer la posici´on y orientaci´on del ex- tremo del robot en funci´on de los valores de los par´ametros de las articulaciones.

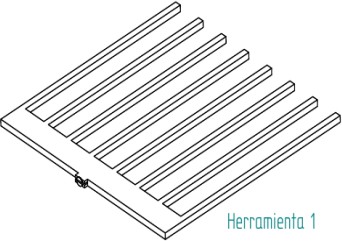


Figura 5: Herramienta 1

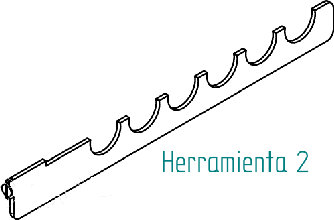
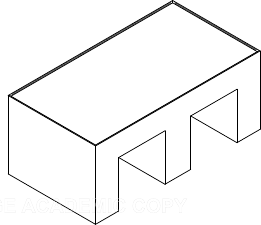


Figura 6: Herramienta 2



Figura 7: Crisol

Esto se logra gracias a la *asignaci´on de sistemas de referencia* en cada una de las articulaciones y a la *lectura de los sensores internos* del robot. Aho- ra s´ı, mediante la composici´on de *transformaciones* b´asicas tanto de traslaci´on como de rotaci´on, ser´a posible determinar la posici´on y orientaci´on del ex- tremo con respecto a un sistema de referencia fijo del mundo, situado por ejemplo, en la base del ro- bot.

*αi*: A´ngulo que existir´ıa entre los ejes *i e i+1* de las articulaciones si estos se cortasen en los puntos de corte de la l´ınea normal comu´n. Este par´ametro en cierto modo mide *”la forma del eslab´on*. Tambi´en se lo conoce como *”A´ngulo de torsi´on del eslab´on”*

*di*: Distancia entre las intersecciones de las nor- males comunes al eje de la articulaci´on *i*, medi- da a lo largo de dicho eje. Esta medida en cierto modo expresa la distancia entre los dos eslabo- nes, marcada por el taman˜o y la forma de la articulaci´on, por lo que se denomina *”Longitud articular”*.

Figura 8: Mesa

*θi*: A´ngulo que existir´ıa entre las l´ıneas nor-

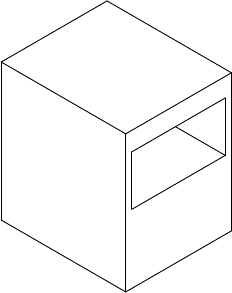


Figura 9: Horno

# Par´ametros Denavit - Hartenberg

Teniendo en cuenta que nuestro robot est´a com- puesto por una concatenaci´on de articulaciones de un grado de libertad y, que dos articulaciones con- secutivas est´an conectadas por un eslab´on (que no es m´as que un objeto r´ıgido), se pueden estable- cer **cuatro par´ametros**, *dos relativos al taman˜o y forma del eslab´on* y otros *dos relacionados con la posici´on relativa entre dos eslabones consecuti- vos*, par´ametros relativos a la articulaci´on que los enlaza.

A continuaci´on se agrega una breve definici´on de los par´ametros:

 *ai*:Distancia entre los ejes *i e i+1* de las articu- laciones a los largo de la normal comu´n. Este par´ametro define en cierto modo el taman˜o del eslab´on, por lo que se le conoce como *”Longi- tud del eslab´on*”.

males comunes al eje de la articulaci´on *i* si se

cortasen en el mismo punto del eje de la ar- ticulaci´on. De alguna forma expresa el ´angulo que forman los dos eslabones, marcado nueva- mente por la forma de la articulaci´on, por lo que se denomina *”´angulo articular”*.

Para la determinaci´on de los par´ametros se debe proceder siguiendo el paso a paso de la **Conven- ci´on de Denavit - Hartenberg** (Ver Referencia)

, la cual propone tanto la asignaci´on de ejes como la de sistemas de referencia.

# Asignaci´on de EJES

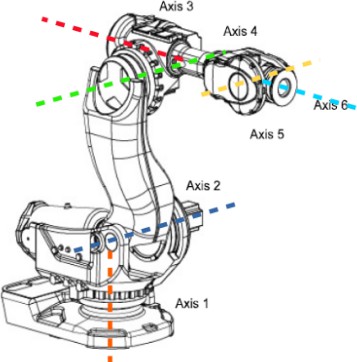


Figura 10: Asignaci´on de ejes

# Asignaci´on de SISTEMAS DE REFERENCIA

Finalmente, una vez realizado los pasos previos, se lleg´o a las siguiente matriz de Denavit - Harten- berg **(DH)**:

Habiendo determinado los par´ametros DH, y te- niendo en cuenta las cuatro transformaciones indi-

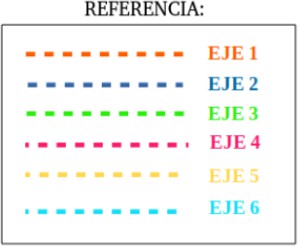


Figura 11: Referencias

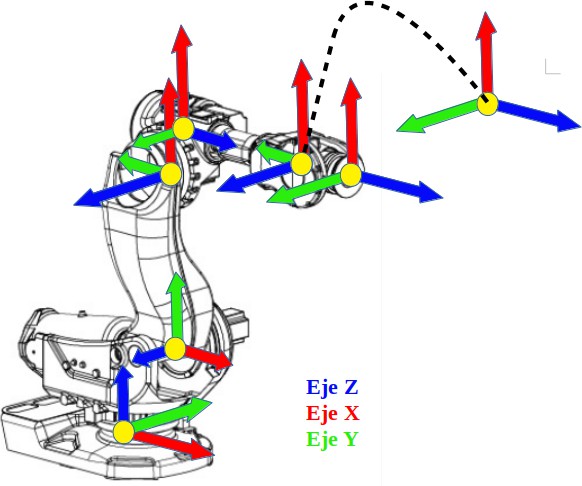


Figura 12: Asignaci´on de sistemas de referencia

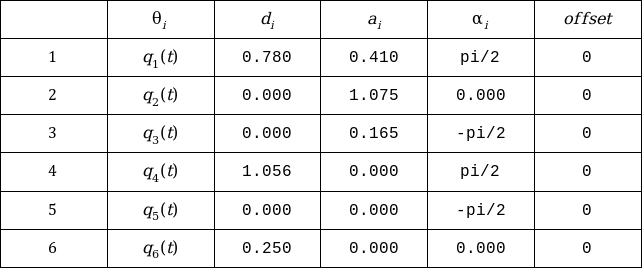


Figura 13: Matriz DH

cadas en la Ec.(1), se obtiene la matriz de transfor- maci´on homog´enea para pasar del sistema *i* − 1 al *i* :

*T i−*1 = [*Rot*(*zi−*1*, θi*)*.T ras*(*zi−*1*, di*)]*.*

*i*

q0 = [0*,*330*,* 2*,*476*,* −1*,*189*,* 2*,*127*,* 0*,*563*,* −2*,*138];

(2)

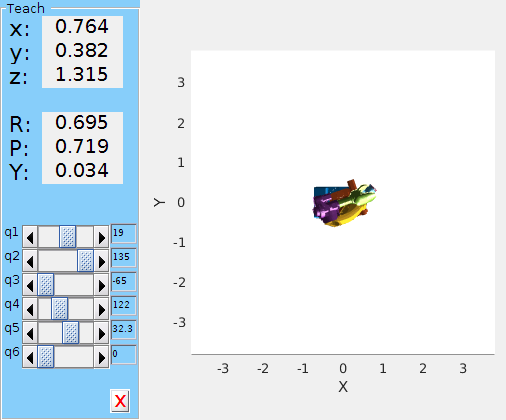


Figura 14: Vista superior

**NOTA:** Las transformaciones homog´eneas tota- les correspondientes a ese arreglo de coordenadas articulares son (Ver Fig. 15): La de arriba corres- ponde al c´alculo SIN considerar la herramienta (en vac´ıo), y la de abajo es considerando la misma. E´sta tiene un largo equivalente de 1300[*mm*]*.*

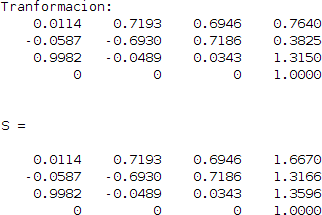
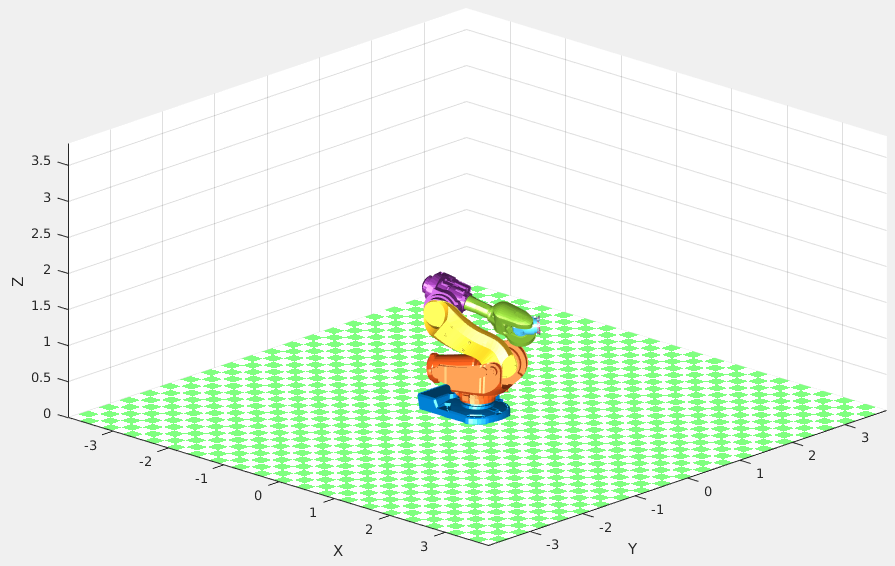


Figura 15: T.H. Directa 0-6

Tras(x*i, ai*)*.Rot*(*xi, αi*)(1)

## Implementaci´on en MALTAB

Escogimos un vector de coordenadas articulares *q*0 para el cual el robot se encontrar´ıa en su posici´on de reposo.

Figura 16: Vista de Reposo

# Cinem´atica Inversa

En la cinem´atica directa resolvimos el problema de determinar la posici´on y orientaci´on del extre- mo del robot segu´n unos valores conocidos de las variables articulares. Sin embargo, desde el punto de vista pr´actico, resulta mas interesante el poder determinar los valores que tienen que tomar las va- riables articulares para que el extremo del robot se encuentre en una posici´on y orientaci´on dada. Esto an´alisis se conoce como: **Problema cinem´atico inverso**. En donde tenemos como variables de en- trada: La posici´on(*x, y, z*), y la orientaci´on (*α, β, γ*). Y como variables de salida el valores de las varia- bles articulares. *qk* = *q*1*, q*2*, ..., qk*

*qk* = *fk*(*x, y, z, α, β, γ*)

*k* = 1*,* 2*,* 3*, ..., n* ∴ *n* : *gdl*

Para robots complejos (m´as de 3 grados de libertad) como es nuestro caso, para resolver este problema, nos basamos en la **Soluci´on de Pieper** la cual nos propone dividir el problema en dos partes (tambi´en conocido como desacoplamiento cinem´atico), esto se debe a que una vez posicio- nada la mun˜eca en el espacio en la que los tres ejes se cortan, el movimiento de las misma en torno a sus ejes no altera la posici´on espacial del punto de corte. Dicho esto, en primer lugar se debe analizar la posici´on (con las tres primeras articulaciones), y luego la orientaci´on (con las u´ltimas tres articulaciones).

N´otese en la (Fig.17) que el punto de corte ”*mun˜eca*”, se define en el punto de intersecci´on

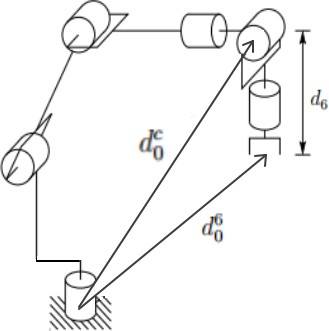
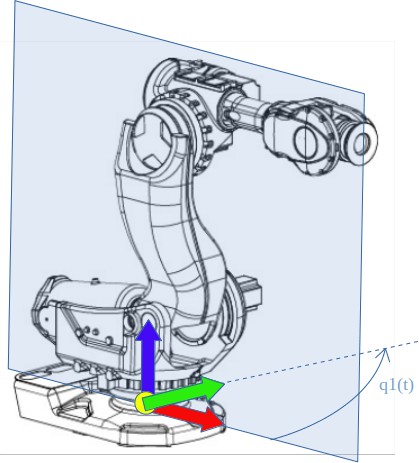


Figura 17: Punto de corte

de los ejes de las u´ltimas tres articulaciones. Pa-

ra poder resolver la primera parte del problema cinem´atico inverso se debe referenciar las coorde- nadas cartesianas del extremo del robot al punto corte, y esto se logra mediante la siguiente expre- si´on (en correspondencia con la (Fig.17)):

*dc* = *d*6 − *d*6*.z*ˆ6 (3)

Figura 18: Robot desacoplado

0 0

## Parte I - Posicionamiento

Una caracter´ıstica de la cinem´atica inversa es que

referencia de la base *S*0 En nuestra implementaci´on

las denotamos como (*Pxm, Pym, Pzm*)

la soluci´on (en la mayor´ıa de los casos y para *k >* 1) no es u´nica. Para esta primera parte, la soluci´on

0 0

*xm x*6

*P*

= *P*

0 0

P

= *P*

*ym y*6

− *d*6*.r*13

− *d*6*.r*23

que propusimos fue basada en el m´etodo geom´etri-

co, haciendo uso de relaciones trigonom´etricas y de

0 0

*zm z*6

P

= *P*

− *d*6*.r*33

la propia geometr´ıa del robot. En ellas intervienen las coordenadas del extremo del robot y sus res- pectivas orientaciones, las coordenadas articulares y las dimensiones f´ısicas de cada eslab´on.

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

Es importante mencionar que, las coordenadas (*Px, Py, Pz*) que all´ı aparecen son las que corres- ponden al punto de corte, respecto del sistema de

A la hora de implementarlo en MATLAB, la

matriz de transformaci´on *T* 0 que utilizamos como dato entrada del problema, la obtuvimos previa- mente de aplicar cinem´atica directa. Expres´andola en forma general:

6

 *r*11 *r*12 *r*13 *Px*6 

0 *r*21 *r*22 *r*23 *Py*6

*T* = 







6 *r*31 *r*32 *r*33 *Pz*6

0 0 0 1

Ahora si, una vez referenciado el punto de cor- te, podemos expresar las coordenadas articulares

*zm*

En donde *cos*(*π* − *q*3) = −*cos*(*q*3), entonces:

en funci´on de las coordenadas cartesianas:

*r*2 + (*P* 1

)2 = *l*2 + *l*3 + 2*.l*2*.l*3*.cos*(*q*3) (6)

*q* = *arctg*( *Pym*

1 *P*

) (4)

Despejamos ahora el *cos*(*q*3):

*xm r*2+(*P* 1

2*.l*2*.l*3

)2*−l*2*−l*3

Esta u´ltima presenta 2 soluciones posibles:

*q*1 = *θ*1 ∴ *q*1 = *θ*1 + *π*

Habiendo definido un valor de *q*1, podemos ob- servar que las siguientes dos articulaciones quedan contenidas en un mismo plano que, para hallar las relaciones trigonom´etricas que las vincule con las coordenadas cartesianas, trasladamos el sistema de referencia a la segunda articulaci´on. Esto se logra,

*cos*(*q*3) = *zm*

Haciendo uso de la siguiente identidad:

*sen*2(*x*) + *cos*2(*x*) = 1

Expresamos la soluci´on en funci´on de la tangente como:

±✓1 − *cos*(*q*3)2

calculando la matriz de transformaci´on *T* 0

1

y refe-

*q*3 = *arctg*(

*cos*(*q*3)

) (7)

renciando el punto de corte a este nuevo sistema

*S*1:

¯1 = (*T* 0)*−*1*.P*¯*m* (5)

*p*

1

*m*

En la siguiente figura se puede observar como se redujo la cadena cinem´atica a una simplificada situaci´on de un robot planar de dos grados de li- bertad.

Por otro lado vemos que la segunda articulaci´on

*q*2 est´a definida por las orientaciones mediante:

*q*2 = *β* − *α* (8)

El a´ngulo *β* corresponde a la orientaci´on del es- lab´on 2 respecto del eje ”*r*”:

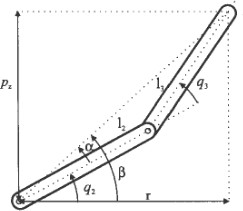
1 1

*P*

*P*

*zm zm*

*Xm*

*β* = *atg*(

*r* ) = *atg*( ±✓(*p*0

)2 + (*p*0

) (9)

)2

*Y m*

El ´angulo *α* corresponde a la orientaci´on del es- lab´on 2 respecto de la hipotenusa ”*d*1 ”:

*m*

*l*3*.sen*(*q*3)

*α* = *atg*( ) (10)

*l*2 + *l*3*.cos*(*q*3)

Habiendo definido estas dos orientaciones, la coordenada articular *q*2 queda definida en funci´on de las coordenadas cartesianas:

*P* 1 *l*3*.sen*(*q*3)

*zm*

*q*2 = *atg*( ±✓(*p*0

)2 + (*p*0

)−*atg*(

)2 *l*

)

+ *l .cos*(*q* )

*Xm Y m*

2 3 3

(11)

Figura 19: Simplificaci´on-Robot planar 2gdl

## Parte 2 - Orientaci´on

Finalmente queda por determinar los valores de las

tres u´ltimas tres articulaciones que orientar´an el

En donde *r* corresponde:

extremo del robot para que el problema cinem´ati- co inverso quede totalmente definido. Para la solu- ci´on de esta segunda parte, nos hemos basado en el

*r*2 = (*p*1

*xm*

)2 + (*p*1 )2

m´etodo matricial, haciendo uso particularmente de las submatrices de rotaci´on.

Y por otro lado, si calculamos la hipotenusa forma- da por *Pzm* y *r*:

*ym*

Si recordamos de cinem´atica directa :

*Rot*0

= *Rot*0*.Rot*3

(12)

(*d*1 )2 = *r*2 + (*P* 1 )2

*extremo*

3 *extremo*

*m zm*

En donde de esta

u´ltima expresi´on la submatriz

Aplicando el teorema del coseno al triangulo for-

0

*extremo*

*Rot*

es dato de entrada, y habiendo resuelto

mado por los dos eslabones y la hipotenusa del

triangulo rect´angulo, obtenemos:

el problema cinem´atico inverso para las primeras

tres articulaciones resulta que *Rot*0 tambi´en es co-

3

*r*2 + (*P* 1

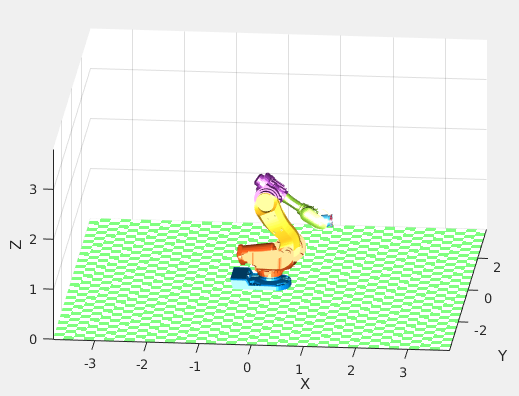
*zm*

)2 = *l*2 + *l*3 − 2 ∗ *cos*(*π* − *q*3)

nocida.

Multiplicando a ambos miembro por (*Rot*0)*−*1:

3

(*Rot*0)*−*1*.Rot*0 = (*Rot*0)*−*1*.Rot*0*.Rot*3

3 *extremo* 3 3 *extremo*

Por u´ltimo, expresamos de forma expl´ıcita la

inc´ognita:

*Rot*3

= (*Rot*0)*−*1*.Rot*0

(13)

*extremo* 3 *extremo*

Las matrices de transformaci´on para las articu- laciones de la mun˜eca son :



*T* 3 = 



 (14)

|  |  |  |  |
| --- | --- | --- | --- |
| *sen*(*q*4) | 0 | *cos*(*q*4) | 0 |
| −*cos*(*q*4) | 0 | *sen*(*q*4) | 0 |
| 0 | −1 | 0 | *l*3 |
| 0 | 0 | 0 | 1 |

4  

 

|  |  |  |  |
| --- | --- | --- | --- |
| *cos*(*q*5) | 0 | *sen*(*q*5) | 0 |
| *sen*(*q*5) | 0 | −*cos*(*q*5) | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 |

Figura 20: Cinem´atica inversa

*T* 4 =   (15)

5





*q*

= [−0*,*077*,* 1*,*986*,* −0*,*990*,* 2*,*975*,* 0*,*602*,* −3*,*053]



*inv*

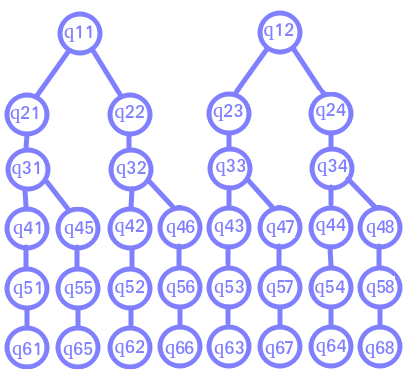


|  |  |  |  |
| --- | --- | --- | --- |
| *cos*(*q*6) | −*sen*(*q*6) | 0 | 0 |
| *sen*(*q*6) | *cos*(*q*6) | 0 | 0 |
| 0 | 0 | 1 | *l*4 |
| 0 | 0 | 0 | 1 |

*T* 5 = 



Se han evaluado las mu´ltiples soluciones y en ba- se a ello pudimos confeccionar un a´rbol con los va- lores de las coordenadas articulares.Fig(21)

6  

(16)

Para determinar los valores de *q*4*, q*5*, q*6 se utiliza la siguiente igualdad:

*Rot*3

= (*Rot*0)*T .*[*x*0

*, y*0

*, z*0 ] (17)

*ext*

3 *ext*

*ext*

*ext*

Resolviendo las ecuaciones, obtenemos:

0

*z*

*q*4 = *arcos*(− *extremo* ) (18)

*sen*(*q*5)

*q*5 = *arcos*(*x*0

*extremo*

*.*(−*sen*(*q*3)*.cos*(*q*3)+*cos*(*q*2)*.sen*(*q*3)+*...*

0

+y

*extremo*

*.*(*cos*(*q*2)*.cos*(*q*3)−*sen*(*q*2)*.sen*(*q*3)))(19)

Figura 21: A´rbol de soluciones

Por simplicidad, se redujo la expresi´on de *q*6:

A continuaci´on presentamos los valores obtenidos para la soluci´on propuesta:

0

*−x*

*a* = *extremo*

*−y*0

*.*(*−sen*(*q*3)*.cos*(*q*3)+*cos*(*q*2)*.sen*(*q*3)) *sen*(*q*5)

*.*(*cos*(*q*2)*.cos*(*q*3)*−sen*(*q*2)*.sen*(*q*3))

*q*1 = [−0*,*0779*,* 3*,*0637]

*q*2 = [1*,*9685*,* −0*,*3873*,* 1*,*9685*,* −0*,*3873]

*b* =

*q*6 resulta:

*extremo*

*sen*(*q*5)

*q*3 = [−2*,*4017*,* 2*,*4017*,* −2*,*4017*,* 2*,*4017]

*q*4 = [3*,*1416*,* 3*,*1416*,* 3*,*1416*,* 3*,*1416]

*q*6 = *arcos*(*a* + *b*) (20)

## Implementaci´on en MALTAB

Para la implementaci´on del problema cinem´ati- co inverso, partimos de una transformaci´on ho- mog´enea calculada por cinem´atica directa, como dato de entrada. Para la misma escogimos un vector de coordenadas articulares (que es el que corres- ponde al posicionamiento del robot para ingresar al horno), para obtener la matriz de transforma- ci´on *T*. A continuaci´on de esquematiza el robot en dicha posici´on. Fig.(20)

*q*5 = [0*,*5972*,* 1*,*2030*,* 0*,*5972*,* 1*,*2030]

*q*6 = [3*,*1416*,* 1*,*9664*,* 3*,*1416*,* 1*,*9664]

Otra alternativa para el c´alculo de la cinem´ati- ca inversa es haciendo uso de la funci´on *objetoRo- bot.ikine* que dispone el *toolbox de Perter Corke*. La misma busca la soluci´on mediante un m´etodo num´erico iterativo.

# Matriz Jacobiana

En muchas aplicaciones es de gran utilidad dis- poner de la relaci´on entre las velocidades de las

coordenadas articulares y las posiciones y orienta- ciones del extremo del robot. La relaci´on entre am- bos vectores de velocidad se obtiene a trav´es de la denominada **matriz Jacobiana**. Fig (22).

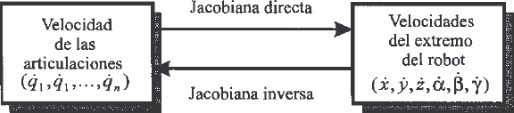


Figura 22: Relaci´on espacio Cartesiano- Articular

Expresando la dependencia de las variable car- tesianas en funci´on de las coordenadas articulares:

*x* = *fx*(*q*1*, ..., qn*)

*y* = *fy*(*q*1*, ..., qn*)

*z* = *fz*(*q*1*, ..., qn*)

*α* = *fα*(*q*1*, ..., qn*)

*β* = *fβ*(*q*1*, ..., qn*)

*γ* = *fγ*(*q*1*, ..., qn*)

Si derivamos estas expresiones respecto al tiempo obtenemos:

汇 *∂fx*

 El nu´mero de columnas es igual al nu´mero de articulaciones del manipulador, que al igual es 6.

## Implementaci´on en MATLAB

Para la obtenci´on de la matriz Jacobiana num´eri- ca hicimos uso del m´etodo objetoRobot.jacob0() el cual pide que se le pase como par´ametro un arreglo de coordenadas articulares.

La matriz Jacobiana correspondiente al arreglo

*q*0 es:

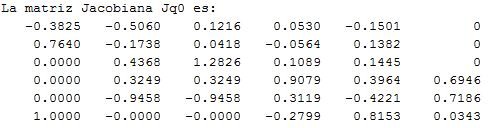


Figura 23: Matriz Jacobiana directa - Num´erica

# Matriz Jacobiana Inversa

Con la matriz Jacobiana inversa buscamos de- terminar las velocidades articulares para que el ex- tremo del robot se mueva a una velocidad, lineal y

*x*˙ =

*∂qi*

汇 *∂f*

*y .q*˙

汇

*.q*˙*i*

angular, determinada. Para obtenerla se proponen los siguiente m´etodos:

*∂qi*

*z*˙ = *∂fz .q*˙

*y*˙ =

*i*

*i*

*∂qi*

汇

*α*˙ = *∂fα .q*˙

*i*

*∂qi*

汇

*β*˙ = *∂fβ .q*˙

*i*

*∂qi*

汇

*γ*˙ = *∂fγ .q*˙

*i*

*∂qi*

Por u´ltimo lo expresamos en forma matricial:

**Primer alternativa:** Supuesta conocida la

relaci´on directa se podr´ıa obtener la matriz inver- sa, invirtiendo simb´olicamente la matriz. Es de dif´ıcil realizaci´on en la pr´actica ya que supone la inversi´on simb´olica de una matriz de 6x6, cuyos elementos son funciones trigonom´etricas resulta complejo. Como **segunda alternativa** puede plantearse la evaluaci´on num´erica de la matriz J para una configuraci´on *qi* dada e invirtiendo num´ericamente ´esta. Sin embargo, se debe tener en cuenta que el valor num´erico de la Jacobiana

*x*˙ *y*˙





 *z*˙ 

 *q*˙1 

va cambiando constantemente (depende de t). La **tercera alternativa** es repetir el procedimiento seguido para la obtenci´on de la Jacobiana directa,

  = *J.*  . 

 

(21)

*α*˙ .





*β*˙ *q*˙*n*





*γ*˙

Siendo **J** la matriz Jacobiana:

pero ahora partiendo de un modelo cinem´atico inverso.

Cabe destacar que existen puntos en donde la matriz Jacobiana no sea invertible por ser su deter-



*J* = 

Observaciones:

*∂fx*

*∂q*1

.

*∂fγ*

*∂q*1

· · · *∂fx*

*n*

*∂q*





. . . .

*. . . ∂fγ*

*∂q*

*n*

minante, denominado Jacobiano, nulo. Estas confi- guraciones en donde el Jacobiano se anula son lla- madas configuraciones singulares.

## Implementaci´on en MATLAB

La matriz Jacobiana inversa correspondiente al

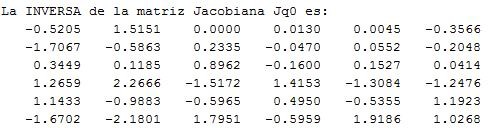
La matriz *J(q)* se denomina matriz Jacobiana y al ser funci´on del tiempo, para cada instante de tiempo se tiene un valor distinto de la ella.

El nu´mero de filas de la matriz igual al nu´mero de grados de libertad, en este caso .

arreglo *q*0 es Fig.(24) :

# Singularidades.

Hacemos menci´on de algunas caracter´ısticas de una configuraci´on singular:

Figura 24: Matriz Jacobiana inversa - Num´erica  A velocidades finitas del extremo pueden co-

rresponderles velocidades infinitas de las arti- culaciones.

 A fuerzas y pares finitos del extremo puede corresponderles fuerzas y pares infinitas del as articulaciones.

 Se pierde al menos un grado de libertad (ali- neaci´on de ejes articulares).

 El movimiento en ciertas direcciones es impo- sible(fuera del espacio de trabajo).

 El rango del Jacobiano es menor al orden de la matriz Jacobiana.

En los puntos del l´ımite del espacio de trabajo del robot existen singularidades, porque all´ı no es posible los movimientos en direcci´on axial del ro- bot (por ejemplo que se encontrase totalmente es- tirado), lo que supone la p´erdida de un grado de libertad en esa direcci´on del espacio, por alineaci´on de 2 o mas de sus ejes articulares. En nuestro robot pasar´ıa 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 posici´on y orientaci´on del efector final de un sistema rob´otico en una confi- guraci´on dada. Esto quiere decir que, el efector final tiene mayor capacidad de movimiento en la direc- ci´on del eje mayor del elipsoide, por otra parte, en la direcci´on del eje menor la capacidad de desarrollar velocidad ser´a menor.La elipsoide de manipulabi- lidad es una herramienta muy u´til para ver gr´afi- camente las posibilidades de velocidades a desarro- llar en un punto de trabajo definido. Lo interesante de esta elipsoide es que tambi´en podemos obser- var gr´aficamente 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.

## Implementaci´on en MATLAB

Haciendo uso del m´etodo *vellipse()* de la clase Se- rialLink del toolbox de Peter Corke se lograron ob- tener las elipsoides de manipulabiliadad para algu- nos puntos de inter´es.

En la Fig.(25)se muestra el elipsoide de manipu- labilidad para la posici´on de reposo. Se observa que

el efector final dispone de mayor capacidad de mo- vimiento en la direcci´on del eje mayor del elipsoide. Por otra parte, en la direcci´on del eje menor la ca- pacidad de desarrollar velocidad ser´a menor. Este elipsoide simboliza la capacidad de manipulaci´on.

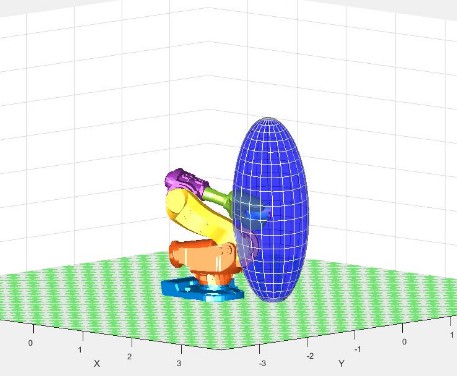


Figura 25: Elipsoide-Punto espacio de trabajo Ahora tomamos un punto singular, por ejemplo

la configuraci´on:

*qsingular*1 = [90*,* 1*,* 90*,* 173*,* 0*,* 0]

Realizando el c´alculo de la matriz Jacobiana para este punto, vemos que 2 columnas (las correspon- diente a la articulaciones *q*4 y *q*6), son coincidentes (linealmente dependientes). Ver Fig.(26). Al pre- sentar esta dependencia lineal, el rango de la ma- triz resulta ser de 5 y por la tanto su determinante resulta ser nulo.

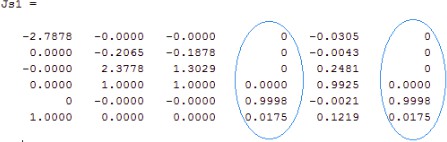


Figura 26: Matriz Jacobiana Singular

Como mencionamos mas arriba en las singulari- dades 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 podr´ıa situar el extremo de robot. Extra´ıdo del datasheet del fabricante.

## Implementaci´on en MATLAB

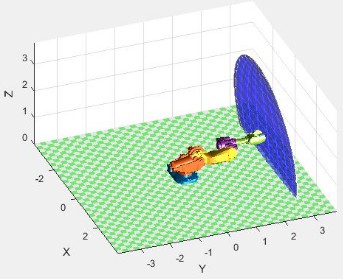


Figura 27: Configuraci´on singular

Para su representaci´on en MATLAB, usamos una funci´on del lenguaje llamada *scatter()* la cual plo- tea una dispersion de puntos en el espacio. Asig- nando adecuadamente la coordenada articular que se desee representar se obtuvieron los siguientes re- sultados:

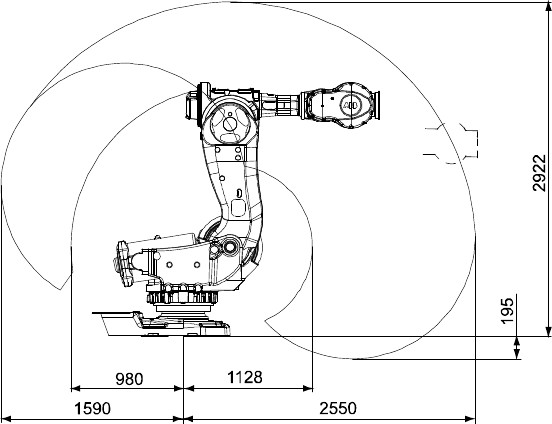


Figura 28: Espacio de trabajo

Si observamos al Fig.(29) podemos notar, que la altura m´axima 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 ho- rizontal y mostramos el alcance m´aximo en esa po- sici´on.

# Planificaci´on de trayectorias

Para realizar la planificaci´on de trayectoria se de- be 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 vendr´an dados por una 6-upla, t´ıpicamen- te *x, y, z, α, β, γ* . Utilizando la cinem´atica inversa, convertimos cada uno de estos puntos en coorde- nadas articulares (*q*1*, q*2*, q*3*, q*4*, q*5*, q*6), asegurando la continuidad en cada uno de estos puntos y asegu- rarse as´ı la trayectoria. Es necesario tambi´en contar

Figura 29: Espacio de trabajo-MATLAB

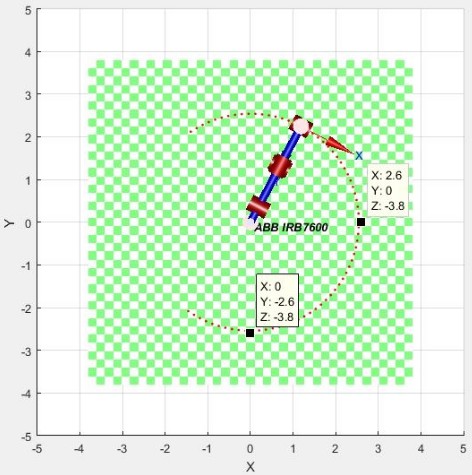
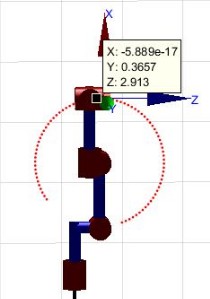


Figura 30: Espacio de trabajo-MATLAB

con m´etodos de interpolaci´on para generar una ex- presi´on *qi*(*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 m´as pr´oxima a la especificada por el programa del usua- rio (en cuanto a precisi´on, velocidad, etc.). La tra- yectoria articular debe ser muestreada para generar referencias al control din´amico. Para el prop´osito de este proyecto fue necesario localizar los puntos caracter´ısticos de cada tarea segu´n su trayectoria de trabajo. Comenzando y finalizando cada tarea en el punto *q*0 Ec.(2) y analizando cada una de

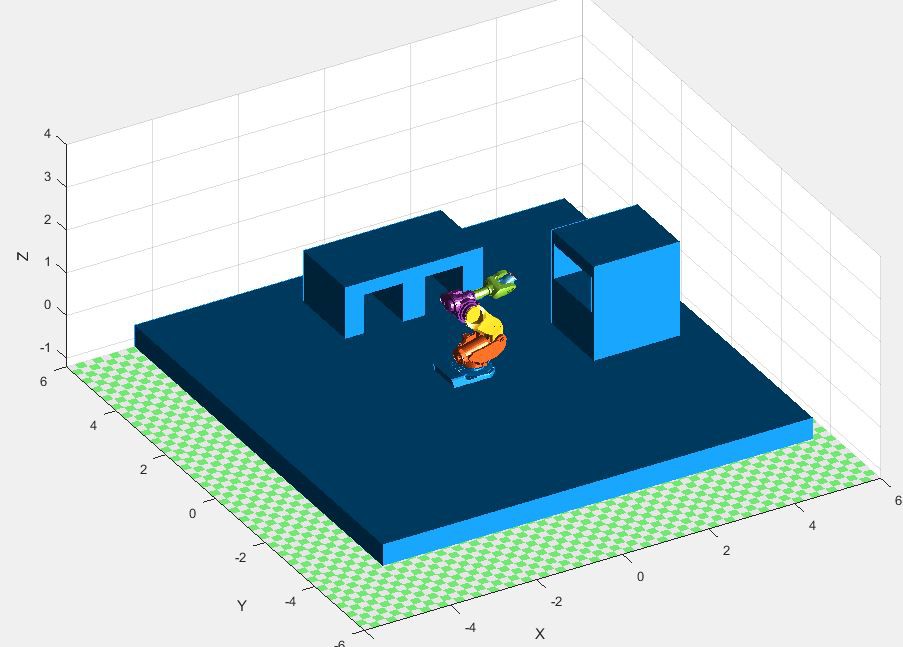
movimientos con los crisoles, ya que estos en su in- terior tienen acero fundido. A partir del m´etodo de cinem´atica inversa *“.ikine()”* se lograron obtener con ´exito y gran precisi´on todos los puntos para la generaci´on de trayectorias. Cada trayectoria empie- za en *q*0 y termina en *q*0 por lo que podemos hacer que cada una suceda a la otra en el orden adecuado segu´n la aplicaci´on. Los puntos m´as significativos son:

Figura 31: Espacio de trabajo-MATLAB

las orientaciones y puntos fundamentales para la realizaci´on del trabajo de colada, pudimos generar 12 tareas con su trayectoria espec´ıfica, en las que mediante la selecci´on de cualquiera de las 2 herra- mientas disponibles el robot estar´a en condiciones de:

 Tomar la herramienta 1 (Ver Fig. 5), posicio- narse en la mesa para retirar las filas de cri- soles, retirarlas e ingresarlas al horno para su fundici´on. Una vez fundidas, este ser´a capaz de retirarlas del horno con el metal en esta- do l´ıquido y las posicionar´a nuevamente en la mesa de donde las retir´o, apoy´andolas suave- mente en la mesa y retirarse a la posici´on *q*0 Ec.(2) para luego dejar la herramienta 1 en su posici´on espec´ıfica.

 Tomar la herramienta 2 (Ver Fig. 6), posicio- narse en la entrada del horno, retirar las suce- sivas 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 ser´an siempre los mismos (normalizados), ya que las trayectorias del robot est´an configurados para ello. En el caso de querer fundir y que no haya que realizar colada (la fundici´on se solidifica en el crisol), podremos retirar las filas completas, pudiendo as´ı utilizar diversos crisoles (iguales) en esta aplicaci´on.

## Implementaci´on 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 mas impor- tante es que debe mantenerse el eje *Z* del efector

final paralelo al plano *x*0 − *y*0 cuando se realicen

 Punto inicial *q*0

 Entrada al horno, con un punto anterior para posicionarse antes de entrar al horno con la herramienta.

 Posici´on 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 situ´an las 7 filas para retirar con la herramien- ta 1 y en otra zona de la mesa el espacio para realizar la colada de una fila con el metal l´ıqui- do.

Como se mencion´o anteriormente, para obtener la configuraci´on articular segu´n la coordenada car- tesiana y la orientaci´on del efector final deseada se utiliz´o el m´etodo “.ikine()”, el cual pide una matriz de transformaci´on homog´enea como argumento. Es- tas matrices se obtuvieron a partir del an´alisis de la parte de rotaci´on de la matriz de transformaci´on homog´enea.

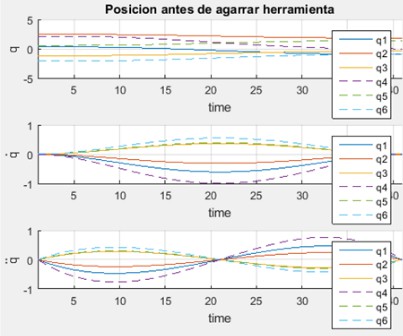
Las matrices de orientaci´on usadas fueron Fig.(32):

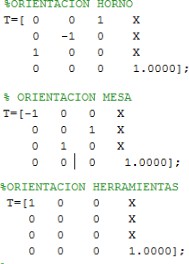
Las tareas que se pueden realizar son 11 en total.

Se pueden observar en la Fig.(33):

Donde en el lugar de las X se coloca la posici´on deseada. Se logra observar que la parte de rotaci´on de las matrices se ve modificada de una orienta- ci´on a la otra. Estas matrices T son las que se les insertan como argumentos al m´etodo ikine para ob- tener todos los arreglos de *q* = [*q*1*, q*2*, q*3*, q*4*, q*5*, q*6] para cada punto de inter´es, y con ellos generar las trayectorias.

Tambi´en se realiz´o el an´alisis de las variaciones de posici´on, velocidad y aceleraci´on 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 aceleraci´on. Algunos ejemplos son indicados a continuaci´on: Por ejemplo para la tarea “1. Agarrar herramienta 1” tenemos 4 movimientos. Ir de *q*0 a la posici´on de la herramienta 1 con un z mayor al z de la herramien- ta, bajar desde este z al z herramienta para realizar



Figura 32: Transformaciones

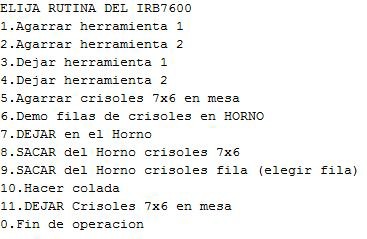


Figura 33: Rutinas de operaci´on.

el acople y retirar la herramienta para finalizar con el movimiento a *q*0 nuevamente. Los movimientos m´as significativos son:

En esta Fig.(34) tenemos los perfiles de posici´on, velocidad y aceleraci´on de cada una de las 6 articu- laciones. Podemos observar que la m´axima veloci- dad la obtuvo *q*4 al igual que la aceleraci´on, en este caso se mueven todas las articulaciones.

En cambio para la realizaci´on del movimiento en la Fig.(35) logramos ver que las articulaciones *q*3 y *q*5 sufren modificaciones en su posici´on, velocidad y aceleraci´on para agarrar la herramienta 1 se de- be mantener la posici´on 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 evi- tar la colisi´on del efector final con la herramienta

Figura 34: Perfil 1

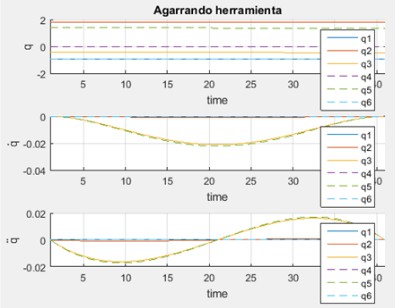


Figura 35: Perfil 2

y poder realizar el acople de forma suave y segura. Comparando las escalas de velocidades y acelera- ci´on de la Fig.(34) y la Fig.35 podemos ver que en el caso de la Fig (35) ´estas son 100 veces m´as 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 realizaci´on de la colada solamente la articulaci´on *q*6 es la que realiza movimiento, en este caso gira 120 para que el metal l´ıquido de los crisoles caiga sobre los moldes apoyados en la mesa.

# Sensores

Un sensor es un dispositivo el´ectrico y/o mec´ani- co que convierte magnitudes f´ısicas (como pueden ser luz, magnetismo,calor, etc), a valores medibles de dicha magnitud. Para nuestro robot el fabricante no provee informaci´on sobre sus sensores internos en el ”Product specification

. De igual manera intuimos que como m´ınimo debe tener un enconder por cada articulaci´on. Por otro lado nos pareci´o conviene proponer (llegado al caso de que no disponga del mismo), un sensor de

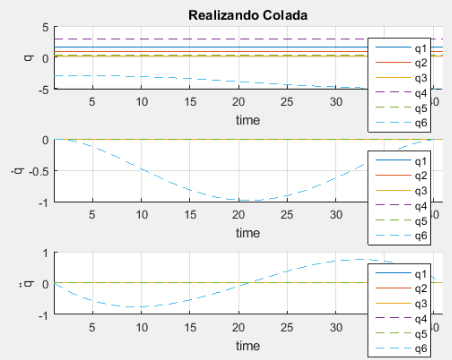


Figura 36: Perfil 3

que los movimientos del robot sean fluidos y sin saltos importantes de velocidad que puedan, o for- zar los motores, o bien en este caso, derramar me- tal fundido mientras el robot maniobra los crisoles. Finalmente, el estudio de la trayectoria no result´o 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 tra- yectorias que el efector final recorre, teniendo en cuenta que se deben evitar colisiones y que los po- sicionamientos deben coincidir con el entorno del robot. En el desarrollo de este proyecto se dej´o a un lado el an´alisis din´amico del robot (ya que exce- de el alcance de la materia). Dicho an´alisis podr´ıa considerarse una mejora completamente necesaria si se quisiera materializar este robot. Como u´lti-

temperatura, debido a que para nuestra aplicaci´on el robot opera el las proximidades del horno. Ase- gur´andonos de esta forma que no se excedan los limites requeridos de operaci´on. Ver Fig.(37)

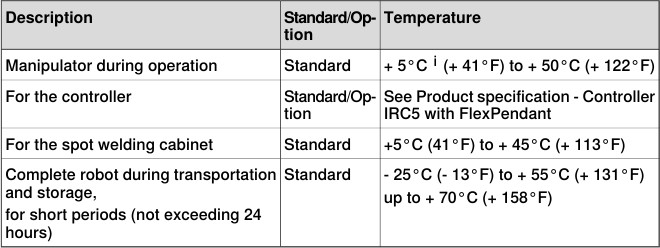


Figura 37: Requerimientos de operaci´on

# Conclusi´on, desaf´ıos y mejoras.

La implementaci´on de este robot en la industria trae consigo una evasi´on a cualquier riesgo que pue- de sufrir un ser humano al trabajar con metales fundidos y en un entorno hostil. Como beneficio secundario, el IRB7600 podr´a cumplir las mismas tareas de forma precisa, r´apida y completamente autom´atica. Como primera instancia, una vez elegi- do el problema a solucionar y el robot que cumplir´a esta tarea, demoramos varios d´ıas en poder encon- trar la matriz de Denavit-Hartenberg, paso inicial para modelar el manipulador. Siempre al comenzar un an´alisis cinem´atico para un robot de esta magni- tud, y nosotros reci´en adentr´andonos en el mundo de la rob´otica, varios conceptos se vuelven confusos y es dif´ıcil iniciar dicho an´alisis. Una vez encontrada la matriz DH el estudio cinem´atico directo e inverso del robot se pudo resolver con la ayuda del software MATLAB y del toolbox “RVCtools” creado por Pe- ter Corke el cual est´a espec´ıficamente disen˜ado para estudiar y/o los aspectos cinem´aticos y din´amicos de una gran cantidad de manipuladores. El an´ali- sis del Jacobiano y de la relaci´on de velocidades fue un tema interesante de tratar. Es importante

mas palabras diremos que, en aspectos generales, no nos result´o dif´ıcil estudiar la cinem´atica de un robot. Sea m´as chico o m´as grande los pasos a se- guir son los mismos siempre y cuando se utilice DH como convenci´on (que fue en la que nos basamos).

# Referencias.

 Fundamentos de Rob´otica - Antonio Barrien- tos

 Rob´otica (3*era* Edici´on) - John J. Craig

 Robots y Sistemas sensoriales - Fernando To- rres

 Video de aplicaci´on URL: h[ttps://www.youtube.com/watch?v=78S0UdsZ994](http://www.youtube.com/watch?v=78S0UdsZ994)

 Product specification IRB7600 - ABB

 Material de estudio por parte de la C´atedra (Trabajos Pr´acticos y apuntes) URL: <http://ingenieria.uncuyo.edu.ar/estudios/materia/182>

 Documentaci´on del lenguaje URL: https://la.mathworks.com/

 Documentaci´on del toolbox de Peter Corke URL:

<http://petercorke.com/wordpress/toolboxes/robotics-> toolboxDocumentation

 Para el relevamiento de las herramientas se uti- liz´o el software de Siemens: Solid Edge Student Version 9. SolidWorks

 Modelos ”.stl”fueron extra´ıdos de la p´agina ofi- cial de ABB para el IRB7600-400-255

 Redacci´on del informe: Plataforma online Overleaf basada en LaTeX.