



**UNCUYO**  
UNIVERSIDAD  
NACIONAL DE CUYO



**FACULTAD  
DE INGENIERÍA**

**UNIVERSIDAD NACIONAL DE CUYO**

**FACULTAD DE INGENIERÍA**

**ROBÓTICA I**

**ANÁLISIS CINEMÁTICO DE ROBOT SERIE DE 6 GRADOS DE  
LIBERTAD**

KUKA KR 120 R2100 nano F exclusive

**Integrantes:**

Pedro Matías Masi - 13205

Tomás Mauricio Suárez - 13139

Corazza, Luis Agustín - 12909

# Índice

<b>1. Resumen</b>	<b>2</b>
<b>2. Introducción</b>	<b>2</b>
<b>3. Especificaciones</b>	<b>4</b>
3.1. Ficha técnica del robot . . . . .	4
3.2. Tarea a realizar . . . . .	5
<b>4. Robot</b>	<b>6</b>
4.1. Convención Denavit & Hartenberg . . . . .	6
4.2. Cinemática Directa . . . . .	7
4.3. Cinemática Inversa . . . . .	9
4.4. Relación de Velocidades . . . . .	11
4.5. Planificación y Generación de Trayectorias . . . . .	13
4.5.1. Robot 1 . . . . .	14
4.5.2. Robot 2 . . . . .	16
4.5.3. Robot 3 . . . . .	18
<b>5. Sensores y actuadores</b>	<b>20</b>
5.1. Actuadores . . . . .	20
5.2. Sensores . . . . .	20
<b>6. Conclusiones</b>	<b>21</b>
<b>7. Referencias</b>	<b>22</b>

## 1. Resumen

Este trabajo consistió en proponer una aplicación industrial y ver su posible realización mediante el uso de robots industriales. El trabajo se enfocó principalmente en el análisis cinemático del robot, es decir que no han sido considerados efectos dinámicos en el estudio.

En este caso, se propuso implementar 3 robots industriales (KUKA KR 120 R2100 nano F exclusive) para utilizarlos en una cadena de montaje en serie de una fábrica de piezas de fundición.

A lo largo del presente documento se describe como fue el proceso de desarrollar la cinemática directa, inversa, el análisis de la relación de velocidades mediante el cálculo del Jacobiano y la detección de singularidades y, finalmente, la planificación y generación de trayectorias de los robots. Por otra parte, también se realizó un breve análisis de los sensores y actuadores con los que los robots deberían contar para llevar a cabo la tarea planteada.

## 2. Introducción

El proceso de forja es un proceso de conformado de metales mediante el uso de martilleo, prensado o laminado. El prensado en caliente produce una deformación plástica a una temperatura y velocidad de deformación que permite la recristalización del material simultáneamente con la deformación, evitando así el endurecimiento por acritud. Para que esto ocurra la temperatura de recristalización del metal debe mantenerse durante todo el proceso.

Debido a las condiciones de temperatura que se somete al metal y debido también al peligro que las herramientas pueden representar para los operarios, resulta conveniente en muchos casos la utilización de robots.

Así se eligió para esta aplicación el modelo KUKA KR 120 R2100 nano F exclusive, que se muestra a continuación:



Figura 1: *Kuka KR 120 R2100 nano F exclusive*

El trabajo a realizar se corresponde con la utilización de tres robots trabajando en modo cooperativo, donde la totalidad de ellos estarán equipados con un efector final del tipo pinza ON/OFF, realizando movimientos en el área de trabajo, con el fin de poder manipular una pieza a ser forjada entre 3 estaciones de forja distintas.

El efector final a utilizar en ambos robots, como ya se mencionó, es un gripper similar al mostrado a continuación:



Figura 2: *Tipo de gripper utilizado en la aplicación*

Durante las simulaciones se hizo uso de un modelo 3d del robot, provisto por Arte Toolbox, un Toolbox para Matlab de robótica. En cuanto al efector final se utilizó un modelo propio, simulando el mostrado anteriormente.

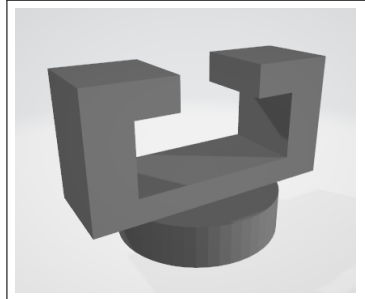


Figura 3: *Gripper modelado en 3D*

### 3. Especificaciones

#### 3.1. Ficha técnica del robot

Las características generales del robot son las siguientes:

- Robot tipo serie
- Grados de libertad: 6
- Tipo de articulaciones: Rotacionales
- Límites articulares:
  - Articulación 1:  $\pm 165^\circ$
  - Articulación 2:  $-135^\circ/45^\circ$
  - Articulación 3:  $-65^\circ/155^\circ$
  - Articulación 4:  $\pm 350^\circ$
  - Articulación 5:  $\pm 125^\circ$
  - Articulación 6:  $\pm 350^\circ$

A continuación se presentan las dimensiones del robot junto con su rango de trabajo:

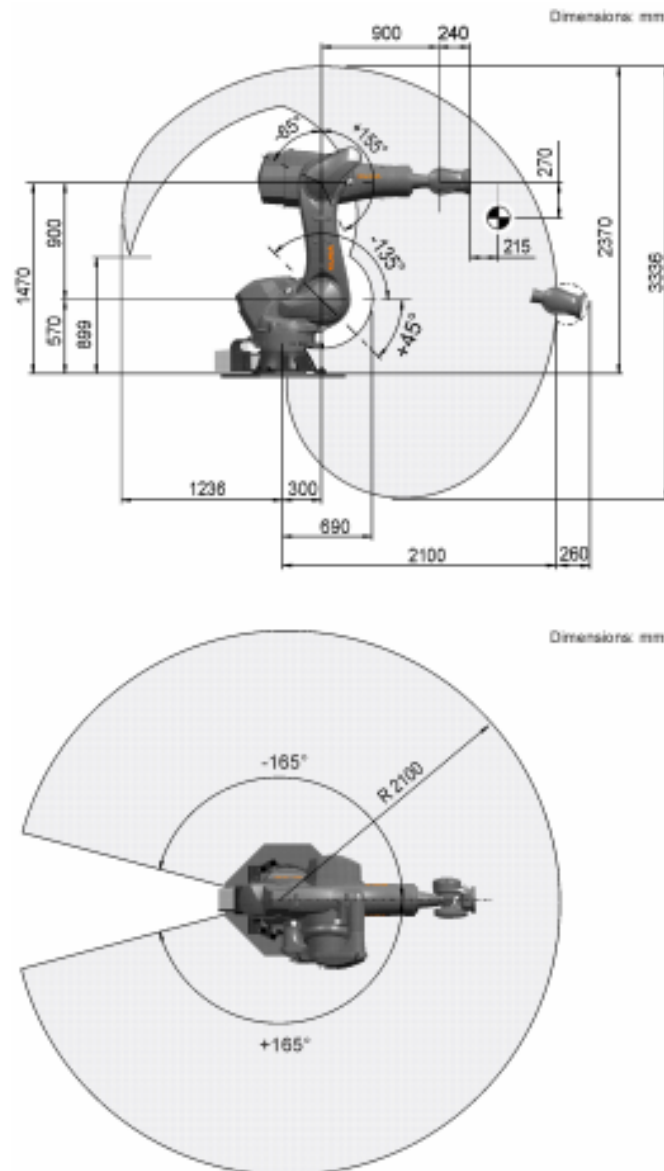


Figura 4: *Dimensiones y área de trabajo*

### 3.2. Tarea a realizar

La aplicación consiste en utilizar tres robots KUKA KR 120 R2100 nano F exclusive de forma conjunta para manipular piezas de fundición en un proceso de fabricación en serie. Todos los robots serán montados sobre el suelo y se ubicarán de forma adecuada en relación a las máquinas circundantes con las que deben trabajar, de manera tal de estar trabajando siempre lejos de los límites del espacio de trabajo.

La tarea a realizar se basa en el siguiente vídeo:

<https://www.youtube.com/watch?v=h1HvXLDjUsw>

Para el primer robot, la tarea a realizar consiste en recoger un premolde de fundición de una especie de plano inclinado de  $30^\circ$  y llevarlo hasta una máquina que aplica presión sobre el premolde para deformarlo.

El segundo robot, recoge esta pieza ya deformada y la ingresa en una nueva máquina que realiza un proceso similar, dándole a la pieza la forma final que se desea.

Finalmente, el tercer robot recoge la pieza final y la deposita en una cinta transportadora.

A continuación se muestra una gráfica en perspectiva de la ubicación relativa de los robots y las máquinas circundantes:

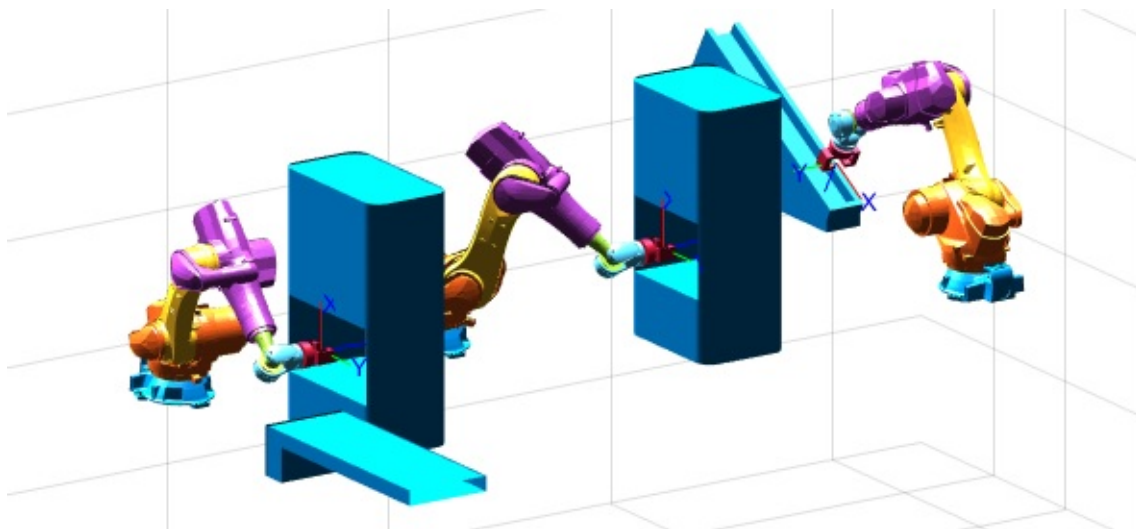


Figura 5: Ubicación de los robots y las máquinas circundantes

## 4. Robot

### 4.1. Convención Denavit & Hartenberg

Basándonos en la siguiente figura junto con las dimensiones del robot se obtiene la parametrización del robot según la convención de Denavit & Hartenberg:

De este modo, la matriz de parámetros de D&H resulta ser:

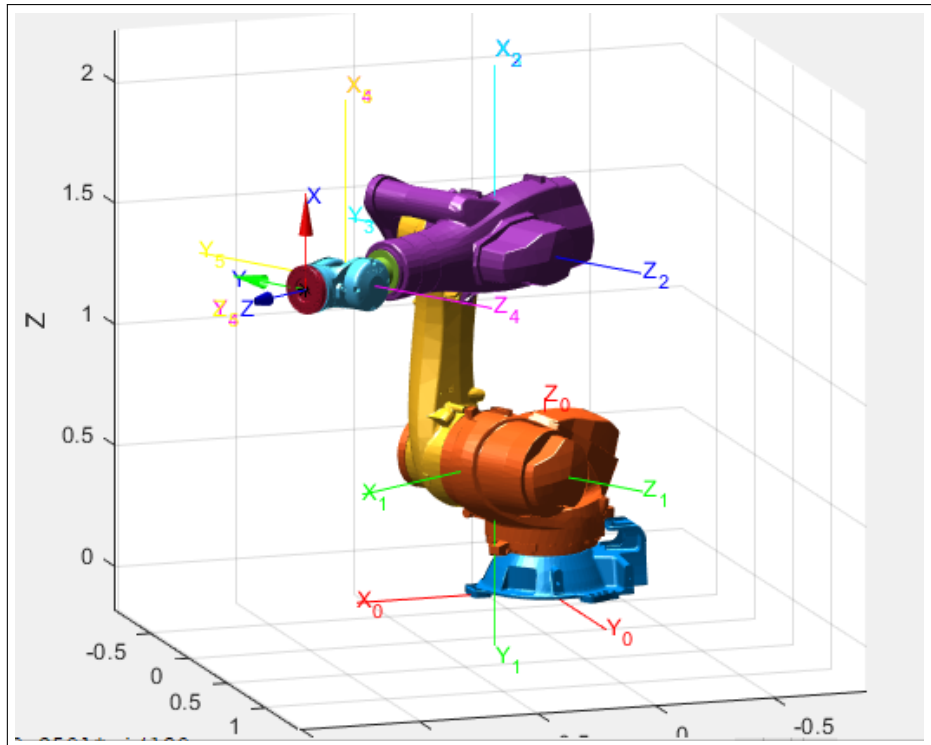


Figura 6: *Sistemas de referencia de cada articulación*

## 4.2. Cinemática Directa

Para la solución de la cinemática directa se hizo uso de algunas de las herramientas matemáticas aportadas por el Robotics Toolbox de Peter Corke.

A partir de conocer la matriz de parámetros de D&H se puede crear un objeto de la clase '**SerialLink**'. Luego diseñamos nuestra propia función que devuelve una matriz de transformación homogénea del efector final o herramienta con respecto a la base, a partir de pasar como parámetro el objeto SerialLink que representa a nuestro robot y el vector articular propuesto.

$${}^{base}T_{tool} = Transf\_Sistemas\_Total(R, q)$$

Donde '**q**' es el vector articular y '**R**' es el objeto SerialLink mencionado anteriormente.

Entre los atributos de la clase SerialLink encontramos que podemos colocar una matriz de transformación homogénea **base**, si queremos establecer al robot en otra posición que no sea el origen de coordenadas, además de una matriz de



Sistema	$\theta$	d	a	$\alpha$	$\sigma$
1	$q_1$	0,570	0,300	$-\pi/2$	0
2	$q_2$	0	0,900	0	0
3	$q_3$	0	0	$-\pi/2$	0
4	$q_4$	0,900	0	$-\pi/2$	0
5	$q_5$	0	0	$\pi/2$	0
6	$q_6$	0,240	0	0	0

Cuadro 1: *Matriz de parámetros D&H*

transformación homogénea, **tool**, entre el sistema {6} y el efector final, en nuestro caso, una herramienta tipo gripper.

A continuación se muestran los comandos utilizados en cada uno de los robots para poder establecer los atributos anteriormente mencionados.

```
R1.tool = transl(0,0,0.2);

R2.base = transl(1,4.2,0)*trotz(-pi/2);
R2.tool = transl(0,0,0.2);

R3.base = transl(1,6.5,0)*trotz(-pi/2);
R2.tool = transl(0,0,0.2);
```

Figura 7: *Atributos base y tool*

Además entre los atributos podríamos encontrar un offset para establecer la posición del robot de otra forma, sin embargo este no es el caso.

La implementación de esta función se muestra a continuación:

Si observamos la fig. 8, vemos que se hace uso de la función '**Transf\_Sistemas**', dicha función devuelve una función de transformación homogénea de sistema ' $i$ ' con respecto al sistema ' $i-1$ '. La función recibe como parámetros un objeto de la clase SerialLink, un numero entero que identifica el sistema  $i$  y un valor angular de la articulación  $i$  dado por  $q(i)$  en ese caso. La implementación se observa a continuación:

```

function T_total = Transf_Sistemas_Total(R, q)
    q = q + R.offset;
    base = R.base.double;
    tool = R.tool.double;
    T=zeros(4,4*length(q));
    for i=1:length(q)
        %Calcula las matrices de un sistema respecto al siguiente
        T(:, (1+4*(i-1)):(4+4*(i-1))) = Transf_Sistemas(R,i,q(i));
    end
    %Calcula la matriz total
    T_total = eye(4);
    for i=1:length(q)
        T_total = T_total * T(:, (1+4*(i-1)):(4+4*(i-1)));
    end
    T_total = base * T_total * tool;
end

```

Figura 8: *Implementación Transf\_Sistemas\_Total*

```

function T = Transf_Sistemas(R, i, q)
    T = trotx(q)*transl(0,0,R.links(i).d)*transl(R.links(i).a,0,0)*trotx(R.links(i).alpha);
end

```

Figura 9: *Implementación Transf\_Sistemas*

### 4.3. Cinemática Inversa

Como solución al problema de la cinemática inversa se utilizó el método de **Pieper**, ya que nuestro robot posee una muñeca esférica, es decir, un punto donde tenemos el cruce de tres ejes de rotación. Este método se basa en realizar un desacople cinemático entre el brazo y la muñeca para posteriormente resolver ambos problemas por separado.

Si denominamos la posición del extremo como:

$${}^0\bar{p}_6$$

Y la posición de la muñeca como:

$${}^0\bar{p}_4 = {}^0\bar{p}_6 - d_6\hat{z}_6$$

Siendo  $d_6$  parámetro de la matriz de D&H de la fila 6, mientras que  $\hat{z}_6$  es el versor en dirección de z de la matriz de transformación homogénea del sistema  $\{0\}$  al sistema  $\{6\}$ .

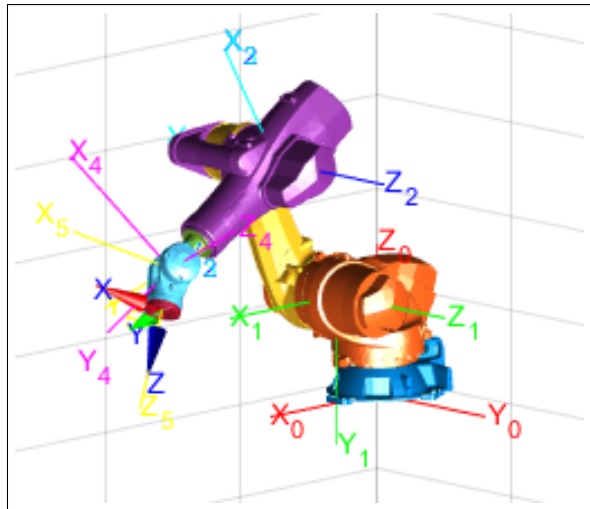


Figura 10: *Ejemplo de posición a alcanzar por el robot*

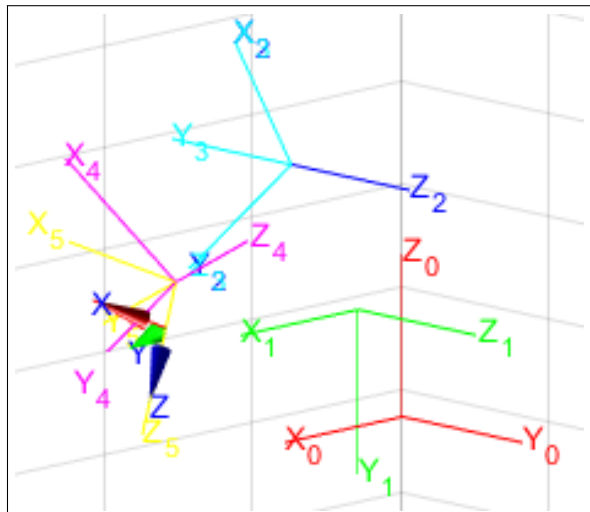


Figura 11: *Sistemas coordenados de las articulaciones*

De esta manera se obtienen dos problemas a resolver, por un lado los valores articulares que conllevan a la posición de la muñeca y por otro lado los valores articulares, que a partir de la posición de la muñeca, conllevan a la posición del extremo.

**Problema 1:**  $(q_1, q_2, q_3) = f({}^0x_4, {}^0y_4, {}^0z_4)$

**Problema 2:**  ${}^0T_6 = {}^0T_3 {}^3T_6 \rightarrow {}^3T_6 = ({}^0T_3^{-1}) {}^0T_6$

La función propia utilizada para resolver el problema de la cinemática inversa es:

$$q_{sol} = CinInv(T, R, q_0, q_{mejor});$$

La justificación del funcionamiento interno de dicha función se encuentra disponible en el anexo.

#### 4.4. Relación de Velocidades

El Jacobiano de nuestro robot resulta ser de 6x6. Analizar simbólicamente dicho Jacobiano mediante MATLAB resultaría imposible, ya que la expresión que se obtendría sería inmanejable y además implicaría un costo computacional muy elevado. De modo que para encarar el problema del Jacobiano se utilizó un análisis numérico.

Para encontrar los puntos singulares dentro del espacio de trabajo lo que se procedió a hacer fue a partir de 10000 puntos aleatorios dentro del espacio de trabajo, conocer aquellos en los que su determinante resultaba nulo. Los resultados, utilizando las herramientas provistas por el Robotic Toolbox de Peter Corke, obtenidos fueron los siguientes:

Columns 1 through 5				
137.2027	75.1754	-30.8343	124.2243	-28.9222
-110.6139	-95.2878	37.1888	-75.2378	-92.7984
-81.9341	-98.6154	19.7394	-73.4722	-81.2955
278.2359	-1.5392	313.8954	200.2732	74.6668
-0.0878	-124.1914	-0.0237	0.0242	0.0333
80.7018	-319.8752	-272.3999	-286.7161	223.1703
Columns 6 through 8				
-4.0060	-157.7330	-163.9079		
-116.7327	-98.9498	-128.0415		
-90.0491	-92.0872	-89.9919		
-244.7756	-213.4466	285.4170		
2.1301	10.4682	-78.2484		
78.7771	-21.9746	-78.4750		

Figura 12: Puntos singulares calculados aleatoriamente

De esta figura, siendo cada columna un vector articular, por un lado podemos observar ciertos patrones que se mencionan a continuación, sin embargo, hay casos donde no se puede ver una relación sencilla entre parámetros que haga que el determinante del Jacobiano se anule. Algunos de los puntos singulares que pueden identificarse a simple vista analizando patrones son:

- $q_3 = -\pi/2$
- $q_5 = 0$
- *Límite espacio articular*

Todos los casos fueron justificados mediante el uso de Matlab, en donde para los 10000 puntos aleatorios generados se pisaron los valores primero de  $q_3$  y luego de  $q_5$ , de modo que los en ambos casos, los 10000 vectores articulares aleatorios resultaron singulares.

La presencia de puntos singulares en dichos valores se da por la pérdida de un grado de libertad, en donde en el primer caso se alinea el eslabón 2 con el 3, mientras que en el segundo caso se alinea el eslabón 3 con el 5.

Pese a la existencia de estos puntos singulares, en la planificación de trayectorias se utilizarán resultados para obtener trayectorias adecuadas que eviten pasar por las singularidades existentes.

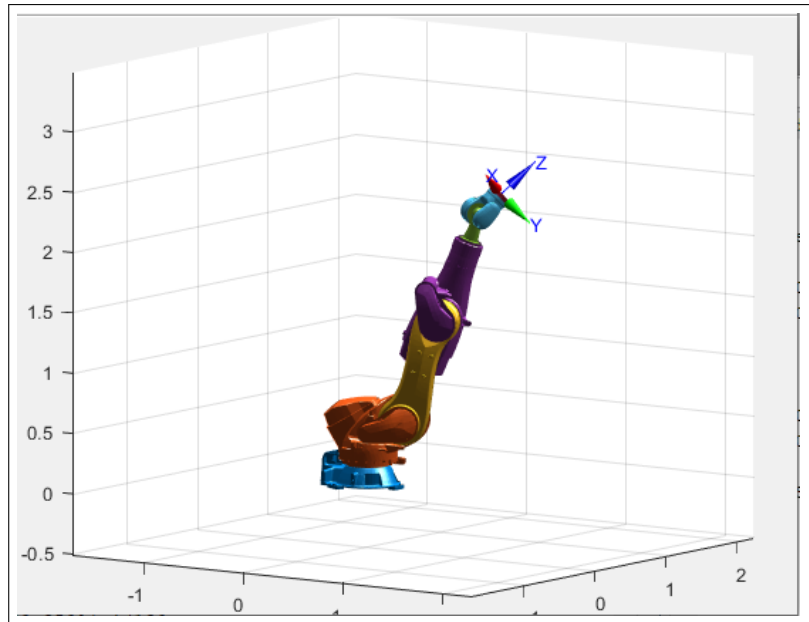


Figura 13: *Ejemplo de punto singular con  $q_3 = -\pi/2$*

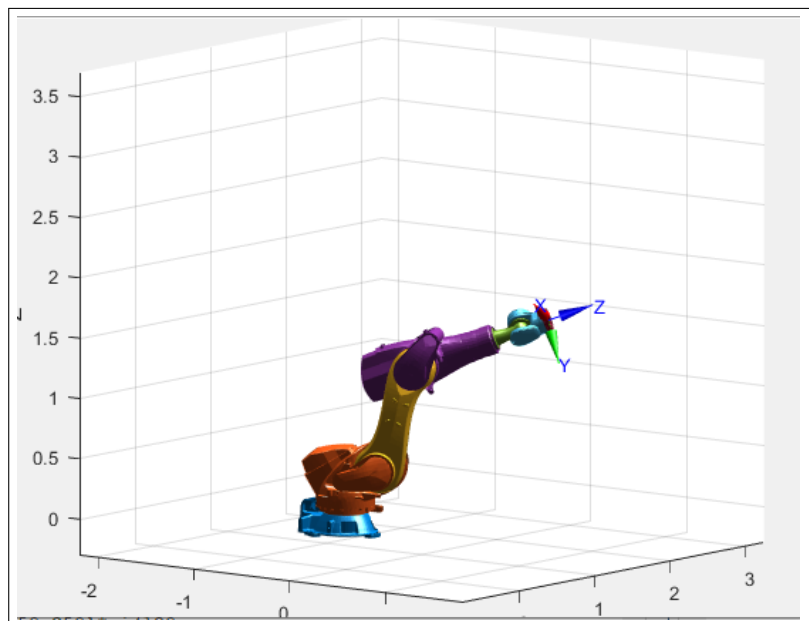


Figura 14: *Ejemplo de punto singular con  $q_5 = 0$*

#### 4.5. Planificación y Generación de Trayectorias

En cuanto a la planificación de trayectorias, el objetivo es que los 3 robots planteados puedan manipular correcta y cuidadosamente las piezas de fundición, sin que se produzcan colisiones con las máquinas circundantes, y evitando además pasar por los puntos singulares que se identificaron anteriormente. Por esta

razón, es necesario que el robot se mueva a una velocidad moderada y que no se produzcan movimientos bruscos en la trayectoria.

Con todo esto en mente, se plantean varios puntos en el espacio para formar parte de las trayectorias de los diferentes robots. Cuando alguno de los robots se encuentre trabajando dentro o cerca de una máquina, generalmente se utilizará una interpolación en el espacio cartesiano para obtener una trayectoria más precisa en este espacio. De esta forma, por ejemplo, cuando los robots deban ingresar a una máquina para dejar o recoger una pieza, lo harán en forma lenta y siguiendo una línea recta en el espacio cartesiano. Por otra parte, cuando el robot ya no se encuentre en peligro de colisionar con alguna máquina, se empleará una interpolación articular, ya que en general producían una trayectoria mucho más limpia y suave a la hora de mover a los robots de un punto intermedio de posicionamiento a otro que las que se obtenían mediante una interpolación cartesiana.

En las siguientes secciones se describirán la selección de puntos en el espacio cartesiano y las herramientas o métodos seleccionados para generar la trayectoria de cada uno de los robots.

#### 4.5.1. Robot 1

Para el primer robot, se definieron 5 puntos en el espacio cartesiano, donde dos de esos puntos corresponden a las posiciones donde el robot recoge y deja la pieza que debe ser manipulada, y los otros tres puntos corresponden a puntos intermedios de posicionamiento que evitan colisiones y también que el robot pase por puntos singulares durante su trayectoria. Las coordenadas articulares correspondientes a estos puntos se pueden ver en la tabla 2:

Punto	$q_1$	$q_2$	$q_3$	$q_4$	$q_5$	$q_6$
0	0	-1,152	-0,1355	0	-0,7641	$-\pi$
1	0	-1,047	-0,001	0	-0,5243	$-\pi$
2	$\pi/2$	-0,823	0,426	$\pi$	-1,174	$-\pi/2$
3	$\pi/2$	-0,581	-0,160	$\pi$	-0,8296	$-\pi/2$
4	$\pi/2$	-0,406	-0,228	$\pi$	-0,936	$-\pi/2$

Cuadro 2: Puntos en el espacio articular para el robot 1

Para pasar de estos puntos en el espacio cartesiano a vectores articulares que representen la posición instantánea de todos los eslabones del robot, se utilizó la función de cinemática inversa desarrollada en la sección 4.3. En particular, a la hora de realizar una interpolación en el espacio articular, se eligió cuidadosamente entre los diferentes vectores articulares posibles (recordar que, en general, se tienen 8 soluciones posibles para el problema de cinemática inversa), de manera tal que la transición de un punto a otro no obligue al robot a pasar por un punto singular.

La interpolación cartesiana se realizó utilizando la función `ctray` incorporada en el toolbox de Peter Corke, mientras que para la interpolación articular se utilizó la función `mstraj`, la cual es una función de aproximación que permite que el robot pase aproximadamente por los puntos intermedios pero con la ventaja de que no necesita detenerse. Se realizó una animación de la trayectoria del robot, la cual se puede ver en el código anexo.

Por otra parte, a continuación se muestran las gráficas correspondientes a la posición, velocidad y aceleración que experimenta el efector final del robot. Es importante remarcar que todas estas gráficas están dadas en función del número de pasos, para transformarla al dominio del tiempo se debe multiplicarla por un diferencial tiempo que represente el tiempo que demora el robot en realizar cada paso. Para el cálculo de las derivadas, se adoptó este diferencial tiempo como  $dt = 0,1$  segundos/paso.



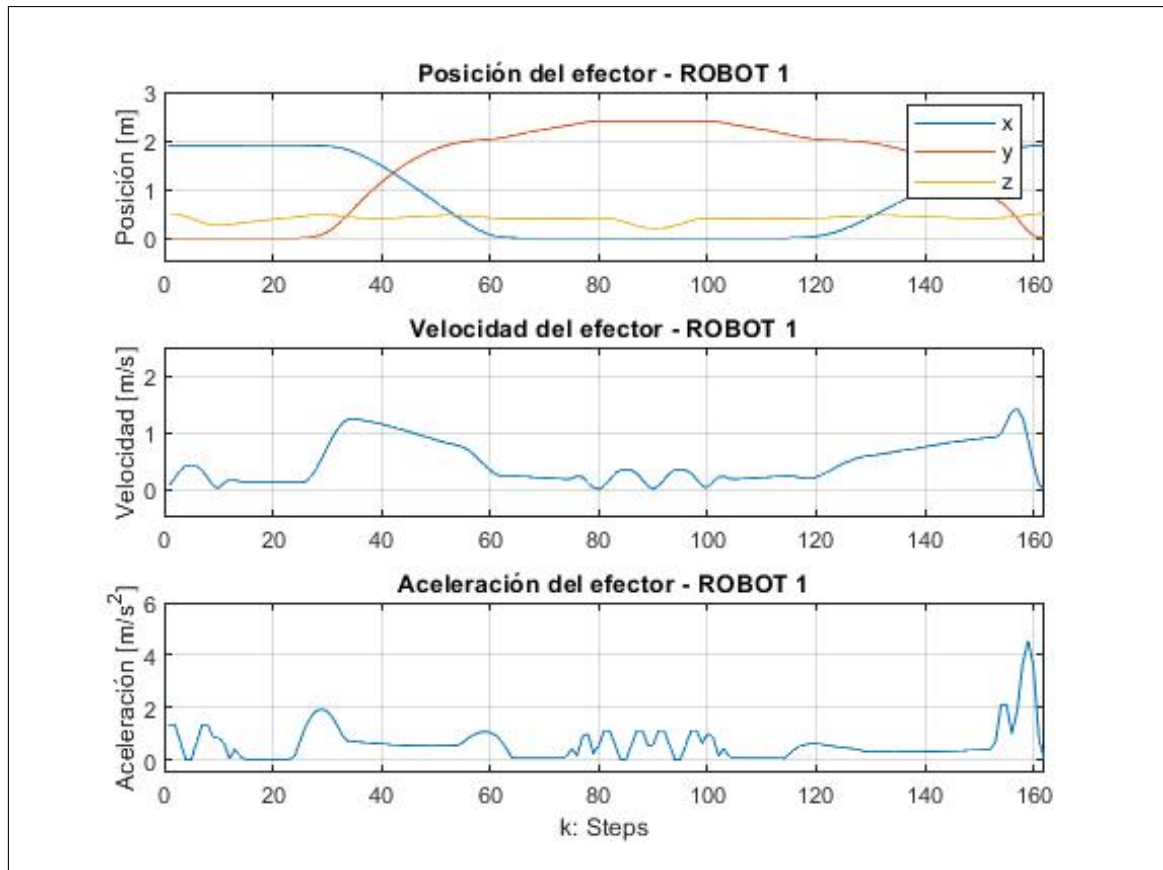


Figura 15: *Evolución temporal de la posición, velocidad y aceleración del efector del robot 1*

Vemos que toda la trayectoria en el espacio cartesiano es bastante suave, a excepción de un pequeño pico de aceleración que se produce hacia el final de la misma, cuando el robot está llegando hacia la posición inicial para luego recoger otra pieza, pero cuya amplitud no es muy grande y tampoco se trata de un movimiento crítico que deba ser realizado con mucha precisión, por lo que la trayectoria obtenida se considera aceptable.

La evolución y el análisis de las gráficas correspondientes a las variables articulares se encuentra detallado en el anexo del presente informe.

#### 4.5.2. Robot 2

En cuanto al robot 2, se tienen en cuenta las mismas consideraciones vistas para el robot 1, la única diferencia es que aquí se utilizó la función `jtraj` para realizar las interpolaciones en el espacio articular, en lugar de `mstraj` como en el caso anterior. Se llegó a esta decisión debido a que la función `mstraj` producía

gráficas de velocidad caóticas y picos de aceleración de amplitud considerable, y para solucionar estos problemas el movimiento tenía que ser excesivamente lento, lo cual tampoco es ideal.

En este caso, se plantearon 7 puntos en el espacio cartesiano, los cuales corresponden a los siguientes vectores articulares (tabla 3):

Punto	$q_1$	$q_2$	$q_3$	$q_4$	$q_5$	$q_6$
0	-0,806	-0,759	0,257	-0,871	1,231	0,3769
1	-0,634	-0,559	-0,209	-0,796	0,977	0,520
2	-0,634	-0,386	-0,277	-0,751	1,051	0,433
3	-1,193	-0,775	0,298	0,420	1,130	-0,188
4	-2,039	-0,784	-0,189	1,2934	1,187	-0,920
5	-2,186	-0,626	-0,487	1,268	1,026	-1,028
6	-2,186	-0,500	-0,478	1,195	1,071	-0,882

*Cuadro 3: Puntos en el espacio articular para el robot 2*

Por otra parte, a continuación se muestran las gráficas obtenidas para la posición, velocidad y aceleración del efector en función del número de pasos (variable temporal):

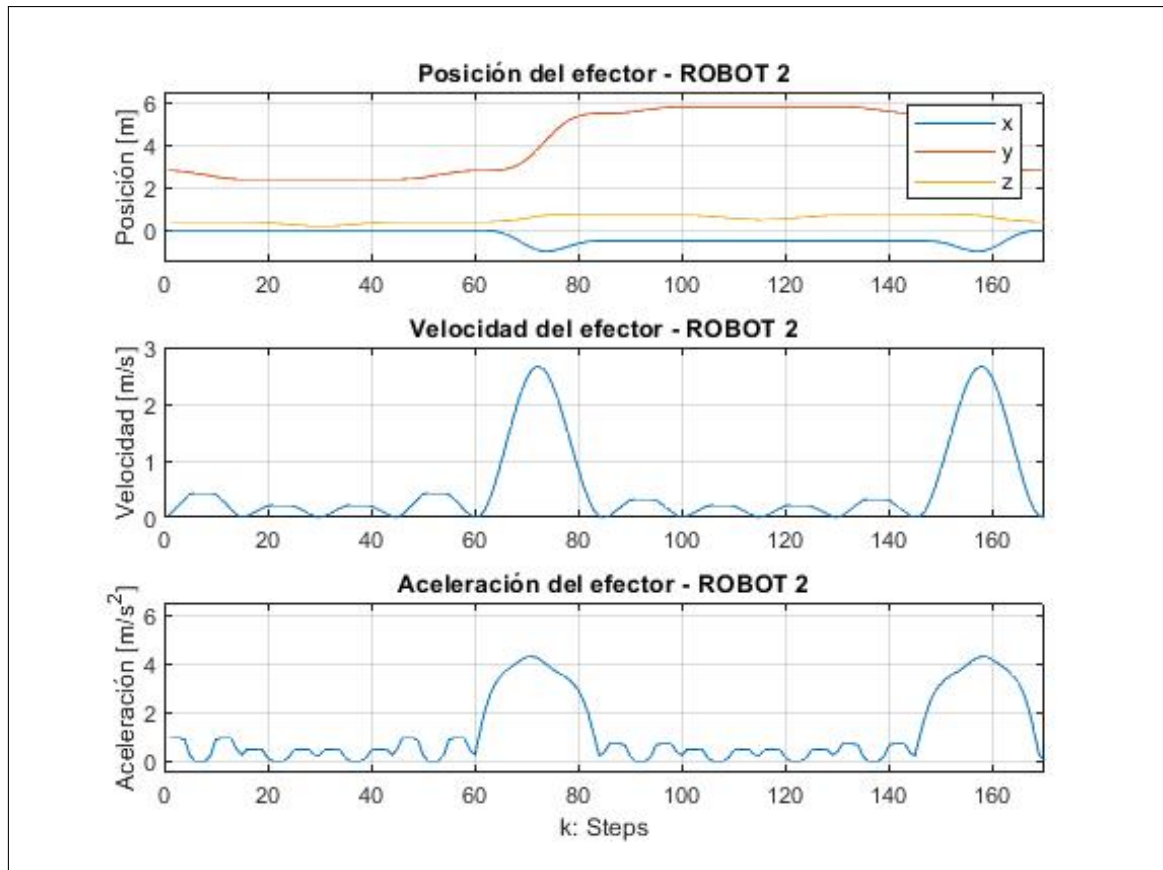


Figura 16: *Evolución temporal de la posición, velocidad y aceleración del efector del robot 2*

Nuevamente, se observa que las trayectorias y curvas de velocidad y aceleración en el espacio cartesiano son suaves y no existen picos de aceleración significativos. Al igual que en el caso anterior, el análisis de las gráficas obtenidas en el espacio articular se encuentra incluido en el anexo del informe.

#### 4.5.3. Robot 3

Se realiza el mismo análisis que en los otros dos casos. Nuevamente, aquí se utiliza la función `jtraj` para interpolar en el espacio articular en lugar de `mstraj`, por las mismas razones que se consideraron en el caso del robot 2. Los puntos que se consideraron para el tercer robot, expresados en el espacio articular del mismo, son:

Punto	$q_1$	$q_2$	$q_3$	$q_4$	$q_5$	$q_6$
0	-1,399	-0,925	0,067	-1,458	1,441	0,850
1	-0,872	-0,469	-0,778	-1,317	0,912	1,171
2	-0,872	-0,354	-0,778	1,228	0,949	1,021
3	$-\pi/2$	-0,708	-0,332	0	0,530	0
4	$-\pi/2$	-0,575	-0,324	0	0,671	0

Cuadro 4: *Puntos en el espacio articular para el robot 3*

En este sentido, las gráficas obtenidas para la posición, velocidad y aceleración del efector del tercer robot en función del número de pasos (variable temporal) se muestran a continuación:

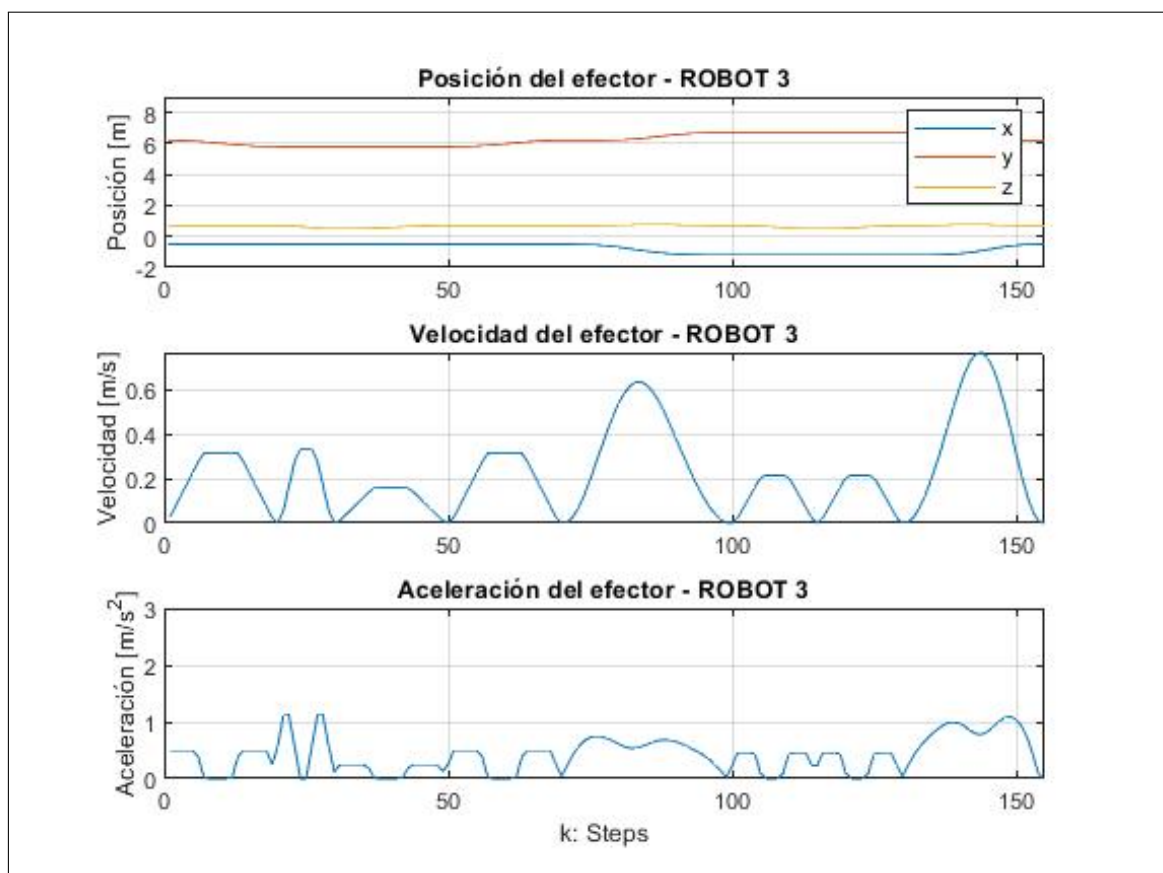


Figura 17: *Evolución temporal de la posición, velocidad y aceleración del efector del robot 3*

Nuevamente, se observa que las trayectorias y curvas de velocidad y aceleración en el espacio cartesiano son suaves y no existen picos de aceleración significativos. Al igual que en el caso anterior, el análisis de las gráficas obtenidas en el espacio articular se encuentra incluido en el anexo del informe.

## 5. Sensores y actuadores

Considerando que el robot seleccionado es el KUKA KR 120 R2100 nano F exclusive y, teniendo en cuenta que el datasheet gratuito del robot no proporciona información acerca de los sensores internos o actuadores que vienen ya incluidos en el robot, nos vemos en la obligación de describir aproximadamente o suponer aquellos actuadores y sensores que deberían ser tenidos en cuenta para la realización de la tarea.

### 5.1. Actuadores

En primer lugar, resulta apropiado pensar que los actuadores que incluye son eléctricos. Considerando el tipo de aplicaciones para las que está pensado el robot según el fabricante, es fundamental conocer el estado de las articulaciones y su posición, es por ello que se plantea el uso de servomotores con sus respectivos drivers, potencias, consumos y encoder para conocer la posición de cada articulación. Además, cada servomotor cuenta con un freno de parada mecánico.

### 5.2. Sensores

En cuanto a los sensores, se espera que el robot cuente con sensores de corriente dada su simplicidad y su costo, que hacen que se utilicen comúnmente en este tipo de sistemas mecatrónicos.

Se consideró la implementación de sensores de temperatura internos, sin embargo, con un estudio del estado de corriente a través del sensor anterior, es posible determinar la temperatura con la que opera la máquina, ya que la elevación de temperatura que puede sufrir el robot se produce principalmente por Efecto Joule. Es por ello que no incluimos estos tipos de sensores.

Por otro lado, por cuestiones de seguridad y protección, se incluirá sensores de proximidad en la parte exterior del robot. Los mismos se colocarán en el efector final y en el brazo para evitar el contacto con cualquier objeto o persona. Si bien existe una amplia gama de este tipo de sensores, se optará por utilizar los ultrasónicos, porque permite la detección a distancias de hasta varios metros.

De esta forma, podríamos establecer una distancia mínima para evitar colisiones y también observar que tan próximos se encuentran los objetos alrededor del robot en todo momento. Como caso particular se podría implementar un sensor de proximidad Siemens 3rg6013-3ad00 200-1300mm.

Se pensó también en implementar un sensor de fuerza que permita hacer un control más preciso de la fuerza de agarre que debe utilizar la pinza o efector final. Sin embargo, dado que siempre se trabajarán con las mismas piezas, las cuales estarán normalizadas, nos resultó conveniente establecer una fuerza a priori para generar el agarre de manera correcta, y así evitar la compra de un sensor adicional.

## **6. Conclusiones**

Como grupo, consideramos que el desarrollo del trabajo integrador ha sido una experiencia enriquecedora, donde hemos podido dejar plasmados la mayoría de los conceptos vistos durante el semestre. Además, también se han visto implicados procesos de tomas de decisiones a lo largo de todo el trabajo, los cuales consideramos que son muy valiosos.

También nos resultó muy interesante todas las funcionalidades que están implementadas en el Toolbox de Peter Corke, que si bien no está pensado para un uso industrial o profesional, permite visualizar muchísimos conceptos cinemáticos relacionados con la Robótica que, de otra forma, serían muy difíciles de comprender y utilizar.

## 7. Referencias

- Aula Virtual Robótica - 2022 - Facultad de Ingeniería - Universidad Nacional de Cuyo
- Robotics, Vision and Control, Fundamental Algorithms in MATLAB - Peter Corke
- Robots y Sistemas Sensoriales-Torres
- <http://www.petercorke.com/robot>