

# Entrenamiento de IA para aplicación a robótica industrial: Generación de trayectorias para ajuste de parámetros de red neuronal

Mikel Merino Olagüe<sup>1</sup>, Javier Ibarrola Chamizo<sup>1</sup>, Jokin Aginaga García<sup>1,2</sup>, Mikel Hualde Otamendi<sup>1</sup>

<sup>1</sup>Departamento de Ingeniería, Universidad Pública de Navarra, España.

<sup>2</sup>Institute of Smart Cities (ISC), Universidad Pública de Navarra, España.

E-mail: [mikel.merino, javier.ibarrola, jokin.aginaga, mikel.hualdeo]@unavarra.es

## Resumen

En el presente trabajo se generan y simulan cinemáticamente una serie de trayectorias robóticas. De dichas simulaciones se obtienen diferentes datos (coordenadas articulares, posición y orientación del elemento terminal, imágenes etc.) de cara a realizar el entrenamiento de una red neuronal para aplicaciones en robótica. El objetivo de la red neuronal es poder generar trayectorias de forma automática a partir de un conjunto de imágenes y coordenadas. Para ello, se han diseñado trayectorias con dos tramos convenientemente unidos mediante curvas de Bèzier, asegurándose la continuidad hasta aceleraciones. Además, entre las posibles trayectorias que pueden llevarse a cabo debido a las distintas configuraciones del robot, se han seleccionado las más adecuadas. Se han evitado colisiones y singularidades y se han utilizado distintos criterios de selección. Se ha diseñado un algoritmo que puede ser utilizado en múltiples aplicaciones adaptándose los diferentes parámetros del mismo.

**Palabras clave:** Robótica industrial, Generación de trayectorias, Singularidades, Inteligencia Artificial

## Abstract

In the present work robot trajectories are generated and kinematically simulated. Different data (joint coordinates, end effector position and orientation, images etc.) is obtained in order to train a neural network suited for applications in robotics. The neural network has the goal of automatically generating trajectories based on a set of images and coordinates. For this purpose, trajectories are designed in two separate sections which are conveniently connected using Bèzier curves, ensuring continuity up to accelerations. In addition, among the possible trajectories that can be carried out due to the different configurations of the robot, the most suitable ones have been selected. Collisions and singularities have been avoided and different selection criteria have been used. The designed algorithm can be used in multiple applications by adapting its different parameters.

**Keywords:** Industrial Robotics, Trajectory Generation, Singularities, Artificial Intelligence

## 1. Introducción

En los últimos años el avance de la robótica industrial ha permitido automatizar operaciones que requieren miles de repeticiones. Dicha automatización ha permitido aumentar la productividad y disminuir el número de errores respecto a los cometidos por los operarios. Sin embargo, en ciertas operaciones la automatización resulta más complicada debido a la

aleatoriedad que puede darse. Por ejemplo, la recolección de objetos que pudieran no encontrarse dispuestos exactamente del mismo modo u operaciones de corte, remachado etc. en objetos con algún tipo de variabilidad.

El rápido desarrollo de la Inteligencia Artificial (IA) está posibilitando automatizar este tipo de tareas con cierta aleatoriedad. Las Redes Neuronales (RN) son

una de las soluciones que más se está desarrollando en este campo.

Las RN tienen múltiples aplicaciones, pero en el caso del presente proyecto se van a utilizar para poder generar trayectorias robóticas a partir de imágenes. Concretamente, la RN es capaz de identificar el objeto que debe coger el robot y generar la trayectoria automáticamente. Para poder llevar a cabo esta aplicación es necesaria una fase de entrenamiento de la RN en la que se le proporcionan imágenes y series temporales agrupadas en un conjunto de datos. Este entrenamiento permitirá ajustar los parámetros de forma óptima.

Este trabajo se centra en la generación de trayectorias para obtener imágenes y series temporales virtuales del robot de cara a realizar la primera fase de entrenamiento. Se ha desarrollado una metodología capaz de generar trayectorias con cierta aleatoriedad, de las que se obtienen los conjuntos de datos necesarios.

## 2. Generación de trayectorias

En robótica una trayectoria es el movimiento que se desea que describa un manipulador desde un punto inicial a un punto final, descritos dichos puntos mediante la posición y orientación de la herramienta o elemento terminal del robot. La generación de trayectorias busca encontrar las funciones que describan la posición, velocidad y aceleración en función del tiempo de las articulaciones, conocidas la posición y orientación inicial y final deseados para el elemento terminal.

En este trabajo van a generarse una serie de trayectorias utilizándose el robot UR10e. Se trata de un robot serie con 6 grados de libertad cuya cinemática se ha descrito utilizando los parámetros Denavit-Hartenberg (D-H)[1-3]. De esta forma, puede describirse la relación entre los sistemas de coordenadas asociados a dos eslabones consecutivos mediante una matriz de transformación homogénea en la que aparecen los cuatro parámetros D-H necesarios.

**Tabla 1.** Parámetros D-H del robot UR10e.

$i$	$\alpha_{i-1}$	$a_{i-1}$	$d_i$	$\theta_i$
1	0	0	$d_1$	$\theta_1$
2	$\alpha_1$	0	$d_2$	$\theta_2$
3	0	$a_2$	$d_3$	$\theta_3$
4	0	$a_3$	$d_4$	$\theta_4$
5	$\alpha_4$	0	$d_5$	$\theta_5$
6	$\alpha_5$	0	$d_6$	$\theta_6$

siendo los parámetros de D-H los siguientes:

- $\alpha_{i-1}$ : ángulo de rotación de  $z_{i-1}$  a  $z_i$  entorno al eje  $x_{i-1}$ .
- $a_{i-1}$ : distancia de traslación de  $z_{i-1}$  a  $z_i$  en el eje  $x_{i-1}$ .
- $d_i$ : distancia de traslación de  $x_{i-1}$  a  $x_i$  en el eje  $z_i$ .
- $\theta_i$ : ángulo de rotación de  $x_{i-1}$  a  $x_i$  entorno al eje  $z_i$ .

El robot tiene la tarea de coger un cable que se encuentra en una mesa siendo variables una serie de parámetros del cable como se comentará más adelante. Dicho objeto determinará cual deberá ser la posición final del elemento terminal.

De cara a la descripción de las trayectorias a recorrer por el robot existen dos espacios distintos: el espacio de operación, formado por las posiciones y orientaciones del elemento terminal en coordenadas cartesianas (pose) y el espacio articular, formado por las coordenadas articulares del robot. Ambos espacios se encuentran relacionados mediante una función  $f$  diferenciable denominada función cinemática [4].

La función  $f$  calcula la posición y orientación del elemento terminal asociada a unas coordenadas articulares determinadas del robot. Se denomina *cinemática directa* a dicha operación y es un cálculo con una única solución que siempre puede resolverse. También puede realizarse la operación inversa y calcularse las coordenadas articulares en función de la posición y orientación del elemento terminal. Se denomina *cinemática inversa* a este cálculo, el cual puede no tener solución o tener múltiples soluciones denominadas configuraciones del robot. En el caso concreto del UR10e existen hasta 8 configuraciones distintas, es decir, puede alcanzarse una misma pose con 8 combinaciones de coordenadas articulares distintas.

En este trabajo van a generarse trayectorias utilizando simultáneamente el espacio de operación y el espacio cartesiano, puesto que se ha dividido la trayectoria en dos tramos distintos. El primer tramo se denomina *aproximación*, trasladándose el robot desde su pose inicial (P1) hasta una pose intermedia (P2) cercana a la pose final (P3). En este primer tramo se busca que el robot realice el recorrido en el menor tiempo posible y está basado en el espacio articular. Se llega a una pose intermedia muy cercana a la final y se alcanza la orientación necesaria para la posterior manipulación del objeto. El segundo tramo se denomina *lento*, trasladándose el robot desde P2 hasta P3. En este tramo se busca que el robot se aproxime con una velocidad reducida y sin variar la orientación del elemento terminal. Para mantener la orientación constante se ha definido este tramo en el espacio de operación.

## 2.1. Cálculo de los puntos de trayectoria

Para generar la trayectoria dadas la pose inicial y final del robot, se comienza calculando la pose intermedia en la que se unen los dos tramos de la trayectoria. Para ello, se ha calculado una pose del robot con la misma orientación que la pose final y con unas coordenadas cartesianas  $x, y, z$  que se calculan con la distancia (en módulo) que se desea entre las coordenadas cartesianas que definen ambas poses.

A continuación, mediante la cinemática inversa que se ha obtenido utilizándose el método geométrico [5] se obtienen las coordenadas articulares de las poses inicial, intermedia y final. Se ha utilizado la misma configuración del robot para los tres cálculos de forma que el robot mantiene la misma configuración durante toda la trayectoria.

De esta forma se tiene las coordenadas articulares del robot, que se denotarán mediante el vector  $\mathbf{q} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]^T$  y las coordenadas cartesianas del elemento terminal que se definirán mediante el vector  $\mathbf{x} = [x, y, z, \alpha, \beta, \gamma]^T$  para las poses P1, P2 y P3. A dichas poses intermedias se las denominará *puntos* a partir de ahora.

## 2.2. Diseño de trayectorias mediante curvas de Bèzier

Las curvas de Bèzier son curvas polinomiales que se aproximan a una serie de puntos denominados puntos de control. Su ecuación es la siguiente:

$$B(t) = \sum_{i=0}^n C_i \cdot \binom{n}{i} t^i (1-t)^{n-i} \quad (1)$$

donde  $t$  es un coeficiente que va de 0 a 1 y  $n$  es el grado de la curva de Bèzier.

En este trabajo se han utilizado curvas de grado 5 puesto que al tener 6 puntos de control permiten garantizar la continuidad hasta las aceleraciones. Mediante estas curvas se han creado dos tramos que definirán la trayectoria completa del robot. El tramo de aproximación de la trayectoria se genera en el espacio articular de forma que sólo será necesaria la cinemática inversa calculada anteriormente en P1 y P2 [6]. Para minimizar el recorrido de las coordenadas articulares, se modifican los valores del punto inicial sumándose y restándose convenientemente  $2\pi$  radianes de forma que se tienen valores de coordenadas equivalentes.

Se han definido un total de 6 curvas de Bèzier puesto que se tienen 6 coordenadas articulares. Para garantizar la continuidad hasta aceleraciones se han definido los puntos de control del vector de puntos de control

$\mathbf{c} = [c_1, c_2, c_3, c_4, c_5, c_6]^T$  de forma que cumplan una serie de condiciones.

Se ha decidido que las velocidades ( $\dot{\mathbf{q}}$ ) y aceleraciones ( $\ddot{\mathbf{q}}$ ) articulares sean nulas en el instante inicial, imponiéndose que los tres primeros puntos de control deben tener el valor de la coordenadas articulares en P1 para cada una de las 6 curvas de forma que  $c_{1i} = c_{2i} = c_{3i} = \theta_{i,P1}$  donde  $i$  se refiere a la  $i$ -ésima coordenada articular. Para asegurar la continuidad en posiciones articulares en P2, debe cumplirse que  $c_{6i} = \theta_{i,P2}$ , es decir, el último punto de control debe ser igual a las coordenadas articulares de P2 para cada una de las curvas de Bèzier.

Para calcular los valores de  $c_{5i}$  necesarios primero se ha impuesto la velocidad en el espacio de operación que desea tenerse en P2,  $\dot{\mathbf{x}} = [\dot{x}, \dot{y}, \dot{z}, 0, 0, 0]^T$ . Las velocidades angulares son nulas puesto que en P2 ya se tiene la orientación final del elemento terminal deseada. Las velocidades en el espacio de operación se han trasladado al espacio articular mediante la ecuación obtenida a partir de diferenciar  $f$ :

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q}) \cdot \dot{\mathbf{q}} \quad (2)$$

donde  $\mathbf{J}(\mathbf{q})$  es la matriz jacobiana respecto de las velocidades articulares. Mediante la Eq. (2), se han calculado las velocidades articulares deseadas y se ha elegido  $c_{5i}$  de forma que la velocidad deseada en el punto final de la curva de Bèzier es igual a la velocidad articular deseada en dicho punto:

$$\frac{n}{T}(c_{6i} - c_{5i}) = \dot{\theta}_{i,P2} \quad (3)$$

donde  $T$  es el tiempo de duración del tramo de trayectoria y  $n$  el grado de la curva de Bèzier.

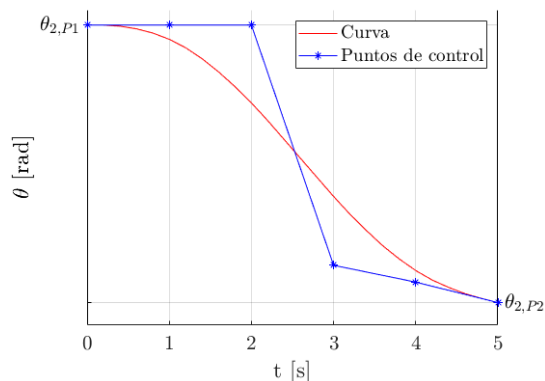
Los valores de  $c_{4i}$  siguen un proceso de cálculo similar a los de  $c_{5i}$ , imponiéndose primero la aceleración en el espacio de operación que desea tenerse en P2, en este caso,  $\ddot{\mathbf{x}} = [\ddot{x}, \ddot{y}, \ddot{z}, 0, 0, 0]^T$ . Para trasladar las aceleraciones del espacio de operación al espacio articular se ha vuelto a diferenciar la función  $f$  obteniéndose la ecuación

$$\ddot{\mathbf{x}} = \mathbf{J}(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \mathbf{u}(\mathbf{q}, \dot{\mathbf{q}}) \quad (4)$$

donde  $\mathbf{J}(\mathbf{q})$  es la matriz jacobiana y  $\mathbf{u}(\mathbf{q}, \dot{\mathbf{q}})$  es un término que depende de las posiciones y velocidades articulares. Mediante la Eq. (4) se han obtenido las aceleraciones articulares deseadas y se ha elegido  $c_4$  de forma que se cumple que la aceleración articular en el punto final de la curva de Bèzier es igual a la aceleración articular deseada:

$$\frac{n(n-1)}{T^2}(c_{6i} - 2c_{5i} + c_{4i}) = \ddot{\theta}_{i,P2} \quad (5)$$

Se han discretizado las 6 curvas obtenidas con un intervalo de tiempo fijo, de forma que se tienen una serie de coordenadas articulares en función del tiempo que definen la trayectoria.



**Figura 1.** Evolución de la coordenada articular  $\theta_2$  en el tramo de aproximación.

El tramo lento se ha definido en el espacio de operación puesto que se desea que el robot se traslade sin que la orientación de su elemento terminal varíe. Por lo tanto, como el elemento terminal varía únicamente su posición son necesarias únicamente 3 curvas de Bézier para  $x$ ,  $y$ ,  $z$ .

En la pose final del robot se desea tener velocidades y aceleraciones nulas definiéndose  $c_{4i}$ ,  $c_{5i}$ ,  $c_{6i}$  de forma homóloga a los puntos de control  $c_{1i}$ ,  $c_{2i}$ ,  $c_{3i}$  del primer tramo de trayectoria. Debe cumplirse la relación  $c_{4i} = c_{5i} = c_{6i} = x_{i,P3}$ . Siendo  $x_{i,P3}$  el valor de la componente  $i$ -ésima del vector  $\mathbf{x}$  en P3.

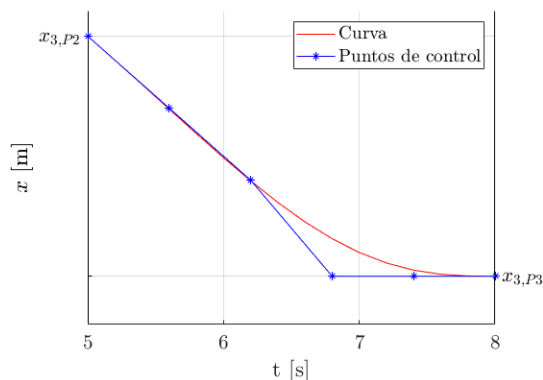
Para asegura la continuidad en P2 se impone  $c_{1i} = x_{i,P2}$  de forma que ambas trayectorias coincidan en el punto intermedio tanto en el espacio de operación como en el espacio articular. La continuidad en velocidades en P2 es más sencilla en este caso puesto que no es necesario utilizar la Eq. (2) y basta con imponer que la velocidad al inicio de la curva de Bézier se igual a la deseada en P2:

$$\frac{n}{T}(c_{2i} - c_{1i}) = \dot{x}_{i,P2} \quad (6)$$

de esta ecuación se obtiene  $c_{2i}$  y seguidamente  $c_{3i}$  imponiendo que la aceleración al inicio de la curva de Bézier sea igual a la deseada en P2:

$$\frac{n(n-1)}{T^2}(c_{3i} - 2c_{2i} + c_{1i}) = \ddot{x}_{i,P2} \quad (7)$$

Las curvas de Bézier obtenidas para este segundo tramo se han discretizado al igual que en el primer tramo y después se ha calculado la cinemática inversa de cada uno de los instantes discretos de la trayectoria.



**Figura 2.** Evolución de la coordenada cartesiana  $x_3$  tramo lento.

De esta forma, se tiene la trayectoria completa descrita en el espacio articular de forma que puede describirse el movimiento del robot de forma unívoca. La trayectoria en el espacio de operación resultaría ambigua puesto que podrían tenerse distintas soluciones articulares en cada punto debido a las diferentes configuraciones del robot.

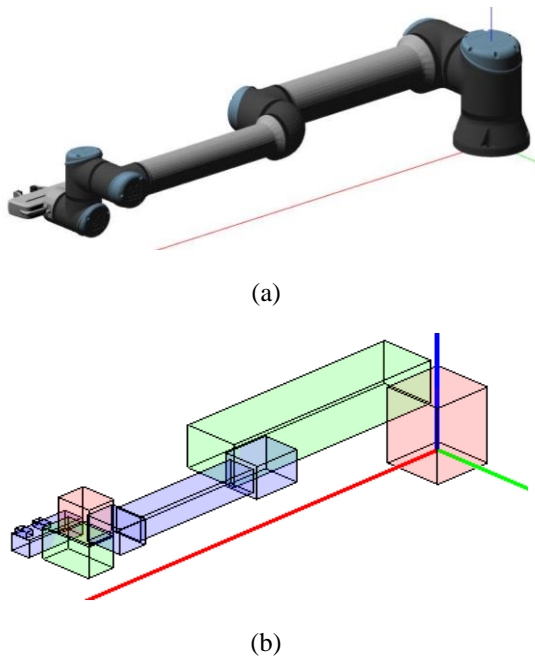
### 3. Selección de trayectorias

Bajo las premisas de la sección 2 se han generado un total de 8 trayectorias para unas poses inicial y final dadas, de forma que cada trayectoria corresponde a cada una de las configuraciones del robot. Estas 8 trayectorias representan 8 soluciones posibles para que el robot se traslade de una pose inicial determinada a una pose final determinada. En esta sección se utilizarán una serie de criterios para seleccionar cual es la mejor entre las posibles soluciones.

#### 3.1. Análisis de colisiones

Las colisiones con algún objeto del entorno y las auto colisiones del propio manipulador son uno de los problemas que puede aparecer al diseñar una trayectoria.

De cara a poder eliminar las trayectorias en las que existe algún tipo de colisión, se ha diseñado un algoritmo de detección de colisiones. El algoritmo, evalúa si existe algún tipo de intersección entre los sólidos que pueden colisionar entre sí y en caso de que la haya desecha la trayectoria.



**Figura 3.** (a) Robot modelado con sólidos originales en Gazebo. (b) Robot modelado mediante sólidos de colisión.

El robot se ha modelado mediante paralelepípedos rectangulares de cara a simplificar y agilizar el algoritmo de detección de colisiones. Como puede verse en la Figura 3 los paralelepípedos del mismo color que se encuentran en contacto entre si modelan un mismo eslabón del robot para lograr un mejor ajuste a la entidad geométrica real. Este mejor ajuste permite que se rechace un menor número de colisiones falsas en las que se tendría contacto entre los sólidos del robot modelo, pero no entre los eslabones reales del robot.

### 3.2. Análisis de singularidades

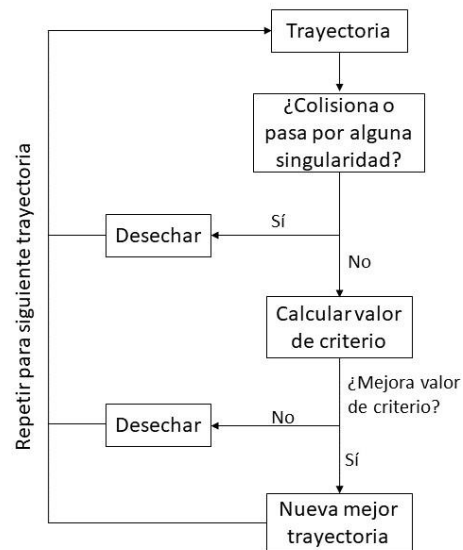
Las singularidades son combinaciones de posiciones articulares en las que un robot pierde uno o varios de sus grados de libertad, no pudiendo trasladarse en una o más direcciones determinadas. Existen dos tipos de singularidades distintas para manipuladores en serie: las *singularidades de frontera* y las *interiores* [7]. El primer caso ocurre cuando el manipulador está en el límite de su espacio de trabajo y se encuentra en una posición completamente extendida o recogida. El segundo caso ocurre cuando para unas velocidades determinadas del elemento terminal las velocidades articulares no están acotadas. Debe evitarse pasar por estas posiciones y por zonas cercanas a las mismas para evitar velocidades excesivas en las coordenadas articulares que podrían ser críticas para la vida del robot.

En robótica se evalúan las singularidades utilizando la matriz jacobiana que relaciona las velocidades articulares con las velocidades en coordenadas cartesianas Eq. (2). Un determinante nulo de esta matriz implicaría que el manipulador se encuentra en una posición singular. De cara a evaluar la proximidad de un robot a una singularidad se ha utilizado el número de condición de la matriz jacobiana. De esta forma se han podido rechazar las trayectorias con algún instante con valor de condicionamiento superior al valor máximo establecido.

### 3.3. Criterios de selección de trayectorias

Entre las trayectorias no desechadas debido a colisiones y singularidades se han diseñado una serie de criterios de cara a seleccionar la mejor de ellas:

- Criterio de minimización de velocidad articular máxima: Se escoge la trayectoria cuya velocidad articular máxima es la menor.
- Criterio de minimización del recorrido del elemento terminal: Se estima la distancia que ha recorrido el elemento terminal sumándose los módulos entre cada uno de los instantes de tiempo de la trayectoria y se escoge la de menor recorrido.
- Minimizar el valor máximo del número de condición: Se escoge la trayectoria cuyo máximo valor del número de condición de la matriz jacobiana sea el menor.



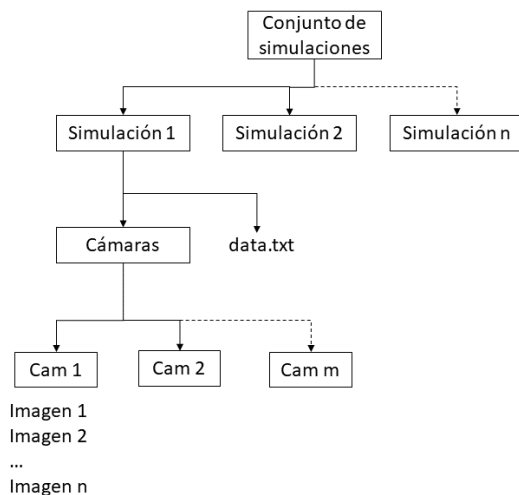
**Figura 4.** Diagrama de selección de trayectorias.

Se han generado conjuntos de datos utilizándose los tres criterios indistintamente, siguiéndose el esquema de la Figura 4. Se comentará en la sección 5 cuáles han sido las diferencias entre ellos.

#### 4. Generación de datos en base a la trayectoria óptima

Siguiendo las secciones 2 y 3 se han generado 8 trayectorias y se ha seleccionado la mejor entre ellas. De cara a alimentar la red neuronal no basta con una única trayectoria, siendo necesarias un gran número de trayectorias en las que se varíe algún parámetro de las mismas. Para generar esta conjunto de trayectorias se han variado aleatoriamente los siguientes parámetros del cable que debe coger el robot: posición, radio, ángulo de apertura, ángulo de torsión de los terminales y longitud del mismo. Así, se consigue variar la pose final que debe alcanzar el robot pudiendo crearse un gran número de trayectorias aleatorias.

Se ha llevado a cabo la generación (sección 2) y selección (sección 3) de un gran número de trayectorias aleatorias obteniéndose en cada una de ellas distintos datos para alimentar la red neuronal con conjuntos de datos formados por varias trayectorias. Los conjuntos de datos generados están compuestos por una serie de imágenes, vídeos y archivos de texto con series temporales (posición, velocidad y aceleración articulares, posición y orientación del elemento terminal etc.).



**Figura 5.** Organización de los conjuntos de datos.

La Figura 5 muestra la estructura de un conjunto de datos completo. Dicho conjunto estará formado por una serie de carpetas conteniendo cada una todos los datos de una simulación de trayectoria completa. Cada una de estas simulaciones se habrá llevado a cabo variándose de forma aleatoria los parámetros del punto final de la trayectoria. De este modo la red neuronal se entrena de forma robusta ante posibles variaciones de la trayectoria.

#### 4.1. Imágenes

Las imágenes del manipulador llevando a cabo la trayectoria en distintos instantes de tiempo es la parte que mayor información aporta a la RN puesto que a partir del entrenamiento mediante estas imágenes será capaz de generar trayectorias de forma autónoma.

La obtención de estas imágenes se ha llevado a cabo utilizándose el software libre de simulación de robótica Gazebo. Se ha creado un entorno virtual de trabajo lo más parecido posible al real. Además, se han instalado cámaras en distintas posiciones y orientaciones según las necesidades de entrenamiento de la red neuronal.

El motor físico ODE integrado en Gazebo permite realizar simulaciones dinámicas e introducir algoritmos de control, pero en este trabajo no se van a utilizar estas características. El software se utiliza como un mero visualizador y herramienta para tomar las imágenes necesarias. Se ha tomado esta decisión debido a que se quiere alimentar la RN mediante trayectorias ideales en las que sólo se tenga en cuenta la cinemática. Tomar imágenes a partir de una simulación dinámica resultaría inadecuado puesto que se introducirían errores en la trayectoria provenientes de diferentes fuentes como pueden ser el integrador numérico, el control utilizado etc.

Por lo tanto, para cada instante de la trayectoria, se ha utilizado un modelo del robot en Gazebo en la posición deseada, se han tomado las imágenes necesarias y se ha pasado al siguiente instante de tiempo. Se ha repetido este proceso hasta obtenerse todas las imágenes correspondientes a los instantes de tiempo de la trayectoria.

#### 4.2. Series de datos temporales

Las series de datos temporales son generadas en el momento en el que la trayectoria óptima ha sido seleccionada. Se guardan en un archivo de texto de forma que puedan abrirse en múltiples softwares de tratamiento de datos. La primera fila del archivo contiene el encabezado con los nombres de todas las variables:

- Tiempo.
- Posiciones, velocidades y aceleraciones articulares.
- Posición y orientación del elemento terminal en coordenadas cartesianas.

El resto de filas representan el valor de cada una de estas variables en un instante de tiempo determinado.

## 5. Resultados

En esta sección van a analizarse los resultados obtenidos tras generarse varios conjuntos de datos susceptibles de ser utilizados para alimentar la RN. El análisis individual de cada una de las trayectorias obtenidas resulta completamente inviable. Así, se va a analizar una única trayectoria a modo ejemplo y después se comentarán algunos datos generales de los conjuntos de datos obtenidos.

### 5.1. Trayectoria ejemplo

La trayectoria ejemplo se genera siguiéndose las secciones 2 y 3. Se dan valores aleatorios a los parámetros del cable obteniéndose la pose final del robot para después generarse las 8 trayectorias correspondientes a cada una de las configuraciones. De las 8 trayectorias 4 han sido descartadas (trayectorias 3-6) por detectarse colisiones mediante el algoritmo presentado en la sección 3.1. La Figura 6 muestra un ejemplo de colisión en el modelo de Gazebo.

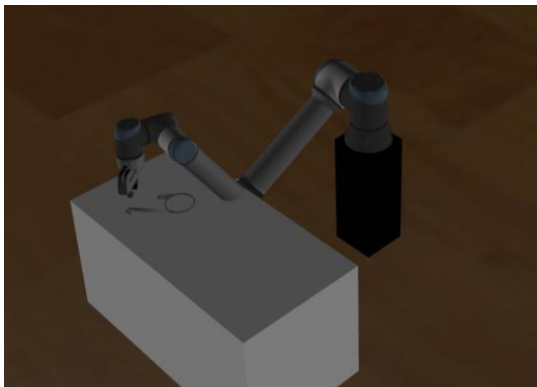


Figura 6. Trayectoria desechada por colisión.

Las colisiones se producen con la mesa en la que se encuentra el cable en configuraciones en las que la articulación entre los sólidos 2 y 3 se encuentra en una posición más baja que el primer sólido del robot (configuraciones *codo abajo*).

Las 4 configuraciones libres de colisiones se han sometido al análisis de singularidades en el que se ha comparado la evolución de los números de condición de sus matrices jacobianas como puede verse en la Figura 7. Se producen dos grandes picos que superan el valor límite establecido en las trayectorias 2 y 8. Estas trayectorias son desechadas por estar demasiado cerca de singularidades.

La elección final entre las trayectorias que dan las configuraciones 1 y 7 se ha realizado en base al criterio de minimizar el valor máximo del número de

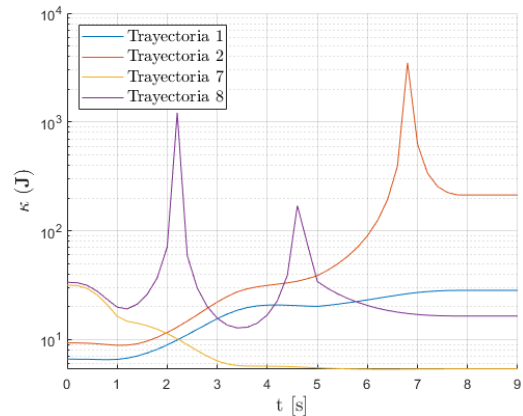


Figura 7. Evolución de los valores del número de condición.

condición. De esta forma la trayectoria 1 ha sido la seleccionada y se ha realizado toda la obtención de imágenes y series temporales presentada en la sección 4 utilizándose dicha trayectoria.

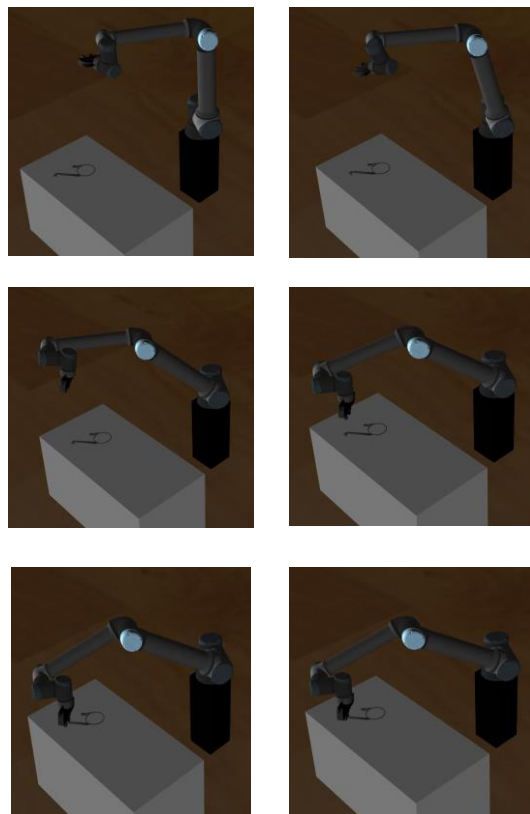


Figura 8. Evolución de la trayectoria seleccionada en distintos instantes de tiempo.

## 5.2. Resultados de un conjunto de datos completo

Se han generado 3 conjuntos de datos compuestos por 1000 trayectorias cada uno de cara a realizar un análisis de las configuraciones utilizadas y de las trayectorias que no han podido llevarse a cabo. La Tabla 2 contiene los resultados obtenidos.

**Tabla 2.** Configuraciones utilizadas.

Configuración	nº de usos	% de uso
1	4284	42.84
2	320	3.2
3	0	0
4	0	0
5	0	0
6	0	0
7	2635	26.35
8	46	0.46
Error	2715	27.15

Se ha comprobado que las configuraciones 1 y 7 son las más utilizadas, siendo la número 1 la más destacada. En cuanto al no uso de las configuraciones 3 a 6 puede decirse que es el resultado esperado puesto que el robot colisiona con la mesa debido a la configuración de *codo abajo* que adopta. A la vista de estos resultados podría modificarse el algoritmo para este caso concreto evitando calcularse las trayectorias asociadas a las configuraciones 3 a 6 de forma que se reduciría el coste computacional. Podrían incluso no calcularse en el caso de las trayectorias 2 y 8 puesto que su uso es también muy reducido.

Por otro lado, las trayectorias erróneas representan un porcentaje significativo entre todas las trayectorias generadas. Los errores se deben a que existen casos concretos en los que todas las configuraciones llevan a trayectorias en las que se produce alguna colisión o el paso por alguna singularidad. Para reducir el coste computacional que supone una trayectoria errónea se ha desarrollado el algoritmo de forma que en el momento que se produce un error se desecha la trayectoria y vuelve a iniciarse la generación completa del siguiente caso aleatorio.

Por último, en esta aplicación concreta, los criterios de selección utilizados han servido esencialmente para elegir una configuración entre la 1 y la 7. Se ha visto, que las diferencias que existían entre los valores de criterio eran mínimas y por lo tanto, puede decirse que en general podrían haberse utilizado directamente ambas configuraciones.

## 6. Conclusiones

Este trabajo presenta una metodología para crear trayectorias robóticas y seleccionar las mejoras de cara

a obtener un conjunto de datos para entrenar una red neuronal. Se han comprobado las ventajas de generar trayectorias de forma cinemática en lo que respecta a simplicidad de generación de las trayectorias e idealidad de las mismas. Se ha creado un algoritmo que permite evitar colisiones y pasos por puntos cercanos a singularidades. Las trayectorias generadas se han utilizado para exportar una serie de datos en forma de imágenes y series temporales que son los que se utilizarán para las labores de entrenamiento de la RN.

Por último, pueden fijarse los valores de las poses inicial y final del robot, las duraciones de los tramos de la trayectoria, las velocidades y aceleraciones deseadas en el punto intermedio e incluso pueden añadirse objetos extra tanto en el entorno gráfico como en el algoritmo de colisiones. Así, la metodología completa puede utilizarse para otras aplicaciones similares.

## 7. Agradecimientos

Este trabajo fue financiado por la “Convocatoria de ayudas a proyectos de I+D del Gobierno de Navarra” bajo los proyectos con Ref. 0011-1365-2021-000080 y Ref. 0011-1411-2021-000023.

## 8. Referencias

- [1] J. Denavit y R. S. Hartenberg, «A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices,» *Journal of Applied Mechanics*, vol. 22, pp. 215-221, 1965.
- [2] B. Siciliano, L. Sciavicco, L. Villani y G. Oriolo, *Robotics: Modelling, Planning and Control*, Springer, 2010.
- [3] J. Angeles, *Fundamentals of Robotic Mechanical Systems*, Springer, 2007.
- [4] I. Zaplana, J. A. Claret y L. Basanez, «Análisis Cinemático de Robots Manipuladores Redundantes: Aplicación a los Robots Kuka LWR 4+ y ABB Yumi,» *Revista Iberoramericana de Automática e informática Industrial*, vol. 15, pp. 192-202, 2018.
- [5] L. Qiang, Y. Daoguo, H. Weidong y W. Yao, «Research on Kinematic Modeling and Analysis Methods of UR Robot,» de *Information Technology and mechatronics Engineering Conference*, 2018.
- [6] J. J. Craig, *Introduction to Robotics. Mechanics and Control*, Pearson Education, 2005.
- [7] L.-W. Tsai, *Robot Analysis*, Wiley, 1999.