

Análisis de las características de un robot industrial y modelo geométrico directo KUKA KR 340 R3330

Cristian Camilo Cuestas Ibáñez, Nicolas Gil Rojas, Jorge Luis Reina Jara

2021-II

1. Características del robot

Capacidad de carga = 340 Kg (nominal) 418 Kg (máxima)

Alcance horizontal = 3326 mm

Alcance vertical = 3871 mm

Repetibilidad = ± 0.08 mm

Software = KR C4 – KR C5

Protección IP = IP65

En la figura 1 se ve la imagen del robot. En la figura 2 está el espacio alcanzable del robot, donde corroboramos el alcance vertical y horizontal expuesto. Además, junto con los ejes ilustrados en la figura 3 es posible calcular los parámetros DH presentes en la tabla 1.



Figura 1. KUKA R340 R3330

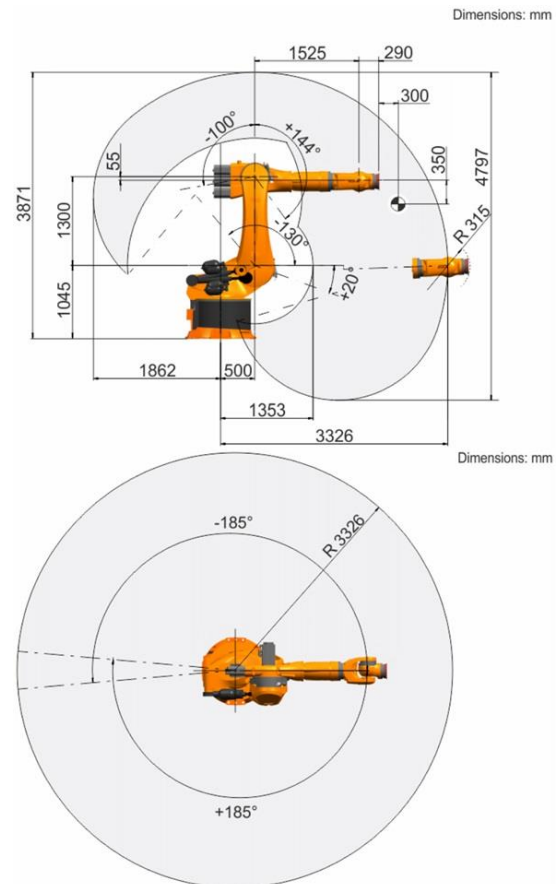
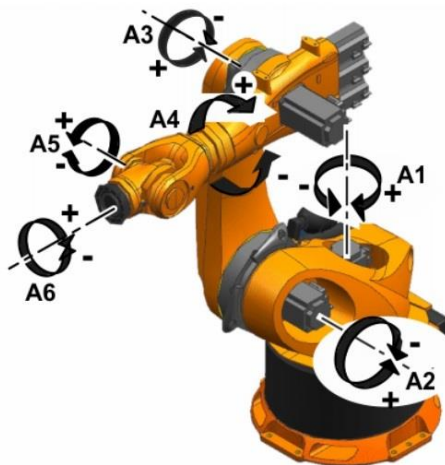


Figura 2. Espacio alcanzable del robot.

Tabla 1. Parámetros DH.

i	a	α	d	θ	offset
1	500	90	1045	q1	0
2	1300	180	0	q2	90
3	-55	-90	0	q3	0
4	0	90	1525	q4	0
5	0	-90	0	q5	0
6	0	0	290	q6	0



Mastering position	
A1	0 °
A2	-90 °
A3	90 °
A4	0 °
A5	0 °
A6	0 °

Figura 3. Ejes de las articulaciones y sentido de giro.

El diagrama de análisis de carga, figura 4, indica qué tan distante puede estar el centro de masa de la carga referente a la herramienta del robot, así, para una carga de 340 Kg, su centro de masa puede estar a 350 mm aproximadamente en los ejes x y y, y a cerca de 395 mm sobre el eje z.

Viendo esto, el robot puede ser usado en cadenas de producción para mover elementos, como lo puede ser en una línea de producción de automóviles mover y ubicar las puertas para su unión al resto del chasis.

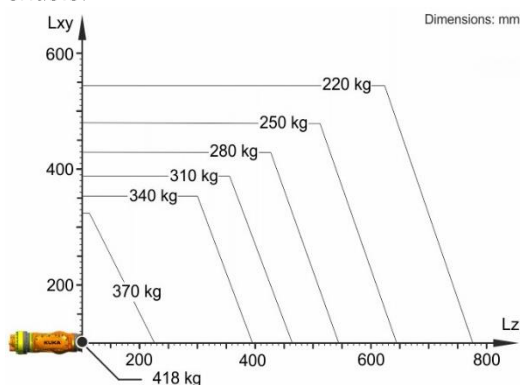


Figura 4. Diagrama de cargas.

2. Modelos geométricos.

Para la obtención de los modelos geométricos mediante los parámetros DH modificados se emplea la información presente en la figura 3 junto con datos provistos en la gráfica del espacio alcanzable, teniendo en cuenta los sentidos de giro establecidos por el fabricante, y obteniendo los parámetros de la tabla 2.

En la figura 5 se ve la representación del robot usando el toolbox de Peter Corke, RVC, mientras en la figura 6 está la misma representación, pero esta vez usando el toolbox de Matlab, RST.

Tabla 2. Parámetros DH modificado.

i	a	α	d	θ	offset
1	0	180	-1045	q1	0
2	500	90	0	q2	0
3	1300	0	0	q3	-90
4	-55	90	-1525	q4	0
5	0	-90	0	q5	0
6	0	90	0	q6	0

Realizar el modelo del robot para ambos toolboxes contó con procedimientos similares. En ambos se crean objetos que representan la geometría de los eslabones, para esto, una lectura de la ficha técnica del manipulador es necesaria para tomar de allí sus dimensiones. Mientras la herramienta propia de MATLAB, el Robotics System Toolbox (de ahora en adelante RST), necesita de varias líneas de código para la creación de el eslabón, pues requiere de la información

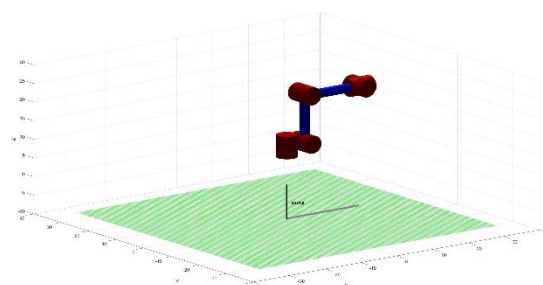


Figura 5. Modelo del robot usando RVC.

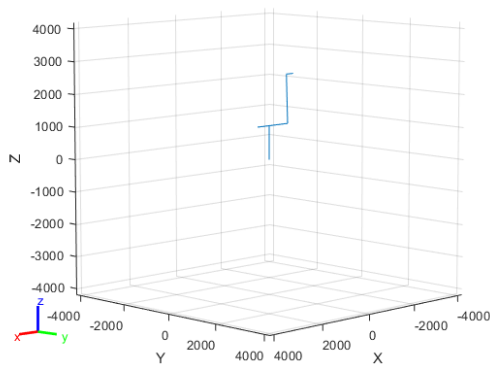


Figura 6. Modelo del robot usando RST.

de su transformación respecto a una base fija, la asociación a un elemento de tipo articulación e incluso un nombre particular para poder ser asociado luego con los demás eslabones de la cadena; el Robotics Toolbox de Peter Corke para MATLAB (de ahora en adelante RVC) emplea una sintaxis más simplificada por medio de la creación de elementos tipo Link que terminan recibiendo toda la información necesaria para definir un eslabón, y además, su articulación asociada. Los elementos creados, los eslabones y articulaciones como cuerpos rígidos para el RST y los Links para el RVC, son reunidos en una sola cadena cinemática a través de métodos distintos, el primero requiere del comando `addBody()` que asocia cada eslabón (hijo) con su igual inmediatamente anterior (padre), dando la posibilidad incluso de crear diferentes ramas en un mismo manipulador; dicha asociación en el RVC se realiza con la función `SerialLink` que simplifica el tratamiento al recibir un simple arreglo con los eslabones de manera independiente. A pesar de sus diferencias, algo es común en ambos, es fundamental para describir apropiadamente el manipulador deseado su análisis de los parámetros de Denavit-Hartenberg modificados (D&H mod), sin estos, se hace mucho más complicado para el RST y casi imposible para el RVC, el cual solicita

explícitamente estos parámetros para poder definir correctamente al manipulador.

3. Modelo geométrico directo

Tras tener el modelo del robot en Matlab usando RVC se generaron el espacio de trabajo teórico del mismo, figura 7, es posible apreciar que si coincide con el provisto por el fabricante.

Para realizar los métodos de la cinemática directa para encontrar la pose del efector final de acuerdo a unos ciertos valores de articulación también es diferente para ambos toolboxes, para el RST, la posibilidad de realizar una imagen con la configuración del robot para una configuración dada se torna poco intuitiva, usando la función del RST `show()` se puede agregar como argumento el robot que se desea visualizar y su configuración, además de otras opciones; precisamente, esta configuración es una estructura que contiene los nombres de la articulación y los valores asociados, de manera que no puede ser configurado sino tan solo entrando en una de las características de la estructura, por medio de la sintaxis `config(i).JointPosition`, algo que no se espera cuando se quiere realizar una modificación rápida a la posición del

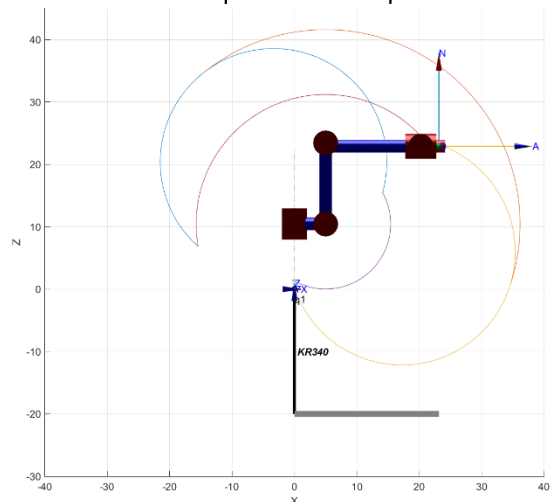


Figura 7. Espacio alcanzable.

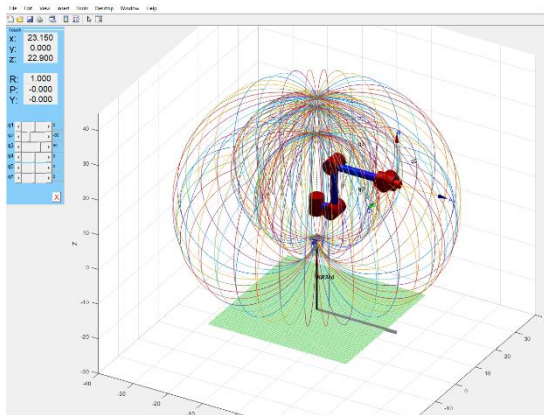


Figura 8. GUI realizada.

manipulador. Por otra parte, el RVC simplifica mucho eso al simplemente recibir un arreglo con los valores de articulación q como argumento de la función `robot.teach()`, es este caso, no solo que muestra visualización del robot en la posición requerida, sino que también despliega unos controles con la posibilidad de modificar en tiempo real estos valores y así observar el robot en movimiento articular.

Adicionalmente, cada toolbox tiene la función propia para la obtención de la matriz de transformación homogénea a partir de una configuración entregada, para el caso del RST, la función es `getTransform()` que solicita el robot, la configuración (estructura con nombres y valores de las articulaciones) y el nombre del elemento donde se quiere analizar dicha matriz, usualmente, el efector final; por el lado del RVC, la función es `robot.fkine()` cuyo único argumento es el arreglo de los valores articulares q .

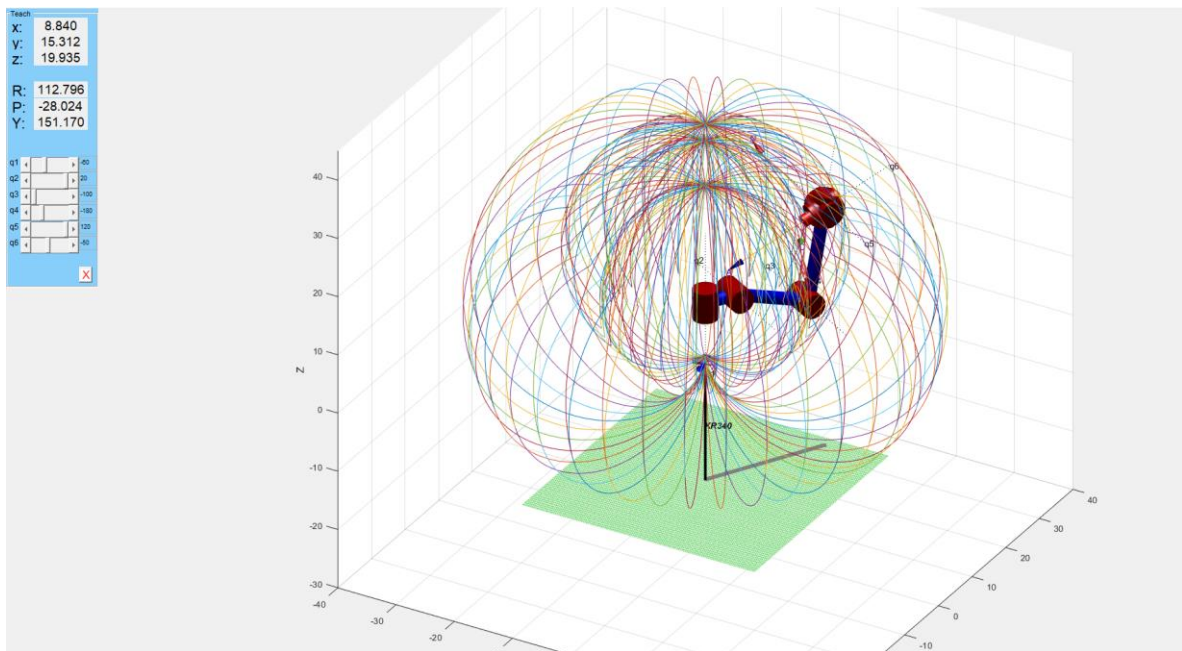
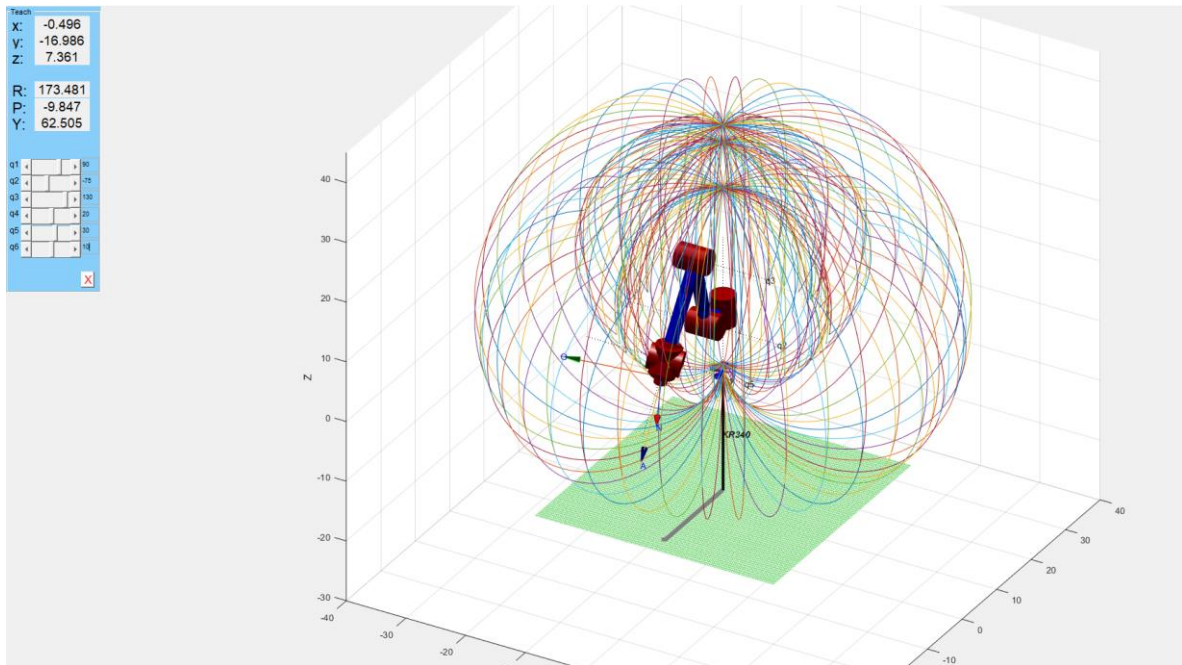
Por último, para el manejo de la interfaz gráfica de usuario, GUI, se abre y se ejecuta el archivo [KUKA_KR340.m](#). Al hacerlo, se muestra una ventana similar a la que se observa en la figura 8. En la parte superior izquierda se ubican, en orden descendente, la orientación del efector referente al origen, luego su representación en ángulos fijos y por último los controles deslizantes para cada una de las articulaciones. Las líneas que se observan indican el espacio alcanzable por la herramienta del robot.

Este trabajo se realizó mediante aportes de los integrantes a un repositorio de Github y que se encuentra en el enlace:

<https://github.com/nicolasgil95/Lab1Robotica>

Allí es posible encontrar algunas de las imágenes acá puestas en mayor tamaño además de encontrar los archivos con el código usado, como el archivo `KUKA_KR340R3330_RST.mlx` donde se encuentra la representación en RST o `RVCmodel.m` que permite la representación en RVC. Se debe resaltar que en los archivos realizados usando RVC las dimensiones se encuentran en dm mientras en los archivos de RST están en mm.

A continuación, se presentan algunos ejemplos de la interfaz donde se varía los ángulos de las articulaciones. Estas imágenes pueden ser encontradas en el git en tamaño original.



Track

X: 19.927
Y: 21.669
Z: 27.843

R: -24.915
P: 6.828
Y: 162.359

q1	4	40.2
q2	4	19
q3	4	26.8
q4	4	210
q5	4	24
q6	4	140

X

