

# **INFORME FINAL**

## **Robótica**

**Alumnos:**

- Pierre Sébastien REINARTZ - V0743 - 5° año
- Víctor SILVA - 10988 - 5° año

## Resumen

En el marco de la materia Robótica 1, llevamos a cabo un proyecto relacionado con el actual contexto mundial de la pandemia. Uno de los conocidos de Victor SILVA, miembro de la profesión licenciatura en enfermería, nos informó de una necesidad. El objetivo de este proyecto era estudiar la cinemática de un brazo robot adaptado para ayudar al personal médico. El sistema se montará en una mesa fija. La idea sería utilizar el brazo robot para realizar varias tareas en la vida diaria de los médicos que se enfrentan a la pandemia: tomar la temperatura, tomar la frecuencia respiratoria.

## Indice

<b>Resumen</b>	<b>1</b>
<b>Indice</b>	<b>2</b>
<b>Introduccion</b>	<b>3</b>
<b>Presentación formal y técnica</b>	<b>4</b>
<b>Denavit y Hartenberg</b>	<b>8</b>
<b>Cinemática Directa</b>	<b>11</b>
<b>Cinemática Inversa</b>	<b>12</b>
<b>Relación de velocidades</b>	<b>15</b>
<b>Planificación y generación de trayectoria</b>	<b>18</b>
<b>Sensores y actuadores</b>	<b>24</b>
<b>Conclusiones</b>	<b>27</b>
<b>Referencias</b>	<b>28</b>
<b>Anexos</b>	<b>29</b>
<b>Adjuntar códigos desarrollados</b>	<b>30</b>

## Introduccion

Debido a la situación actual del mundo creemos que es necesario desde nuestro lugar como futuros ingenieros poder encontrar soluciones que ayuden al bienestar de la sociedad desde un punto de vista tecnológico. Por ello hemos enfocado nuestro proyecto a un brazo robótico, que será usado como un robot de servicio en las guardias de los hospitales. Nos hemos asesorado con una profesional de la salud, enfermera, para poder obtener información al respecto de la situación actual y de las principales falencias del sistema sanitario en este momento. Se pudo detectar que el mayor flujo de personas que asisten al hospital ingresan por la guardia. De las cuales no se tiene conocimiento si están enfermas de COVID-19 o no, ni su grado de urgencia. El problema radica en que el tiempo de espera para ser atendidos es muy extenso, en consecuencia, provoca que los profesionales de la salud queden expuestos al contagio del virus. Además cabe mencionar que la cantidad de personal no es suficiente para realizar estas tareas de forma rápida y efectiva.

Frente a ello proponemos que este robot, "COBOT19" tiene el objetivo de realizar las mediciones de los parámetros que suelen ser tomados por un enfermero en esos casos. Estos parámetros son la temperatura y la frecuencia respiratoria del paciente.

## Presentación formal y técnica

- Robot seleccionado:

Al principio del cursado empecemos el proyecto con la idea que la estructura del robot tendría una configuración geométrica articular con 6 grados de libertad para tener acceso al espacio tridimensional dentro de su espacio de trabajo y respetar la consigna del mínimo de 4 grados de libertad.

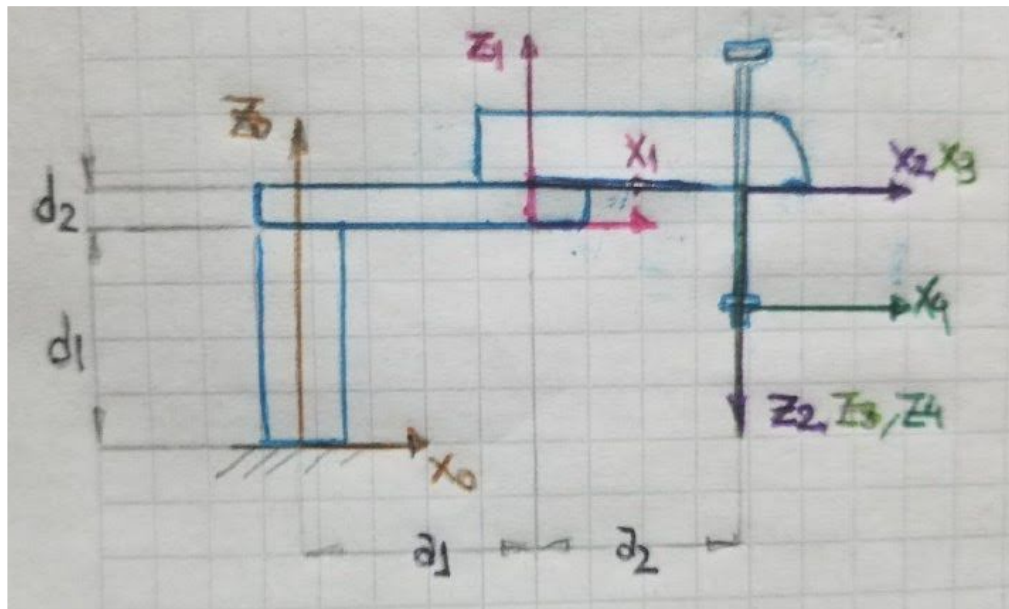
Luego de haber avanzado y profundizado en cuanto a contenidos en las clases, nos hemos dado cuenta de que el robot que habíamos planteado en primera instancia, un robot articular de 6 grados de libertad, tendría grados de libertad de más para las tareas a realizar. Tras mejorar la definición de las tareas planteadas en el trabajo práctico N°1, concluimos que el robot más efectivo para nuestra aplicación sería un SCARA de 4 grados de libertad, debido a que su articulación prismática es fundamental para realizar las tareas. Estaría compuesto por 3 eslabones y usaría actuadores eléctricos.

- Características de la aplicación relacionada con el robot:

- Temperatura: el robot tendrá un termómetro infrarrojo junto a su efector final el cual lo usará para medir la temperatura del paciente.
- Frecuencia respiratoria: el paciente se acostará en una camilla al lado del robot y este pondrá su efector encima del pecho del paciente con un sensor de fuerza para colocarse suavemente y durante 1 minuto para poder medir la Frecuencia Respiratoria con un acelerómetro al subir y bajar en cada respiración.

Todos estos datos serán enviados a una base de datos donde se podrá establecer el orden de urgencia para atender a los pacientes y saber a priori si estos tiene características que podrían coincidir con un infectado de COVID-19.

- Dimensiones:



Estas medidas asignadas a  $d_1$ ,  $d_2$ ,  $a_1$  y  $a_2$ , serían las medidas reales que tendría el robot.

$d_1 = 0,25 \text{ m}$

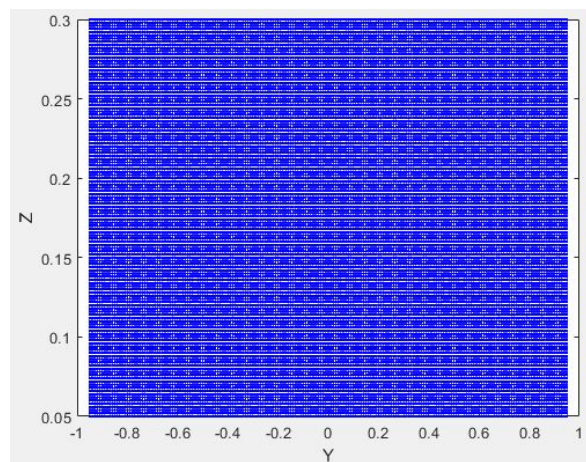
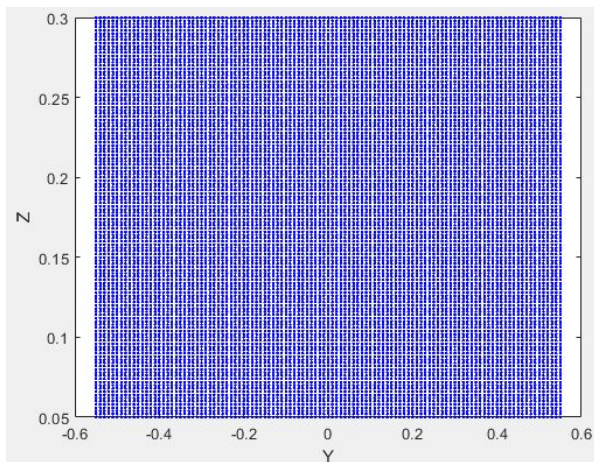
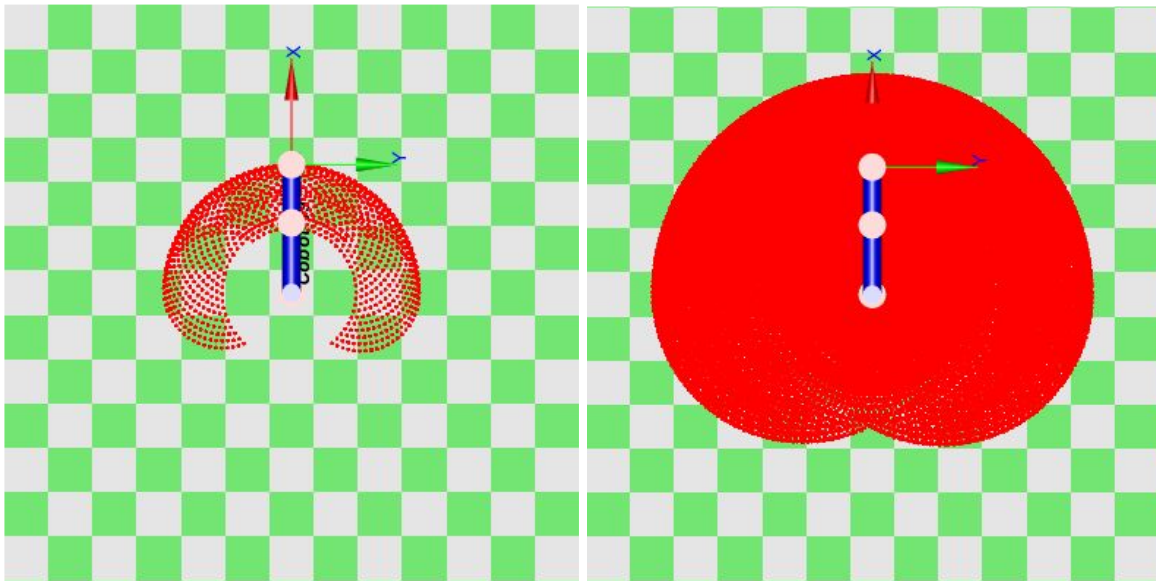
$d_2 = 0,05 \text{ m}$

$a_1 = 0,3 \text{ m}$

$a_2 = 0,25 \text{ m}$

- Espacio de trabajo:

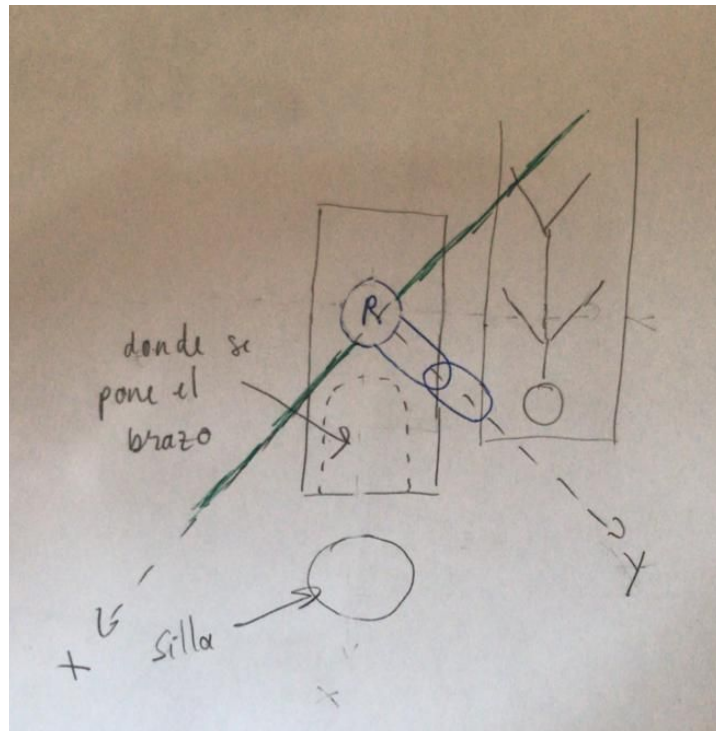
Las dos vistas más representativas del Cobot-19 son la del plano X-Y y la de plano Y-Z. En ambos casos se puede observar a la izquierda sin la herramienta y a la derecha con la herramienta.



Al observar el espacio de trabajo con la herramienta incluida se aprecia que el paciente en la camilla debe tener su abdomen a una distancia máxima de 0.95m con respecto a la base del robot. Para otros pacientes de textura corporal más pequeña el rango de trabajo está cubierto. También se observa que podría colocarse el paciente de un lado de la mesa para la primera medición y luego ir al otro lado de la mesa para la siguiente medición debido a que estaría dentro del espacio de trabajo del robot.

- Distribución elementos en el espacio de trabajo

Considerándolo el robot empotrado a una mesa, tendría de un lado una silla en la cual tomaría la temperatura y del otro lado una camilla donde se debería recostar al paciente luego de la primera medición.

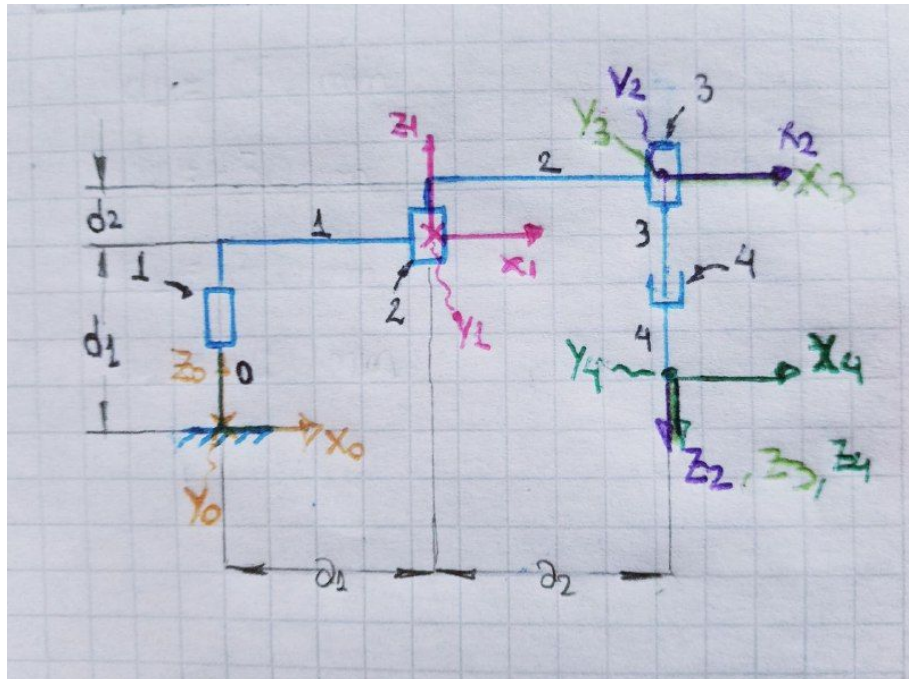




## Denavit y Hartenberg

- Parámetros:

La aplicación de la convención de Denavit Hartenberg nos dio como resultado:



$y_0, y_1$  entrantes,  $y_2, y_3, y_4$  salientes

	$\theta$	$d$	$a$	$\alpha$	$\sigma$
1	$q_1$	0,250	0,300	0	0
2	$q_2$	0,050	0,250	$\pi$	0
3	$q_3$	0	0	0	0
4	0	$q_4$	0	0	1

- Límites articulares, base y tool:

Tras probar varios límites articulares adecuados para la aplicación, elegimos los siguientes:

$$-90^{\circ} \leq q_1 \leq 90^{\circ}$$

$$-120^{\circ} \leq q_2 \leq 120^{\circ}$$

$$-200^{\circ} \leq q_3 \leq 200^{\circ}$$

$$0\text{m} \leq q_4 \leq 0.25\text{m}$$

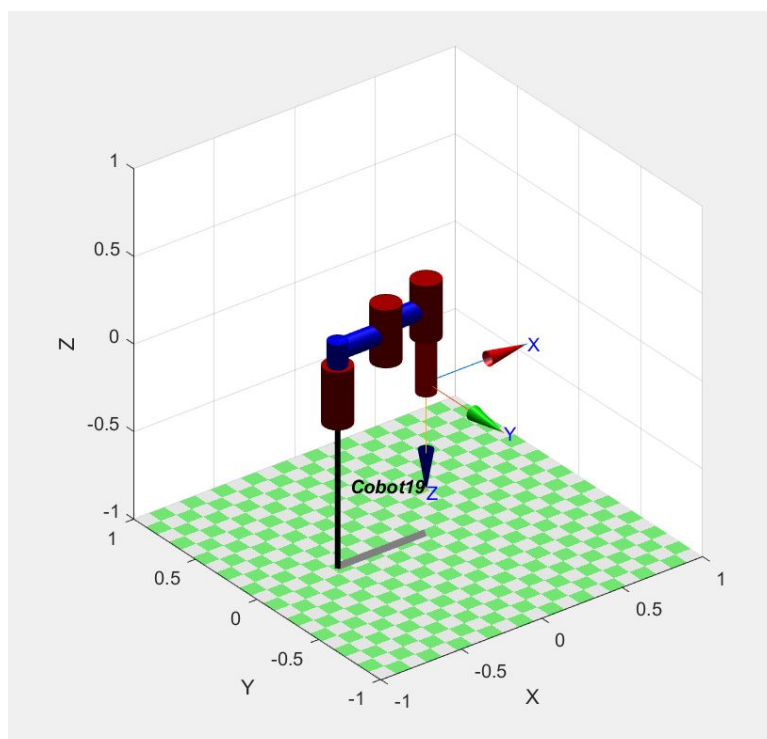
Con  $q_1$  y  $q_2$  se pueden lograr la posición angular adecuada. Los límites articulares de  $q_3$  son suficientes para poder mover la herramienta la cual sería una barra de 40cm con los diferentes sensores para medir la temperatura y frecuencia respiratoria del paciente. Y los límites de  $q_4$  permiten que la articulación prismática baje (eje z positivo hacia abajo) hasta estar cerca de la muñeca de la cual se medirá la temperatura.

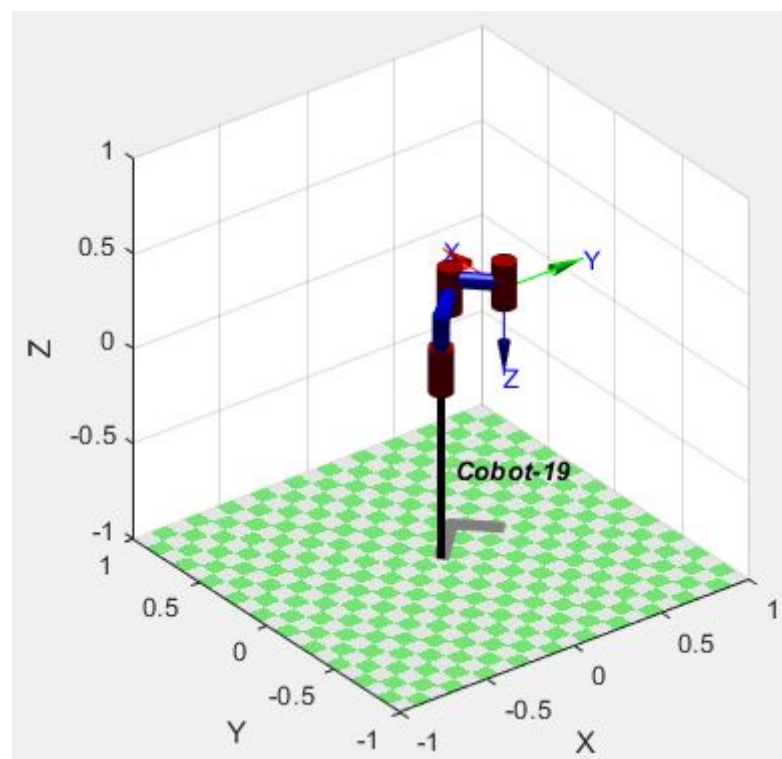
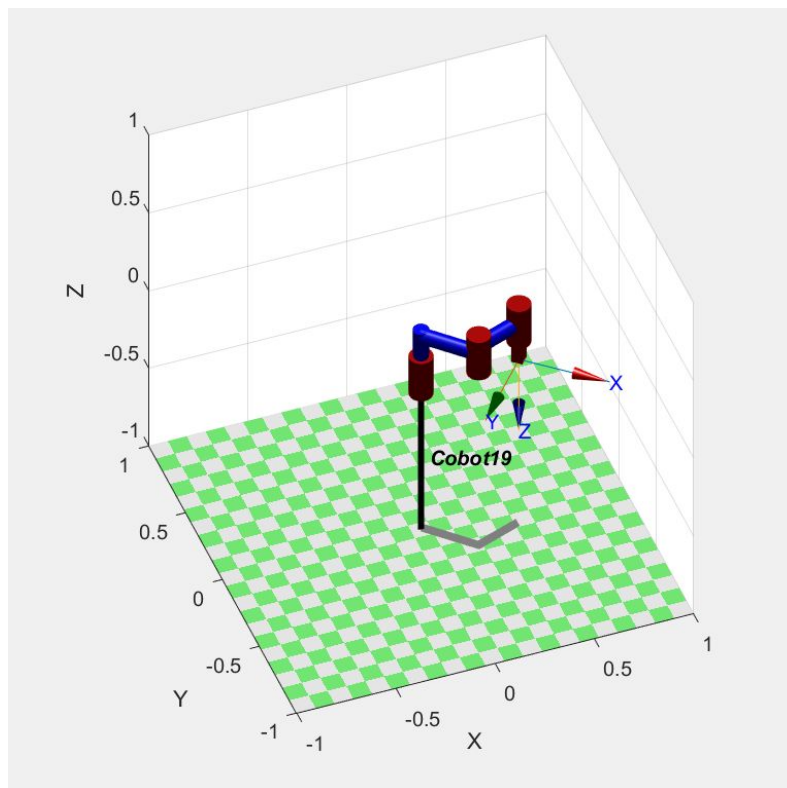
```
R.base = transl(-0.5,0,0);  
R.tool = transl(0.400, 0, 0);
```

Por ejemplo se puede usar una base desplazado de -0.5 m para visualizar de forma más cómoda al robot pero por la continuación del TP hemos decidido que la base sería el origen del espacio de trabajo.

En cuanto a la tool se usó una traslación 0.4 m representarla debido a que la herramienta es una barra de 40cm con los sensores para nuestra aplicación.

- plot del robot en varias posiciones:





## Cinemática Directa

La función de cinemática directa que se desarrolló fue creada para nuestro robot en particular. Le pasamos un vector articular, los parámetros de DH y los límites articulares, y además de la cinemática directa, devuelve un booleano (dentro), para informar si la postura está dentro del espacio de trabajo del COBOT19 y en caso de que no esté se le hacen correcciones hasta que esté dentro del espacio de trabajo más cercano. Como se observa en la siguiente imagen.

```
q =
    0    0    0    0
```

```
limites =
    -2.0944    2.0944
    -2.0944    2.0944
    -3.4907    3.4907
         0    0.2500
```

Está DENTRO del espacio de trabajo

```
E =
    1.0000         0         0    0.4500
         0   -1.0000         0         0
         0         0   -1.0000    0.2700
         0         0         0    1.0000
```

```
dentro =
    1
```

```
q =
         0         0         0    0.3000
```

```
limites =
    -2.0944    2.0944
    -2.0944    2.0944
    -3.4907    3.4907
         0    0.2500
```

Está FUERA del espacio de trabajo

```
E =
    1.0000         0         0    0.4500
         0   -1.0000         0         0
         0         0   -1.0000    0.0200
         0         0 |         0    1.0000
```

```
dentro =
    0
```

En la siguiente imagen se observa que nuestra función de cinemática directa obtiene el mismo resultado que si usáramos fkine.

El vector articular de ejemplo es:

```
qf =
    1.3090   -0.5236         0    0.1000
```

La matriz obtenida con fkine es:

```
Tf =
    0.7071    0.7071         0    0.2544
    0.7071   -0.7071         0    0.4666
         0         0        -1    0.17
         0         0         0     1
```

La matriz obtenida con nuestra función de CD sin la herramienta es:

```
T =
    0.7071    0.7071         0    0.2544
    0.7071   -0.7071         0    0.4666
         0         0   -1.0000    0.1700
         0         0         0    1.0000
```

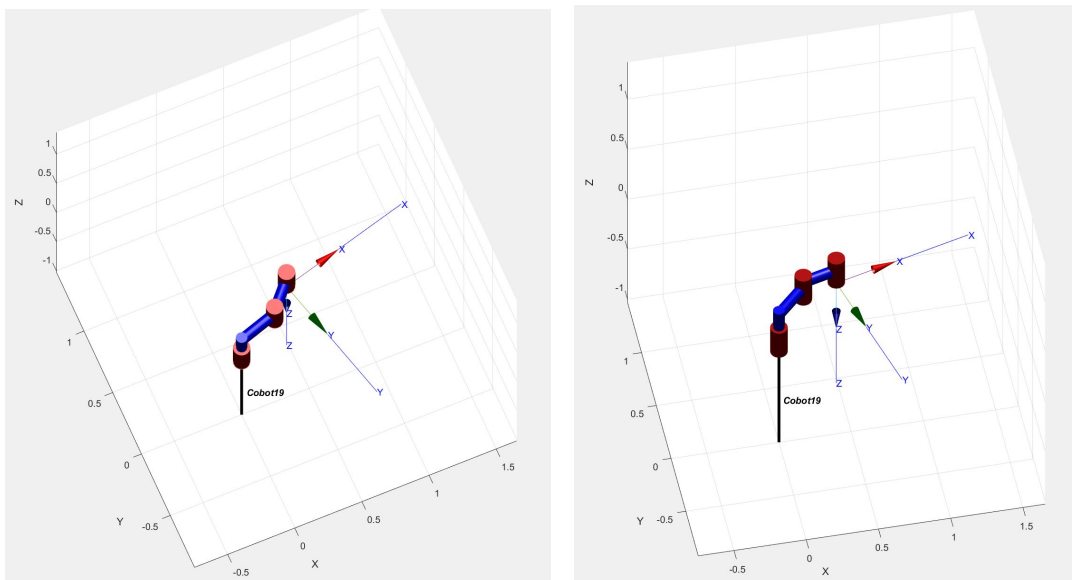
## Cinemática Inversa

Para la cinemática inversa de nuestro robot, un SCARA de 4 grados de libertad desarrollamos una función específica, la cual recibe como parámetros de entrada una matriz homogénea  $TT$  (obtenida a partir de un  $qz$  arbitrario usando la cinemática directa),  $dh$  que son los parámetros de Denavit Hartenberg,  $rqlim$  que establece los límites articulares y  $qstart$  que es la postura actual del robot.

Debido a la morfología de un SCARA, para una posición en el espacio de trabajo siempre existe 2 vectores articulares posible como solución (codo a la derecha, codo a la izquierda). Así que esta función devuelve las dos soluciones posibles ( $q$ : codo derecha,  $qb$ : codo izquierda) y da la información sobre la solución más cercana  $qsol$ .

Además usamos un booleano  $ET$  para informar si la posición inicialmente querida está dentro o fuera del espacio de trabajo. En caso que esté afuera, los valores de las articulaciones que están superiores a los límites articulares correspondientes toman dichos valores límites. Así la función no devuelve una solución exacta sino la solución más cercana que está dentro del espacio de trabajo.

- Al elegir un  $qz$  arbitrario dentro del espacio de trabajo se obtuvieron los siguientes resultados  $q$  y  $qb$ :



Por cada solución se ve el cambio de postura hasta coincidir con la posición y orientación deseadas representada por el frame de color azul.

```

qz =

    0.7854    -0.5236         0    0.1000

Las posibles soluciones son:
q =

    0.3105    0.5236    0.5723    0.1000

qb =

    0.7854    -0.5236   -0.0000    0.1000

==> La solución más cercana es q:    0.3105    0.5236    0.5723    0.1000

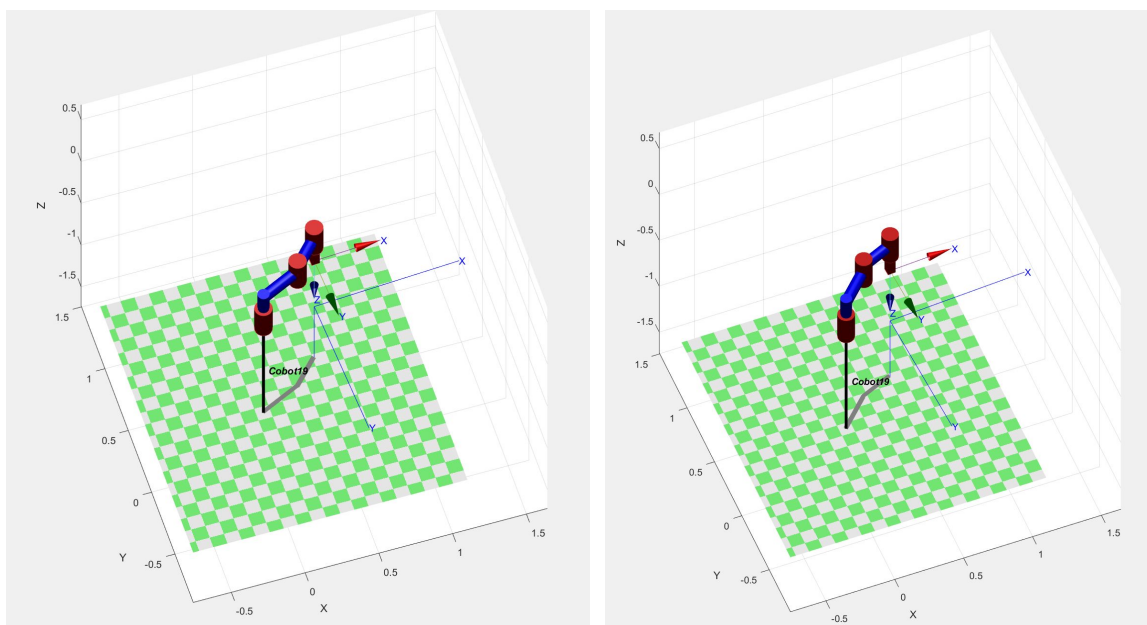
==> ADENTRO DEL ESPACIO DE TRABAJO

Verificación por la solución con ikine:
p =

    0.3105    0.5236    0.5723    0.1000
    
```

Hemos implementado la función de manera que la solución más cercana es la que menos desplazamientos angulares requiere para llegar a la solución, tomando en cuenta la posición inicial del robot.

- Al elegir un  $qz$  arbitrario fuera del espacio de trabajo se obtuvieron los siguientes resultados  $q$  y  $qb$ :



```
qz =  
    0.4387    0.3816    0.7655    0.7952  
Las posibles soluciones son:  
q =  
    0.4387    0.3816    0.7655    0.2500  
  
qb =  
    0.7852   -0.3816    0.3488    0.2500  
  
==> La solución más cercana es q:    0.4387    0.3816    0.7655    0.2500  
  
==> FUERA DEL ESPACIO DE TRABAJO  
  
Verificacion por la solución con ikine:  
p =  
    0.7852   -0.3816    0.3488    0.7952
```

En este ejemplo se muestra que el  $qz$  aleatorio tomó un valor de la articulación prismática  $q4$  fuera de los límites articulares (0 a 0,25m). Por lo tanto la función devuelve 2 soluciones en las cuales se asignaron el valor máximo posible a  $q4$ .



## Relación de velocidades

- Puntos singulares hallados:

Para encontrar los puntos singulares de COBOT19 hemos trabajado de forma simbólica para encontrar el Jacobiano y su determinante.

```
rS =

noname:: 4 axis, RRRP, stdDH, slowRNE, Symbolic
+---+-----+-----+-----+-----+-----+
| j |      theta |      d |      a |      alpha |      offset |
+---+-----+-----+-----+-----+-----+
| 1|      q1|      d1|      a1|      0|      0|
| 2|      q2|      d2|      a2|      pi|      0|
| 3|      q3|      d3|      0|      0|      0|
| 4|      0|      q4|      0|      0|      0|
+---+-----+-----+-----+-----+-----+

JSr =

[- a2*sin(q1 + q2) - a1*sin(q1), -a2*sin(q1 + q2), 0, 0]
[ a2*cos(q1 + q2) + a1*cos(q1), a2*cos(q1 + q2), 0, 0]
[ 0, 0, 0, -1]
[ 1, 1, -1, 0]

DJSr =

-a1*a2*sin(q2)

q2_singular =

0
```

Para poder encontrar su determinante, el Jacobiano debe ser cuadrado. Así reducimos la matriz a una 4x4 (sacando las líneas 4 y 5 que corresponden a rotaciones que no nos interesan) y calculemos el determinante simbólico, que luego simplificamos para obtener su expresión matemática.



Mediante la función “solve(DJrs==0,q2)” se detecta una singularidad. Cómo el determinante obtenido fue:

$$DJrs = -a1 * a2 * \sin(q2)$$

Se observa que cada vez que  $q2 = n * \pi$  nos indicaría que habrá una singularidad (para  $q2 = 0, \pi, 2\pi, \dots$ ).

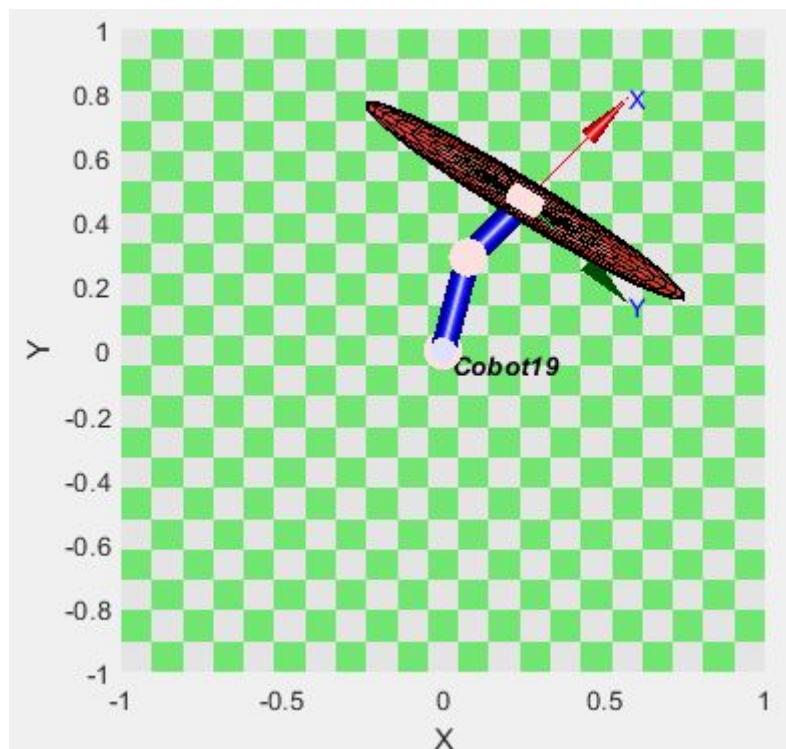
- Comentarios sobre la aplicación que estén relacionados con los puntos:

Según nuestra aplicación el robot no trabaja cerca de los límites de su espacio de trabajo por lo tanto estaremos lejos de esos puntos singulares.

Únicamente cuando el valor de la articulación  $q2$  sea igual a 0, cuando cambia de una posición codo arriba a codo abajo y viceversa, pasaría por un punto singular.

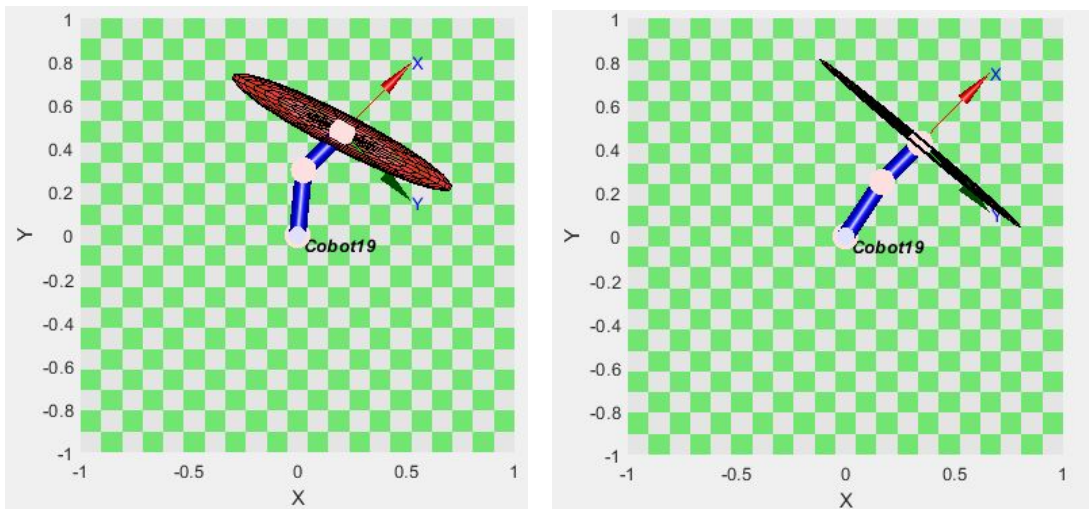
Usaremos la postura en la que el robot medirá la frecuencia cardiaca, y donde el paciente se recostaría sobre una camilla. Se eligió esta postura codo izquierda porque el robot cuando vaya la siguiente postura disminuirá  $q2$ , por lo que se alejará aún más de un punto singular, de modo de que nunca tenga que pasar por uno.

- Plot de la elipsoide:

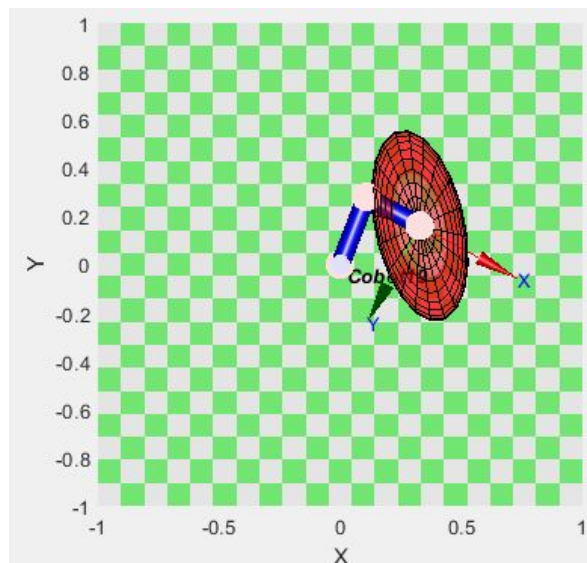


La elipse de manipulabilidad obtenida en esta postura nos muestra que si el robot quisiera moverse en la dirección de  $x$  puede desarrollar una velocidad baja, a diferencia que si se moviera en la dirección de  $y$ .

A continuación se observa como cambia la elipse para algunas posturas cercanas a la de trabajo:



La siguiente tarea a realizar se debería hacer a 90 grados en sentido horario con respecto a la anterior, por lo que si disminuimos  $q_2$ , a continuación se ve como cambia la elipse:

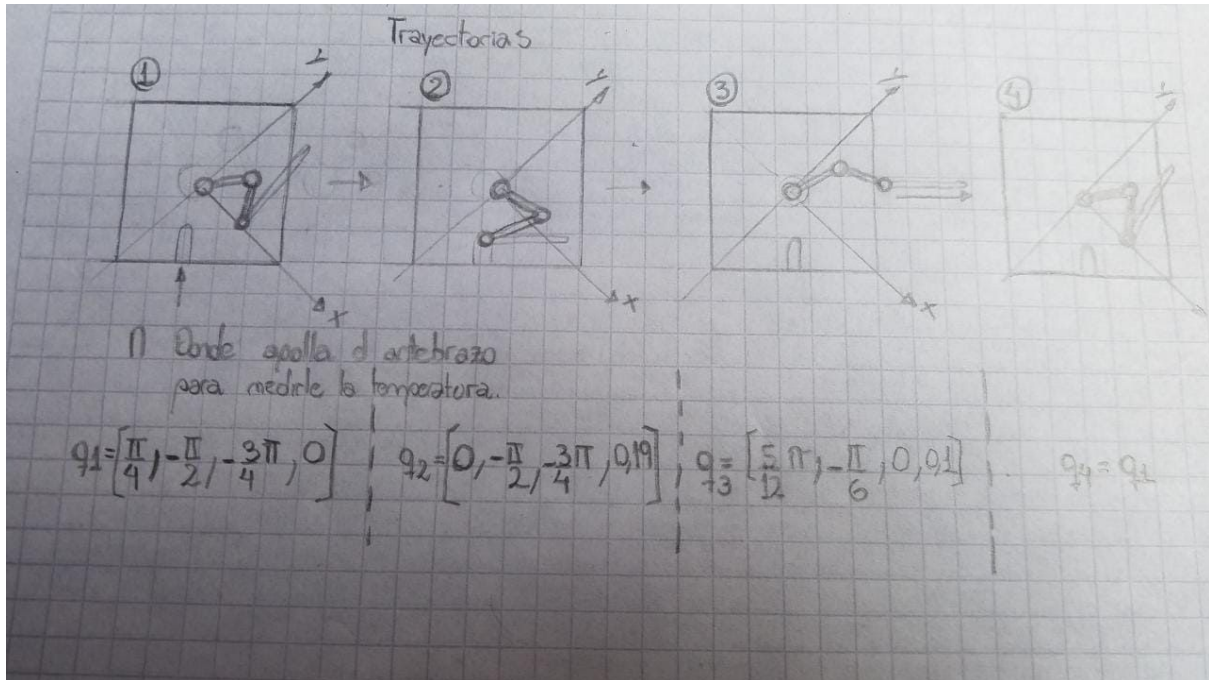


Esto nos permite que el extremo pueda desarrollar mayores velocidades tanto en  $x$  como en  $y$ , además está lejos de los puntos singulares del límite de espacio de trabajo.

## Planificación y generación de trayectoria

- Planificación:

Según nuestra aplicación COBOT19 comenzará el brazo retraído, luego se moverá en sentido horario para tomar la temperatura del paciente, una vez realizada esta tarea se moverá en sentido antihorario mientras el brazo se estira hasta posicionarse para medir la frecuencia respiratoria. Finalmente volverá a la posición inicial.



### Puntos

- $q_0 = [\pi/4, -\pi/2, -3\pi/4, 0]$ ; % posición articular inicial y final del robot
- $q_t = [0, -\pi/2, -3\pi/4, 0.19]$ ; % posición articular para toma de temp
- $q_r = [5\pi/12, -\pi/6, 0, 0.1]$ ; % posición articular para toma de freq respi

El espacio de trabajo está comprendido entre los límites articulares de nuestro robot siendo estos:

$$\begin{aligned} -90^\circ &\leq q_1 \leq 90^\circ \\ -120^\circ &\leq q_2 \leq 120^\circ \\ -200^\circ &\leq q_3 \leq 200^\circ \\ 0m &\leq q_4 \leq 0.25m \end{aligned}$$

Tipo de Trayectoria: Asíncrona (simultánea).

Tiempo/velocidad/aceleración: En nuestra aplicación las aceleraciones y velocidades no son un factor de importancia. Solo hay que cuidar que las velocidades no sean peligrosas para el paciente.

Requisitos : Un requisito específico del COBOT19 es la precisión en los puntos de trabajo y la suavidad.

- Generación de trayectoria:

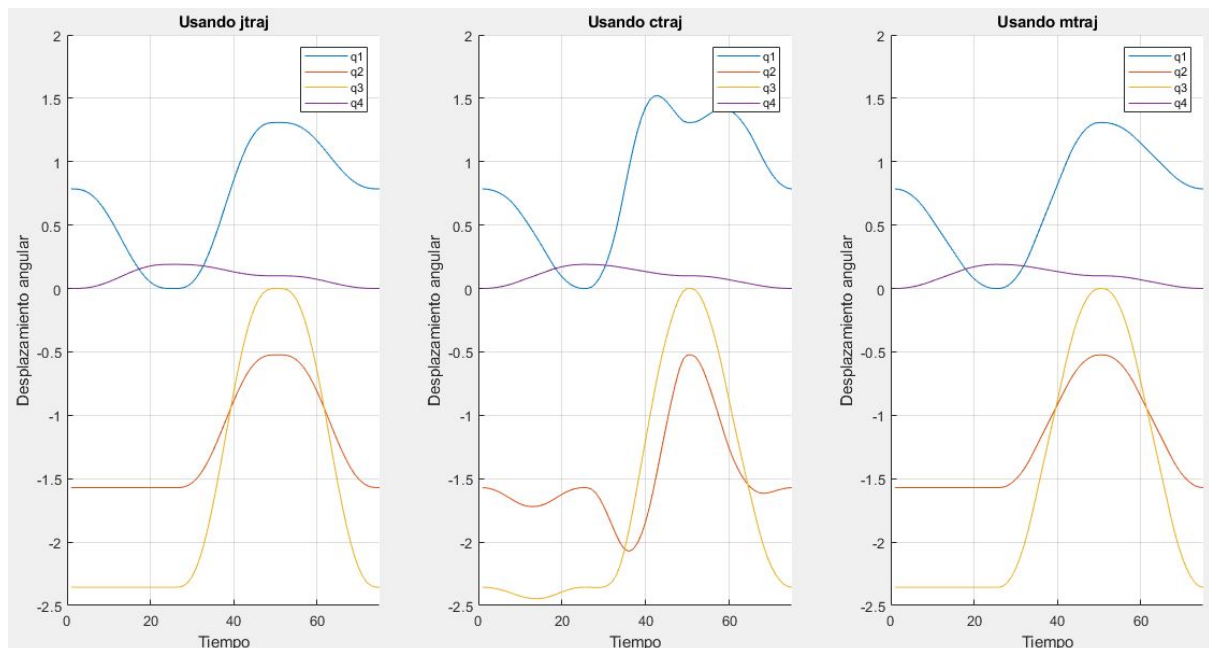
Se implementaron y compararon 3 métodos de interpolación diferentes para la elección final de generación de trayectoria:

Interpolación articular: Hicimos uso de la función `jtraj` para crear trayectorias interpolando entre cada posiciones articular de trabajo obtenidas gracias a la cinemática directa para pasar del espacio cartesiano a articular.

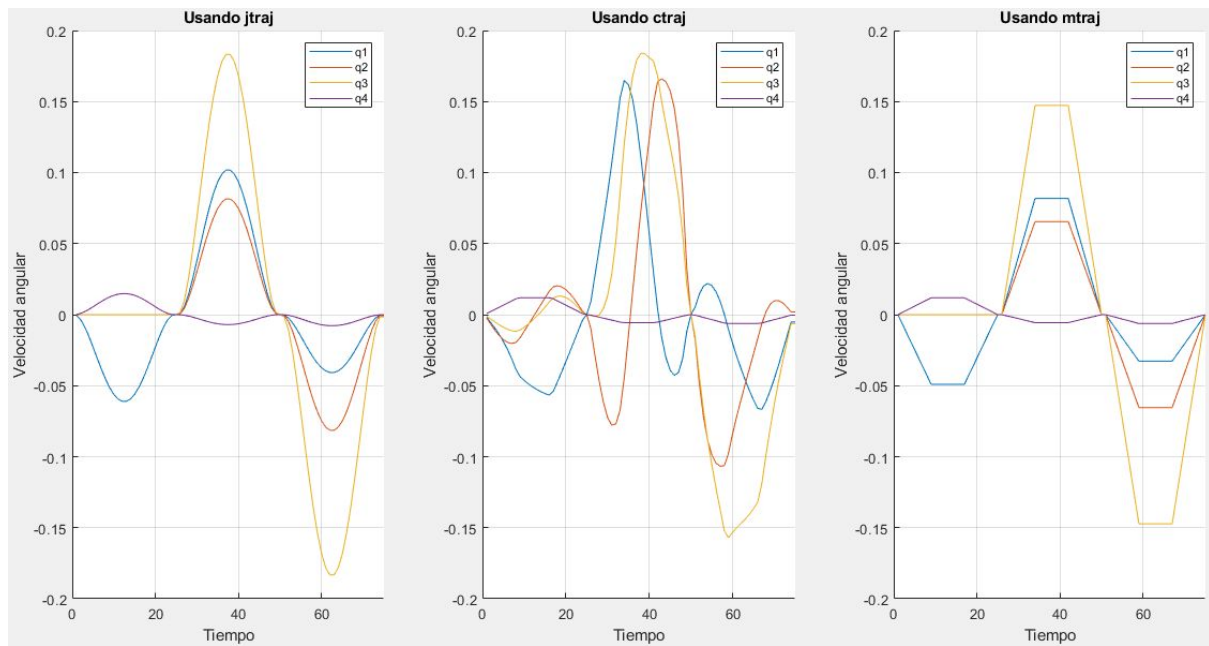
Interpolación cartesiana: De la misma manera usamos la función `ctrj` para crear trayectorias interpolando entre puntos de trabajo en el espacio cartesiano.

Interpolación multidimensional: Probamos una tercera opción con la función `mtraj` que permite crear trayectorias entre las posiciones articulares de trabajo.

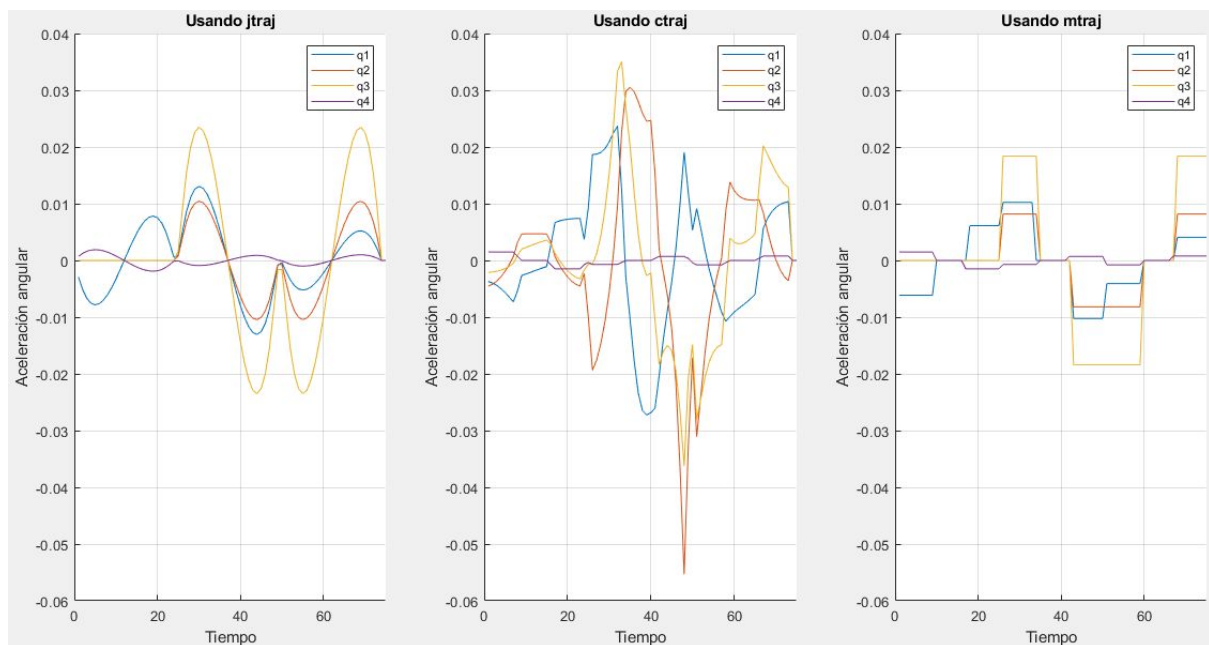
Las gráficas comparativas del desplazamiento, velocidad y aceleración angular entre los tres métodos son:



(Desplazamiento angular)



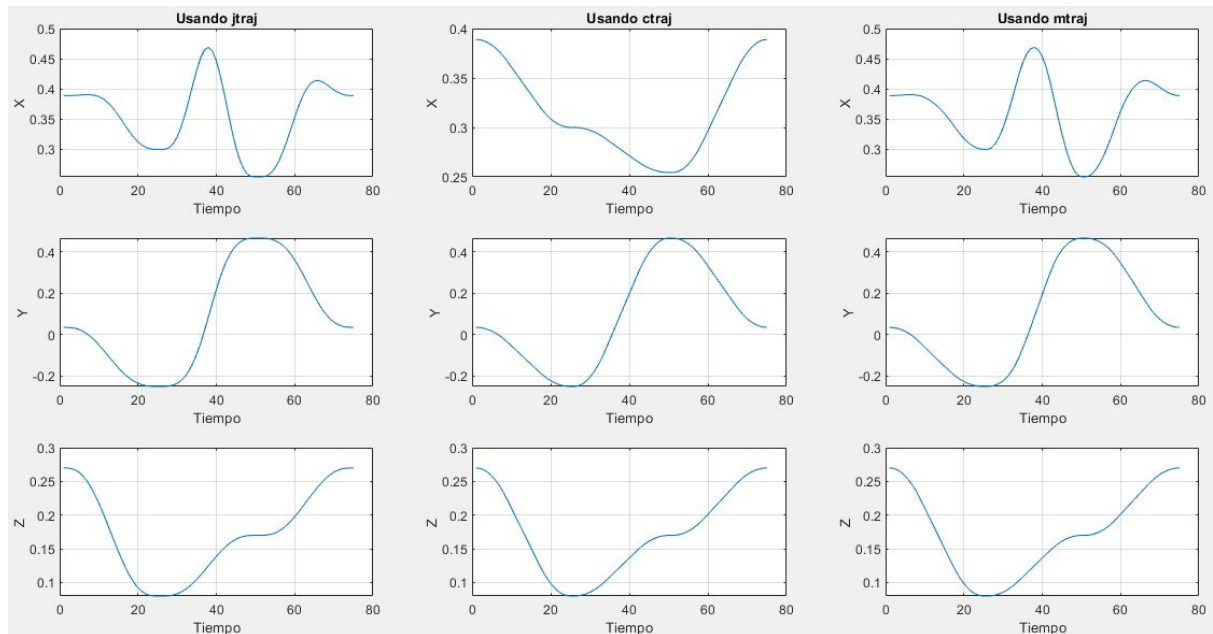
(Velocidad angular)



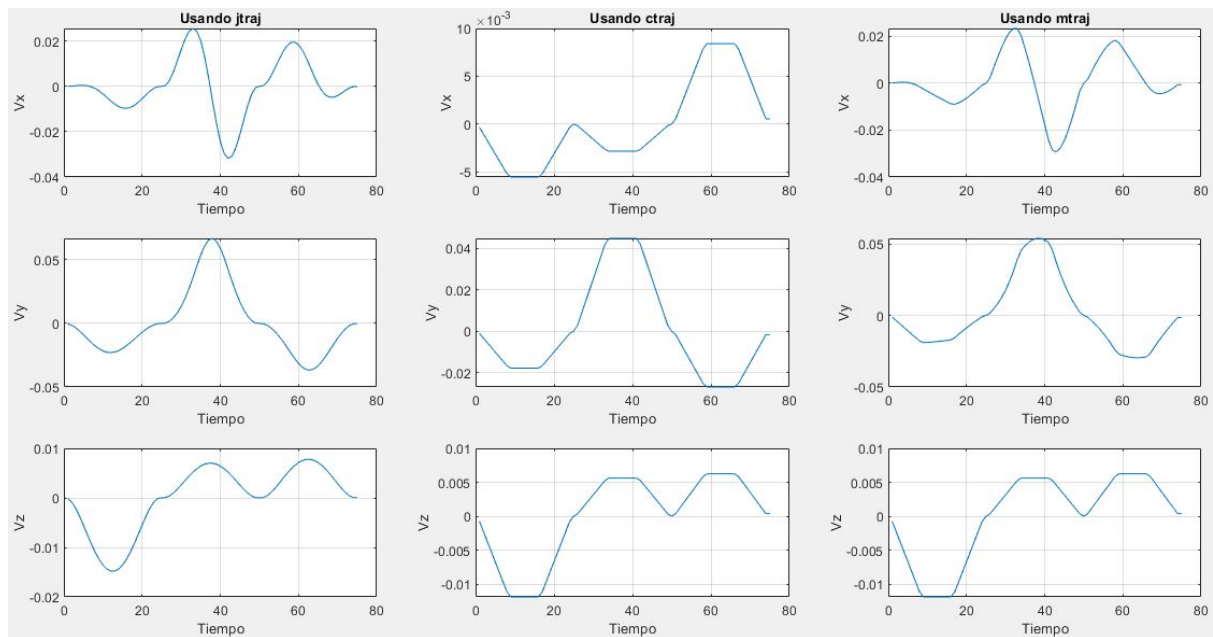
(Aceleración angular)



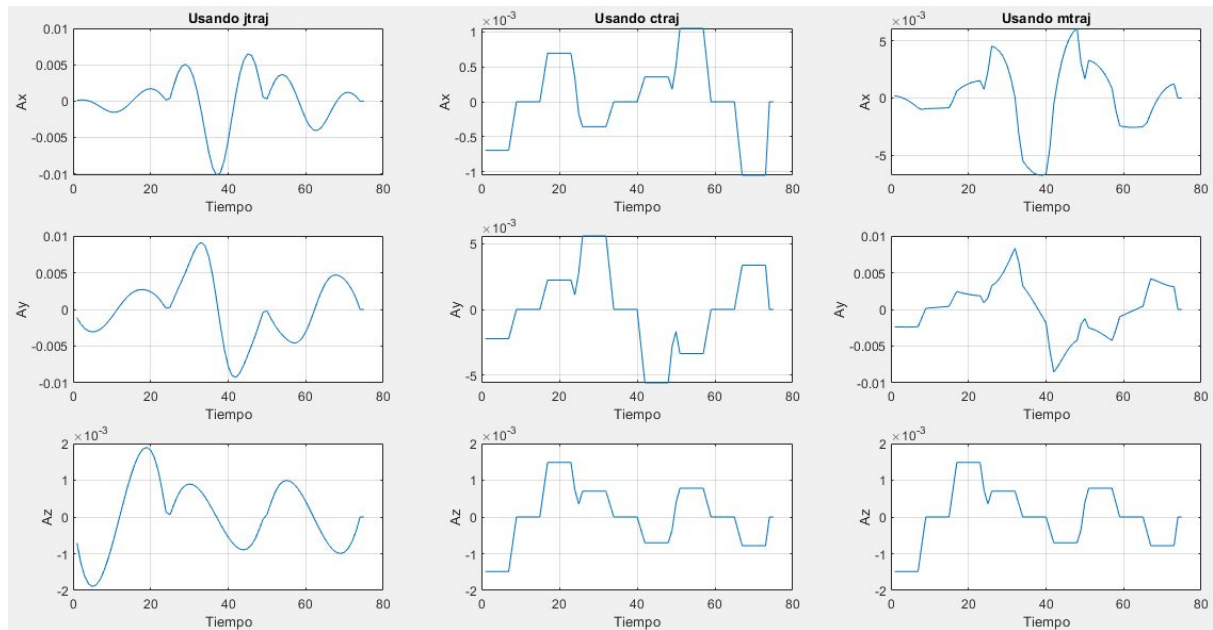
Las gráficas comparativas del desplazamiento, velocidad y aceleración angular entre los tres métodos son:



(Desplazamiento)

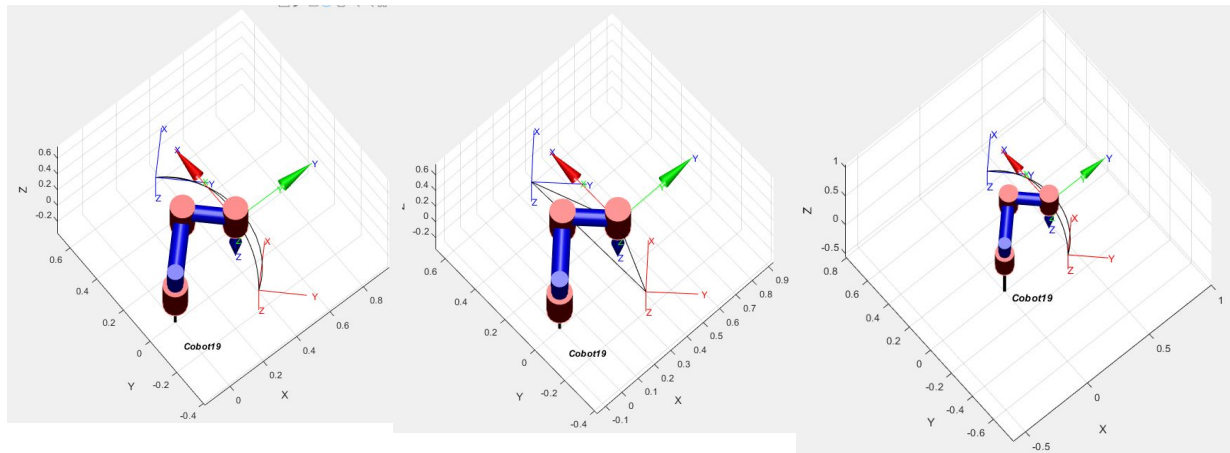


(Velocidad)



(Aceleración)

Las trayectorias realizadas para jtraj, ctraj y mtraj son respectivamente las siguientes:



Por lo tanto, se obtuvieron las posiciones para x, y, z en el espacio de trabajo para cada caso y en la simulación se observan las distintas trayectorias 3D que realizó con los distintos métodos.

Al comparar las gráficas se observa que todos los casos cuando la abscisa es 25 y 50 todas las curvas pasan por esas raíces, esto es debido a que en esos puntos se realizan las tareas de tomar la temperatura (abscisa 25) y medir la frecuencia cardiaca (abscisa 50).

En cuanto al desplazamiento angular las curvas obtenidas con jtraj y mtraj son muy similares, en el caso de ctraj tiene algunas diferencias pero nada a resaltar.

En cuanto a la velocidad angular en el caso de ctraj vemos que se someten las articulaciones  $q_1$ ,  $q_2$  y  $q_3$  altas velocidades mientras que en los otros casos solo la articulación  $q_1$  tiene una velocidad angular grandes mientras que en las otras son menores. Se puede apreciar que en el caso de mtraj tiene una forma más trapezoidal las curvas a diferencia de las de jtraj.

tiene un cambio muy repentino de velocidades en  $q_3$ , esto es debido al problema anterior, mientras que en el caso de mtraj son casi cero las velocidad, por último en el caso del jtraj son mayores que en los demás casos pero son movimientos más continuos.

Y en cuanto a la aceleración angular ctraj necesitará cambios repentinos de aceleración, similar a lo que le ocurre a mtraj, por lo que el método que tiene un comportamiento más continuo es jtraj.

Por lo tanto, al realizar esta comparación entre los 3, para nuestra tarea el mejor interpolador es jtraj.



## Sensores y actuadores

- Actuadores

Servo motores: para las articulaciones 1 y 2 usaremos estos actuadores debido a que se requiere mayor torque para mover la estructura del robot. Para ello usaremos los motores Easy Servo Motor Nema 34 4N.m . Estos motores tienen una entrada para un encoder. Además se usará un driver 2HSS86H para cada uno. Trabaja con 36VDC.



Para las articulaciones 3 y 4 usaremos servomotores pero de menor torque que los anteriores debido a que no es necesario para la aplicación. Se ha optado por los Easy Servo Motor Nema 23 2N.m. Cada motor tendrá su propio driver 2HSS57. En el caso de la articulación 3 le permitirá girar, mientras que en el caso de la articulación 4 (prismática) tendremos un sistema de sinfin para producir el movimiento traslacional. También funcionará con 36VDC.



- Sensores

Internos:

Estos sensores son los que usaremos para el correcto funcionamiento de nuestro robot. Osea encoders para detectar la posición en cada momento. Las articulaciones en su actuador tienen un encoder incremental óptico de 1000 líneas.

Externos:

Estos son los sensores que usaremos para realizar las tareas de nuestra aplicación.

- Sensor de temperatura: para esto usaremos un sensor del tipo láser con el que se pueda medir la temperatura del cuerpo de una persona sin necesidad de hacer contacto físico con ella. Un sensor útil para esta tarea sería el Mlx90614 Gy 906, el cual tiene un tamaño reducido y liviano, además puede medir hasta 10 cm de distancia, pero en nuestro caso trabajaremos a 5 cm la medición.



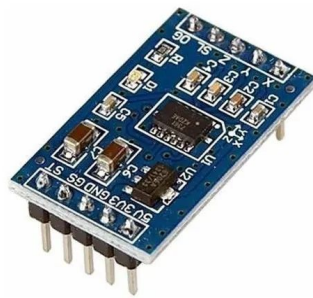
- Sensor de ultrasonido: usado para determinar llegar a la distancia necesaria para que el sensor de temperatura pueda tomar la medición correctamente. Para ello se usaría el sensor HC-SR04, el cual al detectar una distancia de 5cm emitirá una señal que hará que paré la articulación prismática del robot.



- Sensor piezoeléctrico: tendrá el objetivo de detectar que el contacto de herramienta con el paciente sea correcto. Su tamaño y peso son muy bajos.



- Sensor acelerómetro: montado en el otro extremo de la herramienta para la medición de la frecuencia respiratoria del paciente, que al respirar medirá la cantidad de veces que respiró el paciente en un minuto. El sensor a usar será el Mma7361, el cual es liviano y pequeño.



- Fuente de alimentación: Esta debe ser de 36VDC y 40A para que tenga la capacidad de poder alimentar a todos los servomotores si se movieran en simultáneo. También necesita un regulador de voltaje para trabajar con 5V para la parte lógica y de señales donde se maneja la información obtenida por los sensores.



## Conclusiones

Este proyecto nos permitió combinar la competencia teórica y "práctica" mediante el uso de Matlab y de la RTB de Peter Corke.

Trabajar a distancia y con tiempo limitado no siempre fue fácil, pero trabajar en equipo fue realmente beneficioso. Por lo tanto, pudimos mejorar nuestras habilidades en robótica, gestión de proyectos, distribución de tareas y colaboración a lo largo del proyecto. El haber trabajado con Matlab también nos permitió utilizar un software aplicado muy usado en el mundo de la robótica.

Por lo tanto, en este proyecto hemos abordado muchos aspectos de la cinemática en la robótica y de las aplicaciones industriales más comunes.

Debido a la pandemia, desafortunadamente no pudimos probar temas más orientados a la manipulación, sin embargo esperamos continuar este proyecto en un futuro cercano hacia un estudio dinámico e idealmente prototipo.

## Referencias

Bibliografía recomendada por la cátedra

<https://www.mathworks.com/>

Aula abierta - Robótica

## Anexos

<https://datasheetspdf.com/pdf/774392/Melexis/MLX90614/1>

<http://raspoid.com/download/datasheet/HCSR04>

<http://www.datasheet.es/PDF/900699/MMA7361-pdf.html>

## Adjuntar códigos desarrollados

El código principal y todas las funciones desarrolladas se encuentran en un comprimido el cual está adjunto con la entrega de este informe.