



Robótica

Trabajo práctico FINAL-ROBOT

Nombre	Correo Electrónico
Tomas contessi	tcontessi@fi.uba.ar

Índice:

- Descripción del proyecto.
- Especificaciones deseadas.
- Elección de componentes.
- Diseño de partes mecánicas.
- Simulación y obtención de modelos.
- Revisión de las especificaciones.
- Construcción del prototipo.
- Diseño e implementación del software.
- Pruebas sobre el prototipo y verificación de las especificaciones deseadas.
- Lista de componentes.
- Cosas por mejorar.

Descripción del proyecto:

Se desea construir un modelo de robot serie de 4 grados de libertad capaz de realizar tareas del tipo pick and place.

Si bien existen modelos de robot y controladores de este estilo en internet, en este proyecto se diseñarán e implementarán tanto el hardware como el software necesario. Los modelos y el software resultantes serán subidos a un repositorio de GitHub para que cualquiera que lo desee pueda acceder a ellos y continuar el desarrollo o construir su prototipo.

Este proyecto utiliza Solid Works para el diseño de piezas mecánicas, simulación y obtención de características físicas, Arduino como plataforma de desarrollo para hacer de interfaz entre la pc y los motores y Python como lenguaje de programación para el controlador.

Especificaciones deseadas:

Se desea que el prototipo de robot cumpla las siguientes especificaciones:

- Grados de libertad: 4.
- Máxima extensión: 250mm.
- Carga máxima en la brida a máxima extensión: 400g.
- Precisión: $\pm 1\text{mm}$.
- Repetitividad: $\pm 1\text{mm}$.
- Velocidad: 100°/seg.
- Peso del robot: 1kg.
- No se pondrán frenos mecánicos, solo los motores.

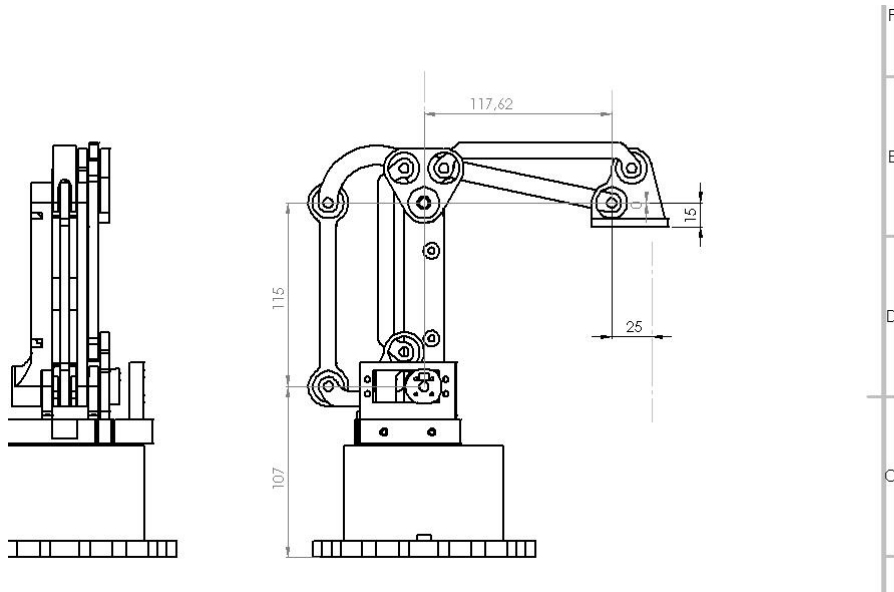
Elección de componentes:

Para la construcción del modelo se eligieron los componentes siguiendo un criterio de precio/calidad y simplicidad, eligiéndose los siguientes:

- Motor para los ejes 1,2,3: servomotor Mg946r.
- Motor para el eje 4: servomotor Sg90.
- Placa de desarrollo: Arduino Nano.
- Rodamientos: rulemanes 6262RS
- Rodamiento eje 1: rulemán 30202JR o 30305JR (se utilizó este en el modelo fabricado).
- Material de fabricación de las partes: PLA.

Diseño de las partes mecánicas:

El diseño del robot tomo inspiración de modelos como el IRB460 de ABB, ya que ese tipo de diseño maximiza la carga útil al poner los motores montados sobre el eje 1 y da la posibilidad de utilizar solo ejes giratorios, a diferencia de modelos como el SCARA. Aunque también tiene la contra de que es un diseño mecánicamente complejo, teniendo los motores de los ejes 2 y 3 montados de manera concéntrica y haciendo uso de un sistema de trapecios (en mi diseño se cambió por romboides) que garantiza que el eje de la brida siempre se mantenga paralelo al suelo.

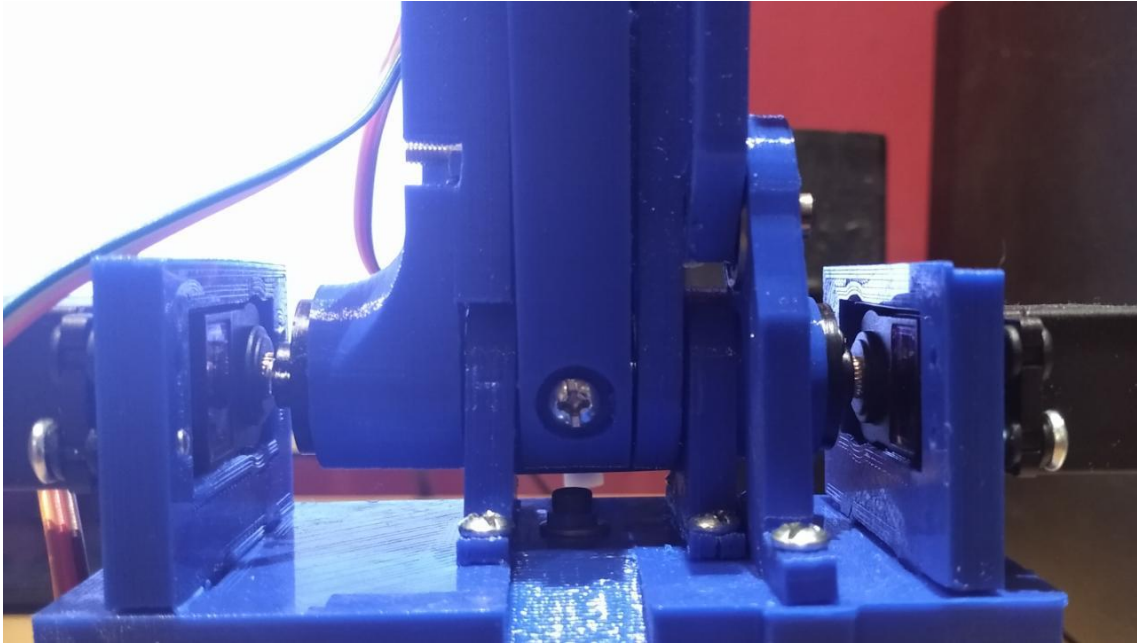


Como software de diseño se utilizó el Solid Works ya que permite diseñar piezas mecánicas de manera intuitiva y precisa.

Al diseñar el robot se buscaba que este sea capaz de plegarse sobre sí mismo, lo cual termino decidiendo gran parte de la forma de los segmentos 2 y 3.

El sistema de trapecios fue reemplazado por un sistema de romboides ya que resulta más simple el diseño aunque se pierde algo de rango de movimiento.

Los ejes de los motores 2 y 3 se alinearon de manera concéntrica montando todo el sistema sobre rodamientos y haciendo el eje 2 del robot parte del eje del motor 3, de esta manera se mantiene la alineación de los motores y a su vez se transfiere la potencia a ambos ejes sin afectar al otro.

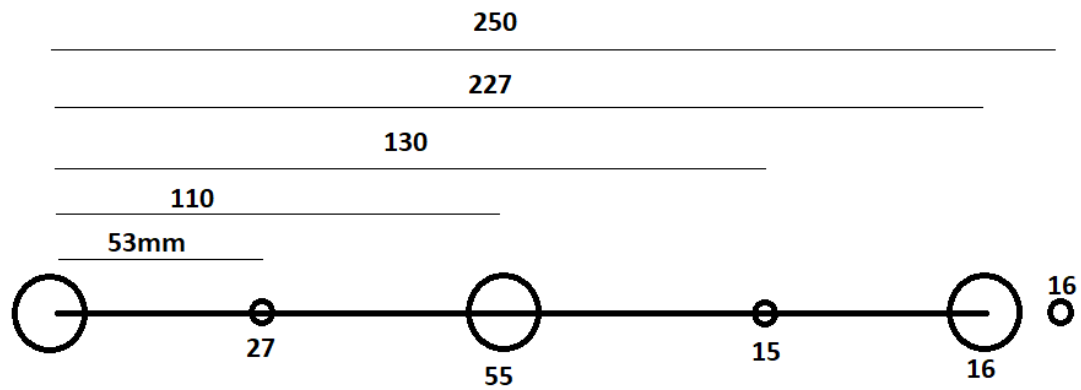


Además, como medida para mejorar la estabilidad, precisión y la resistencia del prototipo se decidió montar todo sobre rodamientos, incluido el eje 1, para el que se usó un rodamiento cónico.

Simulación y obtención de modelos:

Al tener los diseños de las piezas terminados se procedió a realizar un montaje de estas usando la herramienta de ensamble del programa Solid Works, utilizando esta herramienta se simularon los movimientos del robot y se verificó que las piezas encastran y trabajen juntas como se esperaba y que el robot cumpliera las especificaciones de alcance máximo deseadas.

Utilizando el mismo programa se obtuvo la posición del centro de masa de las piezas del brazo y con el programa ultimakerCura, el cual es el que prepara el código para imprimir en 3D las partes se obtuvo el peso estimado que tendrían las partes ya impresas (con una densidad de relleno del 7%). Con esta información se calculó el torque que precisa el motor del eje 1 solo para levantar el robot (este motor es el que limita la capacidad de carga en este diseño).

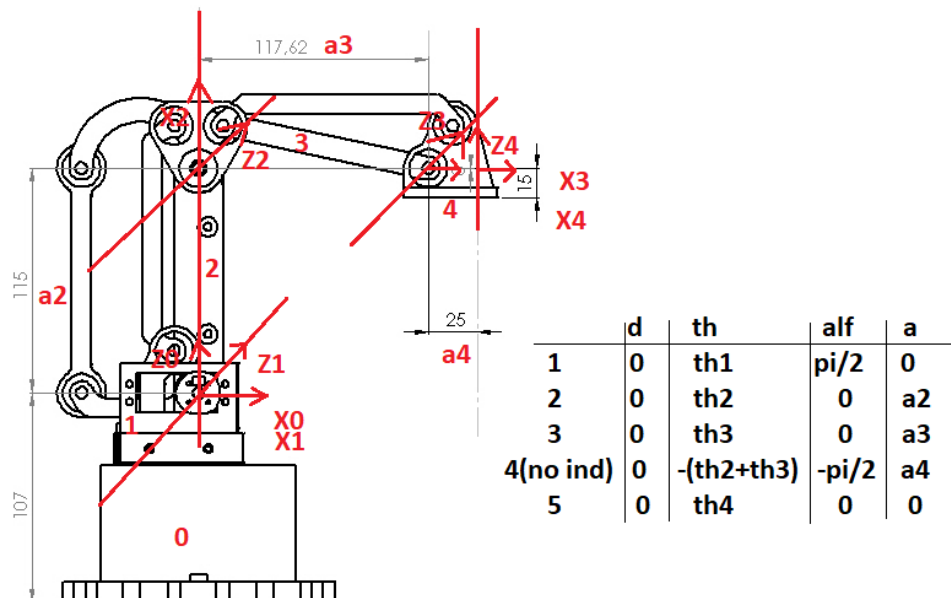


129g toal

xcm=132mm

T=1.7Kg.Cm de torque perdido

Además, teniendo el modelo digital ensamblado se tomaron los parámetros de DH para realizar el modelo cinemático del robot.



Nótese que, si bien el robot tiene 4 grados de libertad y 4 motores, en realidad tiene 5 ejes, solo que uno de los ejes no es independiente.

Además, se imprimió una pieza de 10cm de largo para usarse en la medición del torque de los motores y se corroboró que a la tensión de trabajo estos tenían más de 12kg.cm de torque. (según su datasheet tienen entre 10 y 13kg.cm dependiendo de la tensión)



Revisión de las especificaciones:

Una vez que se tuvo un modelo virtual del robot se procedió a ver si este diseño sería capaz de cumplir con las especificaciones planteadas.

- Del ensamble del modelo se verificó que el robot lograba alcanzar los 250mm en su máxima extensión.
- Al restar el torque necesario para mover el brazo del torque medido del motor se obtiene entonces un torque de 10.3kg.cm y dividiendo esto por 25cm obtenemos una capacidad de carga máxima de 412g.
- Según su datasheet los motores tienen una velocidad de 300º/seg en vacío, lo cual a priori cumpliría la especificación de velocidad, al menos en vacío.

Construcción del prototipo:

El prototipo fue impreso en 3D en PLA, con una altura de capa de 0.15mm y una densidad de relleno del 7%.

Durante la construcción se decidió reemplazar el rodamiento cónico 30202JR por el 30305JR, ya que este último si bien es más caro, el radio interno permite acomodar mejor el eje del motor y al ser más grande proporciona un mejor apoyo a toda la estructura.

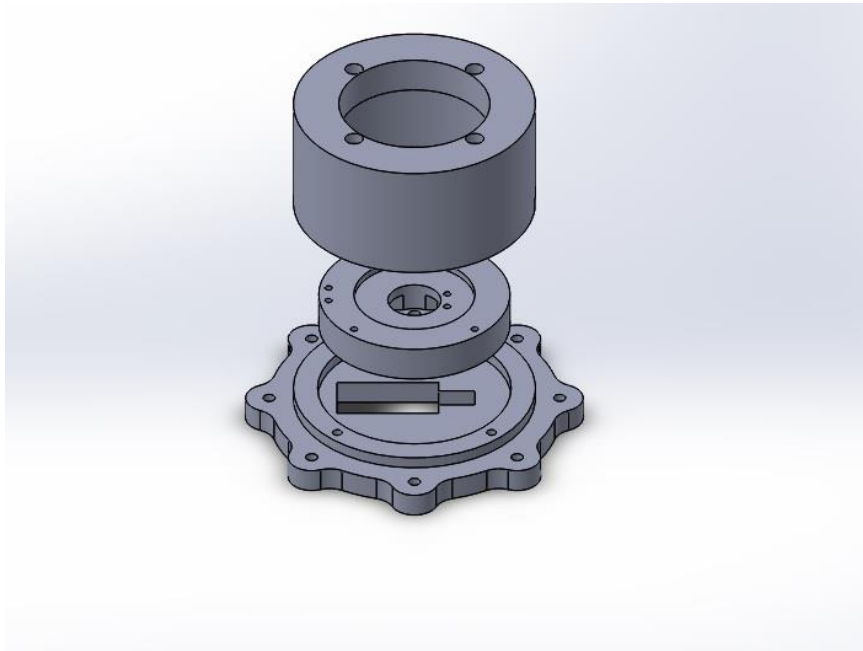
Para la colocación de los motores es recomendable hacer un programa que los haga apuntar a 90º y ponerlos de forma que los 90º queden a la mitad del movimiento de la articulación, igualmente el soft permite cargar como atributo el offset de la colocación de los motores, el

cual es la diferencia entre la posición en la que fue colocado el motor y la posición a la que apunta el segmento del robot según los parámetros de DH.

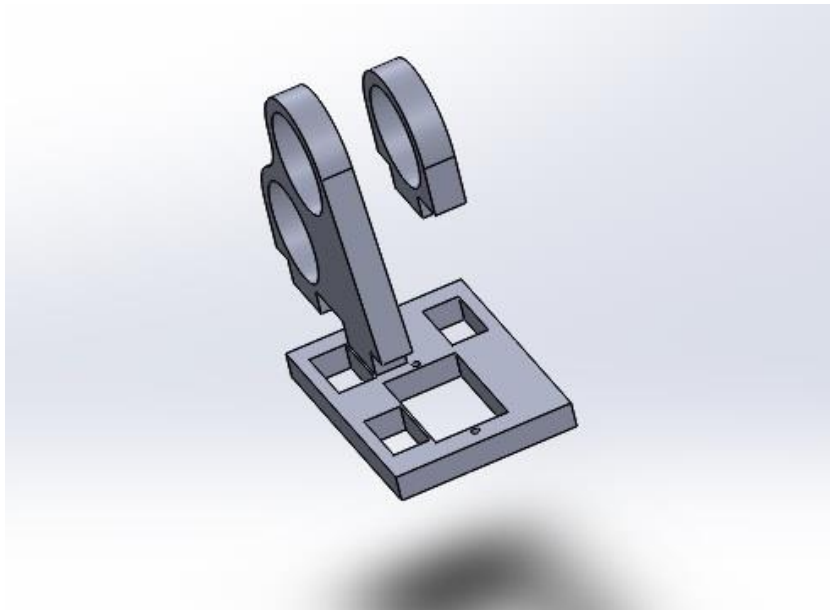
Si bien se dejaron encastrados para colocar tuercas en la plataforma del eje1, el robot fue diseñado para que no sean necesarias y los tornillos rosquen sobre el plástico facilitando el armado. Estos alojamientos para tuercas son por si se falsea la pieza, para que no haya que reemplazarla.

La forma correcta de ensamblar el modelo es:

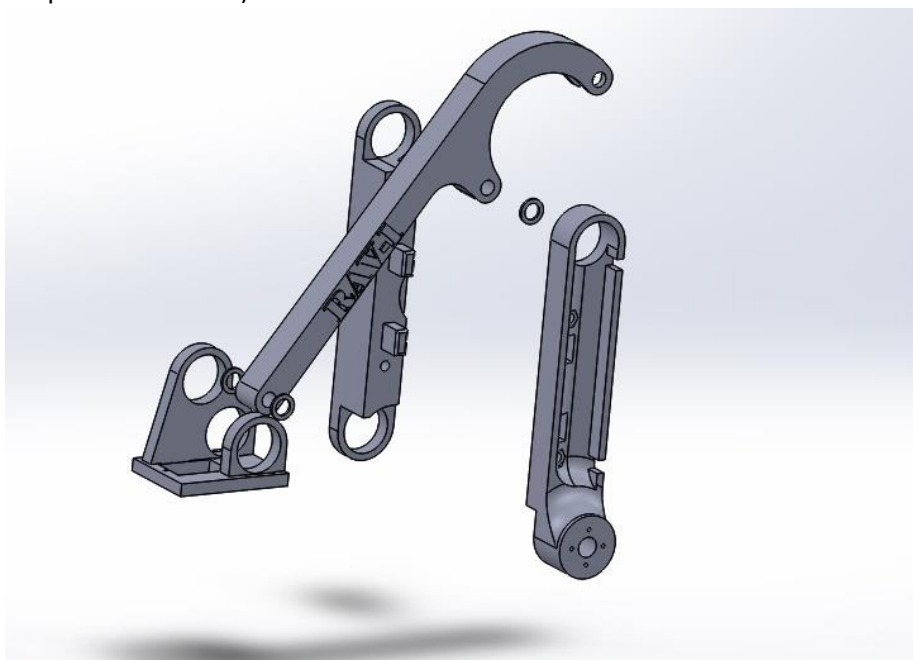
1. Colocar todos los rodamientos en sus alojamientos y verificar que calcen derechos (entran a presión, posiblemente se requiera un tubo de 25mm para poner la pista interna del rulemán cónico en el eje1), también colocar y atornillar los acoples para los motores en sus respectivos encastrados en los ejes, está diseñado para los acoles redondos de 22mm que vienen con los servos).
2. Colocar el motor del eje 1 y ensamblar la base:



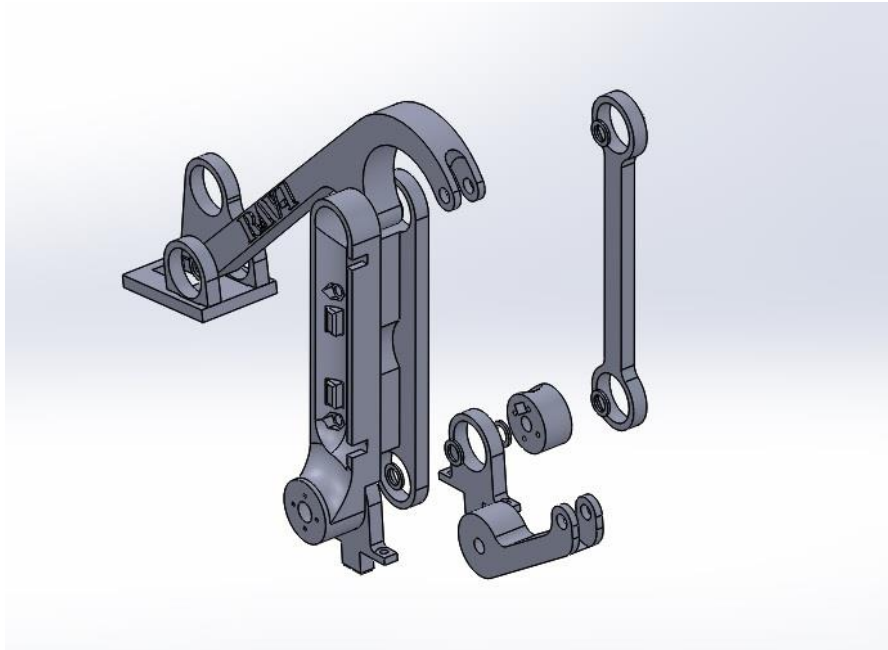
3. Ensamblar el segmento 4 del brazo que se pega con cualquier pegamento tipo la gotita o pegamentos para plásticos. Es recomendable colocar en este paso el motor del eje 4 ya que después será muy difícil con el eje 3 puesto:



4. Ensamblar el eje 2 y montar juntos los ejes 2,3 y 4 (los separadores son para evitar juegos en el robot, aunque puede que dificulten en armado, si se complica mucho también hay unos separadores abiertos con forma de zegger que se pueden colocar después de armado):

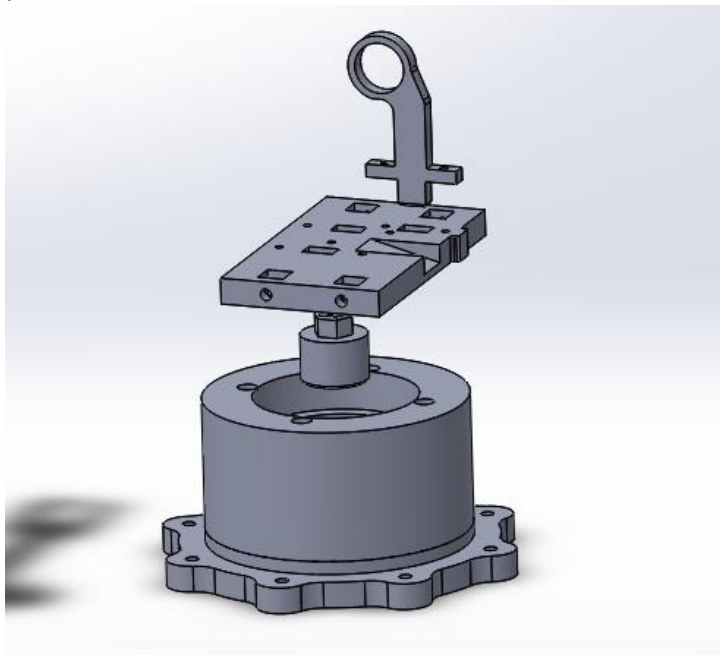


5. Colocar el eje en la pieza tractora del eje 3 y montar la articulación del eje 2, luego también montar la leva que transfiere el movimiento de la palanca al eje 3:



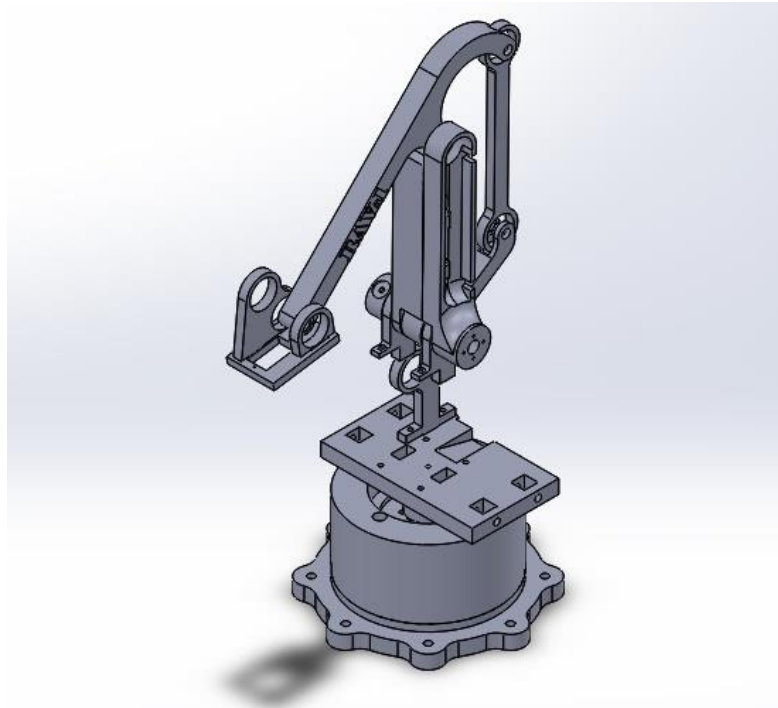
(nota, me acabo de dar cuenta que sería más fácil armar el segmento 2 en esta parte y apretar todo junto, pero estoy describiendo el orden el que lo pensé y arme yo)

6. Colocar el soporte de las paralelas en la plataforma, luego colocar el tractor con el centro del rulemán cónico en el hexágono y ensamblar el rulemán encastrando el tractor en el eje, asegurar toda la pieza atornillando con un tornillo de 40mm dese la plataforma hasta el motor:

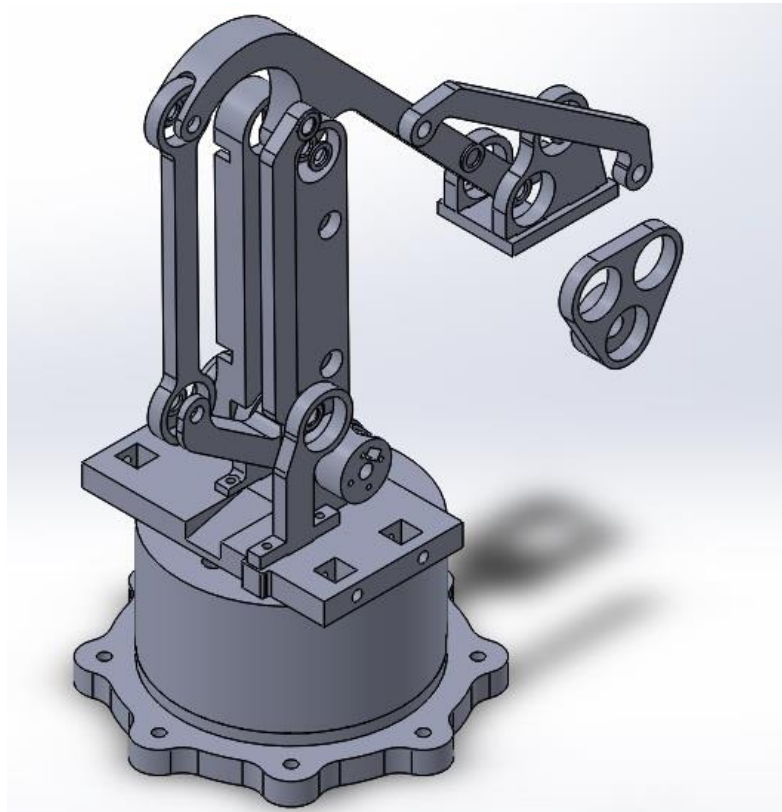


7. Luego atornillar los soportes del eje 2 a la plataforma, es posible que haya que aflojar un poco el soporte de las paralelas o forzarlo un poco porque entra muy justo. En este punto verificar que el eje 2 quede lo más paralelo con la plataforma posible (si

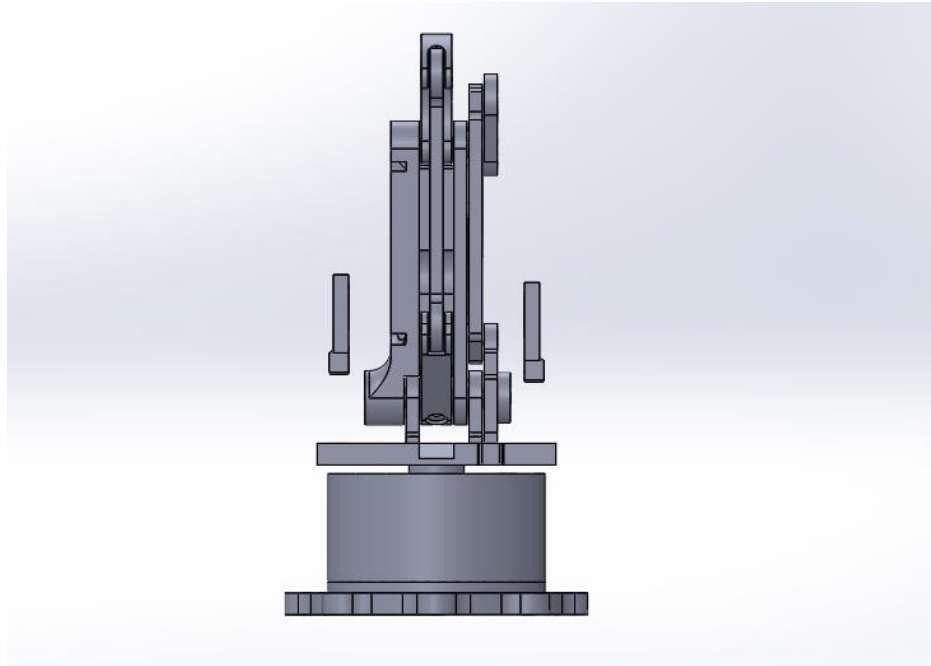
quedaron rebarras o imperfecciones de impresión se pueden corregir con un cuchillo caliente), ya que de quedar desnivelado los motores no quedaran alineados:



8. Colocar las varillas de paralelismo (la de 110mm es la que va con el segmento 2 y la de 117mm es la que va con el 3) y luego el triángulo que las une. Es recomendable asegurar el armado con seguros zegger para que los ejes no se salgan de los rulemanes porque calzan sin juego, pero no apretados.



9. Colocar los soportes de los motores sin atornillarlos, luego calzar los motores en sus acoples y sus soportes y atornillar todo en su lugar para completar el montaje:



Con el robot ya terminado se puede probar de manera manual cargando en el Arduino el programa “joggin_manual_noPC”, el cual permite controlar los ejes con potenciómetros.

Diseño e implementación del software:

Se decidió implementar el software de control en Python por su simplicidad de uso.

Al diseñar la interfaz con el usuario se buscaba que sea lo más similar a la de los programas para robots como el visto en clase, por lo que se intentó replicar el set de instrucciones.

Se implementó un módulo `robot_t`, el cual contiene la definición del objeto robot, este tiene las funciones de los movimientos joint y lineales, así como la cinemática directa e inversa del robot (como se usaron servos no se implementó un controlador como tal para los motores ya que este está embebido en el servo), el robot cuenta también con atributos para definir la herramienta y la transformación de la base a la terna de referencia del robot, también se puede activar el gripper o los motores con estos atributos y setear una pose de home.

En la parte del Arduino se buscó que fuera lo más simple posible, por lo que este solo recibe por medio del puerto USB una línea de comandos de la forma “<th1:th2:th3:th4:gripperState:motorsState>”, la procesa, ejecuta y envía un carácter de acknorage a la pc para que esta le envíe el siguiente comando.

Para programar una rutina en el robot solo se debe importar el módulo, crear el objeto robot, configurar el puerto en el que se conectó el USB (si se desea también se pueden configurar el home, la herramienta, etc.) y luego escribir un programa utilizando los comandos `MoveL` y `MoveJ` para ordenarle al robot que hacer y cambiando el estado del gripper si se desea activarlo o desactivarlo.

Los comandos de movimiento reciben una pose de la forma $[X,Y,Z,th]$ y en el caso del

movimiento lineal el tiempo que se desea que tarde el movimiento.

En el caso de los movimientos joint, como no se implementó aun un control de velocidad estos se ejecutan a máxima velocidad y es conveniente poner un delay entre comandos en el programa para darle tiempo al robot a completarlos.

Los movimientos lineales son procesados en tiempo real paso a paso del ciclo de control, por lo que es posible pararlos a la mitad de la ejecución si se cierra el programa (aún no se implementó un comando de stop presionando una tecla). Debido a que no es posible controlar la velocidad con la que se mueven los servos, estos movimientos se realizan sumando una serie de movimientos joint muy cortos

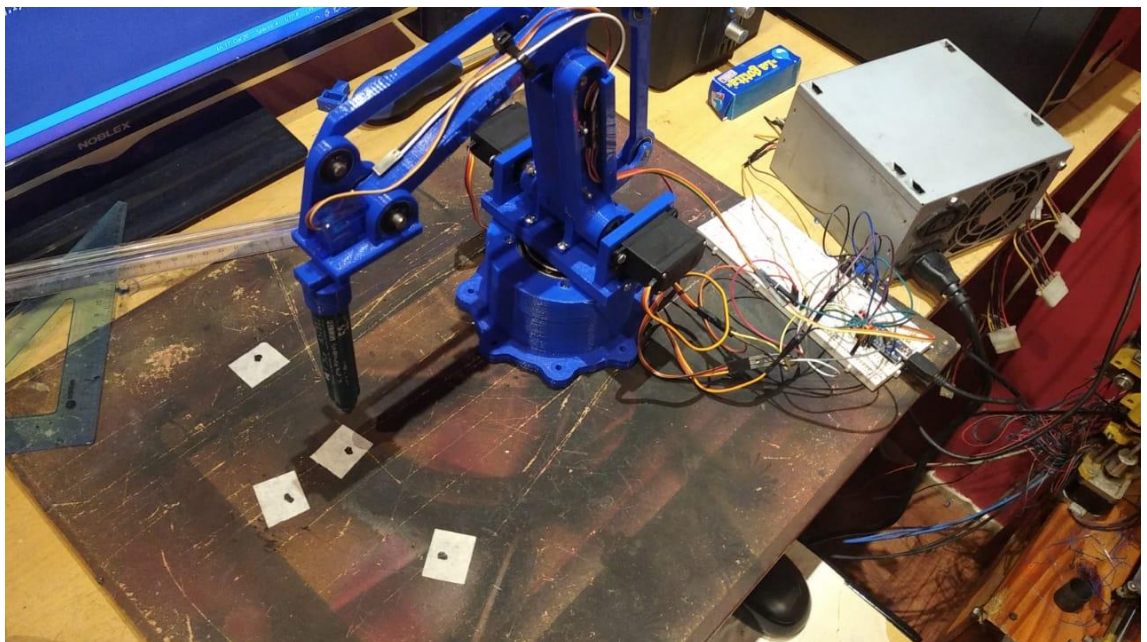
Si bien era posible y quizá más prolijo realizar todo el controlador en una raspberryPi en vez de usar una pc y un Arduino esta opción no se utilizó por motivos de costos.

El software implementado en Python utiliza los módulos pyserial y sympy

Pruebas sobre el prototipo y verificación de las especificaciones deseadas:

- **precisión y repetitividad:**

Se probaron la precisión y la repetitividad del robot haciéndolo repetir 60 veces una secuencia de 4 movimientos en los cuales el robot tenía que marcar 4 puntos en el suelo con un fibron, luego se midió la dispersión de dichos puntos para corroborar la repetitividad y se midió la posición de los clústeres de puntos respecto de las posiciones deseadas para verificar la precisión.



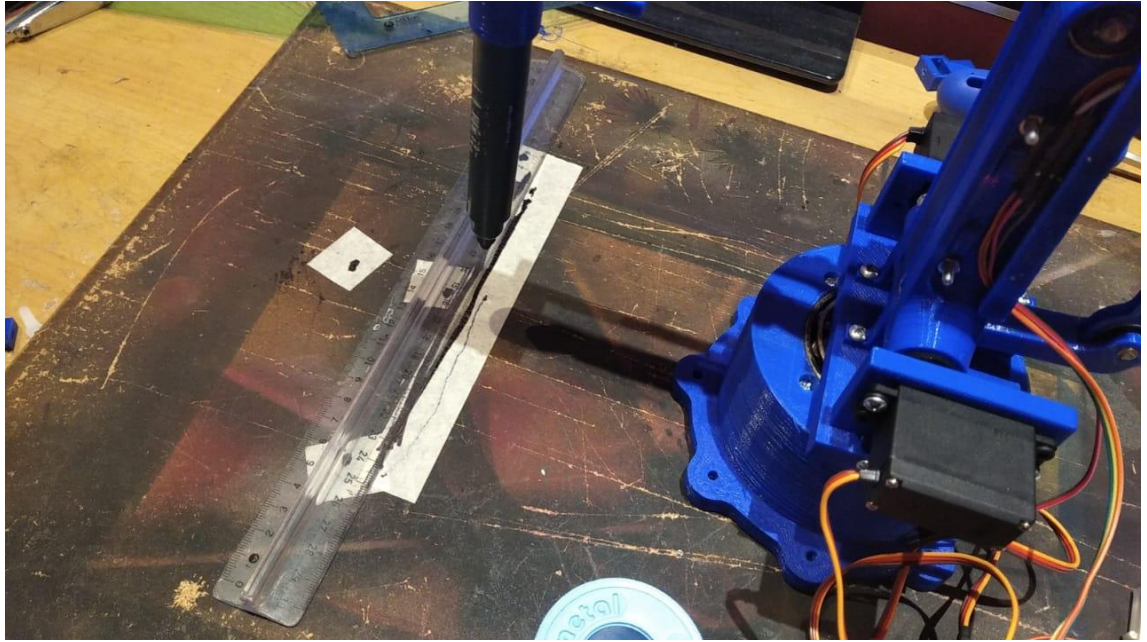
RESULTADO:

PRECISION: $\pm 5\text{mm}$

REPETITIVIDAD: $\pm 3\text{mm}$

- **Linealidad:**

La capacidad de realizar movimientos lineales se probó de manera similar, se hizo al robot trazar 20 líneas rectas y al finalizar se compararon dichas líneas con una regla para verificar que sean trazos rectos.



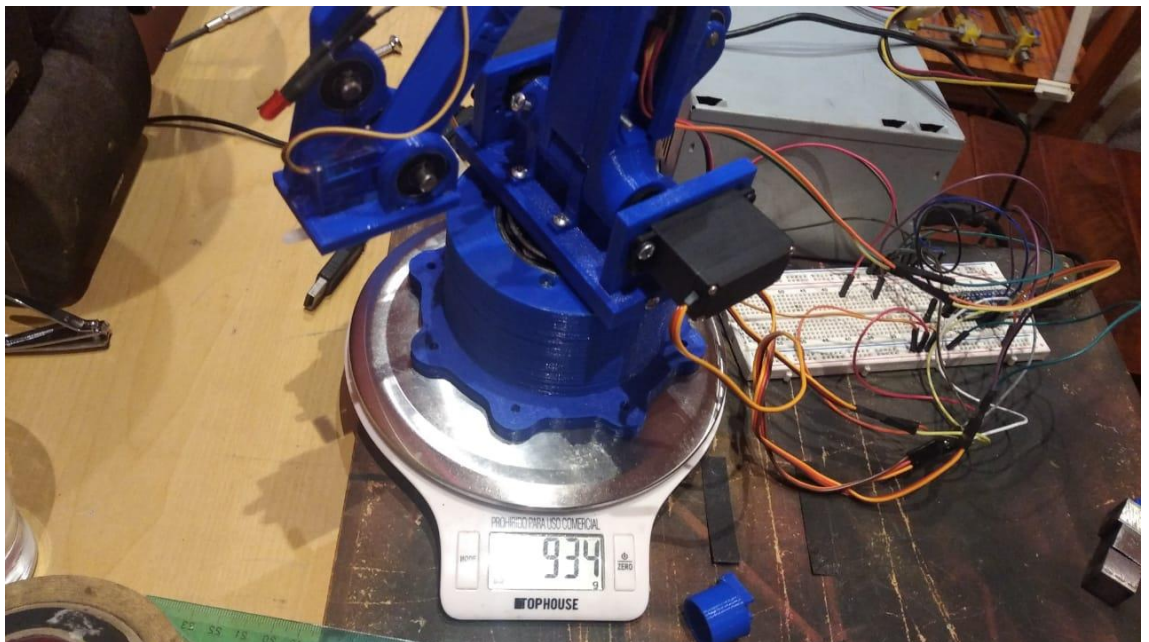
Debido a la precariedad del montaje del fibron la prueba no fue eficaz a la hora de cuantificar la linealidad, solo se pudo verificar que el robot es capaz de hacer movimientos “lineales” bastante consistentes, ya que todos los trazos en ambos sentidos quedaron superpuestos.

- **Capacidad de carga:**

Se realizaron pruebas haciendo al robot levantar cargas conocidas, se verificó que es capaz de levantar cargas hasta 260g, pero no llevarlas a la pose deseada. La máxima capacidad de carga con la que puede alcanzar una pose deseada es 200g



- **Peso total:**
Se corrobora el peso del robot con una balanza y este resultado de 934g



Lista de componentes:

- **Rodamientos:**

15	6262RS
1	30202JR o 30305JR

- **Motores:**

3	MG946R
1	SG90

- **Interfaz:**

1	Arduino Nano con cable USB (cualquier Arduino debería servir)
---	---

- **Burlonería:**

6	tornillos de 3/2in con tuercas
18	tornillos de 1/2in
6	tornillos de 1in
1	varilla rectificada de 6mm
8	seguros zegger de 6mm

- **Varios:**

1	Fuente 5V 5A
1	Cap 22uf
1	Cap .22pf
1	Protoboard
n	Cables

Cosas por mejorar:

En futuras versiones del modelo se debería:

1. Mejorar el diseño de la brida, ya que el eje del servomotor del eje 4 no es estable para montar herramientas.
2. Utilizar motores mas precisos y en los cuales se pueda controlar la velocidad con precisión, porque los servos tienen una resolución de 1º y un juego muerto también de 1º, lo cual genera un gran juego muerto en la herramienta.
3. Implementar el control de velocidad en los movimientos joint.
4. Montar la interfaz en una placa.
5. Solucionar el problema de que el brazo se mueve aleatoriamente al establecer la conexión serie.

6. Agregar feedback de información procedente de sensores en el robot.
7. Hacer los movimientos controlando la velocidad de los motores.
8. Es posible mejorar los servos sacándoles el controlador que traen y controlándolos uno mismo desde el programa, lo cual mejoraría en gran medida el funcionamiento general.

El código, los modelos y todo el material desarrollado para este trabajo pueden descargarse del repositorio:

https://github.com/TomasContessi/TP_FINAL_ROBOTICA.git