

PROYECTO FINAL DE INGENIERÍA

Proyecto Arduino – Brazo robótico

Cascon, Pablo – LU138458

Ingeniería Electromecánica

Corbalan, Maximiliano – LU133712

Ingeniería Electromecánica

Tutor:

Abad, Fernando, UADE

Colaborador/es:

Zambrano, Daniel, UADE

Diciembre 1, 2017



UNIVERSIDAD ARGENTINA DE LA EMPRESA
FACULTAD DE INGENIERÍA Y CIENCIAS EXACTAS

Introducción

Agradecimientos

Queremos agradecer al profesor Daniel Zambrano, por sus consejos y por compartir su experiencia personal con nosotros a fin de ayudarnos en la resolución de diversos problemas que se nos fueron planteando

Al profesor y tutor Fernando Abad, por incentivarnos a querer superarnos en cada etapa del proyecto, por abastecernos de ideas y propuestas diversas para que nosotros evaluemos la mejor opción, y por su inagotable paciencia sin importar día y horario para evacuar nuestras inquietudes.

A nuestros familiares que nos apoyaron en todo momento

A la empresa Agfa Graphics S.A., la cual nos proveyó de una gran cantidad de componentes utilizados en el proyecto sin costo alguno.

Y por último a nuestros compañeros de facultad, siempre dispuestos a ayudarnos en lo que sea necesario, particularmente a Nicolas Dicapua quien nos fue de gran ayuda para introducirnos en el “mundo” Arduino.

Resumen

El proyecto se basa en la construcción de un brazo robótico con tres grados de libertad, basando el diseño mecánico del mismo, en un proyecto anterior realizado en madera. El brazo realiza sus movimientos a través de motores paso a paso los cuales son controlados con una placa Arduino.

El objetivo principal del brazo robótico es que sea robusto, estable y apto para la realización de prácticas para los alumnos de las materias de mecatrónica, e hidráulica y mecánica industrial.

Se encontraron varias fallas en el diseño original a copiar, por lo que nos vimos obligados a modificar ciertas partes, como por ejemplo la colocación de un contrapeso en el extremo opuesto a la pinza y la utilización de cajas de reducción para los motores ya que el esfuerzo que debían hacer los mismos para poder realizar los movimientos era tal que no fue posible encontrar en el mercado un motor con el tamaño adecuado y el torque suficiente para realizar los movimientos adecuadamente.

El modo de operación del brazo, tiene dos opciones, la primera es utilizarlo mediante un joystick siendo el operador quien decide que movimientos realizará el brazo robótico, siendo los sensores de “home” y “fin de carrera” los límites de maniobra del mismo. Y la segunda opción es mediante la cinemática inversa, se ingresan los valores de coordenadas (X, Y, Z) del destino y automáticamente el brazo calcula los movimientos que debe realizar cada motor para llegar a su destino.

Como prácticas para los alumnos de las materias nombradas anteriormente, se realizaron guías con objetivos concretos para que los alumnos puedan llevar a cabo a modo de trabajo práctico algunas se basan en el modelo actual, y otras proponen mejoras y/o modificaciones para que sean implementadas por los alumnos

Finalmente podemos decir que el objetivo principal fue cumplido, aunque podemos afirmar que la plataforma Arduino no es muy confiable y estable. Por lo que para un entorno industrial no es muy recomendable ya que se requiere estabilidad y confiabilidad y en nuestra

experiencia nos hemos encontrado con varias fallas aleatorias debidas ocasionadas por la placa Arduino.

Abstract

The project is based on the construction of a robotic arm with three degrees of freedom, the mechanical structure of it, must be based on an earlier project made in wood.

The arm makes its movements through stepper motors, which are controlled with an Arduino board.

The main objective of the robotic arm is that must be robust, stable and suitable for the practice of students for mechatronics and hydraulics and industrial mechanics

Several failures were found in the original design to be copied. So we were forced to modify certain parts, such as the placement of a counterweight on the opposite end of the clamp and the use of gear boxes for the stepper motors, due the effort that the motor had to do to be able to carry out the movements was such, that was not possible to find in the market a motor with the correct size and enough torque to carry out the movements properly.

The operating mode of the arm has two options, the first one is to use it through a joystick; the operator decides which movements the robotic arm should do, but the sensor “home” and “limit” are the maneuver limits. The second options is by inverse kinematics, the data with the coordinate values (X,Y,Z) of the destination is entry, and the arm automatically calculates the movements that each motor must perform to reach its destination

As practical for the students of the above-mentioned subjects, guides were made with specific objectives, so the students can carry out as tutorials workshops, some are based in the current model, and others propose improvements and/or modifications to be implemented by the students.

Finally, we can say that the main objective was fulfilled, although we can say that the Arduino platform is not very reliable and stable. Therefore, for an industrial environment is not highly recommended as it requires stability and reliability and in our experience, we have encountered several random failures due to the Arduino board.

Contenido

Agradecimientos	2
Resumen	3
Abstract	4
1- Introducción	6
1.1Objetivos	6
1.2Relevancia	7
1.3Estructura del Informe	8
2- Antecedentes	9
A - Robots lineales.....	9
B - Robots Cilíndricos	10
C - Robots Paralelos	11
D - Robots Esféricos	11
E - Robot Scara	12
3- Desarrollo y Metodología	13
3.1 Diseño.....	13
3.2 Construcción	16
3.3 Programación	29
3.3.1 Joystick	29
3.3.2 Control con Teclado	32
3.3.3 Esquema Eléctrico	33
3.3.4 Cinemática Inversa	34
3.3.5 Percepción de Entorno	48
3.4 Práctica Alumnos	51
3.4.1 Añadir un grado de Libertad	51
3.4.2 Comparación de Cálculos	52
3.4.3 Modificación Programa Joystick	52
3.4.4 Modificación Programa de Cinemática Inversa	52
4- Conclusión	53
5- Bibliografía	54
6- Anexos	55

1. Introducción

1.1. Objetivos

El objetivo del Proyecto es realizar un brazo robótico robusto y estable, con tres grados de libertad que sea controlado a través de la plataforma Arduino y que sirva para la realización de diversas prácticas para los alumnos de la facultad que estén cursando las materias de hidráulica y neumática industrial y mecatrónica.

El diseño de brazo deberá estar basado en un prototipo realizado con anterioridad en madera, el cual se tomará como punto de partida para la creación de las piezas principales del mismo.

El brazo debe poder ser controlado por un Joystick y realizar los movimientos que el operador desea. Además debe ser capaz de limitar automáticamente cada movimiento evitando de esta manera que se produzcan daños al equipo.

Por otro lado, el brazo robótico debe ser capaz de realizar movimientos automáticamente teniendo como datos las coordenadas de inicio y de destino del efector final, utilizando como herramienta para el cálculo de su trayectoria la cinemática inversa.

Por último se prepararan guías de trabajos prácticos para que los alumnos de las materias correspondientes puedan involucrarse en el diseño del brazo, entender su funcionamiento y modificarlo, aplicando los conocimientos vistos en las clases teóricas.

1.2. Relevancia

En las industrias de hoy en día, la robótica avanza cada vez más, reemplazando trabajos manuales repetitivos por movimientos sistematizados de brazos robóticos; tal como el que implementaremos en nuestro proyecto.

Las razones principales razones de este cambio tecnológico son que los robots no se cansan, no se equivocan, no necesitan tiempo libre, no se estresan, no se enferman (aunque si se rompen) y por lo general, en tareas simples y repetitivas, son mucho más eficientes que las personas.

Nuestra vida diaria está virtualmente afectada por robots, la robótica crea robots cada vez más prácticos y útiles que facilitan nuestras tareas diarias, la mayor parte de la industria electrónica y automotriz esta automatizada por brazos robóticos que operan con una altísima precisión y a una velocidad mucho mayor que cualquier persona podría lograr.

Hay casos en donde los humanos no pueden trabajar, por ejemplo en ambientes hostiles como el espacio o bajo el agua, en dichos casos la utilización de robots es la única opción para realizar las tareas

El mayor inconveniente que hoy por hoy tienen los robots es su elevado valor de venta, costo de instalación y adaptación a cada industria en particular. Los costos de mantenimiento y la mano de obra calificada para operarlos, si bien son mayores que cualquier máquina convencional, no son factores que influyen a la hora de automatizar una planta industrial.

En lo que respecta al proyecto y teniendo en cuenta la importancia de la robótica en la industria, el brazo robótico que construimos fue concebido con fines educativos para los alumnos que estén promediando el final de sus carreras de ingeniería electromecánica, lo que les dará la oportunidad de tener un contacto directo con este tipo de tecnología y podrán ver plasmado en la realidad muchas de las cosas que hasta el momento solo ven en un marco teórico, como ser sensores de diversos tipos, programación de microcontroladores, motores, control de motores, esfuerzos, relaciones de torque, y cálculos teóricos que podrán llevar a la práctica para poder corroborarlos.

1.3. Estructura del informe

El informe está constituido de la siguiente manera, en un principio a modo de introducción, se muestran los principios de la robótica, algo de historia y los diferentes tipos clásicos de robots.

Luego se procede a detallar cada etapa de la construcción del brazo, decisiones tomadas, problemas y soluciones, planos, componentes, materiales utilizados, forma de construcción y programación.

Finalmente se muestran las propuestas de prácticas para que los alumnos realicen con el brazo robótico y las conclusiones del proyecto.

2. Antecedentes

La robótica lleva ya más de doscientos de años desde sus primeras apariciones en el mundo, se dice que el primer prototipo de robot fue una máquina textil programable con tarjetas perforadas en el año 1801.

A partir de allí, la idea de la robótica comenzó a aparecer en diversas áreas como ser la industria, la exploración espacial, los usos militares, la medicina, etc. Los robots, por lo general están diseñados y construidos para ser controlados por computadoras o por controladores con algoritmos específicos para cada tarea. Los robots pueden clasificarse según su generación, inteligencia, estructura, capacidad de operación y aplicaciones.

Una clasificación más general y estandarizada mundialmente es la siguiente:

- a) Robots lineales
- b) Robots cilíndricos
- c) Robots paralelos
- d) Robots esféricos
- e) Robots SCARA
- f) Robots articulados

a) Robots lineales: básicamente son robots que se mueven sobre un eje lineal a través del cual por medio de motores o actuadores, el robot se mueve en ambas direcciones. Se pueden combinar dos ejes lineales perpendiculares, formando lo que se conoce como “Gantry Robot” o “Robot Grúa” en donde se combinan dos ejes (X; Y) para mover el efector final a través de dos direcciones (como puede verse en al Fig. 1). A este tipo de robots, puede agregarse un tercer eje (X;Y;Z) para obtener un movimiento extra del efector final



Fig 1

- b) Robots cilíndricos: básicamente están compuestos por un movimiento giratorio con respecto a la base del robot y otro movimiento lineal que sube y baja el efector final. Se le puede agregar un movimiento más al efector final que le dé la capacidad de moverse hacia adelante y atrás, cubriendo así mayor campo de operación. (Fig.2)

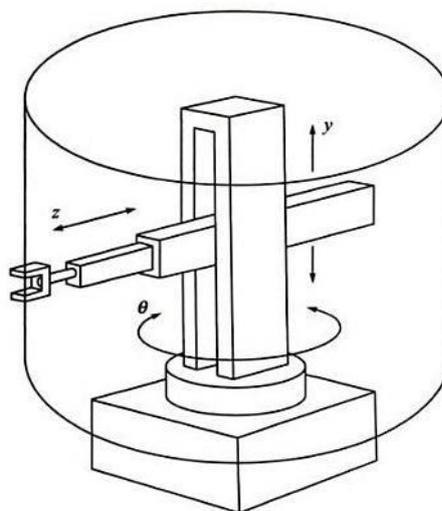


Fig. 2

- c) Robots paralelos: están constituidos por dos plataformas, una que se encuentra en una posición fija y otra que es móvil, ambas plataformas están vinculadas mediante al menos dos cadenas cinemáticas (brazos, o piernas del robot) que son las que le dan el movimiento al robot. (Fig.3)



Fig.3

- d) Robot esférico: se basan en la rotación del brazo sobre la base y el movimiento angular del brazo hacia arriba y hacia abajo, el área que puede recorrer el efector final de este tipo de robot es el del interior de una semiesfera, pudiendo realizar movimientos tanto curvilíneos como rectilíneos combinando los motores de accionamiento. (Fig.4)

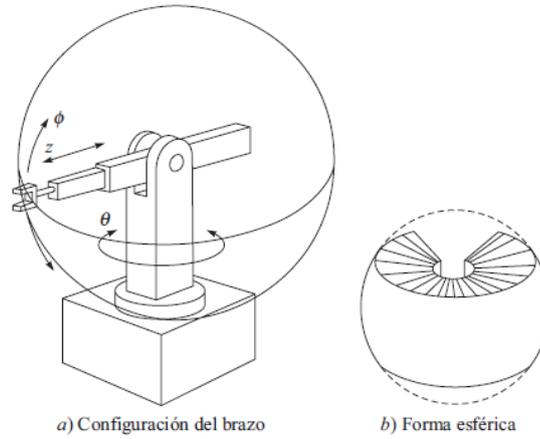


Fig.4

- e) Robot Scara: El nombre proviene del acrónimo “Selective Compliance Assembly Robot Arm (SCARA)” son robots con gran movilidad en los ejes X e Y, pero están limitados en el eje Z. Con cuatro grados de libertad, se caracterizan por movimientos rápidos, precisos y repetitivos. (Fig.5)



Fig.5

Los tipos de robots nombrados están dentro de la clasificación más común, desde ya que hay una gran variedad de robots en el mercado y la industria, dependiendo de la finalidad de cada robot y las aplicaciones de este.

3. Desarrollo y metodología

3.1. Diseño

Basados en el diseño original (ver fig.6) (Proyecto Final De Carlini y Miglio, 2015) evaluamos realizar el nuevo brazo robótico en acrílico o en aluminio, finalmente decidimos realizarlo en aluminio, por ser un material el cual se lo puede mecanizar manualmente sin la necesidad de maquinaria o mano de obra especializada, tiene la robustez necesaria para lograr el objetivo propuesto, es de bajo peso específico (2580 kg/m^3) lo cual ayuda a que los motores que se utilizan no realicen esfuerzos muy grandes, y es estéticamente adecuado para el uso que se le dará



Fig.6 Brazo robótico original

Como se ve en la Fig.6, el brazo robótico original estaba construido en madera, y contaba con 3 grados de libertad (el primer movimiento se encuentra en la base del brazo, realizando movimientos de rotación, cubriendo un campo total de noventa grados; el segundo grado de libertad es el que da los movimientos angulares de la primera palanca y el tercer grado de libertad está relacionado al movimiento del segundo brazo de palanca. Finalmente el efector final con forma de pinza.

Basados en dicho diseño, se procedió a dibujar cada pieza en SolidWorks para obtener luego los planos con los cuales realizaríamos la cotización para la fabricación de las piezas principales

Desde un principio se realizaron cambios en comparación con el diseño original. La base debía ser diferente. Nuestro modelo, al ser de un material más pesado, requiere más esfuerzo del motor para realizar los movimientos. Por lo que se incorporó un rodamiento axial el cual sirve de base para un engranaje que soporta todo el conjunto. Luego, dicho engranaje será accionado por medio de una correa, con la reducción adecuada para permitir el movimiento correcto.

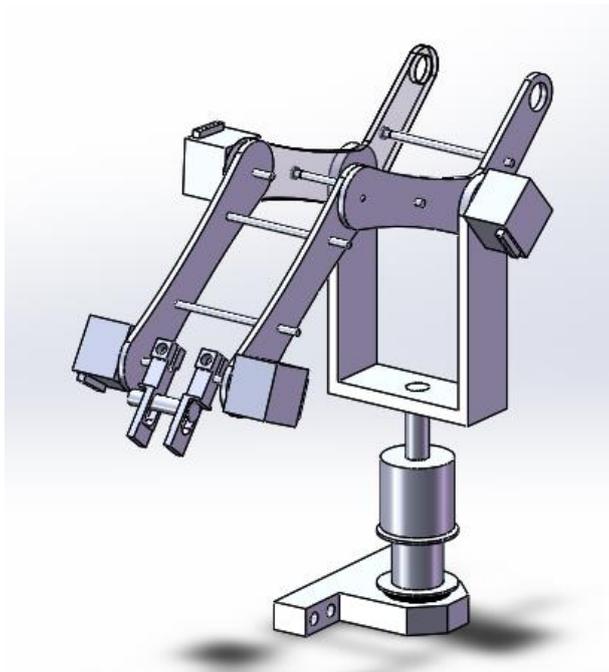


Fig.7 primer diseño del brazo metálico

En la Fig.7 puede observarse un primer diseño de lo que sería el nuevo brazo, difiriendo un poco del diseño original. Además de la base con rodamiento axial, como se mencionó anteriormente, se agregaron soportes entre las caras de los laterales de cada brazo de palanca para tener más firmeza y estabilidad.

En un principio habíamos pensado incorporar como efector final unas mordazas metalizas (ver Fig.8) las cuales, mediante un eje sin fin, generarían el movimiento de abrir y cerrar. Finalmente las mordazas fueron cambiadas por otro modelo de pinza, mejorando así la funcionalidad del brazo robótico (ver Fig.9)



Fig.8 Pinzas primer opción

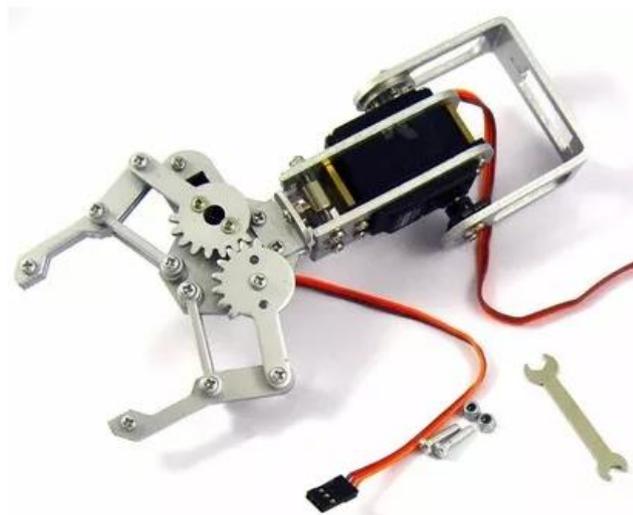


Fig.9 Efecto final opción definitiva

3.2. Construcción

Una vez que tuvimos los planos de las piezas de la estructura principal (véase anexo1) se procedió a buscar en donde fabricarlas. Debido a la complejidad de las mismas, la mejor opción que se encontró fue en empresas de corte por agua, dado que solo sería uno o dos ejemplares de cada pieza. (ver Anexo 2 presupuestos por el trabajo)



Fig. 10. Brazos



Fig. 11. Soporte en U

En las Fig. 10 y 11 pueden verse las piezas fabricadas en aluminio.

Para la base (Fig.12), se utilizó una pieza metálica con un rodamiento axial acoplado a un engranaje y un acople de tipo “manguito” la cual pertenecía a una máquina de Inkjet fabricada por la empresa Agfa Graphics modelo Jeti 3324 (Fig.13), la cual nos fue donada para la preparación del proyecto.



Fig.12 Base del brazo con rodamiento axial



Fig. 13 Agfa Jeti 3324

Para realizar los movimientos de los tres grados de libertad, se consiguieron también como donación de Agfa Graphics S.A. provenientes de un equipo Advantage N-SA (Fig. 14) tres motores paso a paso marca Moons, modelo (véase anexo 3, hoja de datos y características).



Fig. 14 Advantage N-SA

Los motores anteriormente mencionados serían controlados en un principio por la placa Arduino a través de un driver

Motores Paso a Paso (PAP):

Los motores PAP utilizados fueron elegidos por su precisión en el movimiento. La característica mas importante de estos motores es la posibilidad de moverlos un paso por cada pulso que se le aplique. En nuestro caso por cada pulso enviado logramos un movimiento de 1.8° por cada paso, es decir que necesitamos 200 pasos para cada vuelta de motor. Además, poseen la habilidad de poder quedar enclavados en una posición o bien, permanecer libres. Esto lo logra si mantenemos energizadas una o más de sus bobinas.

Básicamente estos motores están contruidos por un rotor y un cierto número de bobinas. Para poder controlar la excitación de estas bobinas necesitamos de un controlador o Driver.

A continuación, ingresamos una imagen de las bobinas.

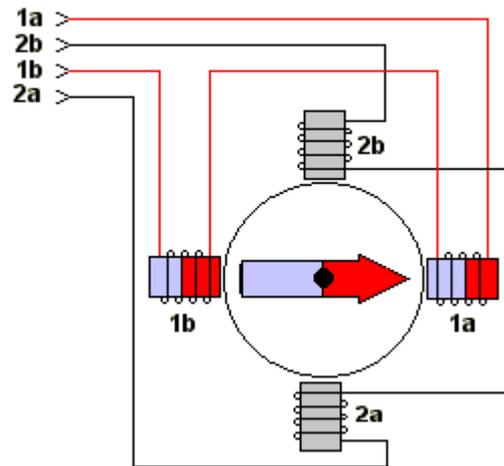


Fig. 15. Bobinas PAP genérico

Según la conexión de estas bobinas el motor PAP se puede comportar como un motor BIPOLAR o UNIPOLAR.

En nuestro proyecto se decidió utilizarlo como motor bipolar por las características y ventajas que entrega en cuanto al torque. Que para mejorarlo fue necesario agregar cajas reductoras para poder aumentar el toque que los motores nos entregaban.

Drivers:

Para el manejo de los motores paso a paso es necesario la utilización de drivers que nos permiten mandar pulsos a las bobinas del motor en el orden adecuado. Ese orden depende de dos factores, el primero si el motor PAP utilizado es unipolar o bipolar y el segundo según el pulso de avance que necesitemos.

En nuestro caso además de tener presente estos dos factores, también tuvimos en cuenta la potencia máxima que nos podrían brindar nuestros motores y de esa manera poder elegir que driver que nos era más conveniente.

Durante los ensayos previos a la construcción del brazo, utilizamos dos drives distintos:

- 1- Pololu DRV 8825
- 2- Pololu A4988

Ambos drivers constan de un integrado llamado puente H el cual permite manejar la lógica necesaria para poder dar movimiento a nuestros motores.

Si bien ambos drivers son de características similares, el Pololu 8825 nos permite tener corrientes de hasta 2,2 A, lo cual nos permite llegar a los valores de potencia que necesitamos.

A diferencia del A4988 que solo puede tener un máximo de 2A.

Si bien el DRV 8825 nos otorga 2A es necesario utilizar un disipador para poder alcanzar esos niveles de corriente. Pero en nuestro caso, además fue necesario agregar dos coolers, ya que sin estos el integrado levantaba mucha temperatura y se quemaba.

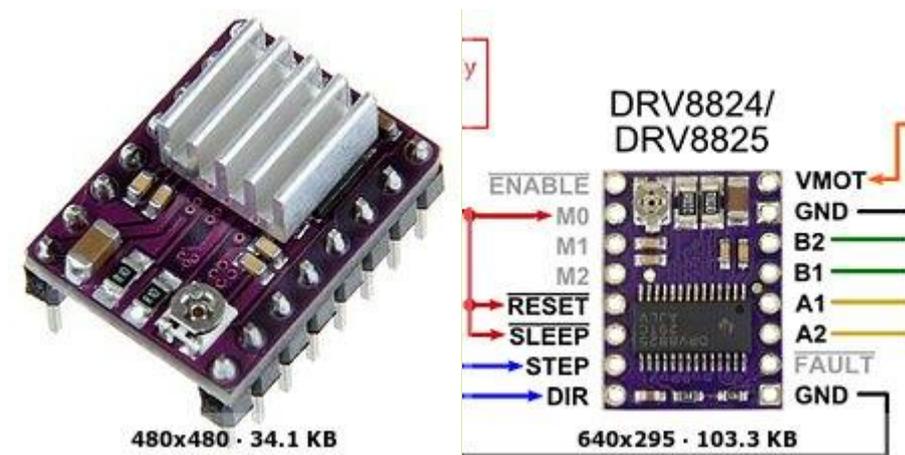


Fig. 16 Driver Pololu 8825

En el Anexo 4 se encuentra el datasheet de los Drivers utilizados.

Otras características de este driver son las siguientes:

- a- Posibilidad de ajustar o limitar la corriente que va al motor.
- b- Posee protección contra exceso de temperatura y sobre corriente.
- c- Posibilidad de controlar motores de 32 micro pasos.
- d- Voltaje de operación del motor entre 8,2 a 45 V.
- e- Voltaje de trabajo (driver) entre 2,5 a 5,25V.
- f- Máxima corriente de trabajo, 1,5A (sin disipador) y hasta 2,2 con disipador.

Si bien la característica principal es la potencia que podemos manejar con estos motores, la posibilidad de estas otras características nos proporcionó mejor manejo de los motores en cuanto precisión de los movimientos.

Limitación de corriente del Drive:

Para su correcto funcionamiento es preciso limitar la corriente y evitar picos de corriente. Durante las pruebas que fuimos realizando, en las cuales varios drivers se quemaron, nos percatamos que no solo limitar la corriente evitaba la destrucción del drive, sino que además evita que se generen movimientos inesperados de los motores.

Basándonos en Foros de internet (<http://foro.cnccontrolapp.com/index.php?topic=774.0>), encontramos la siguiente expresión para limitar la corriente:

$$\text{Corriente limite} = VREF * 2$$

El driver cuenta con un potenciómetro y un pin de referencia (ver Fig. 16)

El potenciómetro debe ser regulado para poder alcanzar la VREF que se obtuvo con la expresión anterior y de esa forma alcanzar la corriente de trabajo necesaria.

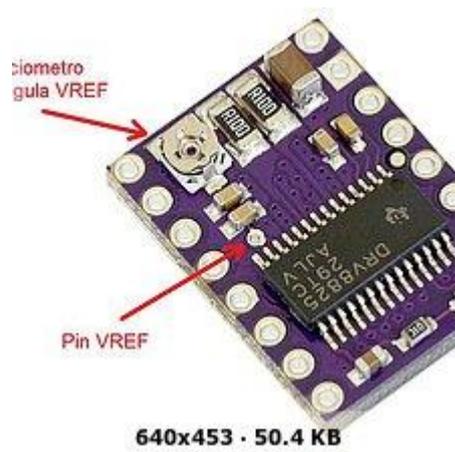


Fig. 17 Potenciómetro y Pin Vref

Disipador y Cooler:

Todos los componentes electrónicos sufren con las altas temperaturas, si no controlamos la temperatura que pueden alcanzar, el componente puede dañarse durante su funcionamiento. Para evitar esto, se utilizan disipadores de calor, en nuestro caso se utilizó un disipador proporcionado por el fabricante del driver que son simplemente placas metálicas con una buena transmisión de calor (Fig. .18).

En nuestro caso también fue necesario agregar ventilación forzada con un cooler, debido a que estábamos cercanos a los límites de corriente establecidos por el fabricante.

Estos cooler fueron donación de la empresa Telefónica S.A, los cuales pertenecían a servidores fuera de uso.



Fig. 18 Disipador y cooler utilizados

Fuente de Alimentación:

Como fuente de alimentación, se consiguió también por donación de la empresa Agfa graphics una fuente regulada y regulable de 17 a 110 volts de salida y 120 a 220 volts de entrada proveniente de un equipo Agfa Galileo VXT (fig. 19)



Fig. 19 Agfa Galileo VXT

Desde un principio el objetivo del proyecto fue controlar el brazo por medio de una placa Arduino, que es una plataforma electrónica de código abierto, que es bastante popular y mundialmente reconocida, por lo que en la gran mayoría de los casos, siempre podremos encontrar elementos compatibles con dicha placa y asegurarnos que todo lo planeado pueda ser ejecutado.

Hay diversos modelos de placas Arduino (fig.20), nosotros por cuestión de costo, y cantidad de puertos disponibles nos decidimos por utilizar la Arduino Mega 2560 (ver anexo 5, Arduino Mega 2560 datasheet).

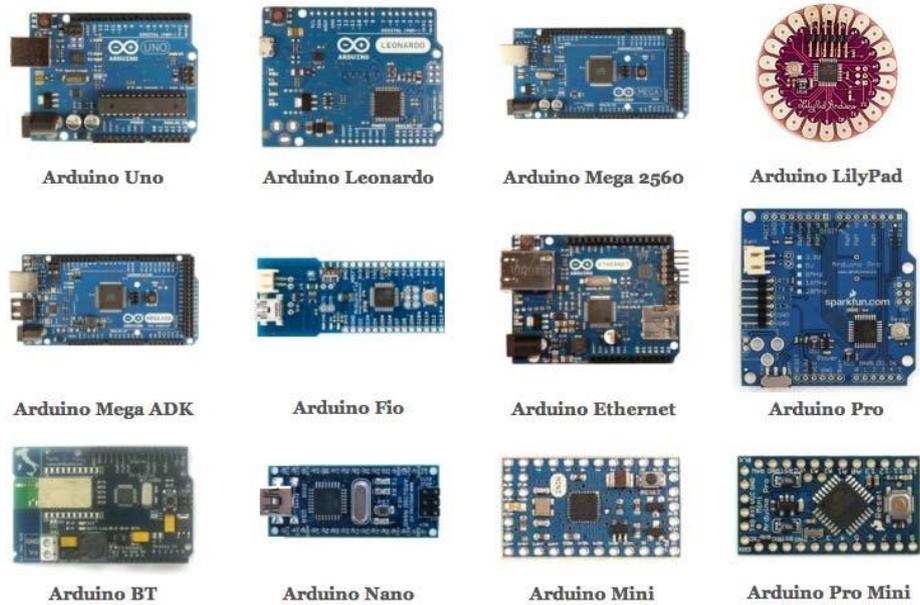


Fig. 20 Placas Arduino

Una vez que logramos juntar todos los componentes fundamentales del proyecto, procedimos a armar las primeras maquetas de lo que sería el brazo robótico (Fig.21).



Fig.21 Primer armado en conjunto

Armado todo el conjunto, comenzamos a realizar diferentes pruebas de movimientos con los motores que teníamos, pero resulto que a el único que realizaba bien sus movimientos fue el motor de la base (Motor M1) cuyo eje lo pusimos solidario a un engranaje mediante el cual, a través de una correa mueve el engranaje principal de la base (Fig. 22).

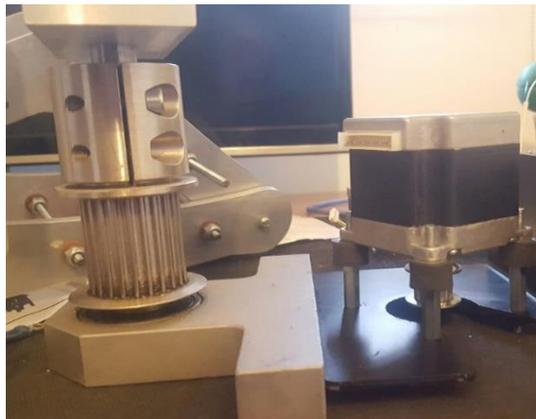


Fig. 22 Motor M1 en su base

Los otros dos motores M2 y M3, no tenían el torque suficiente para levantar el brazo de palanca que le generaba el peso de la estructura del brazo, por lo que nos vimos obligados a tener que introducir cajas reductoras, ya que en el mercado local no pudimos encontrar motores con el torque y el tamaño adecuados para la aplicación que buscábamos.

Las cajas reductoras se sacaron de unos motores en desuso de una impresora offset marca Heidelberg Modelo SM74 (Fig. 23). Motores que su función original es regular la presión de los rodillos de impresión, por lo que la desmultiplicación es muy grande, razón que nos llevó a tener que modificar dicha reducción porque, si bien funcionaba, la velocidad de los movimientos era demasiado lenta por lo que se le quito una etapa de reducción (Fig. 24)



Fig. 23 Impresora Heilderberg SM 74

La reducciones en cuestión están basadas originalmente en tres sistemas planetarios (Fig. 25), que van transmitiendo movimientos uno al otro, y cambiando así la relación de giro de entrada y salida, nosotros para adaptarla a nuestras necesidades le quitamos una etapa quedando un sistema con dos etapas (Fig.26)



Fig.24 Reducción Motor Heilderberg

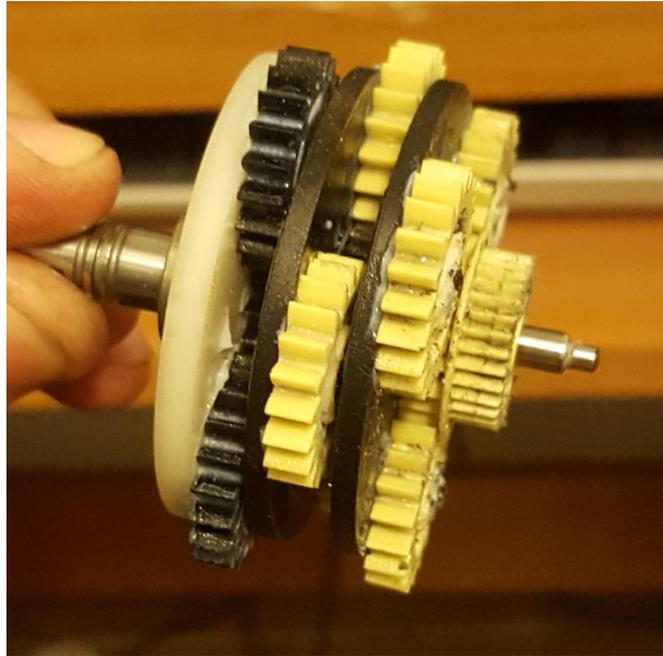


Fig. 25 Sistemas planetarios motor Heilderberg

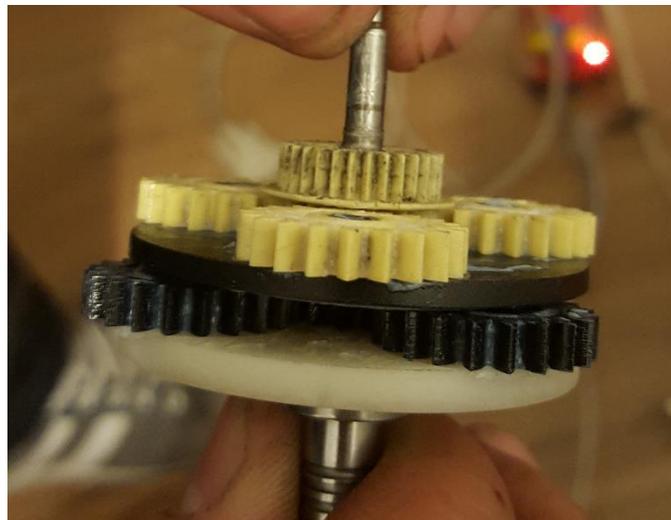


Fig.26 Reducción modificada

Luego de obtener la reducción modificada, se adaptó la carcasa de la manera más económica y efectiva cada caja reductora a su respectivo motor (las carcasas de las distintas reducciones no era la misma, por lo que los acoples de cada una con su respectivo motor, quedo estéticamente distinto).

Como efector final, se decidió cambiar las mordazas del diseño original por una pinza metálica ya armada de fácil obtención en el mercado local, lo cual mejoraría las prestaciones del brazo robótico y no requería de grandes cambios en el diseño (Fig. 27).



Fig.27 Efector final tipo pinza

Una vez ensambladas todas las partes, para lo cual fue necesario el uso de herramientas de mecanizado manual para adaptar todos los acoples y ejes a las piezas que se enviaron a fabricar, comenzamos a realizar diversas pruebas de movimientos a fin de corroborar que todo funcione correctamente. Se agregó un contrapeso en la parte posterior del brazo de palanca del motor M2 para reducir la fuerza que el motor debía realizar y fue necesario el uso de disipadores en los drivers de cada motor además de la colocación de un cooler para forzar el paso del aire y refrigerar los disipadores, ya que los Pololu DRV8825 utilizados, permiten una corriente máxima de 2,2 Amper, y los motores consumen cuando están energizados y sin realizar movimientos hasta 1,8 Amper, por lo que la disipación de temperatura es algo fundamental para que el driver no se queme (problema que se presentó en varias ocasiones). En la Fig. 28, puede verse como quedó el brazo robótico armado y funcional

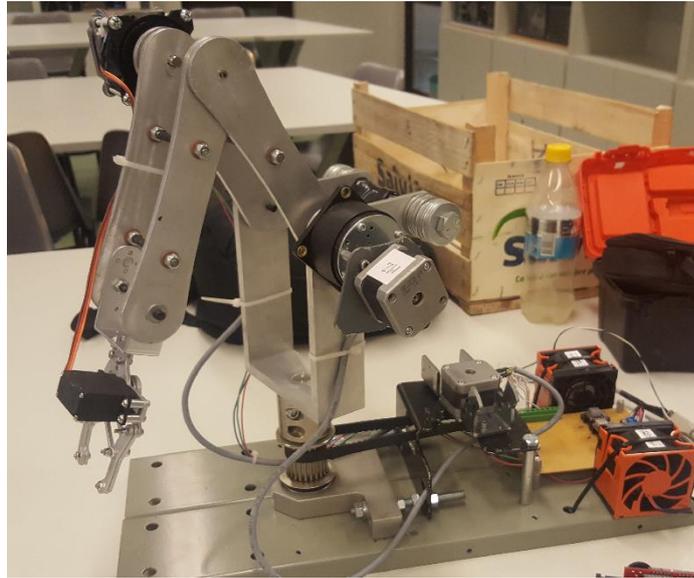


Fig.28 Brazo robótico armado y funcional

3.3. Programación

3.3.1. Control por Joystick

Uno de los objetivos del proyecto es que el brazo pueda ser controlado manualmente mediante un joystick, para ello utilizamos el “Joystick shield V1.A Funduino” (ver Fig.29). El mismo cuenta con 6 botones digitales, los cuales funcionan como pulsadores configurables y con una perilla analógica, la cual tiene un par de ejes X e Y pero que utilizaremos simulando botones digitales, es decir que tomaremos en cuenta solo los extremos del recorrido de cada eje respectivamente para que los motores actúen en consecuencia.

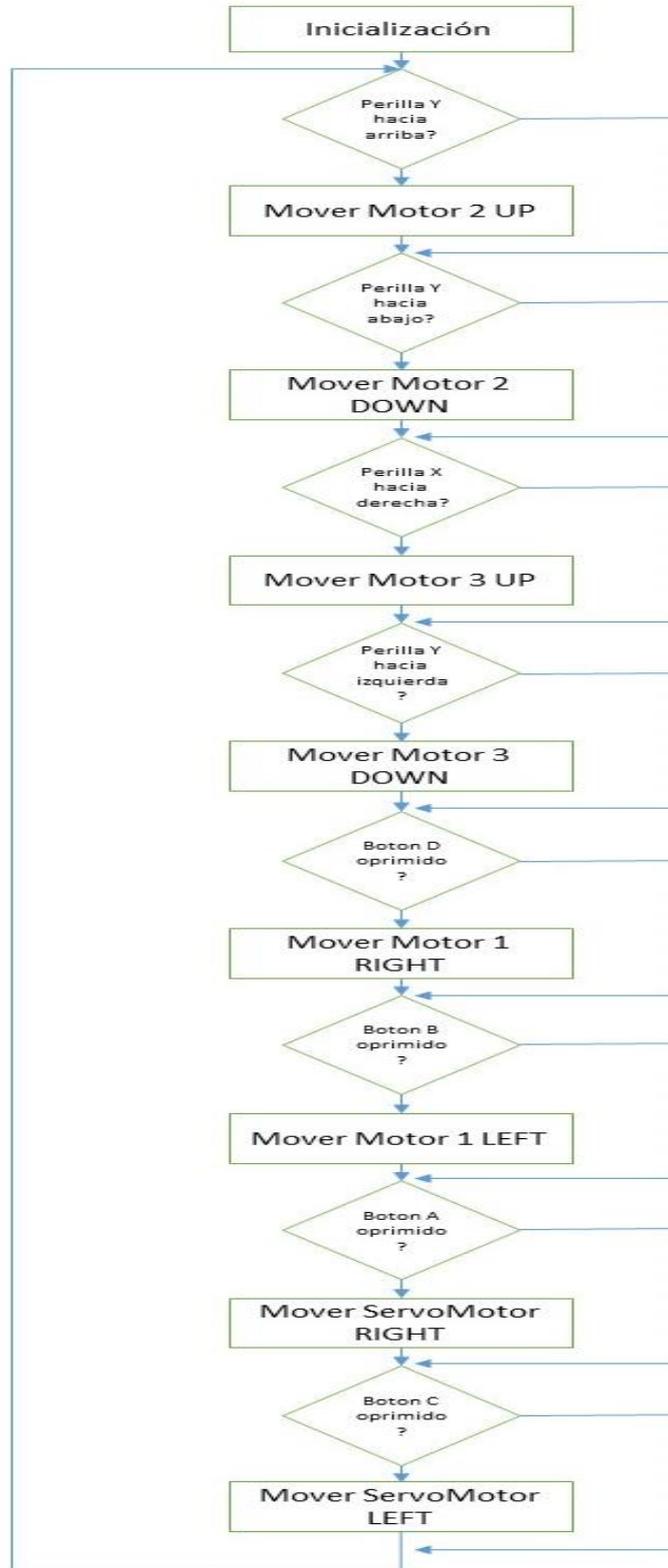


Fig. 29 Joystick shield V1.A

Se utilizaran los botones “B” y “D” para los movimientos a derecha e izquierda respectivamente del motor M1 (giro del brazo), los botones “A” y “C” para abrir y cerrar el efector final, el eje “X” de la perilla analógica para los movimientos hacia arriba y abajo respectivamente del motor M2 y el eje “Y” de la perilla analógica para los movimientos hacia arriba y abajo respectivamente del motor M3.

Se desarrolló el siguiente código Arduino (ver anexo 6) para controlar el brazo robótico con el joystick, en el mismo se verán los comentarios explicando cada etapa del código.

A continuación puede verse el diagrama de flujo del programa del Joystick, en el cual se muestra la idea básica del programa.



Cabe destacar que a cada movimiento se lo tuvo que restringir con sensores de home y fin de carrera (home y away) según corresponda, porque de otra forma los movimientos pueden ser tales que resulten en enredos de cables y posibles daños en el brazo robótico. Se utilizaron switches o pulsadores para limitar los movimientos de los motores M2 y M3 (ver Fig. 30) y sensores ópticos para limitar los movimientos del motor M1 (Fig.31)

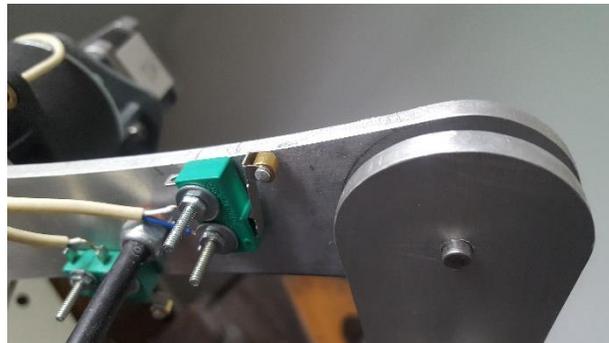


Fig. 30 Sensor microswitch

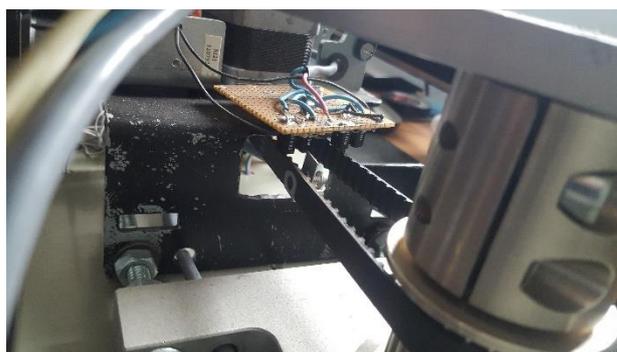


Fig. 31 opto acopladores M1

Para las conexiones de los microswitchs no hay mucho para explicar, simplemente un contacto normal abierto que cuando se acciona, lleva a masa un pin de la placa Arduino dando la señal de que fue activado.

Por el lado de los sensores ópticos, se utilizó el modelo TCST2103 de uso común y comercialmente fácil de conseguir (véase anexo 7 la hoja de datos). se colocó una “bandera” en la correa de transmisión de movimiento para limitar el recorrido y los sensores se conectaron de forma que en su pin de salida dan un “0” o un “1” según estén accionados o no, Ver diagrama de conexión Fig.32, se utilizaron dos sensores, ambos con el mismo circuito independientes uno de otro

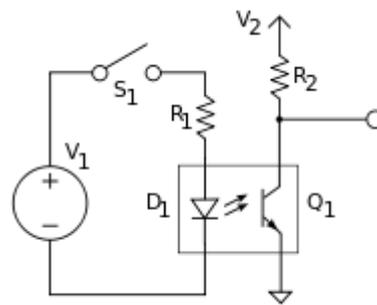


Fig.32 Diagrama de conexión sensor óptico

3.3.2 Control con Teclado

Por medio de un teclado de membrana, se podrán ingresar las coordenadas de ubicación del objeto y las coordenadas donde dejar ubicado el mismo, el cual nuestro brazo podrá calcular mediante su algoritmo de cinemática inversa. Este ingreso de coordenadas será visualizado por medio de una pantalla lcd de 16x2 conectada con nuestro Arduino (ver Fig. 33).

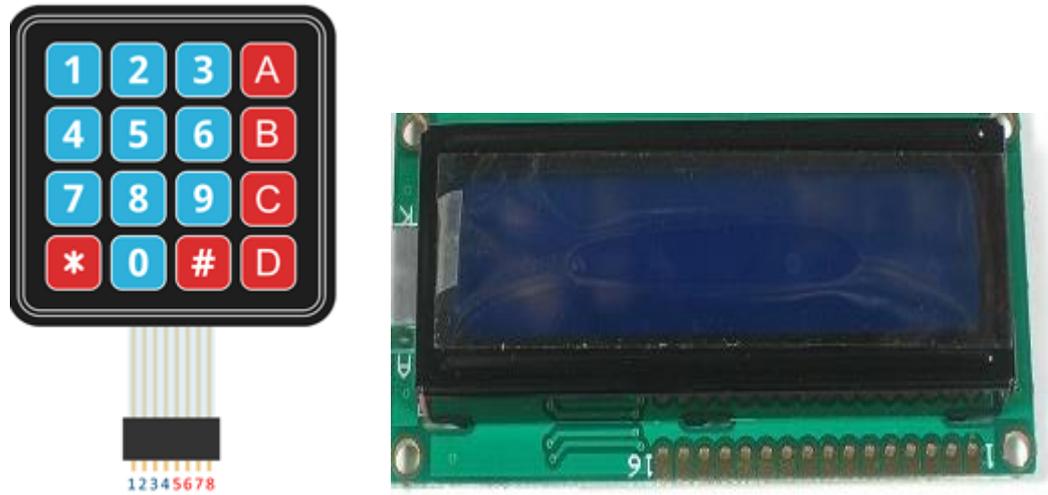


Fig. 33 Teclado y pantalla LCD

3.3.3 Esquema eléctrico del Brazo

En la Fig. 34, puede verse el esquema eléctrico del brazo robótico, en el que se detalla básicamente la placa electrónica que fue construida para realizar las conexiones, ya que al ser un equipo con movimientos y vibraciones, los falsos contactos es un problema al que de esta forma podemos evitar. Se utilizaron zócalos para los drivers, ya que existe la posibilidad de que alguno deba ser reemplazado en el futuro, y para todas las conexiones de los componentes externos como ser la fuente de alimentación, coolers, switches, y motores se utilizaron borneras, de manera tal, que si se requiere de alguna eventual intervención técnica, sea fácil desarmar el equipo.

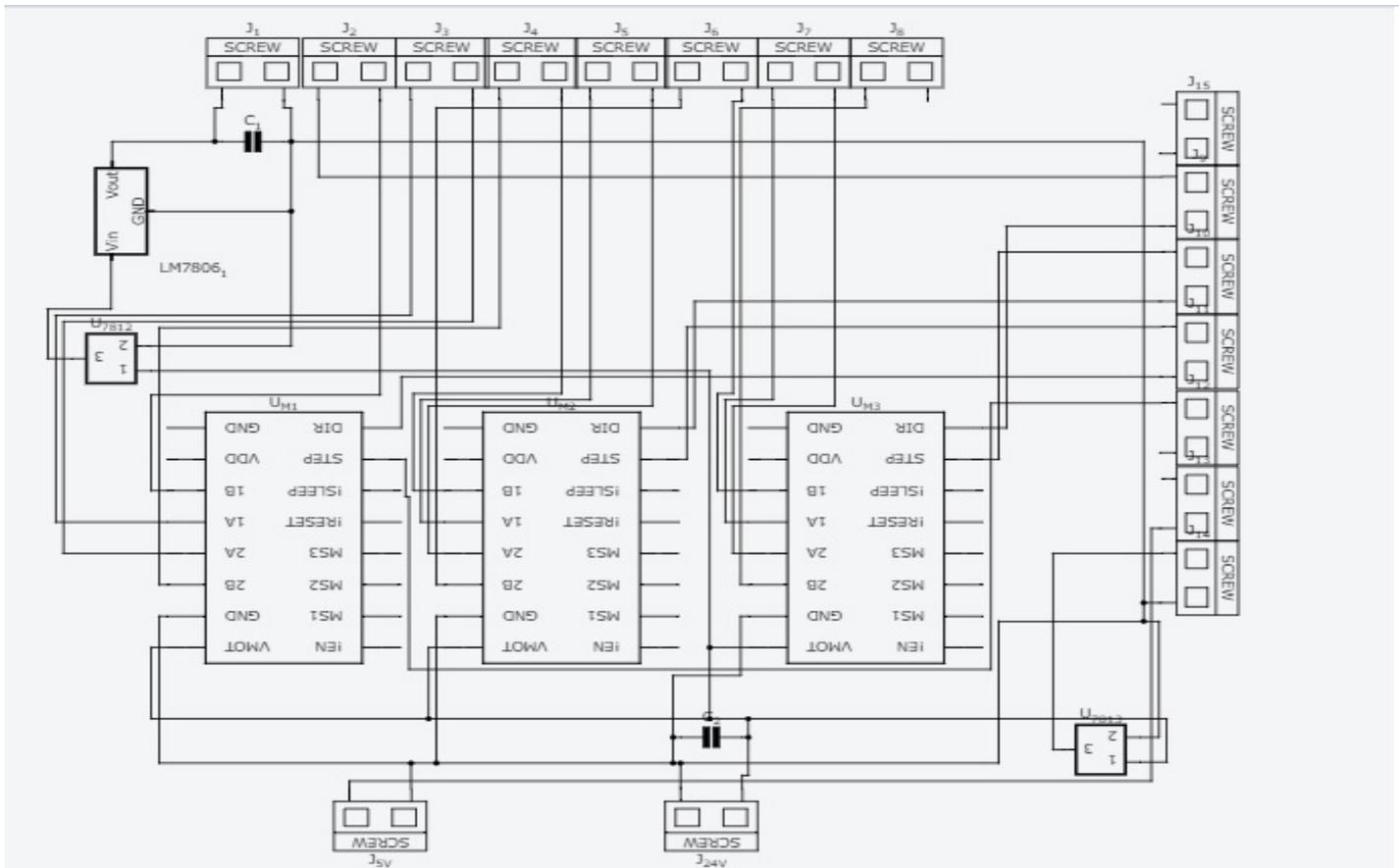


Fig. 34 Esquema eléctrico

3.3.4 Cinemática inversa

El propósito de nuestro trabajo no solo era la construcción de un brazo robótico, sino que además pueda realizar movimientos para ubicar un objeto en el espacio tridimensional.

Para poder lograr esto recurrimos al método de cinemática Inversa.

La cinemática inversa de manipuladores nos brinda la solución para el problema de como calcular la posición y la orientación de nuestro brazo para poder alcanzar algún objeto.

Basándonos en la descripción que utiliza el libro ROBOTICA (John J, Craig. Tercera Edición, 2006), podemos entender la problemática de la cinemática Inversa y sus diversos métodos.

En los estudios de la robótica nos preocupamos constantemente por la ubicación de los objetos en el espacio tridimensional. Estos objetos son los vínculos del manipulador, las piezas y herramientas con las que se trabaja y los demás objetos en el entorno del

manipulador. A un nivel básico pero importante, estos objetos se describen mediante solo dos atributos: “posición y orientación”

Naturalmente, un tema de interés inmediato es la manera en que la representamos estas cantidades y las manipulamos matemáticamente.

Para poder describir la posición y orientación de un cuerpo en el espacio, siempre adjuntamos rígidamente un sistema de coordenadas, o trama, al objeto. Después describimos la posición y orientación de esta trama con respecto a algún sistema de coordenadas de referencia.

Cualquier trama puede servir como sistema de referencia dentro del cual se pueda expresar la posición y orientación de un cuerpo, por lo que pensamos en transformar o cambiar la descripción de estos atributos de un cuerpo de una trama a otro.

Entrando en detalle, la descripción de una posición, una vez establecido el sistema de coordenadas, podemos ubicar cualquier punto en el universo con un vector de posición de orden (3×1) . Como es común que definamos muchos sistemas de coordenadas, además del sistema de coordenadas universal, los vectores deben etiquetarse con la información que identifique en cual sistema están definidos. Tomando referencia a la notación utilizada en la Materia Mecatrónica, se escribe con un subíndice a la izquierda que indica el sistema de coordenadas al que se hace referencia.

Sumado a esto, no solo necesitamos representar solo un punto en el espacio, sino que también describir la orientación de un cuerpo en el espacio. Entonces para describir la orientación de un cuerpo, adjuntaremos un sistema de coordenadas al cuerpo y luego le daremos una descripción de este sistema de coordenadas relativa al sistema de referencia.

Para esto utilizaremos una matriz, llamada matriz de rotación, que describe una trama en forma relativa a otra trama.

Habiendo descrito la posición y la orientación, podemos inferir sobre las operadoras que estas mismas pueden utilizar, que son las mismas operaciones matemáticas que se utilizan para correlacionar puntos entre tramas pueden interpretarse también como operadores que trasladan puntos, giran vectores o hacen ambas cosas.

Especificando una traslación, desliza un punto en el espacio una distancia finita a lo largo de una dirección vectorial dada. Es decir, la interpretación matemática es tan simple como cuando un vector se desplaza “hacia adelante” o que la trama se movió “hacia atrás”. Las matemáticas involucradas en ambos casos son idénticas.

La otra operación sería la rotación, que como aclaramos anteriormente, es una matriz que opera sobre un vector y convierte ese vector en uno nuevo por medio de la rotación, operación matemática conocida de Álgebra Lineal.

Una vez abordados y entendido que es una trama y su manera de operar con la misma, es momento de introducirnos en la Cinemática de los manipuladores.

La cinemática es la ciencia que trata el movimiento sin considerar las fuerzas que lo ocasionan. Dentro de la cinemática se estudian la posición, velocidad, aceleración y todas las derivadas de mayor orden de las variables de posición (respecto al tiempo o cualquier otra variable). Por ende, el estudio de la cinemática de los manipuladores se refiere a todas las propiedades del movimiento, las geométricas y las basadas en tiempo.

Los manipuladores consisten de vínculos casi rígidos, los cuales están conectados por articulaciones que permiten el movimiento relativo de los vínculos adyacentes. Estas articulaciones generalmente se instrumentan con sensores de posición, los cuales permiten medir la posición relativa de los vínculos adyacentes. En el caso de las articulaciones giratorias o angulares, estos desplazamientos se conocen como ángulos articulados.

El número de grados de libertad que posee un manipulador es el número de variables posición independientes que tendrían que especificarse para poder localizar todas las piezas del mecanismo. Este es un término general que se utiliza para cualquier mecanismo. En lo general el número de articulaciones es igual al número de grados de libertad.

En el extremo libre de la cadena de vínculos que conforman el manipulador se encuentra el efector final. En nuestro caso, el efector final es una pinza, pero puede ser diversas cosas dependiendo del uso que se le va a dar.

Entrando a las problemáticas dentro del estudio de mecánica se conoce como cinemática directa y cinemática inversa (nuestro caso).

Para poder entender la diferencia entre estas dos problemáticas, nos introduciremos en la cinemática directa sin detallar tanto y en la cinemática inversa, que es nuestra problemática nos detendremos y haremos una explicación más detallada.

Entonces, la cinemática directa, que es el problema geométrico estático de calcular la posición y orientación del efector final del manipulador. Específicamente, dado un conjunto de ángulos articulares, el problema de la cinemática directa es calcular la posición y orientación de la trama de la herramienta relativa a la trama base.

Por otro lado, la problemática de la cinemática Inversa se plantea de la siguiente manera: dada la posición y orientación del efector final del manipulador, calcule todos los conjuntos posibles de ángulos articulares que podrían utilizarse para obtener esta posición y orientación dada. Este es un problema fundamental en el uso práctico de los manipuladores.

Este es un problema geométrico algo complicado, que se resuelve de manera rutinaria miles de veces diariamente en el sistema humano y en otros sistemas biológicos. En el caso de un sistema artificial como un brazo robótico, necesitamos crear un algoritmo en la computadora de control que pueda realizar este cálculo. En ciertos casos, la solución de este problema es el elemento más importante en un sistema manipulador.

Podemos pensar en este problema como en una asignación de “ubicaciones” en el espacio cartesiano 3D, a “ubicaciones” en el espacio de articulaciones internas del robot. Esta necesidad surge naturalmente siempre que se especifica un objetivo en coordenadas de espacio 3D externas.

El problema de la cinemática inversa no es tan simple como el de la cinemática directa. Debido a que las ecuaciones cinemáticas son no lineales, su solución no es siempre sencilla (o incluso posible) en una forma cerrada. Además, surgen preguntas sobre si existe una solución o existen múltiples soluciones.

El estudio de estas cuestiones hace que apreciemos lo que la mente humana y el sistema nervioso logran realizar cuando, al parecer inconscientemente, movemos y manipular objetos con nuestros brazos y manos.

La existencia o inexistencia de una solución cinemática define el espacio de trabajo de un manipulador dado. La falta de una solución significa que el manipulador no puede obtener la posición y orientación deseadas, ya que se encuentran fuera del espacio de trabajo del manipulador.

Además de tratar con los problemas de posicionamiento estático, es conveniente analizar los manipuladores en movimiento. Cuando, al realizar el análisis de velocidad de un mecanismo, es conveniente definir una matriz conocida como el Jacobiano del manipulador. El Jacobiano especifica la asignación de las velocidades en el espacio de articulaciones a las velocidades en espacio cartesiano. La naturaleza de esta asignación cambia a medida que varía la configuración del manipulador. En ciertos puntos, a los cuales llamaremos Singularidades, esta asignación no es invertible. Es importante que los usuarios tengan una comprensión de este fenómeno.

Otro campo a tener en cuenta es la dinámica, que se dedica al estudio de las fuerzas que se requieren para ocasionar el movimiento. Si bien, nuestro objetivo es la cinemática inversa (nuestro esquema es cinemático puro), abordaremos la dinámica sin profundizar, ya que no podemos pasarla por alto cuando diseñamos un brazo mecánico de las características del nuestro. Para poder acelerar un manipulador desde una posición inerte, deslizarlo a una velocidad constante del efector final y finalmente desacelerarlo hasta detenerlo completamente, los actuadores de las articulaciones deben aplicar un complejo conjunto de funciones de momento de torsión. La forma exacta de las funciones requeridas de momento de torsión de un actuador depende de los atributos espaciales y temporales de la ruta tomada por el efector final y de las propiedades de masa de los vínculos y de la carga útil; de la fricción en las articulaciones y demás factores que intervienen en la dinámica.

Muchos de nosotros hemos experimentado la sensación de levantar un objeto que es realmente mucho más ligero de lo que esperábamos. Este mal cálculo de la carga útil puede producir un movimiento de levantamiento inusual. Este tipo de observación indica que el sistema de control humano es más sofisticado que un esquema climático puro.

Un segundo uso de las ecuaciones de dinámica del movimiento es la simulación. Al reformular las ecuaciones dinámicas, de manera que la aceleración se calcule como una función del momento de torsión de un actuador, es posible simular como se moverá un manipulador bajo la aplicación de un conjunto de momentos de torsión de un actuador.

Si bien el uso de una placa Arduino, la cual su capacidad de procesamiento es limitada, no nos permite realizar este tipo de simulación.

Retomando la cinemática inversa, considerando las limitaciones de nuestra placa, nos encontramos con varias problemáticas previas al realizar cualquier tipo de cálculo, que debemos resolver para poder trasladar en el momento que se planteen dichas ecuaciones.

Una pregunta que nos surge en el inicio es la existencia de alguna solución que surja otra sobre el espacio de trabajo del manipulador. En general, el espacio de trabajo es el volumen de espacio que puede ser alcanzado por el efector final del manipulador. Para que exista una solución el punto de destino debe estar dentro del espacio de trabajo: espacio de trabajo diestro es el volumen de espacio que puede alcanzar el efector final del robot con todas las orientaciones; esto es, en cada punto del espacio de trabajo diestro, el efector final puede orientarse arbitrariamente. El espacio de trabajo alcanzable es el volumen de espacio

que puede alcanzar el robot en por lo menos una orientación. Evidentemente, el espacio de trabajo diestro es un subconjunto del espacio de trabajo alcanzable.

Con el fin de lograr una explicación más resumida, supongamos que tenemos un manipulador de dos vínculos (el nuestro es de tres vínculos), si $L_1=L_2$ (siendo el tamaño de cada vínculo), entonces el espacio de trabajo alcanzable consiste en un disco de radio $2L_1$. El espacio de trabajo diestro consiste en un solo punto, el origen. Si $L_1 \neq L_2$, entonces no hay espacios de trabajo diestro y el espacio de trabajo alcanzable se convierte en un anillo con un radio exterior de L_1+L_2 y un radio interior de $|L_1-L_2|$. Dentro del espacio de trabajo alcanzable hay dos posibles orientaciones del efecto final. En los límites del espacio de trabajo solo hay una orientación posible.

Estas consideraciones de espacio de trabajo para el manipulador de dos vínculos suponen que todas las articulaciones pueden girar 360° . Esto casi nunca sucede para mecanismos reales. Cuando los límites de las articulaciones son un subconjunto de los 360° completos, es obvio que el espacio de trabajo se reduce en forma correspondiente, ya sea en extensión o en el número de orientaciones posibles que puedan obtenerse.

Cuando un manipulador tiene menos de seis grados de libertad, no puede alcanzar posiciones y orientaciones para un destino arbitrario en el espacio 3D. En muchas situaciones realistas se emplean manipuladores con tres, cuatro o cinco grados de libertad que operan fuera de un plano, pero que evidentemente no pueden alcanzar destinos arbitrarios. Cada uno de esos manipuladores debe estudiarse para comprender su espacio de trabajo. En general, el espacio de trabajo de dicho robot es un subconjunto de un subespacio que puede asociarse con cualquier robot en particular. Dada una especificación de trama de destino general, surge un problema interesante en relación con los manipuladores que tienen menos de seis grados de libertad: ¿cuál es la trama de destino más cercana que puede obtenerse?

El espacio de trabajo también depende de la transformación de la trama herramienta, ya que generalmente es la punta de la herramienta sobre la que se discute cuando hablamos de puntos alcanzables en el espacio.

Además de la existencia o no de una solución, dependiendo del alcanza que nuestro brazo posee según sus restricciones, también debemos tener presente la problemática de las distintas soluciones que pueden existir al resolver nuestras ecuaciones de cinemática.

Nuestro brazo planar de 3 articulaciones angulares tiene un espacio de trabajo diestro en el plano extenso, teniendo una buena longitud de vínculos y rangos de

articulaciones extensos, ya que cualquier posición en el interior de espacio de trabajo puede alcanzarse con cualquier orientación. En la siguiente la Fig. 35 las líneas punteadas indican otra trayectoria.

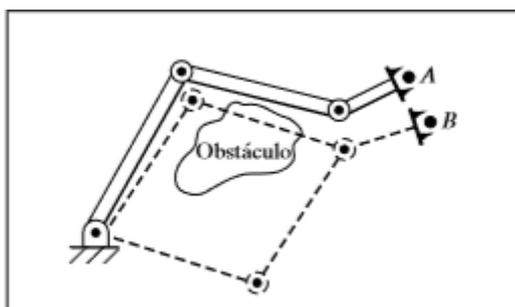


Fig. 35 Extraída del Libro Robótica, John J, Craig. Tercera Edición, 2006

El hecho de que un manipulador tenga múltiples soluciones puede ocasionar problemas, ya que el sistema tiene que ser capaz de elegir una. Los criterios sobre los que debe basarse una decisión varían, pero una elección muy razonable sería la solución más cercana. No obstante, la noción de “cercano” podría definirse de varias formas. Esto dependería si hay obstáculos, los grados de libertad del brazo.

El número de soluciones depende del número de articulaciones en el manipulador, pero también es una función de los parámetros de vínculo y los rangos de movimiento permisible de las articulaciones. En general entre más parámetros de vínculos distintos de cero tengamos, más formas habrá de llegar a un cierto destino. Entonces, cuantos más parámetros distintos de cero, mayor será el máximo número de soluciones.

Todo esto nos lleva a tener diferentes métodos de solución. A diferencia de las ecuaciones lineales, no hay algoritmos generales que puedan emplearse para resolver un conjunto de ecuaciones no lineales. Al considerar métodos de solución, es conveniente definir lo que constituye la “solución” de un manipulador dado.

Se considera que un manipulador tiene solución si las variables de articulación pueden determinarse mediante un algoritmo que nos permita determinar todos los conjuntos de variables de articulaciones asociados con una posición y orientación dadas.

EL punto principal de esta definición es que, en el caso de múltiples soluciones, requerimos que sea posible calcular todas las soluciones. Por lo tanto, no consideramos que

ciertos procedimientos iterativos numéricos resuelvan el manipulador en la medida en que no se garantiza que esos métodos encuentren todas las soluciones.

Dividiremos todas las estrategias de solución de manipuladores propuesta en dos clases amplias: soluciones de forma cerrada (analítica) y soluciones numéricas. Debido a su naturaleza iterativa, las soluciones numéricas son generalmente mucho más lentas que las soluciones de forma cerrada correspondiente; de hecho, son tan lentas que para casi todos los fines prácticos no estaremos interesados en el enfoque numérico para la solución de problemas cinemáticos. La solución iterativa numérica para las ecuaciones cinemáticas es todo un campo de estudio en sí y se encuentra más allá de nuestro alcance.

Restringiendo nuestro enfoque a los métodos de solución cerrada. En este contexto, “forma cerrada” significa un método de solución basado en expresiones analíticas o en la solución de un polinomio de grado 4 o menor, de tal forma que basta con cálculos no iterativos para llegar a la solución. Dentro de la clase de soluciones de forma cerrada, distinguimos dos métodos para obtener la solución: algebraico y geométrico. Estas distinciones son algo confusas: cualquier método geométrico a considerar aplica por medio de expresiones algebraicas, por lo que los dos métodos son similares. Los métodos tal vez difieren solo del enfoque.

Un importante resultado en la cinemática es que, de acuerdo con nuestra definición de solubilidad; todos los sistemas con articulaciones angulares y prismáticas que tengan un total de seis grados de libertad en una sola cadena en serie pueden resolverse. No obstante, esta solución general es numérica. Solo en casos especiales los robots de 6 grados de libertad pueden resolverse analíticamente. Estos robots para los que existe una solución analítica se caracterizan ya sea por tener varios ejes de articulaciones perpendiculares o muchas iguales a 0 ± 90 grados. Generalmente el cálculo de soluciones numéricas consume mucho tiempo cuando se le compara con la evaluación de expresiones analíticas; por lo tanto, se considera muy importante diseñar un manipulador de manera que exista una solución de forma cerrada.

Una condición suficiente para que un manipulador tenga una solución de forma cerrada, es que tres ejes de articulación adyacentes se intersequen en un punto.

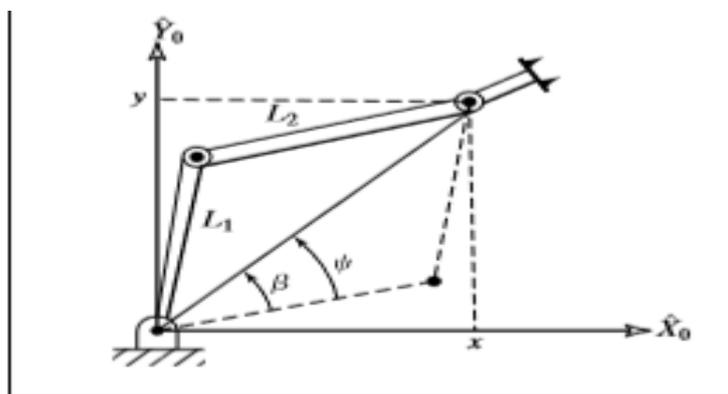
Método Geométrico:

Para poder lograr nuestro objetivo y teniendo en cuenta las problemáticas descriptas, decidimos utilizar el método geométrico.

Si bien dentro de la cinemática Inversa hay varios métodos, la simplicidad de método geométrico nos permite poder procesar las ecuaciones que el método utiliza con el Arduino, ya que con matemáticas más complejas el Arduino no sería capaz de procesar.

En un método geométrico para encontrar la solución de un manipulador, tratamos de descomponer la geometría espacial del brazo en varios problemas de geometría plana.

Para muchos manipuladores, esto puede hacerse con mucha facilidad, debido a que ángulos de articulación pueden resolverse utilizando las herramientas de la geometría plana. Para el brazo con tres grados de libertad que diseñamos, como dicho brazo es planar podemos aplicar directamente la geometría plana para encontrar la solución. Se muestra en la Fig. 36.

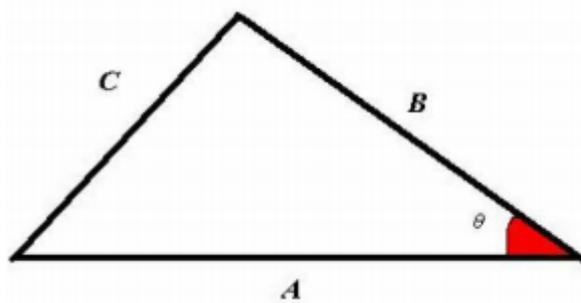


Posibles soluciones

Fig. 36 Extraída del Libro Robotica, John J, Craig. Tercera Edición, 2006

Considerando el triángulo que se forma, podemos aplicar la ley de los cosenos y resolver para θ_1 y θ_2 .

Ley de los cosenos:



Siendo que: $A^2 + B^2 - 2AB\cos(\theta) = C^2$

Esto aplicado a nuestro caso nos da:

$$X^2 + Y^2 = l_1^2 + l_2^2 - 2l_1l_2 \cos(180 + \theta_2)$$

Ahora; $\cos(180 + \theta_2) = -\cos(\theta_2)$, por lo que tenemos:

$$C_2 = \frac{X^2 + Y^2 - l_1^2 - l_2^2}{2l_1l_2}$$

Para que este triángulo exista, la distancia al punto de destino $\sqrt{X^2 + Y^2}$ debe ser menor o igual que la suma de las longitudes de los vínculos, $l_1 + l_2$. Esta condición se comprobaría en este punto en un algoritmo computacional para verificar la existencia de soluciones. Esta condición no se satisface cuando el punto de destino está fuera del alcance del manipulador. Suponiendo que exista una solución, esta ecuación se resuelve para aquel valor de θ_2 que se encuentre entre 0 y 180 grados, ya que solo para estos valores existe el triángulo que describimos en la figura.

Para poder despejar θ_1 encontramos expresiones para los ángulos ψ y β según lo indicado en nuestra figura. En primer lugar, β puede estar en cualquier cuadrante, dependiendo de los signos de X y de Y. Por ello debemos usar una función arco tangente de dos argumentos:

$$\beta = \text{Atan2}(Y, X).$$

De nuevo aplicamos la ley de los cosenos para encontrar ψ :

$$\text{Cos } \psi = \frac{X^2 + Y^2 + l_1^2 - l_2^2}{2l_1\sqrt{X^2 + Y^2}}.$$

Aquí el arco coseno debe resolverse de manera que $0 < \psi < 180$ grados, para poder preservar la geometría que nos lleva a la última ecuación planteada ($\cos \psi$). Estas consideraciones son típicas cuando se utiliza un método geométrico; debemos aplicar la fórmula que derivamos solamente sobre el rango de variables tal que se preserve la geometría. Entonces tenemos que:

$$\theta_1 = \beta \pm \psi.$$

En donde se utiliza el signo positivo si $\theta_2 < 0$ y el signo negativo si $\theta_2 > 0$.

Teniendo presente todas estas consideraciones, llevando todo esto mencionado a nuestro brazo, considerando las variables y longitudes diseñadas en nuestro modelo, podemos aplicar este método para su resolución obteniendo el siguiente resultado:

$$q_1 = \cos^{-1}\left(\frac{p_y}{p_x}\right)$$

$$r^2 = p_x^2 + p_y^2$$

$$r^2 + p_z^2 = l_2^2 + l_3^2 + 2l_2l_3 \cos q_3$$

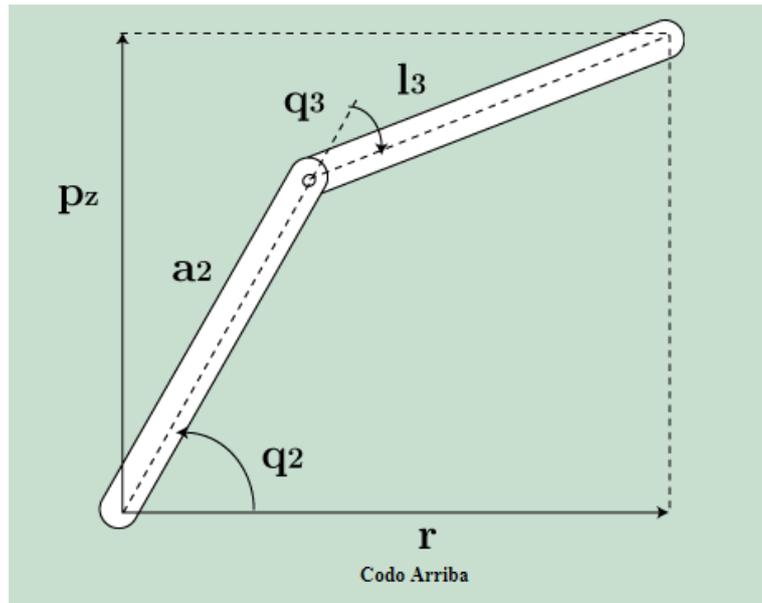
$$\cos q_3 = \frac{p_x^2 + p_y^2 + p_z^2 - l_2^2 - l_3^2}{2l_2l_3}$$

$$\sin q_3 = \pm \sqrt{\cos^2 q_3}$$

$$q_3 = \tan^{-1}\left(\frac{\pm \sqrt{1 - \cos^2 q_3}}{\cos q_3}\right)$$

Siendo: $\cos q_3 = \frac{p_x^2 + p_y^2 + p_z^2 - l_2^2 - l_3^2}{2l_2l_3}$

Como puede observarse en la resolución de q_3 , existen dos soluciones posibles según el signo que se le dé al resultado de la raíz, esto se refiere a que el brazo puede alcanzar el mismo punto con el “codo” hacia arriba o hacia abajo. Ver Fig. 37



Nosotros utilizaremos el “codo hacia arriba” para nuestro proyecto.

Fig. 37 Extraída de la Web: <http://www.Kramirez.net/>

En las siguientes imágenes (Fig. 38 y Fig. 39) se ilustrará nuestro modelo:

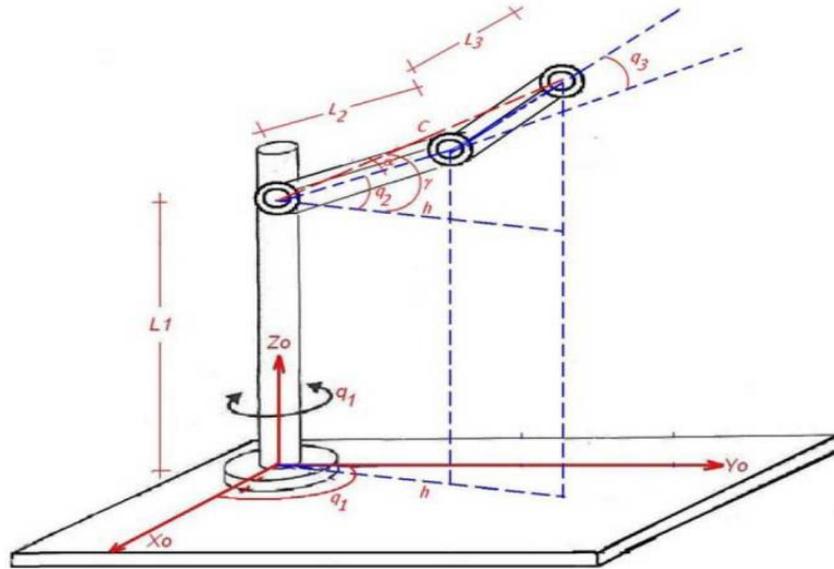


Fig. 38 Extraída de la Web: <http://www.Kramirez.net/>

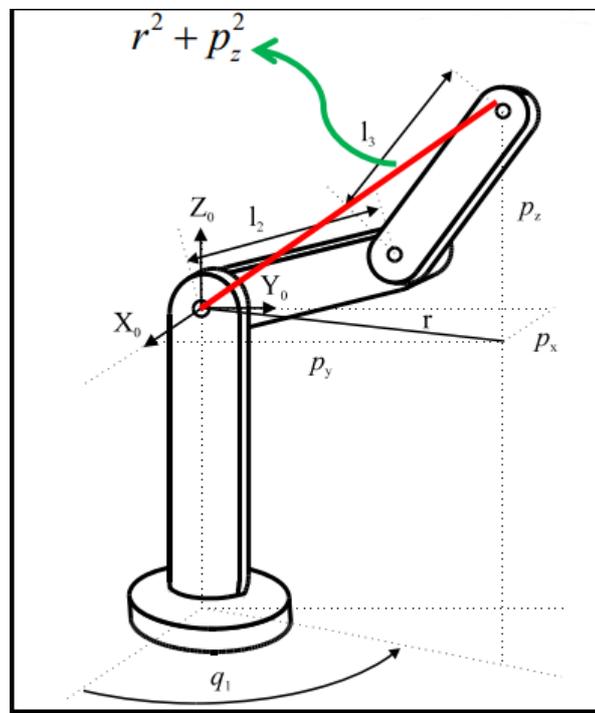


Fig. 39 Extraída de la Web: <http://www.Kramirez.net/>

Como resultado de este método, poder extraer 3 ecuaciones que nos permiten tener los valores de los ángulos necesarios para poder alcanzar el objeto y que la placa Arduino es capaz de resolver.

Ecuaciones de Método Geométrico:

$$1) q_1 = \arctg \frac{P_y}{P_x}$$

$$2) q_2 = \arctg \left(\frac{l_3 * \text{sen } q_3}{l_2 + l_3 \cos q_3} \right) + \arctg \left(\frac{p_z}{\sqrt{p_x^2 + p_y^2}} \right)$$

$$3) q_3 = \arctg \left(\frac{\sqrt{1 - \cos^2 q_3}}{\cos q_3} \right)$$

Otros Métodos de Cinemática inversa:

Si bien estos métodos no fueron utilizados, es importante tenerlos presentes en caso de que se quiera cambiar la placa Arduino por alguna con más procesamiento.

- A- Matrices de transformación homogénea,
- B- Desacoplamiento cinemático
- C- Algebra de tornillos
- D- Cuaternios
- E- métodos iterativos.

Todos estos métodos tienen en cuenta la forma del robot y la existencia de distintas posibles soluciones, no siempre existe una solución de forma cerrada.

En base a estas consideraciones surgen las siguientes posibles soluciones en cinemática inversa:

A- Procedimiento genérico a partir de los parámetros D-H, en honor a Denavit-Hartenberg.

Método iterativo

Problemas de Velocidad y Convergencia.

B- Búsqueda de solución cerrada:

- 1- Posibilidad de resolución en tiempo real.
- 2- Posibilidad de selección de la solución más adecuada.
- 3- Posibilidad de simplificaciones
- 4- No siempre es posible.

Entrando un poco en detalle, no mucho, como se aclaró en la presentación de estos métodos, vamos a indagar en los siguientes métodos:

Matriz De Transformación Homogénea:

En principio es posible tratar de obtener el modelo cinemático inverso de un robot a partir del conocimiento de su modelo directo. Es decir, suponiendo conocidas las relaciones que expresan el valor de la posición y orientación del extremo del robot en función de sus coordenadas articulares, obtener por manipulación de aquellas de las relaciones inversas.

Sin embargo, en la práctica esta tarea no es trivial siendo en muchas ocasiones tan compleja que obliga desecharla.

Desacoplamiento Cinemático, los procedimientos vistos permiten obtener los valores de las 3 primeras variables articulares del robot, aquellas que posición su extremo en las coordenadas (p_x, p_y, p_z) determinadas, aunque pueden ser igualmente utilizadas para la obtención de las 6 a costa de una mayor complejidad

En general no basta con posicionar el extremo del robot en un punto del espacio, sino que es preciso conseguir que la herramienta se oriente de una manera determinada.

Para ello, los robots cuentan con otros tres grados de libertad adicionales, situados al final de la cadena cinemática y cuyos ejes, generalmente, se cortan en un punto, que informalmente se denomina muñeca del robot. Si bien la variación de estos tres últimos grados de libertad

origina un cambio en la posición final del extremo real del robot, su verdadero objetivo es poder orientar la herramienta del robot libremente en el espacio. Este método saca partido de este hecho, separando ambos problemas: Posición y orientación.

Para ello, dada una posición y orientación final deseadas, establece las coordenadas del punto de corte de los 3 últimos ejes (muñeca del robot) calculándose los valores de las tres primeras variables articulares (q_1, q_2, q_3) que consiguen posicionar este punto. A partir de los datos de orientación y de los ya calculados (q_1, q_2, q_3) se obtiene los valores del resto de las variables articulares.

Cuaternios:

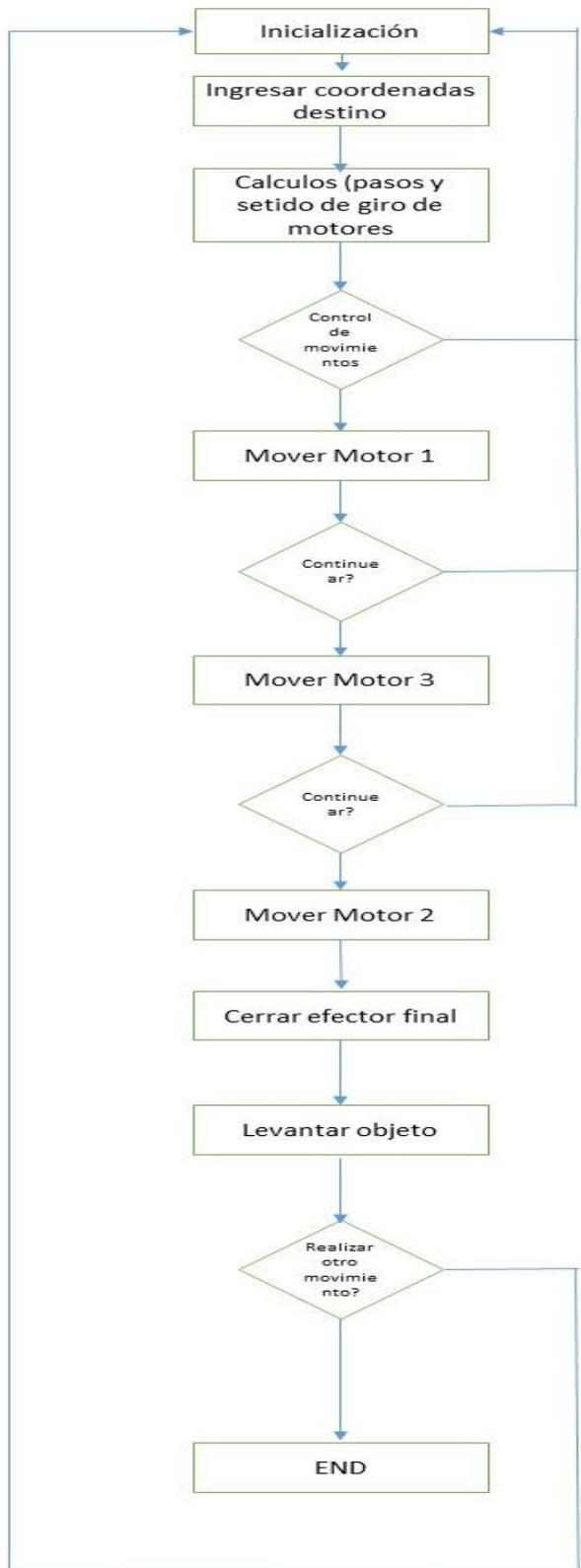
Los cuaternios Q están compuestos por cuatro componentes (q_0, q_1, q_2, q_3) que representan las coordenadas del cuaternio en una base $\{e, i, j, k\}$. A la componente q_0 del cuaternio se le suele denominar “parte escalar” y al resto, “parte vectorial”. Los cuaternios son herramientas matemáticas de gran versatilidad para trabajar con giros y cambios de orientaciones.

3.3.5 Percepción del entorno:

La mayoría de los robots solo cuentan con los sensores necesarios para controlar su posición y, por lo tanto, no son capaces de reconocer su entorno. Esto no suele ser un problema, puesto que la mayoría de los robots se encargan de tareas simples y repetitivas que se han programado previamente. Cuando se quiere crear una máquina más versátil, por ejemplo, que trabaje con piezas diferentes, será necesario dotarlo de sistemas de reconocimiento de su entorno.

En nuestro caso fueron utilizados sensores tipo switch para determinar las limitaciones de trabajo del brazo. Los sensores utilizados fueron:

A continuación, puede verse el diagrama de flujos del programa que responde a los movimientos dados por la cinemática inversa y en el anexo 8, se encuentra el código de dicho programa.



3.4. Prácticas para alumnos

3.4.1. Añadir un grado de libertad al robot

Los alumnos deberán agregar un grado de libertad al brazo robótico existente, se propone y se adjuntan elementos necesarios (motores, soportes, sujeciones, etc.) para hacerlo en el efector final.

Los alumnos deberán cumplir los siguientes ítems:

- Modificar mecánicamente el brazo robótico existente para agregar un grado de libertad
- Modificar eléctricamente el brazo robótico existente para que dicho grado de libertad agregado funcione correctamente
- Agregar sensores de home y away para limitar los movimientos de dicho motor
- Modificar el programa de control del Joystick usando los botones libres (F y E) para que el nuevo motor se mueva al ser accionado dichos botones
- Modificar el programa de cálculo de coordenadas para adaptarlo al nuevo esquema de movimiento

3.4.2. Comparación de cálculos

Los alumnos deberán comparar matemáticamente si los movimientos realizados por la cinemática inversa calculada por el Arduino son correcta, sacar desviaciones entre la teoría y la práctica y evaluar posibles caminos u opciones para las diversas coordenadas y diferentes opciones de movimientos han de realizarse.

3.4.3. Modificar el programa del Joystick actual para evitar cruces de motores

El programa actual tiene ciertos bugs que los alumnos deberán detectar y corregir. Como guía podremos decir que con el software actual es posible combinar ciertos movimientos de motores los cuales anulan movimientos de los otros. Esto se debe a que los sensores de home de ambos motores están conectados en serie para simplificar el cableado, lo mismo sucede

con los sensores de away. Los alumnos deberán buscar la manera de evitar este tipo de situación

3.4.4. Modificar el programa de cinemática inversa para realizar movimientos consecutivos sin necesidad de volver al home

El programa actual tiene unas variables virtuales llamadas “Encoder 1, Encoder 2 y Encoder 3” dichas variables son básicamente un contador que incrementa o decrementa su valor según el motor correspondiente (M1, M2, o M3) gire en sentido horario o anti horario. La idea de estos contadores, es para que los alumnos por medio de la programación, puedan realizar un programa con “Encoders Virtuales” de tal forma que sea posible saber en todo momento la posición de cada motor, y por consiguiente poder realizar movimientos consecutivos según coordenadas ingresadas sin necesidad de volver a la posición de origen

Experiencia en el armado y desarrollo del brazo:

Durante el desarrollo de este proyecto nos encontramos con varios desafíos que fuimos superando hasta lograr el correcto funcionamiento del brazo.

Esto problemas fueron principalmente en la parte física a la hora de ensamblar las piezas y también en el desarrollo de la programación de la placa Arduino.

Por el lado de ensamblado nos topamos en principio con la problemática de unir las cajas de reducción con los ejes del motor, ya que ambas partes eran de máquinas diferentes y poseían los enganches necesarios para cada máquina. Esto nos obligo a fabricar uniones para los ejes basándonos en el concepto de los acoples de manguito.

Una vez resulto esto, necesitábamos acoplar el motor de la base con el rodamiento axial. Por este lado nos topamos con el problema de que no teníamos una polea y una correa de las dimensiones que teníamos ya armada en la base del motor. Debido a esto, buscamos correas y poleas de una medida estándar lo más cercano a nuestro diseño y acomodamos el mismo a la longitud de nuestra correa.

Por ultimo y una de las problemáticas que nos demoro mucho fue la elección del correcto driver para poder controlar los motores paso a paso. Haciendo varias pruebas en las cuales quemamos mas de un driver logramos encontrar el adecuado, pero aun así se nos seguían quemando y en algunos casos funcionaban durante un periodo de tiempo y luego empezaban

a saltar pasos o hacer cosas sobre el movimiento totalmente absurdas. Luego de leer mucho entendimos que la temperatura que alcanzaban por el exceso de corriente necesario para poder mover los motores hacía que su funcionamiento sea totalmente errático. Para poder solucionarlo incorporamos los coolers.

Una vez superados estos problemas físicos, entramos en el mundo de la programación del Arduino la cual nos trajo varios problemas por la limitación en su capacidad de procesamiento. No solo eso, sino además que esta placa tiene varios problemas como por ejemplo, notábamos que de repente y sin ninguna explicación lógica, el programa dejaba de responder de la misma forma en que lo venía haciendo, hecho que nos ocurrió en reiteradas ocasiones y por el cual nuestras primeras teorías fueron que la causa eran falsos contactos eléctricos y ruido eléctrico provocado por las bobinas de los motores. Pero finalmente notábamos que el problema se solucionaba simplemente reiniciando la placa Arduino o recargando nuevamente el mismo programa que tenía cargado previamente.

El problema de usar el reset varias generaba que todo vuelva al inicio y que nuestra rutina del brazo se demore en realizar y que por ende nuestros motores estén mayor tiempo en funcionamiento y que los drivers recalienten.

Esta problemática, si bien, no logramos erradicarla totalmente. Su aparición disminuyo bastante luego de agregar una fuente de alimentación independiente de 5v para la placa Arduino únicamente y utilizando una fuente diferente para los drivers, motores, colores, etc. Teniendo ya todo montado y funcionando en forma manual usando el joystick, teníamos que resolver la programación de la cinemática inversa.

Con las ecuaciones de cinemática inversa y con la programación de estas en el lenguaje de Arduino, no lográbamos hacer que nuestro brazo vaya a la ubicación ingresada por coordenadas. Este problema nos llevo a replantear y modificar las ecuaciones varias veces y a la idea de usar encoders para poder ubicar el brazo en un sistema de coordenadas conocido. Luego de varias pruebas y modificaciones logramos encontrar cual era el problema sin tener que recurrir a algún otro dispositivo para poder ubicar nuestro brazo. La solución fue ponerle una función de HOME que lleva el brazo a esa posición. Dicha posición es conocida y hace de referencia para los cálculos de cinemática inversa.

Conclusiones:

Podemos decir que hemos logrado realizar el objetivo del proyecto, si bien hubo complicaciones no contempladas como por ejemplo la necesidad de utilizar cajas reductoras para los movimientos de los motores M2 y M3, pudimos sortear dichos problemas y solucionarlos de una manera efectiva y funcional.

En lo que respecta a la plataforma Arduino, podemos decir que es bastante amistosa, hay suficiente información en internet como para poder aprender a utilizarla sin inconvenientes, que es muy versátil y viene por default preparada para una infinidad de tareas diferentes. Sin embargo, en varias ocasiones, se nos presentaron problemas aislados como ser, movimientos sin sincronización o falta de movimientos en ciertos momentos los cuales se resolvían simplemente recargando el programa nuevamente en la placa Arduino, lo que nos da la pauta que no es un sistema muy confiable ya que por alguna razón pierde o se corrompe el código que tiene cargado y es necesario recargarlo nuevamente, por lo que su uso en un ambiente industrial es inviable ya que no brinda la seguridad requerida.

Por otro lado, los controladores de los motores paso a paso están trabajando muy al límite y en ocasiones presentan movimientos erráticos debido a exceso de temperatura, por lo que deben estar bien ventilados durante toda su operación.

Si bien esta placa Arduino Mega250 no es fiable debido a las fallas anteriormente mencionadas, sería muy interesante poder utilizar otra placa tipo por ejemplo un Raspberry Pi, con el mismo diseño de brazo, ya que ese tipo de placas son más confiables y tienen una capacidad de procesamiento superior a la Arduino, proporcionándonos la posibilidad de utilizar otro método de cinemática inversa más complejos. De nuestra parte no utilizamos ese tipo de placa, ya que no estaríamos respetando la consigna de nuestra Tesis que fue bien explicitado el uso de Arduino. Sumado a esto, la poca experiencia que teníamos sobre Arduino, no nos permitió evaluar esta posibilidad previa al inicio de nuestro proyecto.

En referencia a las prácticas para los alumnos de Mecatrónica, creemos que les estamos dejando una herramienta muy valiosa para poder ver en la práctica gran parte de los temas vistos en la teoría y además la posibilidad de interactuar con un prototipo funcional y poder realizar modificaciones en el mismo con el fin de mejorarlo, comprobar sus limitaciones y entender su funcionamiento.

Bibliografía:

- 1- Introducción a la robótica. Thompson Paraninfo, 2005.Madrid. ANGULO, J., ANGULO, I., ROMERO, S.
- 2- Fundamentos de robótica. McGraw-Hill, 1997. Madrid. BARRIENTOS, A., PEÑÍN, L.F, BALAGUER, C.
- 3- ROBOTICA (John J, Craig. Tercera Edición, 2006)
- 4- MATLAB. Release Notes. MathWorks Corporation
- 5- Página oficial de Arduino <http://www.arduino.cc/>
- 6- [http://coffeebrain.org/wikicontrol/index.php?title=\(1620-IELE3330-TC\) Proyecto: Brazo Rob%C3%B3tico con tres grados de libertad Grupo J](http://coffeebrain.org/wikicontrol/index.php?title=(1620-IELE3330-TC) Proyecto: Brazo Rob%C3%B3tico con tres grados de libertad Grupo J)
- 7- <http://www.Kramirez.net/>

ANEXOS