

Design and Implementation of a Pick and Place System Using a 4 DOF Robotic Arm Integrated into a Longitudinal Horizontal Movement Base

Marcelo Jaime Quispe Ccachuco, Dr.¹, César Pio Castillo Cáceres, Dr.², Alejandro Flavio Ponce Solórzano, Br.³, y Jorge Luis Castro Valdivia, Dr.⁴.

^{1,4}Universidad Católica de Santa María, Perú, mquispec@ucsm.edu.pe, jlcastro@ucsm.edu.pe

^{2,3}Universidad Católica de Santa María, Perú, ccastillo555@yahoo.es, alejandro.ponce@ucsm.edu.pe

Abstract— The demand of the industry has motivated articulated robotic arms to increase their work area, giving rise to the need to add a greater amount of longitudinal movements in the robots. The mobile manipulators that are manufactured today have a gyration ring driven by a motor at its base and move on two toothed rails that allow it to move along the rails with precision, in other more complex systems the manipulator robot goes on top of a mobile robot doing navigation.

The present project consists of providing a form of integrated longitudinal movement, of the translational displacement type, to a 4DOF robotic arm. This translational movement will be generated by a stepper motor connected through an endless screw with which the translational base will slide where the robotic arm will be attached, like CNC laser printing and cutting machines. The purpose of the system is to have a cyclical pick and place sequence. To validate the design, simulations of efforts present in each component and the torque necessary to move them, in the maximum state of reach of the robot, were carried out, obtaining the most critical points of torsion moments and forces generated in the base and the screws. robot clamping. The equations that govern the positioning of the entire system were calculated, the numerical values of these equations were implemented in a Matlab code, which sends the signals to the Arduino to control the movement of the servomotors.

The program implemented in Matlab allows programming the autonomous and repetitive movement by fixed coordinates and variants for specific movement points, successful performance tests were carried out, for a mobile base of 900 mm the time of 30 seconds round trip was used, you can do faster, but you lose precision in scrolling.

Keywords—Servomotor, PWM, mobile manipulator, DOF, simulation.

Digital Object Identifier: (only for full papers, inserted by LACCEI).
ISSN, ISBN: (to be inserted by LACCEI).
DO NOT REMOVE

Diseño e Implementación de un Sistema Pick and Place Empleando un Brazo Robótico de 4 GDL Integrado a una Base de Movimiento Longitudinal Horizontal

Marcelo Jaime Quispe Ccachuco, Dr.¹, César Pio Castillo Cáceres, Dr.², Alejandro Flavio Ponce Solórzano, Br.³, y Jorge Luis Castro Valdivia, Dr.⁴.

^{1,4}Universidad Católica de Santa María, Perú, mquispec@ucsm.edu.pe, jlcastro@ucsm.edu.pe

^{2,3}Universidad Católica de Santa María, Perú, ccastillo555@yahoo.es, alejandro.ponce@ucsm.edu.pe

Resumen- La exigencia de la industria ha motivado que los brazos robóticos articulados aumenten su área de trabajo, dando lugar a la necesidad de agregar una mayor cantidad de movimientos longitudinales en los robots. Los manipuladores móviles que se fabrican en la actualidad tienen un anillo de giro accionado por un motor en su base y se desplazan sobre dos rieles dentados que le permite desplazarse a lo largo de los rieles con precisión, en otros sistemas más complejos el robot manipulador va encima de un robot móvil haciendo una navegación.

El presente proyecto consiste en proporcionar una forma de movimiento longitudinal integrado, de tipo desplazamiento traslacional a un brazo robótico de 4GDL. Este movimiento traslacional será generado por un motor paso a paso conectado a través de un tornillo sin fin con el cual se deslizará la base traslacional donde se tendrá sujeto el brazo robótico, semejante a las máquinas CNC de impresión y corte laser. El propósito del sistema es tener una secuencia tipo pick and place de manera cíclica. Para validar el diseño se realizaron simulaciones de esfuerzos presentes en cada componente y el torque necesario para moverlos, en el estado máximo de alcance del robot, obteniendo los puntos más críticos de momentos de torsión y de fuerzas que se generan en la base y los tornillos de sujeción del robot. Se calcularon las ecuaciones que gobiernan el posicionamiento de todo el sistema, los valores numéricos de estas ecuaciones fueron implementados en un código de Matlab, el cual, envía las señales al Arduino para controlar el movimiento de los servomotores.

El programa implementado en Matlab permite programar el movimiento autónomo y repetitivo por coordenadas fijas y variantes para puntos de movimiento específicos, se realizaron pruebas de funcionamiento con éxito, para una base móvil de 900 mm se empleó el tiempo de 30 segundos ida y vuelta, se puede hacer más rápido, pero se pierde precisión en el desplazamiento.

Palabras clave - Servomotor, PWM, robot manipulador móvil, GDL, simulación.

I. INTRODUCCIÓN

Actualmente las industrias se concentran cada vez más en automatizar gran parte de sus procesos industriales con el fin de disminuir todos aquellos riesgos que potencialmente afectan la integridad del ser humano [1]. Todo ello con el fin de generar una mayor eficiencia en su producción, realizando procesos con

un menor coste y generando una mayor rentabilidad. Esto da como resultado la necesidad de gestionar e implementar una innovación en el desarrollo de nuevos prototipos, diseños o mejoras para un determinado proyecto.

En una célula robotizada normalmente se encuentra un robot manipulador de 6 grados de libertad, la limitación de estos robots es su volumen de trabajo, ya que estos robots deben anclarse al suelo para tener bien su referencia; sin embargo, hay procesos en los que se requiere ir de un punto a otro llevando un componente. En recientes investigaciones [2-4] coinciden en llamar “mobile manipulator” a un brazo robótico sobre un robot móvil que usa ruedas omnidireccionales. La dificultad de estos sistemas es la planificación de trayectorias y la navegación.



Fig. 1 Robot manipulador on rail – Kuka, en la industria maderera

La empresa KUKA que fabrica los robots manipuladores para el ensamblado de autos ha innovado con un robot manipulador sobre un riel y lo ha denominado “robot manipulador on linear rail”. Como resultado, el manipulador de brazo robótico reposiciona horizontal o verticalmente su enlace de base a una ubicación deseada debido al cambio en los requisitos de trabajo, a este sistema [5] le denomina: “Linear manipulator”. Los sistemas nombrados anteriormente usan dos rieles dentados y el robot tiene en su base un anillo de giro que

le permite desplazarse con precisión a lo largo de los rieles de un punto a otro [6].

Según como estén colocados los rieles, el robot puede desplazarse en forma vertical u horizontal. Hay manipuladores móviles que se usan para la inspección en subestaciones aisladas por gas, en esa aplicación el robot de inspección va montado en un riel en el techo y el robot va de cabeza [7].

En el presente proyecto se presenta una propuesta de movimiento usando un tornillo sin fin en la base del robot en vez de rieles y rueda dentada. Las impresoras 3D por ejemplo y muchas máquinas de control numérico usan motores de paso unidos a un tornillo sin fin para tener una precisión de sus movimientos. En consecuencia, se construyó el brazo robótico con acrílico para que sea lo más liviano posible.

Este diseño tiene la ventaja de que solo se necesite de un robot manipulador y no varios en un proceso productivo, porque esta base móvil permitirá que el área de trabajo sea mayor. La desventaja de este diseño está en la dificultad de resolver el problema de la cinemática inversa del sistema [8]. Para esto sería necesario analizar el brazo robótico con un grado de libertad más en la base y eso le daría un grado de libertad más al sistema original; luego hacer un programa de control de movimiento para todo el sistema.

II. MARCO TEÓRICO

A. Sistema de Control Autónomo

Uno de los más populares programas entonces para el modelado robótico es Matlab, que ya sea con el uso de toolbox o de sus librerías de origen, en el entorno simulink es posible realizar la implementación de un esquema que emula manipuladores robóticos, donde es posible estudiar el movimiento, calcular controladores, entre otras [9].

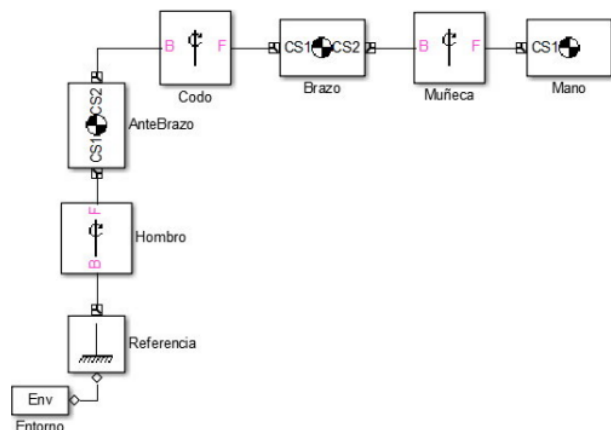


Fig. 2 Manipulador robótico expresado en bloques en Matlab [9]

En las etapas del desarrollo del movimiento autónomo, los valores que dependen del contexto de operación son las velocidades y las aceleraciones en las cuales se trabaja, estos son generados a partir de un sistema que define el tipo de accionamiento que se van a dar en los motores y servomotores para el movimiento controlado del brazo robótico, para el

posicionamiento de cada forma, operación y su realización del mismo procedimiento en diferentes secciones.

B. Fundamentos de la robótica

La estructura de un robot está compuesta por una serie de eslabones unidos por articulaciones de tipo rotacional o prismáticos y su generación de movimientos, estos movimientos quedan definidos por los siguientes elementos matemáticos: Las coordenadas cartesianas (x, y, z) y las coordenadas angulares (α, β, γ).

Tres características importantes de los robots manipuladores son la exactitud, repetibilidad y resolución. La repetibilidad es la medida de la habilidad de un robot para regresar a la misma posición. La exactitud se refiere a la capacidad de un robot para situar el extremo de su muñeca en un punto señalado dentro del volumen de trabajo. La resolución espacial se define como el incremento más pequeño de movimiento que puede ejecutar un robot.

Los movimientos de cada una de las articulaciones del robot manipulador pueden ser rotacional o prismático. Entonces existen 4 configuraciones posibles para un robot de 3 grados de libertad. Si se emplean servomotores se tendría la configuración 3G, que significa que son 3 rotacionales. La herramienta del robot que es una pinza generalmente también se cuenta como un grado de libertad. Por lo tanto, el robot que se va a trasladar de manera horizontal es un robot de 4 GDL, porque tiene 3 movimientos rotacionales y una pinza que se acciona con un servomotor para la apertura y cierre.

El desarrollo cinemático de los fundamentos de la robótica es el estudio del movimiento del robot, un análisis de la posición, velocidad y aceleración de cada uno de los elementos del robot calculados sin considerar las fuerzas que causan el movimiento, las relaciones entre los movimientos y las fuerzas y movimientos de torsión que se generan constituyen el problema de la dinámica, generando un estudio de la manipulación mecánica conocido como cinemática directa.

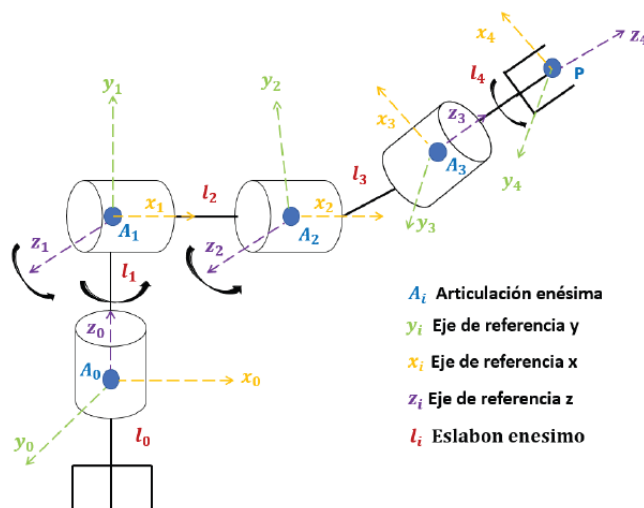


Fig. 3 Robot de articulación angular de 4 GDL [1]

Para el desarrollo de la cinemática directa se dan los métodos de análisis en los cuales se detallan los marcos de referencia necesarios para el modelamiento del prototipo robótico de 4 grados de libertad, efectuado a través de la transformación de matrices, geometría o transformación de coordenadas, para el desarrollo de un sistema de referencia al análisis matricial del robot generando los valores del control autónomo.

Del control cinemático se selecciona el tipo de trayectoria que las articulaciones o el efector final deben seguir. Uno de los métodos tradicionales es la trayectoria punto a punto, que se divide en movimiento eje a eje, simultaneo de ejes y asíncronas [8]. En base al desarrollo del cálculo de la cinemática directa en un robot articular, se desarrolla posterior a esto el cálculo de la cinemática inversa con el fin de determinar las coordenadas articulares a través de puntos de posición y orientación de las coordenadas finales.

La selección para el desarrollo de una cinemática inversa se resuelve por diferentes métodos, a través de las relaciones geométricas o relaciones en base a las coordenadas articulares con la finalidad de obtener las coordenadas extremas del robot en referencia a la localización espacial final, para su desarrollo se conforma con respecto a las coordenadas de origen, a través de las perspectivas del brazo abordando una vista periférica de todo el robot realizando un análisis trigonométrico con el fin de calcular los grados de articulación para utilizar las variables de posición y orientación como los valores de entrada.

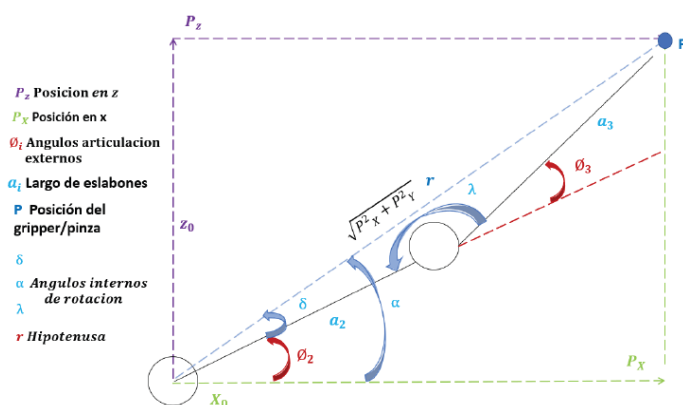


Fig. 4 Vista periférica para un análisis cinemático inverso [1]

En la relación cinemática no se tiene en cuenta las fuerzas o pares que actúan sobre el robot (actuadores, cargas, fricciones, etc.) y que pueden originar el movimiento de este [12], donde esta además va a depender del número de articulaciones para cada brazo en específico en la cual la velocidad será expresada para cada sistema de referencia en cada valor articular variable relacionándolas con las velocidades en el espacio, llamado como modelo diferencial donde la resolución de sus valores se realiza a través de una matriz Jacobiana en relación con la matriz homogénea hallada a través de Denavit-Hartenberg.

Una vez definidos los parámetros de velocidad en cada punto del brazo, se puede pasar a determinar las aceleraciones, las cuales son de vital importancia para el control de fuerza de manipuladores a lo largo de una trayectoria y permiten determinar el estado de aceleración de la herramienta [1], para ello es necesario determinar la aceleración que se requiere en cada ángulo de posición y la orientación, donde se debe realizar una derivada parcial a la Jacobiana generando de la misma una matriz final. La aceleración que dependerá del total de articulaciones y del tipo de movimiento realizado.

C. Características del riel de movimiento traslacional

En el mercado se encuentran diferentes sistemas de rieles con velocidad constante para la captura de imágenes, y algunos presentan dos rieles motorizados contracción por correa y motores paso a paso [14], por ello el desarrollo de un sistema propio a través de un tornillo de potencia donde la fuerza es suministrada por una fuente de poder, en el que a través del principio de movimiento tornillo de potencia-tuerca, se realiza el movimiento del tornillo de potencia.

Para un desarrollo de movimiento lineal, las estimaciones necesarias en el movimiento deben ser las calibraciones por la incertidumbre, los valores de velocidad de entrada y la aceleración del sistema construido, en la velocidad para un sistema se planteó un mecanismo de tornillo sin fin por la precisión que este puede dar y una estabilidad a una velocidad constante y variable dependiendo del control a realizar, para la selección se debe tener en cuenta el tipo de fuerza y los siguientes aspectos, el primero es la velocidad lineal que puede dar el sistema, esto se da a través de una tuerca de bronce que se une con la plataforma.

En cada una de las articulaciones es necesario la generación de un torque para el desarrollo del movimiento con velocidades constantes y/o variables, con una resolución alta para la disposición de un control constante de este, a la par que se desarrolle el movimiento generado mediante el software Matlab en base a la generación del movimiento determinado a partir de rutas de movimiento, el cual generará un movimiento hacia el motor paso a paso de manera autónoma.

III. MATERIALES Y MÉTODOS

A. Diseño del brazo robótico

Para el diseño del sistema se buscó tener el mayor rendimiento con el menor costo posible. Para la estructura de las articulaciones del robot se utilizó acrílico de 3 mm de grosor. Estas piezas de las articulaciones fueron cortadas con láser y ensambladas usando pernos y tuercas.

Para el diseño del sistema base de traslación del robot se tuvo en consideración que las fuerzas resultantes en la función de movimiento no puedan generar desalineamientos ni vibraciones, por eso se colocaron 2 guías de acero inoxidable paralelas al tornillo de potencia, además de un acople flexible en el eje del motor paso a paso. El movimiento lineal generado por el tornillo de potencia se dará en una longitud de 900 mm,

a pesar de estas consideraciones siempre se tendrá una fricción que se debe vencer cada vez que se realice los movimientos, esto generará una fuerza de fricción que el motor paso a paso deberá vencer, para realizar todo este diseño se utilizó el Software SolidWorks

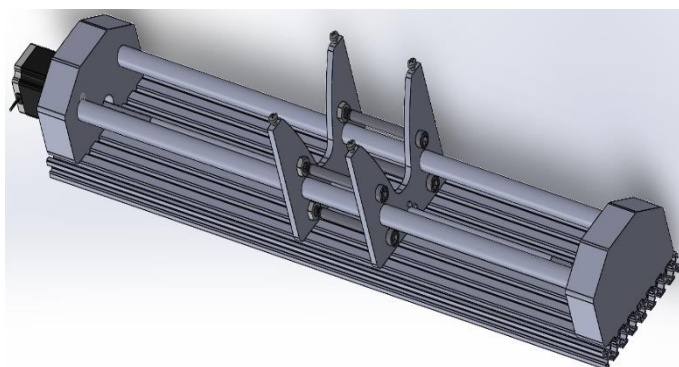


Fig. 5 Plataforma Móvil sobre riel de tornillo de potencia

La masa que debe trasladar el tornillo de potencia es de 1.055 Kg, el torque máximo de soporte suministrado por el motor paso a paso es de 9 kg.cm, de 200 pasos por revolución, es decir en cada paso gira 1.8°.

Un driver o controlador de paso TB6600 permite resoluciones más altas al permitir ubicaciones de pasos intermedios. Esto se consigue energizando las bobinas con niveles de corriente intermedios. Para obtener esas variaciones de pasos por revolución el controlador tiene 3 switches S1, S2 y S3 que se pueden configurar manualmente, de la siguiente manera.

TABLA I
CONFIGURACIÓN DEL DRIVER TB6600

S1	S2	S3	Resolución de micropasos
ON	ON	ON	NC
ON	ON	OFF	Paso completo
ON	OFF	ON	1/2 paso
OFF	ON	ON	1/2 paso
ON	OFF	OFF	1/4 de paso
OFF	ON	OFF	1/8 de paso
OFF	OFF	ON	1/16 de paso
OFF	OFF	OFF	1/32 de paso

En este proyecto se usó la configuración de 1/2 paso, para obtener 400 pasos por revolución, que equivale a 0.9° por paso. La señal que recibe el Driver TB6600 es una señal de pulsos de amplitud de 5 voltios, frecuencia fija pero el ancho de pulso, esto es lo que se conoce como modulación de ancho de pulso o en inglés PWM, el driver convierte esa señal en una corriente variable para los motores de paso. Este controlador también recibe una señal para el cambio de dirección de los motores de manera independiente.

El Arduino es el que se encarga de generar la señal PWM, de acuerdo con la lógica de control, en el caso de los servomotores variando el ancho de pulso se tienen diferentes

posiciones angulares que van desde 0° hasta 180° estos servomotores no requieren ningún driver por ser de bajo torque. En caso del motor DC se requiere de un driver L298N, conocido como puente H. Y en el caso de los motores de paso se requiere el driver TB6600, que ya se describió antes. El Arduino también debe calcular la cantidad de pasos que debe dar el motor de paso para llevar el robot de un punto a otro.

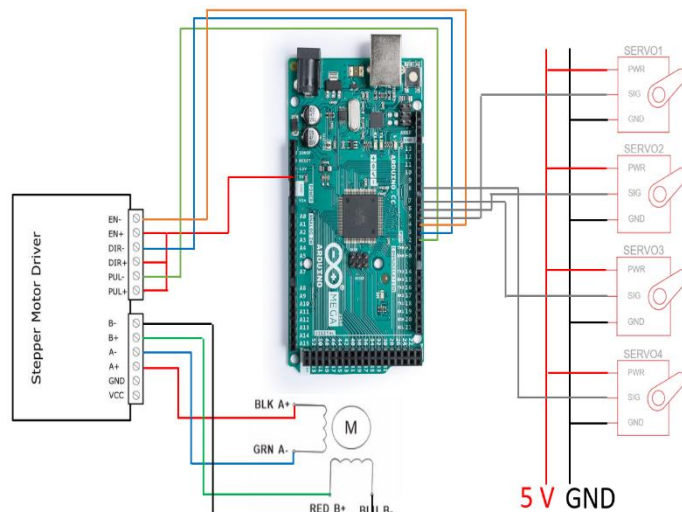


Fig. 6 Esquema de conexiones del brazo robótico

En el desarrollo de la conexión de los componentes se sobrepasa el estimado de 6 pines para el PWM utilizados comúnmente en un Arduino UNO, la necesidad de toda la conexión del robot articular incluyendo el motor paso a paso y el motor con encoder nos trajo la necesidad de realizar un cambio hacia un Arduino MEGA para tener a disposición 14 pines de PWM disponibles y con ello un incremento en la resolución de modulación por ancho de pulso como también una amplia capacidad de memoria además de contar con 6 timers.

En la base se tendrá el material del Perfil V-Solt 2080 diseñado para la realización de estructuras mecánicas y mecanismos, con ancho de 80 mm y un peso de 15g/cm se tomó como el material suficiente para el soporte del brazo y de la carga necesario para el movimiento con las siguientes especificaciones, la plataforma de un diámetro de 140 mm requerirá de dos perfiles en paralelo cada uno de ellos con 900 mm de largo como también el ajuste de los encajes de los soportes en los extremos que se sujetan mediante su ranura en forma de V diseñados específicamente para estas.

La función principal de los soportes de los extremos de la base móvil es la sujeción del tornillo de potencia a través del rodamiento fijado en el soporte, estos además lleva también la sujeción de los ejes guía de acero que deben estar fijos sin tener ningún tipo de movimiento ante posibles vibraciones del sistema para no generar aun mayor fricción en los componentes, los soportes también llevan la sujeción del motor paso a paso colocado directamente para el movimiento del tornillo de potencia a través del acople flexible entre motor y el tornillo.



Fig. 7 Diseño en acrílico del brazo robótico

Para el diseño en acrílico se tomó en cuenta los esfuerzos mecánicos en los puntos críticos del sistema, dados en el prototipo, los cuales son generados por el agarre del gripper en el punto máximo de longitud de alcance, donde este generó un torque en todo el diseño del robot, para ello se debe ver los puntos de sujeción que unen cada pieza y realizar un esquema para observar si tiene la tendencia a quebrarse o romperse. Para la unión de acrílicos se dio a través de tornillo y roscas de sujeción que están a su vez unidas a perfiles L de aluminio que generan un agarre entre cada una de estas partes del proyecto, para la sujeción del motorreductor con encoder se diseñó específicamente para su agarre de manera vertical la cual genera un peso que se sostiene mediante tornillos directamente en la plancha de acrílico.

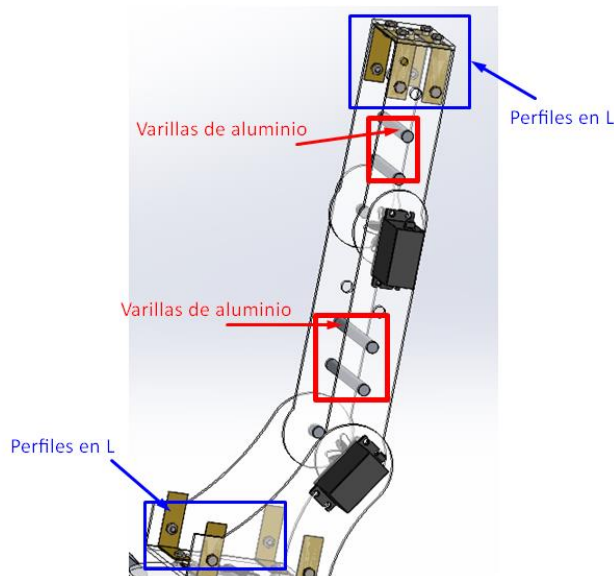


Fig. 8 Piezas para el diseño del brazo robótico

En el diseño de un nuevo componente o sistema se debe considerar los tipos de mecanismos que lo componen, diseñar y fabricar un conjunto seguro y eficaz para que cumpla con los requisitos y necesidades del proyecto, para el desarrollo y conocimiento total del mecanismo, en el movimiento rotacional del tornillo la tuerca se mantiene fija, generando un movimiento longitudinal en todo lo largo, este sistema de movimiento rotacional genera un movimiento lineal en ambas direcciones y será controlado mediante el Software de Matlab donde el sistema de movimiento nos permite tener un mayor control y además de generar ecuaciones que gobiernen el sistema a través del ancho del diente para tener una posición precisa en cada momento.

B. Generación del modelamiento robótico

Un manipulador robótico posee ciertas restricciones físicas según la configuración de cada grado de libertad diseñada; es decir, angular o prismática. La configuración angular abarca un mayor volumen de trabajo y presenta una articulación con movimiento rotacional impulsada por un motorreductor y dos angulares gobernadas a través de los servomotores, de esta forma el brazo articulado puede realizar el movimiento conocido como interpolación lineal, para lo cual requiere mover simultáneamente dos o tres de sus articulaciones, siendo el movimiento natural el de interpolación por articulación, tanto rotacional como angular [12].

En la Figura 9 el brazo mecánico posee un total de tres ejes, dos de ellos se encuentran ubicados en el eslabón angular intermedio y uno en el eslabón angular final, siendo el más esforzado uno de los ejes que se encuentra en el primer eslabón citado ya que este se vincula con uno de los servomotores y se encarga del movimiento del eslabón angular intermedio. Al ser este eje el más crítico, se puede afirmar que con el diseño de esta asegura que los dos ejes restantes cumplan con las condiciones operativas del robot [15].

En el desarrollo de la representación de la cinemática para obtener la posición del punto terminal se tiene la disposición de las 4 articulaciones que interfiere en la posición final del robot, a través del algoritmo de Denavit - Hartenberg donde cada fila de la tabla II tiene que ser transformada en una matriz homogénea obtenido del producto de 4 matrices homogéneas básicas según la ecuación.

TABLA II
PARÁMETROS D-H PARA EL BRAZO ROBÓTICO

Articulaciones	θ_i	d_i	a_i	α_i
1	θ_1	d_1	a_1	$\pi/2$
2	θ_2	0	a_2	0
3	θ_3	0	a_3	$-\pi/2$
4	θ_4	d_4	0	0

En el desarrollo a partir de Denavit - Hartenberg para las matrices de transformación homogénea se desarrollará la aplicación de algoritmos con la finalidad de obtener una

cinemática directa en dirección del control directo del robot a través de las coordenadas angulares.

En el diseño del robot propuesto, se considera el método geométrico para el desarrollo en las 4 articulaciones generando los valores de los ángulos en el primer arco tangente desarrollado en la ecuación (1) para el movimiento en base a las coordenadas finales que se dan en el extremo del robot.

$$\theta_1 = \tan^{-1}(y/x) \quad (1)$$

A través de ello hallar a través de las coordenadas finales los ángulos en el eslabón θ_2 para el segundo eslabón y θ_3 en el tercer eslabón, que, mostrando la disposición geométrica para el cálculo de los ángulos requeridos, para ello se requiere del estudio de identidades trigonométricas relaciones e igualdades entre cada ángulo.

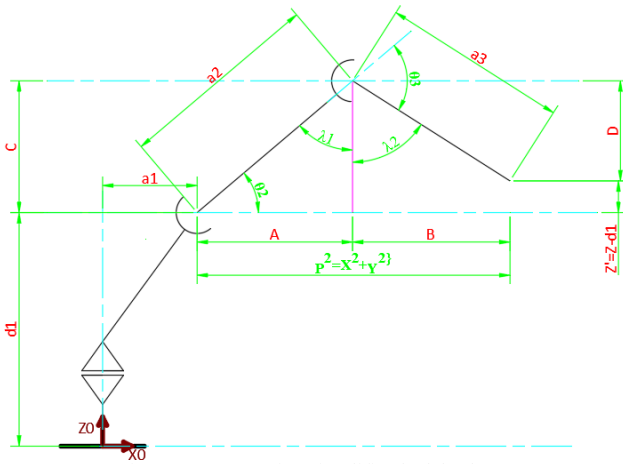


Fig. 9 Representación simplificada del robot

Para el desarrollo de las demás articulaciones en relación al posicionamiento final realizado en el extremo del robot se plantea para el tercer ángulo la ecuación (2) para cumplir con el posicionamiento a través de la razón trigonométrica de la diferencia de dos ángulos en un rango positivo de 0 a 180° en referencia al mismo alcance que se tiene en los servomotores utilizados para su diseño.

$$\theta_3 = \cos^{-1} \left(\frac{x^2 + y^2 + Z'^2 - a_2^2 - a_3^2}{2 \cdot a_2 \cdot a_3} \right) \quad (2)$$

En base a la segunda articulación se desarrolla en configuración a tener un movimiento codo arriba sugiriendo un movimiento de agarre y desplazamiento esto también con el motivo evitar posicionamientos fuera de rango como también el de máximo ángulo posible que se tienen en los servomotores, llevando a replantear la segunda articulación θ_2 a desarrollar una ecuación de movimiento vista en la ecuación (3)

$$\theta_2 = \tan^{-1} \left(\frac{a_3 \sin(\theta_3)}{a_2 + a_3 \cos(\theta_3)} \right) + \tan^{-1}(Z'/P) \quad (3)$$

El desarrollo en la cinemática inversa se da el ángulo de cada articulación, generando de esta manera hace posible que solo colocando las coordenadas finales a alcanzar por el robot posible la entrega de cada ángulo de articulación requerido para realizar el movimiento en un sistema de coordenadas según el origen.

C. Diseño y análisis de partes en CAD

En un diseño en CAD se utilizará el software de SolidWorks para el desarrollo de las partes de este y tener en este los análisis estáticos de la estructura de los eslabones principales del robot, se contará con un Gripper para la sujeción de objetos y su respectivo desplazamiento, para ello también se destacó el acrílico como el material predominante en el robot debido a su bajo costo y su fácil uso, además de ser un material con dureza y flexibilidad necesaria y tener un peso relativamente bajo para un diseño de medidas relativamente grandes.

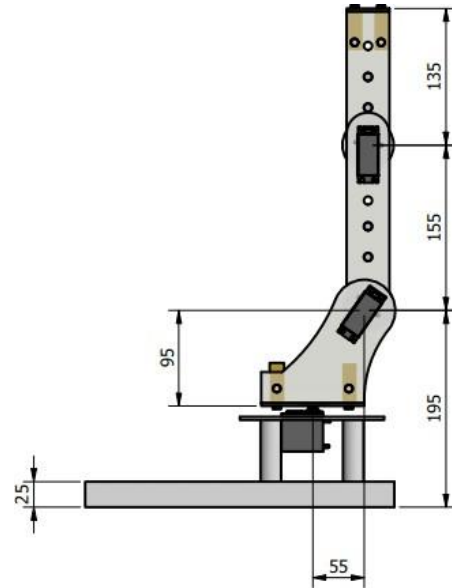


Fig. 10 Medidas en mm del tamaño del robot

Con este software se pudo analizar las propiedades físicas del robot, ya que permite hacer una selección del material, además se pudo observar que el torque necesario para mover el robot era suficiente de acuerdo al datasheet de los servomotores, incluso con las limitaciones impuestas en el mismo robot teniendo solamente movimientos de 180° por cada articulación, que al tener una base móvil aquello no acarrea más problemas que son solucionados por la movilidad sostenida en el robot, donde también se es medido los pesos de cada parte de la base móvil en pos de obtener cuanto torque puede estar dispuesto a soportar antes de que se realice un volcamiento en la sección del agarre.

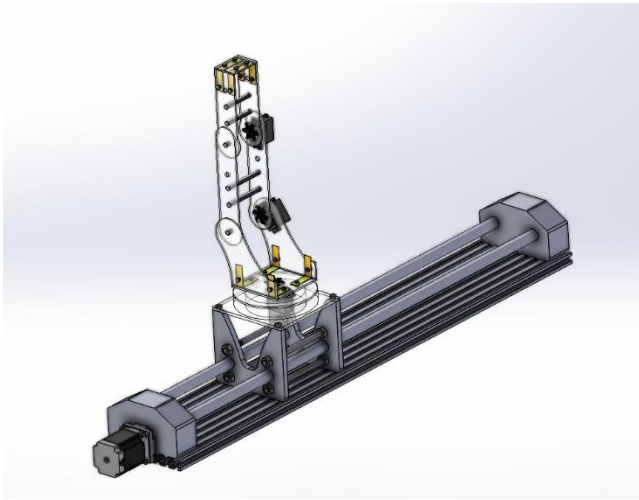


Fig. 11 Simulación del prototipo de robot en base móvil

D. Análisis estático de las estructuras

Para el desarrollo del análisis requerido para atender el peso máximo de sujeción que tendrá el robot se hará enfocado al máximo posible de soporte de fuerza que se tendrá en la garra para ellos a través del análisis estático desarrollado en el Software de SolidWorks donde se aplicarán las cargas en puntos definidos como también el enmallado del modelo.

Dado que el material predominante es el acrílico en cortes seccionales para la reducción de peso como la perforación para los tornillos de sujeción, con todo ello se debe además tener en consideración la ubicación del centroide del elemento dado que a través de este se desarrollarán las medidas para calcular los valores de la aceleración y velocidad con respecto a la inercia de cada elemento como en conjunto para el movimiento.

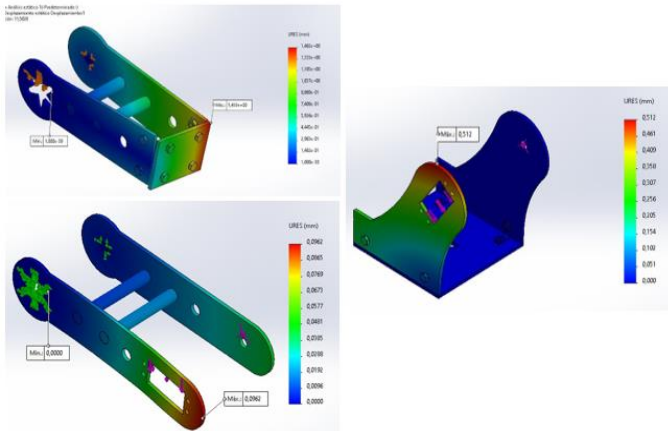


Fig. 12 Simulación del prototipo de robot en base móvil

Con ello se observa los puntos de mayores esfuerzos generados en cada parte del robot y del eslabón, dando como resultado según el punto mínimo de esfuerzo para la parte más comprometida un límite elástico de 186 MPa y como mayor esfuerzo resultante de 171 MPa, concluyendo a partir del alcance más crítico del eslabón soportando una carga.

E. Análisis de implementación de control

El posicionamiento en el espacio de trabajo según la cinemática directa e inversa del robot fue codificado en el software Matlab con la finalidad del control de este a través de la conexión Matlab-Arduino el cual usa el enlace de comunicación de recepción- transmisión asincrónica universal (UART), donde fue codificado el movimiento y control.

La comunicación con Arduino tiene la finalidad de realizar el control en todos los dispositivos de movimiento. Para el control del motor DC con encoder se utilizó el controlador HL298. Para el control del motor paso a paso se utilizó el controlador TB6600, para el control del movimiento a través de un proceso de movimiento repetitivo en un tiempo determinado mediante la modulación por ancho de pulso (PWM).

En el desarrollo del código se tuvo un ciclo de movimientos en los cuales se realiza el transporte de un objeto con la finalidad de demostrar el desarrollo de todas las articulaciones realizando un movimiento completo, los ciclo de operaciones dependerá del código para la operación, denotando el movimiento secuencial en un tiempo determinado en cada punto del sistema requiriendo de un movimiento diferente como también el cambio en los parámetros de posicionamiento actuales, tomando en cuenta el código para cada punto de acción.



Fig. 13 Sujeción del objeto a través del proceso de transporte.

IV. RESULTADOS Y DISCUSIÓN

El movimiento de cada articulación, contando con la base, requirió una nueva forma de analizar el tipo de movimiento; esto debido a la cantidad de variables a tomar en cuenta como el tipo de controlador para cada movimiento, la máxima rotación de 180° que puede realizar un servomotor, además de realizar la operación con diferentes tipos de generación de movimiento.

El desarrollo de las acciones de cada servomotor, motor paso a paso y motor con encoder se da a través de una misma señal de protocolo de comunicación UART para la conexión

entre Matlab-Arduino, a partir de esta comunicación se realizan todos los movimientos en cada punto a través de una modulación por ancho de pulso (PWM) para cada tipo de accionamiento, utilizando un dispositivo de mayor capacidad como el Arduino MEGA con el cual pasar todas las señales de PWM para el movimiento, la programación en Matlab se da en los aspectos de lenguaje en M a través de un paquete de soporte que contiene en este los comandos para el desarrollo del movimiento. Las ecuaciones que contiene este código en Matlab fueron obtenidas por la cinemática inversa usando el algoritmo de Denavit - Hartenberg.

La secuencia de movimientos de prueba fue la siguiente:

1. Posición de inicio para el robot
2. Abrir la garra en 0°
3. Mover motor paso a paso 480 mm alejándose del motor
4. Mover la segunda articulación de los dos servos a 180°
5. Se cierra la garra para agarrar la pieza en 90°
6. Se levanta la garra con la pieza en 153°
7. Mover motor paso a paso 480 mm acercándose al motor
8. Mover el motorreductor en 45° sentido horario visto desde arriba
9. Mover la segunda articulación de los dos servos en 117°
10. Mover el tercer servomotor en 180° para soltar la pieza.
11. Abrir la garra para soltar la pieza
12. Mover el motorreductor en sentido antihorario en 45° visto desde arriba

Y el código en Matlab fue el siguiente:

```
clear all
a = arduino('COM7', 'Mega2560', 'Libraries',
'Servo');
EN = 'D11'; % Salidas PWM conectadas al driver
L9110s
DIR = 'D12'; % Se dan las direcciones que se van a
tomar para el robot; 1=motor 0=otro extremo
PUL = 'D13'; % La cantidad de pulsos necesarios
para el movimiento del robot
AIA = 'D6' ; % Salidas PWM conectadas directamente
al motor para su movimiento horario
AIB = 'D7' ; % Salidas PWM conectadas directamente
al motor para su movimiento antihorario
writePWMPulse(a,AIA,5); %Ajustamos la amplitud
de las señales PWM a 5V
writePWMPulse(a,AIB,5); %Ajustamos la amplitud
de las señales PWM a 5V
writePWMPulse(a,PUL,5); %Ajustamos la amplitud
de las señales PWM a 5V
%% Variables para mover los servos
% Se colocan el valor y el nombre que tendrán cada
servo que en total son 4
s1 = servo(a, 'D2', 'MinPulseDuration', 5.44*10^-
4, 'MaxPulseDuration', 24.0*10^-4);
s2 = servo(a, 'D3', 'MinPulseDuration', 5.44*10^-
4, 'MaxPulseDuration', 24.0*10^-4);
s3 = servo(a, 'D4', 'MinPulseDuration', 5.44*10^-
4, 'MaxPulseDuration', 24.0*10^-4);
s4 = servo(a, 'D5', 'MinPulseDuration', 5.44*10^-
4, 'MaxPulseDuration', 24.0*10^-4);
%% Valores de inicio
angle1=0.9; % Angulo de inicio de los servos 1 y 2
primeros
```

```
angle2=1-angle1;
angle3=1;
angle4=0.5; % el 0.5 es el ángulo de cerrado
mientras que el 0 el ángulo abierto
%% 1. Primera posición de inicio para el movimiento
del robot
writePosition(s1, angle1); % Escribe el valor del
ángulo que se va a tener
current_pos1 = readPosition(s1); %Lee la posición
que se dio al servo
current_pos1 = current_pos1*180; % Se multiplica *
180 para generar el ángulo en grados
fprintf('Current motor position is %d degrees\n',
current_pos1);
writePosition(s2, angle2);
current_pos2 = readPosition(s2);
current_pos2 = current_pos2*180;
fprintf('Current motor position is %d degrees\n',
current_pos2);
writePosition(s3, angle3);
current_pos1 = readPosition(s3);
current_pos1 = current_pos1*180;
fprintf('Current motor position is %d degrees\n',
current_pos1);
writePosition(s4, angle4);
current_pos1 = readPosition(s4);
current_pos1 = current_pos1*180;
fprintf('Current motor position is %d degrees\n',
current_pos1);
pause(1)
%% 2. Código que abre la garra en 0°
clear angle4
angle4=0;
writePosition(s4, angle4);
current_pos1 = readPosition(s4);
current_pos1 = current_pos1*180;
fprintf('Current motor position is %d degrees\n',
current_pos1);
pause(1)
%% 3. Movimiento motor paso a paso con una longitud
de 48 cm alejándose del motor
writeDigitalPin(a,EN,1);
writeDigitalPin(a,DIR,0); % HIGH va hacia el
motor; LOW va hacia el otro extremo
writePWMDutyCycle(a,PUL,0.4); % va en sentido
horario si miramos desde arriba
pause(24); %Tras la pausa se registra la velocidad
writeDigitalPin(a,EN,0); %apagamos el motor
pause(1);
%% 4. Código que mueve nuevamente la segunda
articulación de los dos servos a 180°
clear angle1 angle2
angle1= 1;
angle2= 1-angle1;
writePosition(s1, angle1);
current_pos1 = readPosition(s1);
current_pos1 = current_pos1*180;
fprintf('Current motor position is %d degrees\n',
current_pos1);
writePosition(s2, angle2);
current_pos2 = readPosition(s2);
current_pos2 = current_pos2*180;
fprintf('Current motor position is %d degrees\n',
current_pos2);
pause(1)
```

```

%% 5. Se cierra la garra para agarrar la pieza en
90°
clear angle4
angle4= 0.5;
writePosition(s4, angle4);
current_pos1 = readPosition(s4);
current_pos1 = current_pos1*180;
fprintf('Current motor position is %d degrees\n',
current_pos1);
pause(1.5)
%% 6. Se levanta la garra después de haber agarrado
la pieza en 153°
clear angle1 angle2
angle1= 0.85;
angle2= 1-angle1;
writePosition(s1, angle1);
current_pos1 = readPosition(s1);
current_pos1 = current_pos1*180;
fprintf('Current motor position is %d degrees\n',
current_pos1);
writePosition(s2, angle2);
current_pos2 = readPosition(s2);
current_pos2 = current_pos2*180;
fprintf('Current motor position is %d degrees\n',
current_pos2);
pause(1)
%% 7. Movimiento motor paso a paso con una longitud
de 48 cm acercándose al motor
writeDigitalPin(a, EN,1);
writeDigitalPin(a, DIR,1); % HIGH va hacia el
motor; LOW va hacia el otro extremo
writePWMDutyCycle(a,PUL,0.4); % va en sentido
horario si miramos desde arriba
pause(24); %Tras la pausa se registra la velocidad
writeDigitalPin(a, EN,0); %apagamos el motor
pause(1);
%% 8. Movimiento del motorreductor en 45° sentido
horario visto desde el punto de arriba
writeDigitalPin(a,AIA,0);
writePWMDutyCycle(a,AIB,1); % va en sentido
horario si miramos desde arriba
pause(0.4); %Tras la pausa se registra la velocidad
writeDigitalPin(a,AIA,0); %apagamos el motor
writeDigitalPin(a,AIB,0);
pause(0.4);
%% 9. Código que mueve la segunda articulación de
los dos servos en 117°
clear angle1 angle2
angle1 = 0.65;
angle2= 1-angle1;
writePosition(s1, angle1);
current_pos1 = readPosition(s1);
current_pos1 = current_pos1*180;
fprintf('Current motor position is %d degrees\n',
current_pos1);
writePosition(s2, angle2);
current_pos2 = readPosition(s2);
current_pos2 = current_pos2*180;
fprintf('Current motor position is %d degrees\n',
current_pos2);
pause(1)
%% 10. Movimiento del 3 server en 180° para
soltarlo en donde debe
clear angle3
angle3 = 1;
writePosition(s3, angle3);

```

```

current_pos1 = readPosition(s3);
current_pos1 = current_pos1*180;
fprintf('Current motor position is %d degrees\n',
current_pos1);
pause(1)
%% 11. Se abre la garra para soltar la pieza
clear angle4
angle4=0;
writePosition(s4, angle4);
current_pos1 = readPosition(s4);
current_pos1 = current_pos1*180;
fprintf('Current motor position is %d degrees\n',
current_pos1);
pause(1)
%% 12. Movimiento del motorreductor en sentido
antihorario en 45° visto desde el punto de arriba
writeDigitalPin(a,AIB,0);
writePWMDutyCycle(a,AIA,1); % va en sentido
horario si miramos desde arriba
pause(0.4); %Tras la pausa se registra la velocidad
writeDigitalPin(a,AIA,0); %apagamos el motor
writeDigitalPin(a,AIB,0);
pause(0.4);
%% 13. Código que mueve nuevamente la segunda
articulación de los dos servos a 162°
clear angle1 angle2
angle1=0.9;
angle2=1-angle1;
writePosition(s1, angle1);
current_pos1 = readPosition(s1);
current_pos1 = current_pos1*180;
fprintf('Current motor position is %d degrees\n',
current_pos1);
writePosition(s2, angle2);
current_pos2 = readPosition(s2);
current_pos2 = current_pos2*180;
fprintf('Current motor position is %d degrees\n',
current_pos2);
pause(1)

```

En el desarrollo del control sobre el Software Matlab se realizó la generación de los valores para la configuración de los movimientos pick and place, utilizando los drivers de conexión Matlab-Arduino que proporcionan el protocolo de comunicación UART, en la generación de comandos que controlen los servomotores, motor paso a paso y motor con encoder realizando de manera secuencial los movimientos de manera infinita con arreglos. Para variar los ángulos de los servomotores se varía el ancho de pulso.

La sujeción de un objeto dependerá de su tamaño y del peso, según las pruebas realizadas el máximo peso posible de carga sin que ningún servo falle es de 1 kg esto tomando un rango óptimo para desplazar cualquier objeto menor en ese peso, para el movimiento realizado en las pruebas se utilizó un componente de peso de 0.03 kg.

La velocidad de movimiento no es rápida debido a las propias restricciones del motor paso a paso que reduce la velocidad considerablemente, esto debido a la forma en la cual el motor puede ser accionado que es a partir de señales de pulso, por ende, se requiere de una forma única de movimiento con el cual el proceso solamente puede darse en un tiempo como

también dependerá de todos los pasos que se deban realizar, ralentizando aún más el proceso.

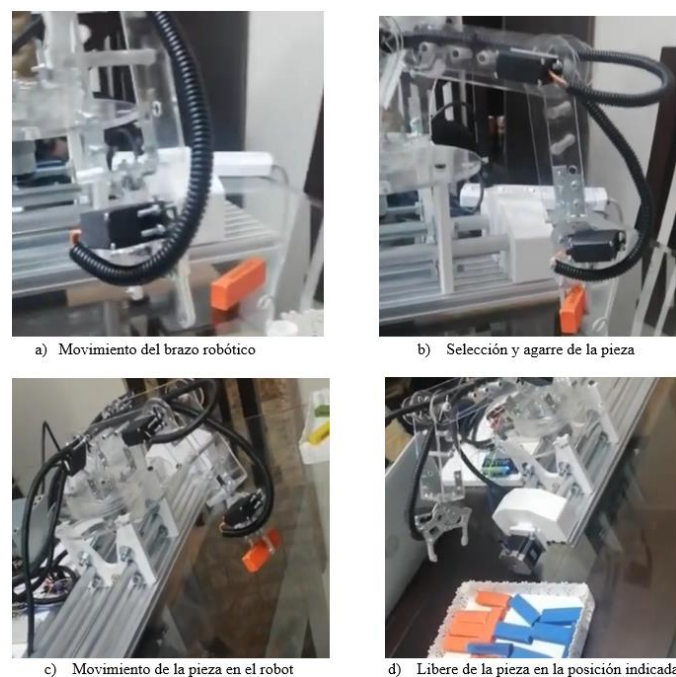


Fig. 14 Sujeción del objeto a través del proceso de transporte.

El desarrollo del movimiento abarca todas las articulaciones, y con ello los servomotores, el motorreductor y para el movimiento longitudinal el motor paso a paso. Se realiza todo el proceso de movimiento Pick and Place, este sistema se puede reconfigurar para tener pasos intermedios, movimientos simultáneos de los eslabones, etc.

Es importante mencionar que el costo de la fabricación del sistema fue de USD 270.00 aproximadamente.

V. CONCLUSIONES

Se diseñó y construyó un robot manipulador de 4 grados de libertad, que se puede desplazar 900 mm sobre una base móvil con una gran precisión, capaz de posicionarse en cualquier punto intermedio.

El diseño se validó estructuralmente con el software Solidworks y con las pruebas de funcionamiento, los servomotores utilizados fueron suficientes debido a la estructura de acrílico del robot.

El análisis de cinemática inversa permitió obtener los parámetros adecuados para el control de movimiento del brazo robótico, se programó un movimiento de articulación por articulación para el robot, para las pruebas realizadas.

El tiempo de un ciclo de trabajo ida y vuelta sobre una base móvil de 900 mm fue de 30 segundos, tomando las posiciones extremas de la base como puntos de paso del robot.

REFERENCIAS

- [1] J. Gutiérrez, "Diseño y análisis del funcionamiento estructural de un brazo robótico de nivel industrial de cuatro grados de libertad" *Ing. Mecánica y afines*, vol. 15, pp. 11-33, 2019.
- [2] Stefan-Octavian Bezrucav, Nils Mandischer, Burkhard Corves, "Artificial Intelligence Task Planning of Cooperating Low-Cost Mobile Manipulators: A Case Study on a Fully Autonomous Manufacturing Application" *Procedia Computer Science*, vol. 217, 2023, pp 306-315, <https://doi.org/10.1016/j.procs.2022.12.226>.
- [3] Jan Carstensen, Torben Carstensen, Malte Pabst, Fabian Schulz, Jan Friederichs, Simon Aden, Daniel Kaczor, Jens Kotlarski, Tobias Ortmaier, "Condition Monitoring and Cloud-based Energy Analysis for Autonomous Mobile Manipulation - Smart Factory Concept with LUHbots", *Procedia Technology*, vol. 26, 2016, pp 560-569, ISSN 2212-0173, <https://doi.org/10.1016/j.protcy.2016.08.070>.
- [4] Gustavo Caiza, Carlos A. Garcia, Jose E. Naranjo, Marcelo V. Garcia, "Flexible robotic teleoperation architecture for intelligent oil fields", *Heliyon*, vol. 6, Issue 4, 2020, e03833, ISSN 2405-8440, <https://doi.org/10.1016/j.heliyon.2020.e03833>.
- [5] Sandeep Ameet Kumar, Ravinesh Chand, Ronal Pratil Chand, Bibhya Sharma, "Linear manipulator: Motion control of an n-link robotic arm mounted on a mobile slider", *Heliyon*, vol. 9, Issue 1, 2023, e12867, ISSN 2405-8440, <https://doi.org/10.1016/j.heliyon.2023.e12867>.
- [6] Kavin Kathiresh Vijayan, Ola Jon Mork, Lars A.L. Giske, "Integration of a Case Study into Learning Factory for Future Research", *Procedia Manufacturing*, vol. 31, 2019, pp 258-263, ISSN 2351-9789, <https://doi.org/10.1016/j.promfg.2019.03.041>.
- [7] Yi Xiao, Yu Yan, YiSheng Yu, Biao Wang, YunHua Liang, "Research on pose adaptive correction method of indoor rail mounted inspection robot in GIS Substation", *Energy Reports*, Volume 8, Supplement 6, 2022, pp 696-705, ISSN 2352-4847, <https://doi.org/10.1016/j.egy.2022.03.155>.
- [8] L. A. Mejía-Calderón, A. González-Echeverry, and H.F. Quintero-Riaza, "Cinemática inversa de un robot redundante tipo Scorbot – ER Vplus: influencia de índices de desempeño", *Iteckne*, vol. 14, no. 1, p. 24, 2017, doi: 10.15332/iteckne.v14i1.1627.
- [9] Sergio Velásquez, Jesús Luces y Ronny Velásquez, "Modelado y simulación de un brazo robótico mediante el uso de simmechanics de Matlab", *Ciencia e Ingeniería*, vol. 2, no. 2, 2015, ISSN 2389-9484.
- [10] J. Jiao, W. Tian, W. Liao, L. Zhang, and Y. Bu, "Processing configuration off-line optimization for functionally redundant robotic drilling tasks", *Rob Auton Syst*, vol. 110, pp. 112-123, 2018, doi: 10.1016/j.robot.2018.09.002.
- [11] C. A. P. C. Pedro Alexander Contreras Parada, Cristhian Iván Riaño Jaime, "Robotic Module for Lulo (Solanum Quitoense) Classification Using Computer Vision", *Inge CUC*, vol. 10, no. ISSN 0122-6517, pp. 51-62, 2014.
- [12] A. Farias, L. Moraga, and A. Martinich, "Integración De Una Célula Flexible De Mecanizado, De Tipo Docente Integration of a Educational Flexible Manufacturing Cell," *Ingeniare. Revista chilena de ingeniería*, vol. 15, no. 1, pp. 27-34, 2007.
- [13] A. Villa Fernández, "Diseño y fabricación de un escáner 3D sin contacto basado en las plataformas Arduino y Matlab," Universidad de Cantabria, Cantabria, 2020. Accessed: Feb. 03, 2023. [Online]. Available: <http://hdl.handle.net/10902/19659>.
- [14] J. A. Cortés Osorio, D. A. Muñoz Acosta, and C. D. Lopez Robayo, "Diseño y construcción de un riel electromecánico para el estudio de la cinemática de imágenes con difuminación lineal uniforme", *Inge CUC*, vol. 16, no. 1, pp. 80-94, 2020, doi: 10.17981/ingecuc.16.1.2020.06.
- [15] J. Otero Yugat, S. Enrique Rodríguez, and J. Javier Gutiérrez, "Diseño y construcción de un brazo mecánico de tres grados de libertad", *Scientia et Technica Año XIV*, vol. 39, no. 0122-1701, pp. 1-6, 2008