



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Manipulador Móvil Híbrido
Autónomo: Coordinación de
Movimientos para el Transporte de
Materiales**

TESIS

Que para obtener el título de

INGENIERO MECATRÓNICO

P R E S E N T A N

PACHECO JIMÉNEZ ANA JANET
ÁVILA GARCÍA ALBERTO



DIRECTOR DE TESIS
DR. VÍCTOR JAVIER GONZÁLEZ VILLELA

Ciudad Universitaria, Cd. Mx., abril 2017



UNAM – Dirección General de Bibliotecas

Tesis digitales

Restricciones de uso

DERECHOS RESERVADOS ©

PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de video, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente de donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Jurado

Presidente:	M. I. RIGEL GÁMEZ LEAL
Vocal:	DR. VÍCTOR JAVIER GONZÁLEZ VILLELA
Secretario:	DRA. MARÍA DEL PILAR CORONA LIRA
1 ^{er} Suplente:	M. I. LIVIER BAEZ RIVAS
1 ^{do} Suplente:	M. I. PATRICIO MARTÍNEZ ZAMUDIO

Dedicatorias

Ana

A mi mamá por brindarme su apoyo y aliento a lo largo de toda mi carrera profesional

A mi papá por enseñarme a ser una persona autosuficiente y segura de lo que puedo lograr

A mis hermanas por motivarme a terminar mis estudios y a mi sobrina por regalarme y contagiarme de su alegría

A mi tío Edgar por apoyarnos y orientarnos en los aspectos técnicos que surgieron

A mis amigos por acompañarme a recorrer estos cinco años con alegría

A mi compañero y “amigo” de tesis por alentarme a no rendirme y creer que lo puedo lograr

Beto

A mis padres por sus ánimos y aliento que siempre me dieron

A mi hermana por su ejemplo y ayuda que me ha brindado

A mi familia y amigos por el estímulo y ánimo que me transmitieron

A mis profesores por su guía

A mi compañera de tesis por su mucha paciencia y abundante cariño

Agradecimientos

Queremos agradecer especialmente a nuestro tutor de tesis el Dr. Víctor Javier González Villela por su apoyo, guía y consejos a lo largo de la realización de este trabajo.

También al M.I Noé Alfredo Martínez, M.I Erik Peña Medina y al Dr. Patricio Martínez Zamudio, por concedernos su tiempo y su ayuda incondicional; motivándonos para finalizar esta tesis.

Agradecemos en lo que corresponde a DGAPA, por apoyo brindado para la realización de este trabajo, a través del proyecto PAPIIT IN117617: “Robótica intuitiva, adaptable, reactiva, híbrida y móvil aplicada al servicio, el rescate y la medicina”

Resumen

El presente documento describe el estudio realizado a un robot manipulador móvil híbrido para el transporte de materiales; modelo funcional que está constituido por la unión de tres sistemas robóticos que son: un robot móvil omnidireccional, un paralelo delta en el plano y un manipulador serial. Dicho estudio, comprende el análisis cinemático del sistema y la implementación de éste en Matlab Simulink; así como una propuesta de coordinación de movimientos para la ejecución de la tarea y los resultados obtenidos en dicha propuesta.

El modelo funcional implementado llevó a cabo la ejecución de la tarea definida de manera autónoma, ya que para el control de la planeación de la trayectoria se emplearon campos potenciales; además que el espacio de trabajo del robot se trató de un ambiente inteligente controlado por un sistema de visión artificial.

Abstract

This document describes the study of a mobile hybrid robot manipulator for material handling; this functional model is composed by three robotic systems which are: a omnidirectional mobile robot, a planar parallel robot and a serial manipulator. This study consists of the system kinematic analysis, its correspondent Matlab Simulink implementation and a motion planning coordination proposal to task performance and the results obtained. The implemented functional model executed the defined task autonomously, since the trajectory planning control was based on potential fields approach; besides, the robot workspace was constituted by an intelligent environment controlled by artificial vision.

Tabla de Contenido

Jurado	ii
Dedicatorias	iii
Agradecimientos	iii
Resumen	iv
Abstract	iv
Introducción	1
Contexto histórico	1
Robot manipulador	3
Tipos de articulaciones	3
Manipulador serial o cadena abierta	4
Manipulador paralelo	6
Manipulador híbrido	8
Robot móvil	9
Tipos de ruedas	10
Configuración de robots móviles con ruedas	12
Estado del Arte	16
Definición del problema	19
Objetivo	19
Hipótesis	20
Justificación	20
Alcances	20
Análisis Cinemático	21
Cinemática del Manipulador Serial	22
Cinemática del Robot Paralelo Delta Plano	25

Cinemática del Robot Móvil Omnidireccional	30
Implementación	35
Visión Artificial.....	35
ReacTIVision.....	36
Espacio de Trabajo TUIO.....	36
Procesamiento de Datos en Matlab	37
Recepción de Datos	37
Módulo de Visión.....	38
Definición de un Objetivo.....	42
Campos Potenciales.....	42
Cinemática de Plataforma Móvil.....	47
Compensaciones.....	47
Procesamiento de Datos de Salida.....	48
Envío de Datos.....	49
Conexión Wifi	51
Función del Shield Wifi de Arduino	51
Función de Arduino	52
Descripción del Hardware.....	53
Arduino Mega 2560	56
Shield Wifi de Arduino.....	57
Fuente de Alimentación	57
Tarjeta MD25.....	58
Motores 37D 50:1 con Encoder	59
Etapa de Reducción de Amplitud de PWM.....	60
Etapa de Potencia IBT 2	61
Servomotores.....	62
Etapa de Potencia de los servomotores.....	63

Implementación del PIC16F88.....	63
Coordinación de Movimientos.....	66
Dimensiones del Robot.....	66
Cinemática Inversa del Manipulador Paralelo-Serie.....	68
Descripción del Movimiento.....	69
Inicio de la Rutina.....	69
Aproximación a la Meta	70
Aproximación Concluida y Toma del Objeto	74
Retorno a la Postura Inicial.....	74
Simulaciones, Pruebas y Resultados	75
Comentarios Finales.....	80
Discusión.....	80
Conclusiones.....	80
Trabajo a futuro	81
Apéndices	82
Código de los Reductores de Frecuencia	82
Bits de Configuración ‘Config.h’	82
Código Principal ‘Encoder4.c’	82
Código de la Plataforma Móvil Omnidireccional.....	85
Código del Manipulador	88
Código Principal de Wolfram Mathematica	92
Código para Manipulación de Coordenadas y Posturas del Manipulador	112
Imágenes Adicionales de la Tarea a Realizar	117
Índice de Ecuaciones.....	119
Índice de Tablas.....	121
Índice de Imágenes.....	121
Referencias	125

Introducción

Contexto histórico

Si bien la robótica es y ha sido uno de los temas predilectos de la ciencia ficción, cuyo origen de la palabra ciertamente yace en ella, su historia y evolución inició desde hace ya varios siglos con el hombre y sus máquinas simples, cuyo desarrollo a través del tiempo y el constante deseo de crear máquinas que fuesen capaces de facilitar el trabajo realizando las tareas pesadas y repetitivas, fueron los precursores de lo que hoy llamamos robótica.

Las primeras máquinas similares a lo que hoy conocemos como robots son los autómatas, cuyas raíces más antiguas están en Grecia, donde existían estatuas con movimiento por medio de energía hidráulica, creados por inventores como Herón de Alejandría quien ideó los primeros autómatas autorregulados, capaces de tomar decisiones basadas en su diseño, sin necesidad de que fueran supervisadas por un ser humano [1].

Otro autómata destacado fue creado por el inventor y polímata Leonardo Da Vinci, quien en la Edad Media ideó varios inventos mecánicos entre los que destacan a su autómata conocido como el *caballero mecánico* en 1495, cuyo propósito era imitar ciertos movimientos de los seres humanos, como mover los brazos, la cabeza o incluso sentarse. Respecto a otros inventos por el mismo da Vinci, destacan unos planos del que es considerado como el primer automóvil de la historia o primer robot programable; cuyo mecanismo consistía en una rudimentaria configuración diferencial que le permitía controlar la dirección.

A pesar de ello, la palabra robot fue acuñada por primera vez por Karel Capek, en su obra satírica, Los Robots Universales de Rossum en el año de 1921, definiéndola como las máquinas que emulan los movimientos de los humanos y que realizan incansablemente ciertas tareas. Aunque se podría ampliar a sistemas electrónico mecánicos cuya función es ejecutar tareas programadas.

Por citar un ejemplo, de manera oficial la Robotics Industries Association (RIA) define a un robot industrial como un manipulador multifuncional reprogramable diseñado para desplazar materiales, piezas, herramientas o dispositivos especiales, mediante movimientos variables programados para la ejecución de una diversidad de tareas [2].

Se puede entonces definir a la Robótica como la ciencia que integra un conjunto de disciplinas que involucran a la mecánica, electrónica, ciencias de la computación, teorías de control, biomecánica, por mencionar algunas; que en conjunto tienen como objetivo el diseño y la

implementación de máquinas de precisión con un cierto grado de autonomía, llamados robots.

A continuación, se muestra el esquema básico de un robot; identificando en éste, sistemas mecánicos, actuadores, sensores y el sistema de control como elementos básicos para cerrar la cadena de actuación-medición-actuación. [3]

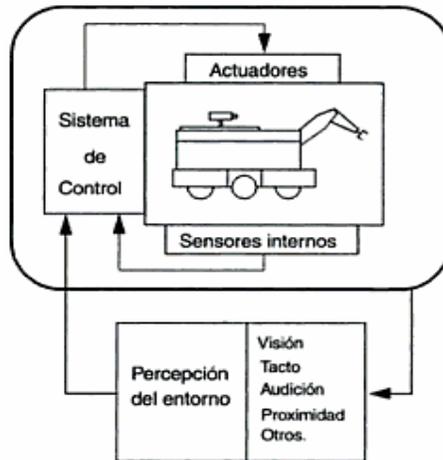


Ilustración 1 Esquema general de un robot [3]

Es importante mencionar que existen muchas y muy variadas formas de clasificar a los robots; tal puede ser el caso que se muestra a continuación, de la tesis de licenciatura del Ing. Miguel Ángel Cárdenas Verdugo [4].

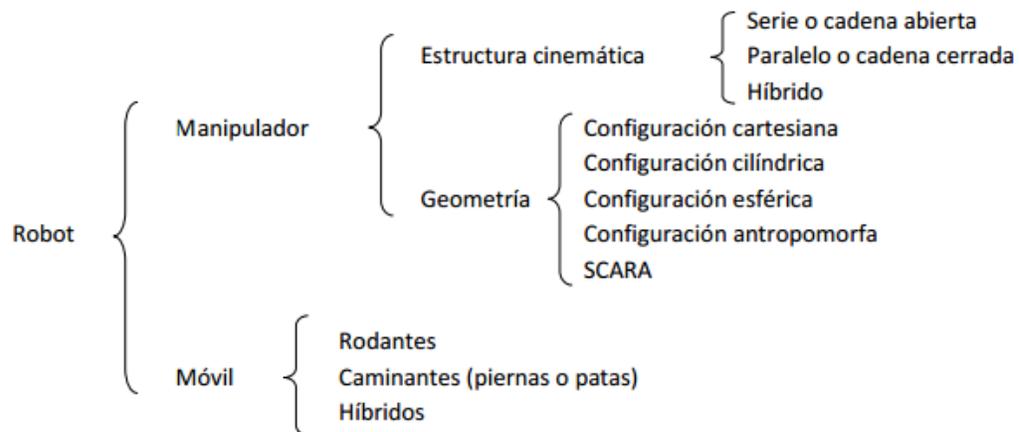


Ilustración 2 Clasificación de los robots, propuesta desde la tesis de Cárdenas Verdugo [4]

Robot manipulador

Un robot manipulador es un ensamble entre eslabones sólidos y articulaciones, los cuales se encuentran referidos a una base rígida. Siendo las articulaciones las que permiten un movimiento de rotación o traslación entre dos eslabones consecutivos y las que determinan los grados de libertad del manipulador.

Generalmente la base del manipulador es rígida y se puede encontrar de manera horizontal, vertical o suspendida, sujeta a una plataforma no siempre estática ya que cuando se mueve, comúnmente lo hace a lo largo de un eje para sincronizar el movimiento del robot con el de otros equipos. De esta manera el movimiento de la base, sumado al movimiento tridimensional del manipulador proporciona más grados de libertad que, los que tendría el manipulador por sí solo. [5]

Tipos de articulaciones

Mecánicamente, un robot está constituido por una serie de eslabones unidos por articulaciones las cuales dotan y restringen al mismo tiempo los movimientos entre eslabones consecutivos según su geometría.

Recordando el número de grados de libertad (GDL) que puede tener un sistema, se definen como el número de parámetros independientes que se necesitan para definir unívocamente su posición en el espacio en cualquier instante. En el plano se requiere de tres parámetros dos coordenadas lineales (x , y) y una coordenada angular (α). En el espacio se requiere de seis: tres distancias (x , y , z) y tres ángulos (α , β , γ). [6]

La siguiente tabla muestra los diferentes tipos de articulaciones con sus respectivas características.

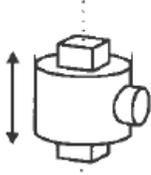
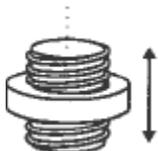
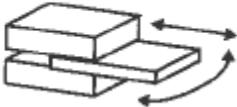
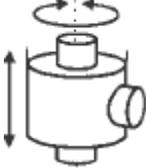
Tipo de articulación	Movimientos
Prismática (1 GDL) 	Traslación a lo largo del eje de la articulación que cambia su posición respecto a este eje, pero no su orientación.
Rotación (1 GDL) 	Rotación alrededor de una línea (eje de articulación) cambiando únicamente su orientación angular.
Tornillo (1 GDL) 	Traslación a lo largo del eje de la articulación.
Planar (2 GDL) 	Movimiento de desplazamiento en el plano
Cilíndrica (2 GDL) 	Movimiento de rotación alrededor de un eje y movimiento de traslación paralelo a éste.
Esférica (3 GDL) 	Rota de tal manera, que cada punto tiene un movimiento alrededor de un punto estacionario común. Está restringido a moverse en una superficie esférica.

Tabla 1 Tipos de articulaciones [5], [7], [3]

Manipulador serial o cadena abierta

Los robots manipuladores son, esencialmente, brazos articulados; cadenas cinemáticas abiertas formadas por un conjunto de eslabones unidos mediante juntas [3]. Este tipo de robots guardan cierta similitud con la anatomía del brazo humano, por lo que en ocasiones,

para hacer referencia a los distintos elementos, se usan términos como cuerpo, brazo, codo y muñeca. [5]

A continuación, se muestran las configuraciones básicas de estos manipuladores seriales.

Tipos de configuraciones

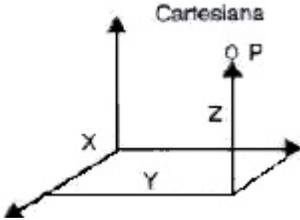
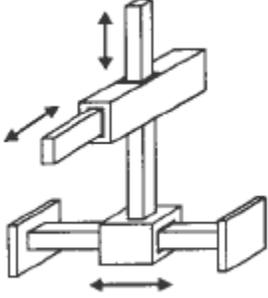
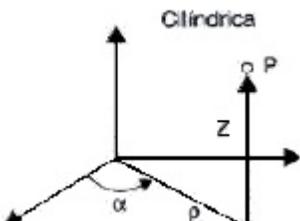
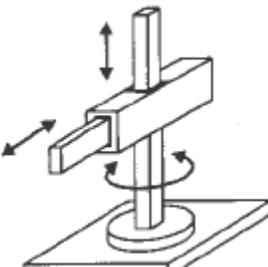
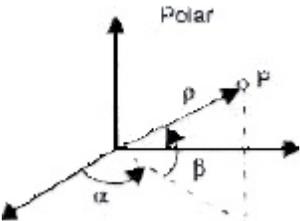
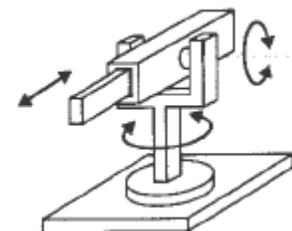
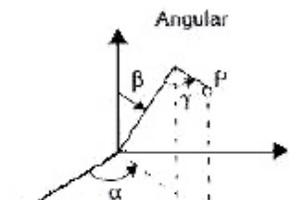
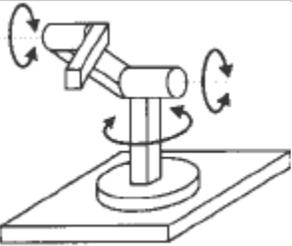
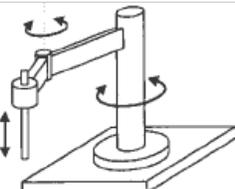
Tipo de configuración	Descripción
<p>Cartesiana</p> 	 <p>Caracterizada por tener tres articulaciones prismáticas (3D o estructura PPP). La especificación de la posición de un P se expresa en coordenadas cartesianas (x, y, z)</p>
<p>Cilíndrica</p> 	 <p>Esta configuración tiene dos articulaciones prismáticas y una de rotación (2D, 1G); la primera articulación suele ser de rotación (RPP). P (posición final) se especifica en coordenadas cilíndricas (ρ, φ, z)</p>
<p>Esférica</p> <p>Polar</p> 	 <p>Caracterizada por dos articulaciones de rotación y una prismática (2G, 1D o estructura RRP)</p>
<p>Antropomorfa</p> <p>Angular</p> 	 <p>Es una estructura con tres articulaciones de rotación (3G o RRR). La posición del extremo final se especifica en coordenadas angulares</p>
<p>Scara</p>	 <p>Constituida por dos articulaciones de rotación con respecto a dos ejes paralelos, y una de desplazamiento en sentido perpendicular al plano.</p>

Tabla 2 Tipos de configuración

Manipulador paralelo

Según Ilian Bonev [8] el primer robot paralelo del que se tiene conocimiento es una plataforma sobre la que estaban colocados los asientos de un teatro con el fin de introducir un movimiento que diese una apariencia más real al espectáculo; sin embargo, esta plataforma no llegó a construirse. La primera estructura paralela conocida industrialmente fue un hexápodo con lados de longitud variable el cual simulaba el proceso de aterrizaje de un avión diseñada por Gough en 1947 (Gough, 1956; Gough y Whitehall, 1962). Mientras que el robot Delta fue propuesto por Clavel (1989).

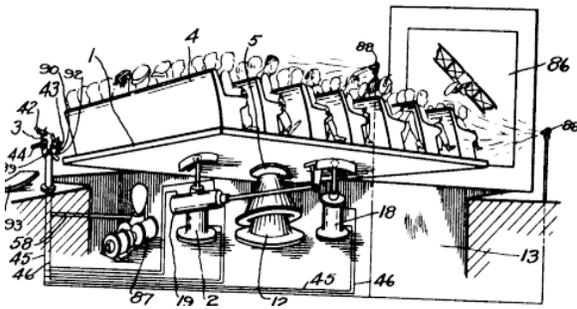


Ilustración 3 Posiblemente el primer robot paralelo patentado en 1931 (US Patent No. 1,789,680) [8]

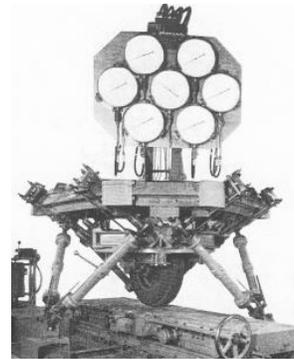


Ilustración 4 Plataforma Gough en 1954 (Proc. IMechE, 1965-66) [8]

La robótica paralela es la rama de la robótica que tiene por objeto de estudio el diseño y construcción de manipuladores paralelos. En un manipulador paralelo, el efector final está conectado a la base móvil a través de varias cadenas de enlaces interconectadas a su vez a una plataforma fija; a diferencia de un manipulador serial que cuenta con una sola cadena que une a este efector final. Cabe destacar que muchas de estas articulaciones no son actuadas, y son comúnmente llamadas articulaciones pasivas que tienen muchos grados de libertad (GDL) como las juntas esféricas, universales entre otras. Dos de los robots paralelos más populares son los hexápodos usados en muchos de los simuladores de movimiento y el llamado robot Delta generalmente usado por la velocidad en la operación pick-and-place. [9]

Clasificación de robot paralelos

Planares

Estos robots paralelos se caracterizan porque el movimiento solo puede darse en el plano, por tanto, éstos sólo pueden tener dos o tres grados de libertad; es decir, movimientos de traslación (x , y) y rotación en (z) eje perpendicular al plano. Los cuales, son accionados mediante actuadores lineales y rotacionales.

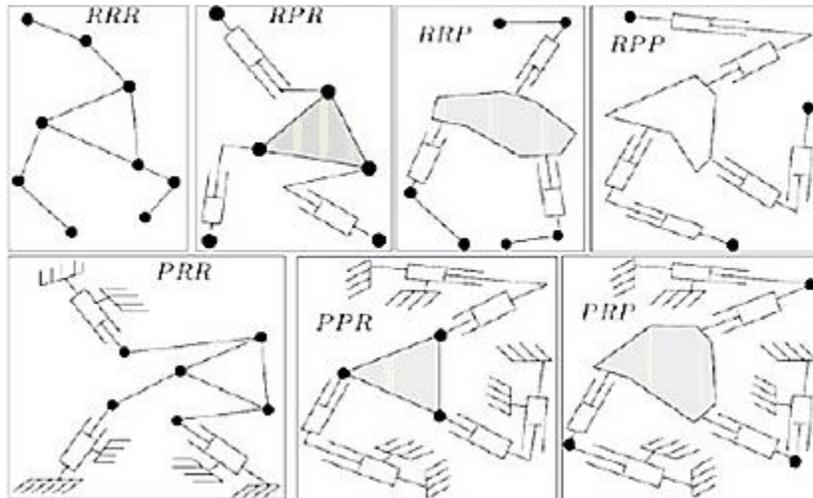


Ilustración 5 Ejemplos de distintos robots paralelos planares con actuadores prismáticos y giratorios

Espaciales

Estos manipuladores paralelos como su nombre lo indica, su movimiento está definido en todo el espacio 3D.

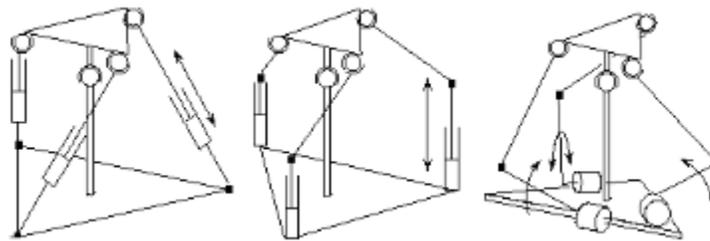


Ilustración 6 Ejemplos de robots paralelo espacial

En estos ejemplos, las articulaciones entre eslabones que van desde el efector final hasta la base se realizan con juntas universales, esféricas o prismáticas.

Según un artículo publicado por la revista Iberoamericana de Automática e Informática Industrial (RIA), Rafael Aracil en su artículo titulado *Robots paralelos: máquinas con un pasado para una robótica del futuro*, junto con su equipo mencionan que de manera general las plataformas paralelas presentan las siguientes ventajas:

- Los accionamientos de potencia se conectan directamente de la base del robot al efector final. Debido a esto, los accionamientos de potencia sirven de elementos estructurales y actúan de manera simultánea, lo que les da la capacidad de manipular cargas superiores a su propio peso.
- Por tanto, el elevado coeficiente carga/potencia de estos mecanismos proporciona una alta eficiencia energética.

- Las estructuras paralelas son mecanismos que ofrecen una alta rigidez y bajo peso. Lo que en consecuencia hace que presenten ciertas ventajas superiores a los manipuladores serie en cuanto a precisión.
- Presentan elevadas velocidades de operación, en comparación con cualquier otro tipo de estructura robótica.

Sin embargo, y dependiendo la aplicación final, estas estructuras robóticas presentan algunas características que podrían considerarse desventajas, como el hecho que realizar un análisis cinemático resulta más complicado por las numerosas cadenas que lo conforman, obligando a recurrir a una gran cantidad de sensores, para establecer un lazo de control. Además, que el espacio de trabajo de éstos es pequeño comparado con los de cadena abierta y no se tiene un modelo dinámico general, ya que debe resolverse para cada topología.

Manipulador híbrido

Se entiende por manipulador híbrido como una combinación entre cadenas cinemáticas abiertas y cerradas, es decir, un manipulador constituido por mecanismos paralelos y seriales interconectados. Como se menciona en el artículo *Workspace of a hybrid (parallel-serial) robot manipulator* (2006) de Tanev T. los manipuladores híbridos vencen las limitaciones del espacio de trabajo reducido y al mismo tiempo conservan la rigidez y alta velocidad de operación de un mecanismo paralelo.

Clasificación de manipuladores híbridos

La clasificación que a continuación se muestra fue propuesta por Sukhan Lee y Sungbok Kim en su artículo *Efficient inverse kinematics for serial connections of serial and parallel manipulators* (1993); la cual muestra de manera general las cuatro combinaciones posibles que pueden surgir entre la conexión de manipuladores serial-paralelos.

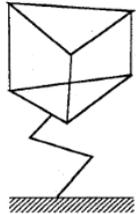
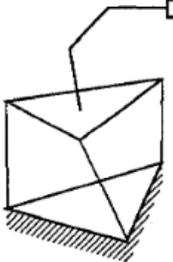
Tipo	Descripción	Imagen
Paralelo-Paralelo	Configuración en la cual dos brazos paralelos son conectados en serie como lo muestra la figura. Originalmente propuesto para hacer que los manipuladores fuesen rígidos, livianos y que operaran con movimientos rápidos.	
Serial-Paralelo	A un manipulador serial le es montado un paralelo; obteniendo un sistema de alta precisión.	
Paralelo-Serial	Conexión inversa a la anterior, la cual puede ser útil para mejorar la estabilidad y fuerza del punto de conexión paralelo serie. Configuración que fue empleada en el presente documento.	
Serial-Serial	Configuración convencional de los manipuladores seriales.	

Tabla 3 Ejemplos de robots híbridos

Robot móvil

Se define a la robótica móvil como la rama de la robótica que tiene por objeto de estudio el diseño y construcción de sistemas electromecánicos capaces de desplazarse de manera autónoma sin estar sujetos físicamente a un solo punto; cuyo sistema de desplazamiento es proporcionado mediante dispositivos de locomoción. [10]

Los robots móviles están clasificados de manera muy general de acuerdo al tipo del sistema de locomoción que emplean, resultando en muchas y muy variadas formas de moverse; es por ello que la selección del sistema de locomoción es un aspecto importante para el diseño de un robot móvil, debido a ello han surgido robots móviles que pueden caminar, deslizarse, correr, brincar en el caso de tratarse de robots terrestres, y nadar o volar en caso de los acuáticos y aéreos respectivamente; cabe destacar que muchos de estos mecanismos de

locomoción han sido inspirados por sus contrapartes biológicas. [11] En el caso de los robots terrestres se pueden identificar tres formas básicas de locomoción: mediante orugas, patas o ruedas; sin embargo, por fines de interés general de este trabajo únicamente se profundiza en los robots móviles cuya locomoción es por medio de ruedas.

Algunas de las ventajas al implementar un sistema de locomoción por ruedas es por una parte, la facilidad de construcción y control, contando con estructuras menos complejas además del hecho de que es posible transportar cargamento pesado en comparación con los robots que usan patas además poseen mayor estabilidad dado que siempre están en contacto con una superficie en contraste con los robots caminantes; sin embargo y a pesar de todas estas ventajas éstos aún presentan problemas al recorrer terrenos muy accidentados.

Tipos de ruedas

La elección del tipo de ruedas que se acopla al robot móvil se encuentra estrechamente ligado con su modelo cinemático, ya que según sea el caso, éstas le proporcionan más o menos grados de libertad. De manera general las ruedas que son empleadas para estos robots se encuentran clasificadas en dos grandes grupos ya sea del tipo convencionales o del tipo especiales como lo muestra el siguiente diagrama.

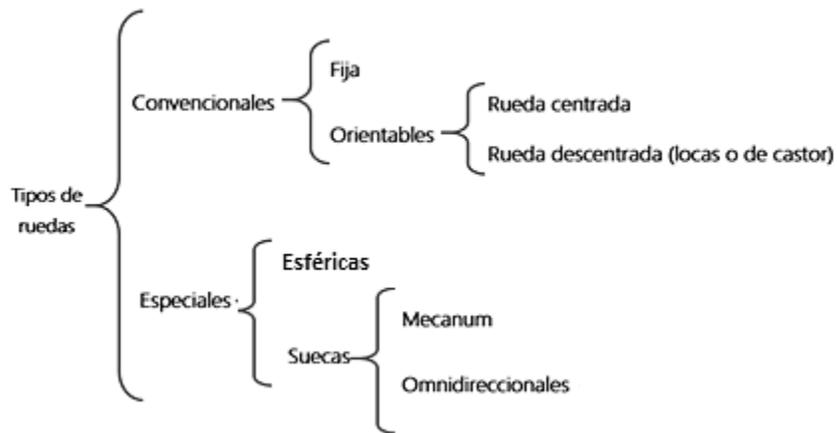
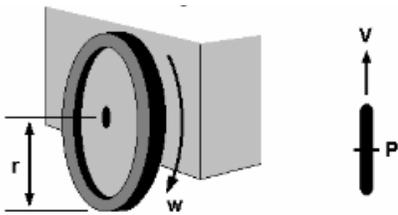
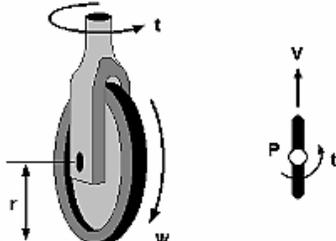
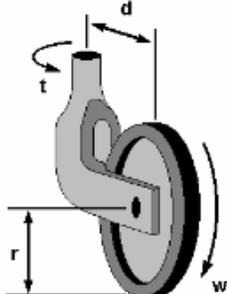
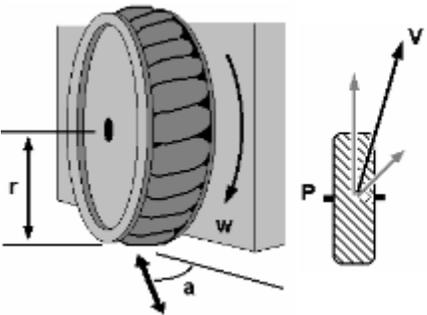
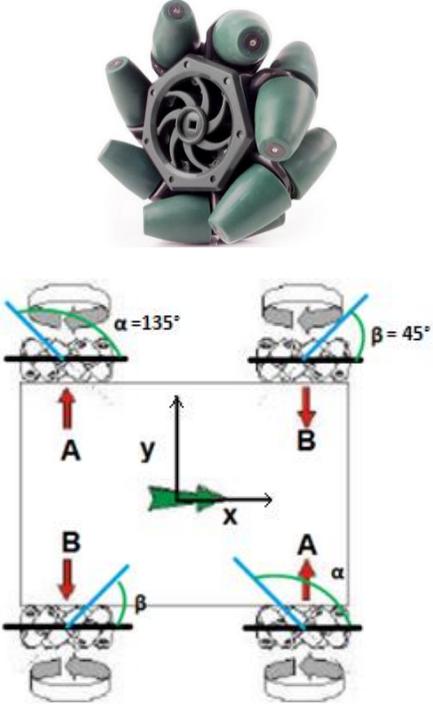


Ilustración 7 Clasificación de ruedas

Dentro de esta clasificación se puede identificar de manera adicional a las ruedas del tipo pasivas y las motorizadas que dotan al robot de tracción y movimiento; así como las del tipo directriz ruedas que como su nombre lo indica son de dirección y orientación controlable. [12]

Tipo de rueda	Descripción
<p data-bbox="396 279 524 310">Rueda fija</p>  <p data-bbox="315 562 609 594"><i>Ilustración 8 Rueda fija [12]</i></p>	<p data-bbox="706 279 1339 604">El eje de la rueda (punto P) está fijo a la estructura del robot que por lo general está se encuentra asociada al sistema de tracción del robot. Es necesario analizar dos movimientos uno, asociado al desplazamiento lineal a lo largo del plano de la rueda (punto de contacto con el suelo) y dos el movimiento ortogonal al plano de la rueda.</p>
<p data-bbox="293 648 630 680">Rueda orientable centrada</p>  <p data-bbox="248 976 675 1050"><i>Ilustración 9 Rueda orientable centrada [12]</i></p>	<p data-bbox="706 648 1339 877">Tanto las ruedas orientables centradas como las descentradas se caracterizan por rotar alrededor del eje vertical (perpendicular al plano de contacto). Este eje de rotación vertical permite orientarla con respecto al vehículo [3]</p> <p data-bbox="706 888 1339 1024">Donde r es el radio, P el eje de rotación, v la velocidad lineal y w la velocidad angular y t la rotación respecto al eje vertical.</p>
<p data-bbox="264 1094 657 1167">Rueda descentrada o rueda de castor</p>  <p data-bbox="289 1493 638 1524"><i>Ilustración 10 Rueda castora [12]</i></p>	<p data-bbox="706 1094 1339 1514">También conocida como rueda castor es una rueda orientable con respecto a la estructura, tal que la rotación del plano de la rueda es alrededor de un eje vertical el cual no pasa a través del centro de la rueda, ya que existe una distancia d entre el eje vertical de orientación y el de rotación de la rueda. Su principal función es la de dar estabilidad a la estructura mecánica del robot como rueda de dirección. [13]</p>

<p>Ruedas suecas omnidireccionales</p>  <p><i>Ilustración 11 Ruedas suecas omnidireccionales [12]</i></p>	<p>La rueda omnidireccional se define como una rueda estándar a la cual se le ha dotado de una corona de rodillos, cuyos ejes de giro resultan perpendiculares a la dirección normal de avance. Por sí misma tiene los tres grados de libertad que hay en el plano. [14]</p>
<p>Ruedas suecas tipo Mecanum</p>  <p><i>Ilustración 12 Rueda Mecanum</i></p>	<p>Estas son similares a las mostradas en la figura anterior con la excepción que los rodillos poseen una rotación de cierto ángulo; ya sea de 45° si se trata de una rueda tipo B o 135° si es del tipo A; tal y como se muestra en la figura. Teniendo así dos grados de libertad (lineal y angular).</p>

Configuración de robots móviles con ruedas

Se puede definir un robot móvil con ruedas (RMR) como un sistema electromecánico controlado, que utiliza como locomoción ruedas de algún tipo, y que es capaz de trasladarse de forma autónoma a una meta preestablecida en un determinado espacio de trabajo. Entendiendo por autonomía de un robot móvil, al dominio que tiene éste para determinar su curso de acción mediante un proceso propio de razonamiento con base en sensores, en lugar de seguir una secuencia fija de instrucciones. [15] Una vez aclarado el concepto de RMR se

describen las principales configuraciones cinemáticas de éste tipo de robots; entiéndase como configuración cinemática al arreglo en el cual el análisis sólo toma en consideración los movimientos en el espacio y tiempo sin tomar en cuenta las fuerzas que lo producen.

Triciclo clásico

Es el diseño conceptual más simple para un robot móvil, consiste en tener una sola rueda central motorizada quien a su vez es de direccionamiento, requiere también de dos ruedas castoras pasivas en la parte trasera para lograr así una estabilidad en el plano. [16]

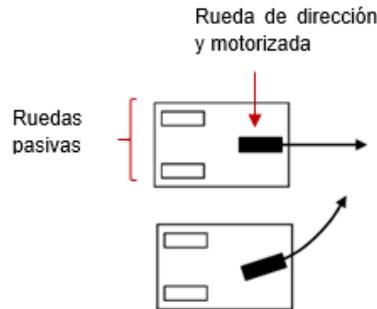


Ilustración 13 Triciclo clásico [16]

Diferencial

La configuración diferencial consta de dos ruedas motorizadas de manera independiente, es decir, que cada una de ellas cuenta con un motor acoplado; además suelen tener ruedas pasivas orientables descentradas (ruedas locas) para así evitar cabeceos y mantener la horizontal del robot. Al contar con dos motores que se manejan independientemente, resulta más compleja su operación a diferencia de otras configuraciones que cuentan con un solo motor de tracción.

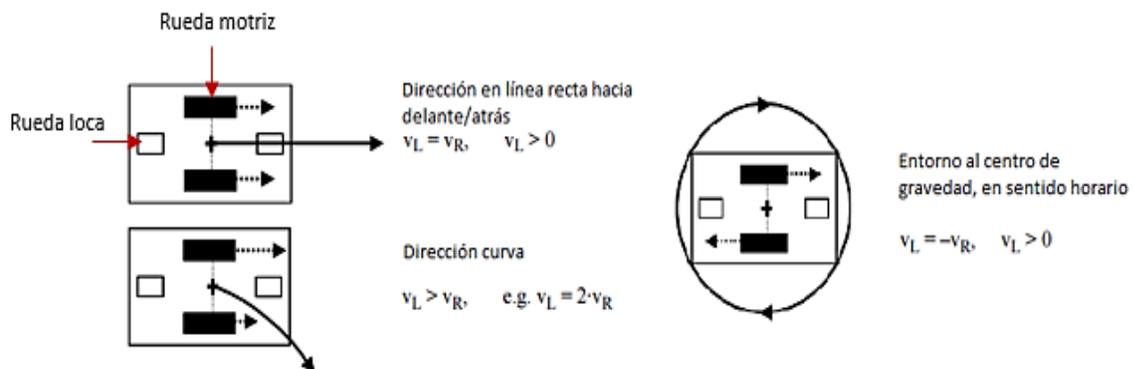


Ilustración 14 Configuración diferencial [16]

Direccionamiento Ackerman

Este tipo de configuración es la que actualmente se emplea en los automóviles, la cual consiste en dos ruedas traseras de tracción y en dos ruedas delanteras de direccionamiento. Este tipo de direccionamiento obedece al principio de que las ruedas deben formar círculos concéntricos asociados a un centro de rotación instantáneo, para ello, el ángulo de giro de la rueda interior debe ser mayor que la exterior, tal y como se muestra a continuación

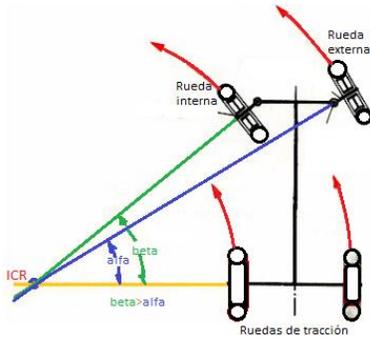


Ilustración 15 Configuración Ackerman

El desplazamiento en línea recta no presenta ningún problema dado que las ruedas traseras son impulsadas por un eje en común; sin embargo, una de las desventajas que presenta es la limitación en las maniobras pues presenta restricciones no holónomas; es decir, que el robot no puede moverse instantáneamente de manera lateral, debido a la dirección de deslizamiento de las ruedas. [12]

Locomoción síncrona

En este tipo de locomoción todas las ruedas actúan de forma síncrona por un motor; el cual define la velocidad del robot, es decir, que es el encargado únicamente del desplazamiento ya que la orientación es proporcionada por un segundo motor mediante una cadena, como se ilustra a continuación. [17]

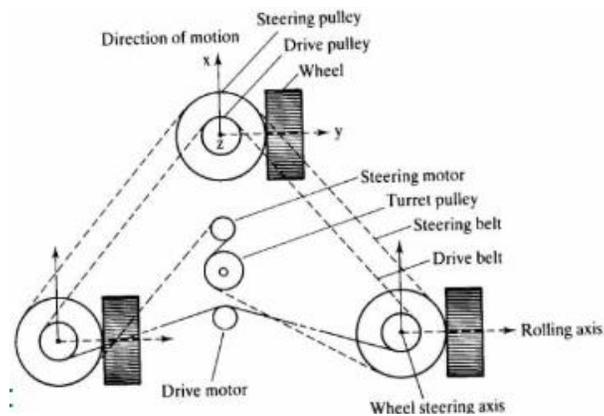


Ilustración 16 Locomoción síncrona [12]

Una de las ventajas que presenta esta configuración es que puede ser orientado instantáneamente en cualquier dirección, por lo que es correcto calificarlo como de tipo omnidireccional.

Configuración omnidireccional

El robot con configuración omnidireccional presenta restricciones del tipo holónomas, es decir, que puede moverse en todas direcciones, cambiando su dirección de desplazamiento en cualquier instante de tiempo; siendo ésta la principal ventaja de estos robots frente a los que no lo son.

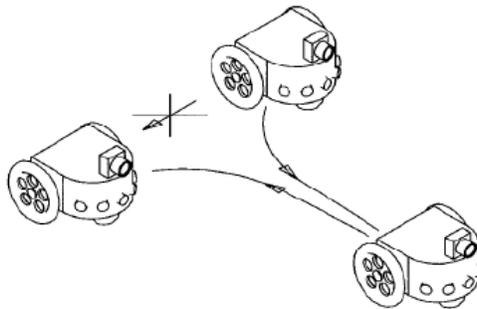


Ilustración 17 Robot diferencial no holonómico [14]

Un sistema omnidireccional es aquel que posee la misma cantidad de desplazamientos posibles, que la cantidad de coordenadas generalizadas necesarias para localizar al sistema de forma única en un plano (x, y, θ) . Así, un robot omnidireccional es aquel que puede ejecutar movimientos en cualquiera de las tres componentes (x, y, θ) , es decir, movimientos de traslación frontales y de reversa, laterales y de rotación al mismo tiempo y con total libertad, a partir de un estado de inmovilidad. [18]

La configuración omnidireccional se puede lograr de dos formas:

- Usando ruedas convencionales, en una locomoción síncrona en dónde las ruedas por si mismas no son las que le dan esta propiedad omnidireccional, sino el mecanismo en cuestión.
- Usando las llamadas ruedas especiales ya sea del tipo de esféricas o las populares ruedas suecas en cualquiera de sus dos presentaciones (Mecanum u omnidireccionales) las cuales no imponen restricciones cinemáticas en el chasis del robot.

Estado del Arte

Dada la configuración que se tiene en el robot, existe una buena cantidad de trabajos previos a los que se puede referir, pues dicha configuración cuenta con tres tipos de robots.

Una referencia que se tomó en el presente texto es la realizada por Tanigaki, Fujiura, Akase y Imagawa [19], donde se desarrolla un robot que cosecha cerezas, dado que la temporada es corta y representa un peligro para los agricultores pues los cerezos son árboles de gran tamaño, además de la ineficiencia que ello representa.

Por ello implementaron un sistema robótico compuesto de un manipulador, y varios sistemas de visión artificial (láser, infrarrojo y cámaras 3D), mediante los cuales se pretendió realizar la actividad de cosechar de forma automática, segura y eficiente.

El trabajo concluye con que es un avance considerable en el área de la automatización para la agricultura, además claro de la robótica inteligente.

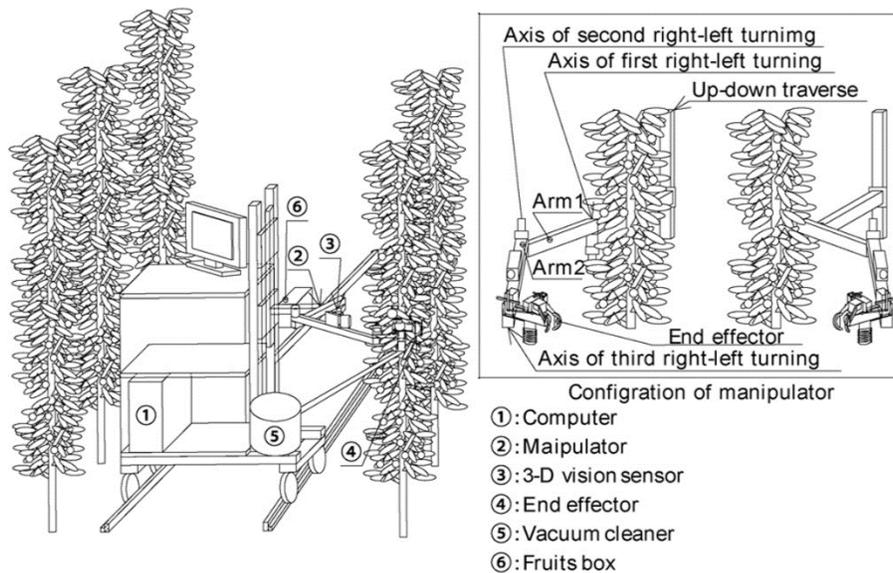


Ilustración 18 Esquema de robot cosechador de cerezas [22]

Respecto a la robótica móvil se tienen los siguientes trabajos como referencia.

El trabajo [4], presenta un robot móvil zoomórfico, y consta de 6 patas motorizadas el cual tiene por objeto, estudiar el movimiento de un insecto y aplicarlo a una máquina con el fin de poderlo implementar a diferentes sistemas, es decir escalarlo y usarlo en varias formas.

De igual forma se analiza cinemática y dinámicamente al robot para controlar su movimiento de manera adecuada. La forma en la que se desarrolla el estudio, deja como base el control de un robot omnidireccional hexápodo, con varias rutinas de movimiento y con la posibilidad de ampliar dicho control a una mayor cantidad de motores, además de especificaciones mecánicas y matemáticas para su análisis.

Introducción



Ilustración 19 Modelo funcional final de robot hexápodo [4]

La tesis de maestría del Dr. Patricio Martínez Zamudio [20] estudia el comportamiento de un manipulador paralelo delta híbrido en la realización de trayectorias que resolvieran las ecuaciones de lazo del robot, resultado de un análisis cinemático completo de esta configuración. Dicho estudio tiene como objetivo proponer nuevas técnicas de análisis para la generación de conocimiento en esta área de la robótica híbrida. Cabe mencionar que resultado de esta unión de robots paralelos se muestran ventajas como el aumento del espacio de trabajo del robot paralelo por sí solo.

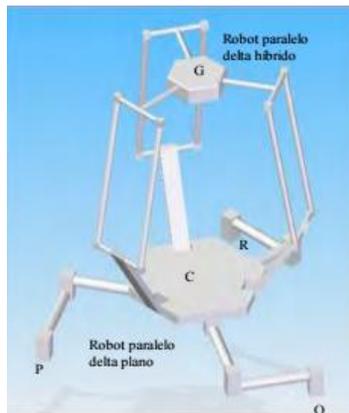


Ilustración 20 Robot híbrido manufacturado para la tesis del Dr. Patricio Martínez Zamudio [20]

Adicionalmente se realizó una revisión del artículo de S. Ali A. Moosavian, Alireza Pourreza y Khalil Alipour [21], que cabe mencionar es el único que se halló con una configuración similar a la que se estudia en este texto, pues consiste en una unión de un robot móvil 2,0 un robot paralelo delta y un manipulador serial.

El objetivo de dicho estudio es analizar la cinemática y dinámica del robot al momento de tomar algún objeto pesado. De forma que se obtienen los resultados de sus análisis al realizar simulaciones y cálculos del robot.

Para realizar lo anterior, los autores proponen la división del robot en tres etapas, la plataforma móvil, el robot paralelo y el manipulador serial, con lo que se obtiene de forma modular la

cinemática del robot. Y, respecto a la dinámica, utilizando el enfoque de Newton-Euler y de Lagrange, se dividió el sistema en dos, la parte móvil en un lado y el manipulador paralelo serial en otro.

Finalmente se comprobó el modelo obtenido mediante entradas y restricciones en las simulaciones, que avalaron los datos resultado de su estudio.

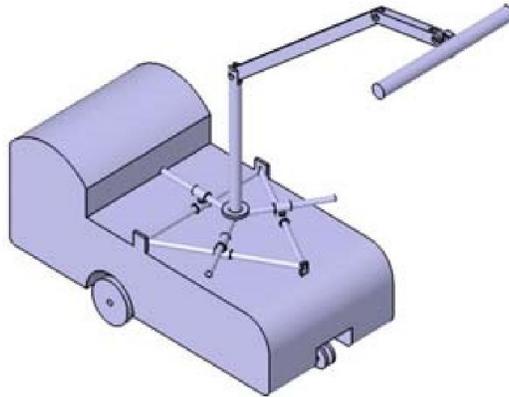


Ilustración 21 Modelo en CAD del estudio de un robot manipulador híbrido móvil [21]

Otra configuración que es importante destacar se muestra en la tesis de licenciatura presentada en 2013 por Ignacio Carlos Cruz López y Juan de Dios Flores Méndez [22], se demuestra el control y manejo de un sistema robótico híbrido compuesto por dos robots paralelos, un robot delta plano 3R, y un robot manipulador paralelo espacial.

Dicho documento, expone los resultados obtenidos en la realización de pruebas con un modelo físico y la forma en que se coordinan los movimientos. El modelo físico resulta ser un sistema robusto que provoca una gran estabilidad en el manipulador y en la ejecución de las trayectorias.

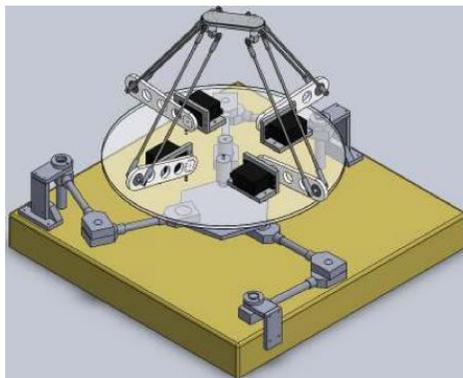


Ilustración 22 Modelo en CAD del modelo funcional para la tesis de Ignacio Carlos Cruz López y Juan de Dios Flores Méndez [20]

Igualmente, en el ámbito de la robótica móvil se tiene la siguiente tesis realizada por Esteban Suárez y Alina Sánchez [14].

En dicho trabajo se muestra el desarrollo del control de un robot móvil omnidireccional de ruedas tipo mecanum, con ello, sentando las bases para el control de este tipo de robots, ahora más comunes y con bastantes aplicaciones.

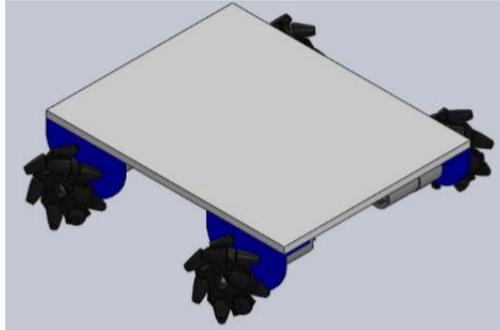


Ilustración 23 Plataforma móvil omnidireccional utilizada en la tesis [14]

Definición del problema

Como anteriormente se define, la robótica híbrida es la rama de la robótica que estudia la combinación entre diferentes estructuras, las cuales conforman un único sistema; buscando con ello aprovechar las cualidades de dichas configuraciones. A pesar de esto, realizar estas combinaciones implica un mayor grado de complejidad en su análisis e implementación, lo cual se traduce en una escasez de estudios relacionados con el área.

Este trabajo presenta el estudio de un manipulador móvil híbrido, constituido por una base con ruedas omnidireccionales tipo mecanum, un robot paralelo en el plano y un manipulador serial. Surgiendo con ello un área de oportunidad relacionada con la coordinación y la distribución de los movimientos de cada robot, para llevar a cabo la tarea definida.

El presente trabajo abarca únicamente el modelado cinemático del sistema, la coordinación de movimientos y implementación de un modelo funcional.

Objetivo

Se tiene como objetivo principal implementar un modelo funcional de un robot manipulador móvil híbrido, que permita estudiar la coordinación de sus estructuras mediante la repartición de tareas para el transporte de materiales.

Para lo anterior, se tienen las siguientes metas:

- Implementar modelo físico funcional.
- Desarrollar y aplicar el modelado cinemático.

- Proponer la coordinación de movimientos entre todas las estructuras que comprenden al robot híbrido.
- Generar conocimiento nuevo y de alto impacto en la rama de la robótica híbrida.

Hipótesis

Con lo anteriormente expuesto, se plantea la siguiente hipótesis.

Es posible distribuir la tarea de transporte de materiales, de una manera proporcional, entre los diferentes componentes que constituyen a un manipulador móvil híbrido omnidireccional-paralelo-serie, aprovechando la suma de cualidades que presentan cada uno de estos componentes.

Justificación

Dado que la robótica híbrida es una rama de la robótica poco explorada, no hay muchos estudios en arreglos similares al que se presenta. Por tanto, este trabajo, puede ampliar los estudios en el área de la robótica móvil e híbrida, al generar conocimiento nuevo en dichos temas.

Esta propuesta tiene como característica que toma ventaja de las cualidades particulares de cada robot para formar un sistema que minimiza las limitaciones de cada parte, e incrementa sus ventajas. De forma que el conjunto, resulta ser mejor que cada elemento por separado.

Alcances

Efectuar la repartición de tareas en un robot móvil híbrido autónomo, para el transporte de materiales.

Análisis Cinemático

El presente estudio muestra el desarrollo matemático desde un enfoque cinemático, y dado que uno de los objetivos es probar las teorías base y generar conocimiento nuevo, no es realizado un estudio dinámico del sistema.

Dado ello, el procedimiento para describir matemáticamente a la cinemática del sistema es el siguiente: al analizar la plataforma móvil se realiza en el espacio de las velocidades, por lo que se emplea el método de propagación de velocidades descrito en el libro *Introduction to Robotics Mechanics and Control*, de John J. Craig [24], de esta forma se tiene una referencia inercial que define el movimiento de la plataforma móvil en velocidades.

El centro de dicha plataforma, resulta la referencia del robot paralelo delta de 3 GDL, el cual se analiza desde el punto de vista de las posiciones mediante el método de las matrices de rotación y traslación puras, para la cinemática directa del robot paralelo y mediante el método de sumas vectoriales y reducción polinómica se obtiene finalmente, la solución de ese sistema, es decir, la cinemática inversa. Cabe mencionar que nuevamente la solución del punto del efector final del robot paralelo es la referencia del manipulador serial.

Para la solución de la cinemática directa del manipulador serial de 4 GDL, se emplean nuevamente las matrices de rotación y traslación puras; resolviendo la cinemática inversa del manipulador serial considerándolo como una cadena más del paralelo, con lo que se logra tener consistencia entre cada sistema. Y se logra la solución general del mismo.

Se define a una matriz de transformación, sea de rotación o traslación, como la matriz que contiene la información de una transformación lineal, aplicada a un vector en un espacio vectorial inicial, en su vector imagen en otro espacio vectorial final. Las matrices de transformación, las cuales no deben ser confundidas con las matrices de rotación ni traslación, están compuestas por cuatro submatrices, las cuales contienen toda la información del sistema en cuestión; el factor de escalamiento representado por una matriz de 1x1, el vector de perspectiva representado por una matriz de 1x3, el vector de traslación, definido en el vector de 3x1, y la matriz de rotación de 3x3. Comenzando con las matrices de rotación completas, se tienen rotaciones en los 3 ejes, X, Y, Z, se hace hincapié en que la perspectiva, escalamiento y traslación son cero o unitarias, pues solamente se tiene la información de rotación, por lo cual se definen las 3 diferentes matrices:

$$Q_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad Q_y = \begin{pmatrix} \cos \alpha & 0 & \sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad Q_z = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Ecuación 1 Matrices de rotación sobre los ejes X, Y y Z respectivamente

Donde se puede apreciar, α es el ángulo que rota un sistema respecto a otro. Por otro lado, las traslaciones se definen mediante la siguiente matriz, donde la matriz de rotación resulta ser la matriz identidad, el vector perspectiva es nulo y el factor de escalamiento es unitario:

$$T_{xyz} = \begin{pmatrix} 1 & 0 & 0 & P_x \\ 0 & 1 & 0 & P_y \\ 0 & 0 & 1 & P_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Ecuación 2 Matriz de traslación

Donde el punto P, es el punto al que se desea trasladar el sistema, con sus respectivas componentes en X, Y, Z.

Cinemática del Manipulador Serial

Para realizar el análisis del manipulador serial, fueron propuestos los siguientes sistemas de referencia:

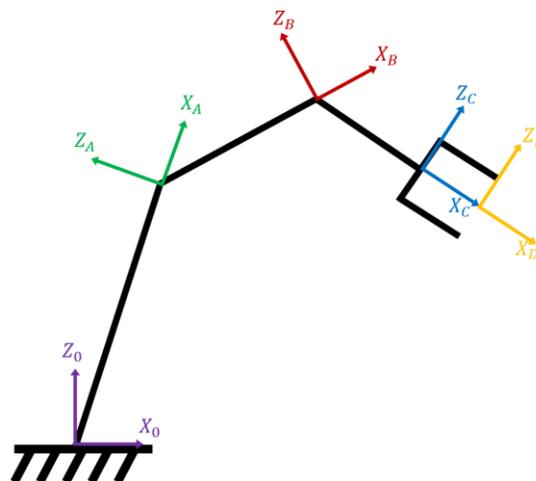


Ilustración 24 Sistemas de referencia del manipulador serial

Posteriormente se definieron las siguientes variables articulares, es decir, los ángulos.

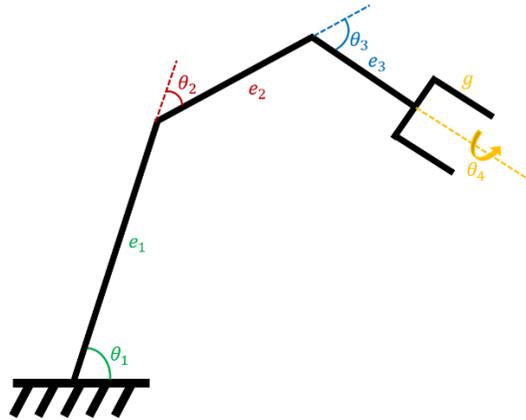


Ilustración 25 Definición del manipulador serial con base en sus variables

Dado que es implementada la solución mediante una suma vectorial, se tienen las siguientes transformaciones respecto de un sistema a otro:

$$T_{0A} = Q_y(\theta_1) T_x(e_1)$$

Ecuación 3 Transformación de A visto desde 0

$$T_{AB} = Q_y(\theta_1) Q_y(\theta_2) T_x(e_2)$$

Ecuación 4 Transformación de B visto desde A

$$T_{BC} = Q_y(\theta_1) Q_y(\theta_2) Q_y(\theta_3) T_x(e_3)$$

Ecuación 5 Transformación de C visto desde B

$$T_{CD} = Q_y(\theta_1) Q_y(\theta_2) Q_y(\theta_3) Q_x(\theta_4) T_x(g)$$

Ecuación 6 Transformación de D visto desde C

Por lo que para definir cada sistema visto desde el sistema inercial del manipulador se tienen las siguientes sumas:

$$T_{0B} = T_{0A} + T_{AB}$$

Ecuación 7 Sistema B visto desde el sistema inercial

$$T_{0C} = T_{0B} + T_{BC}$$

Ecuación 8 Sistema C visto desde el sistema inercial

$$T_{0D} = T_{0C} + T_{CD}$$

Ecuación 9 Sistema D visto desde el sistema inercial

De esta forma las ecuaciones que definen al punto en el efector final, con base en las variables articulares, es decir los 4 ángulos theta, visto desde la base del manipulador, se tiene:

$$X = e_1 \cos(\theta_1) + e_2 \cos(\theta_1 + \theta_2) + (e_3 + g) \cos(\theta_1 + \theta_2 + \theta_3)$$

Ecuación 10 Coordenada en X del efector final desde la base del manipulador

$$Y = 0$$

Ecuación 11 Coordenada en Y del efector final desde la base del manipulador

$$Z = e_1 \sin(\theta_1) + e_2 \sin(\theta_1 + \theta_2) + (e_3 + g) \sin(\theta_1 + \theta_2 + \theta_3)$$

Ecuación 12 Coordenada en Z del efector final desde la base del manipulador

Con lo anterior se tienen las ecuaciones del efector final con base en las variables articulares, que son los primeros tres elementos de la cuarta columna de la matriz generada de las anteriores sumas; es decir, el vector posición. Note que las ordenadas tienen un valor de cero dado que el manipulador de 4 grados de libertad, no tiene un giro en su base, por tanto, está confinado a un espacio de trabajo en el plano XZ. Dicho giro se proporciona mediante la base delta plano. Con lo anterior se tiene resuelta la cinemática directa.

En el caso de la cinemática inversa, se desea conocer los valores de las variables articulares con base en el efector final, para resolverlo, se emplea el método de reducción polinómica.

A continuación, se parte de las ecuaciones obtenidas en la cinemática directa, del efector final:

$$X = e_1 \cos(\theta_1) + e_2 \cos(\theta_1 + \theta_2) + (e_3 + g) \cos(\theta_1 + \theta_2 + \theta_3)$$

$$Z = e_1 \sin(\theta_1) + e_2 \sin(\theta_1 + \theta_2) + (e_3 + g) \sin(\theta_1 + \theta_2 + \theta_3)$$

Ecuación 13 Punto del efector final del robot serial

Recordando que $\theta_1 + \theta_2 + \theta_3 = \varphi$:

$$X = e_1 \cos(\theta_1) + e_2 \cos(\theta_1 + \theta_2) + (e_3 + g) \cos(\varphi)$$

$$Z = e_1 \sin(\theta_1) + e_2 \sin(\theta_1 + \theta_2) + (e_3 + g) \sin(\varphi)$$

Ecuación 14 Punto del efector final empleando la restricción cinemática

Con lo anterior se despejan, en ambas ecuaciones, los términos de $e_2 \cos(\theta_1 + \theta_2)$ para X y $e_2 \sin(\theta_1 + \theta_2)$ para Z:

$$e_2 \cos(\theta_1 + \theta_2) = X - e_1 \cos(\theta_1) - (e_3 + g) \cos(\varphi)$$

$$e_2 \sin(\theta_1 + \theta_2) = Z - e_1 \sin(\theta_1) - (e_3 + g) \sin(\varphi)$$

Ecuación 15 Despeje auxiliar para la solución de la cinemática inversa

Se elevan ambas ecuaciones al cuadrado y se suman para emplear la identidad trigonométrica de $\cos^2(\theta) + \sin^2(\theta) = 1$; simplificando así dos incógnitas:

$$e_2^2 \cdot (\cos(\theta_1 + \theta_2))^2 = (X - e_1 \cos(\theta_1) - (e_3 + g) \cos(\varphi))^2$$

$$e_2^2 \cdot (\sin(\theta_1 + \theta_2))^2 = (Z - e_1 \sin(\theta_1) - (e_3 + g) \sin(\varphi))^2$$

Ecuación 16 Ecuación elevada al cuadrado para implementar las identidades trigonométricas

Recordando que φ es una constante, la única incógnita en la ecuación anterior es θ_1 , por lo que se hace uso de una identidad para el seno, coseno y tangente de dicho ángulo:

$$\sin(\theta_1) = \frac{2t}{1+t^2} \quad \cos(\theta_1) = \frac{1-t^2}{1+t^2} \quad t = \tan\left(\frac{\theta_1}{2}\right)$$

Ecuación 17 Identidades trigonométricas implementadas en la cinemática inversa

$$(X - e_1 \cos(\theta_1) - (e_3 + g) \cos(\varphi))^2 + (Z - e_1 \sin(\theta_1) - (e_3 + g) \sin(\varphi))^2 - e_2^2 = 0$$

Ecuación 18 Polinomio cuadrático de solución de θ_1

Obteniendo una ecuación de segundo grado en función de t, que es sustituida por la tercera identidad trigonométrica, y se despeja finalmente a θ_1 , por ser una ecuación de segundo grado, se tienen dos soluciones para θ_1 para el manipulador.

$$\theta_1 = 2 \tan^{-1}(\pm t)$$

Para determinar θ_2 , se sustituye el valor de θ_1 en cualquier igualdad de la Ecuación 15 que tomando a las abscisas resulta:

$$\theta_2 = \arccos\left(\frac{X - e_1 \cos(\theta_1) - (e_3 + g) \cos(\varphi)}{e_2}\right) - \theta_1$$

Ecuación 19 Solución para θ_2

Respecto al manipulador serial, se tienen estos análisis, posteriormente se establece la forma en que se enlazan con la siguiente etapa que se refiere al robot paralelo.

Cinemática del Robot Paralelo Delta Plano

El análisis que se desarrolló para el robot paralelo, emplea el método utilizado para resolver la cinemática directa del manipulador serial, es decir sumas vectoriales, por lo cual se definen los siguientes sistemas de referencia:

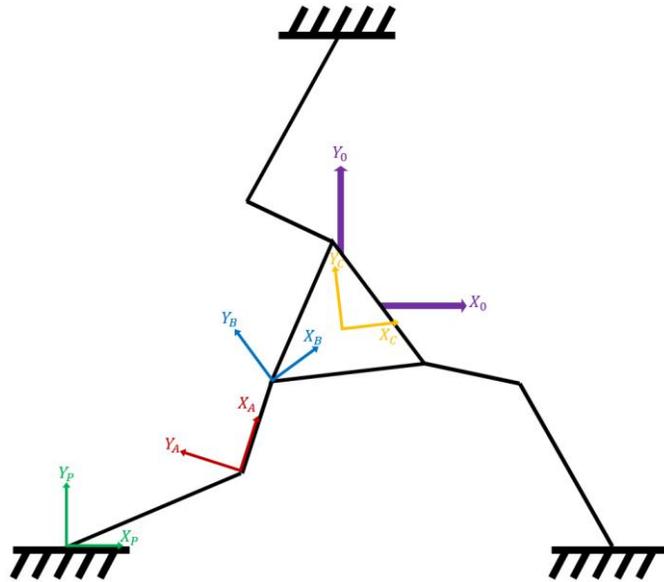


Ilustración 26 Sistemas de referencia del robot paralelo

Dichos sistemas de referencia son, el sistema 0 en morado, que es el punto de referencia del robot y que a su vez resulta ser también el centro de la plataforma móvil, el sistema P en verde, el inicio de la primera cadena cinemática, el A en rojo, sistema referido al final del primer eslabón cuyo eje X es paralelo al segundo eslabón, el sistema B en azul cuyo eje X apunta siempre al centro del delta, y finalmente el sistema C en amarillo cuyo eje X y la arista del delta son paralelos.

Dada la simetría del robot, los sistemas son repetidos en el resto de las cadenas con ciertas restricciones; a continuación, se muestra el robot con las variables que en este se presentan:

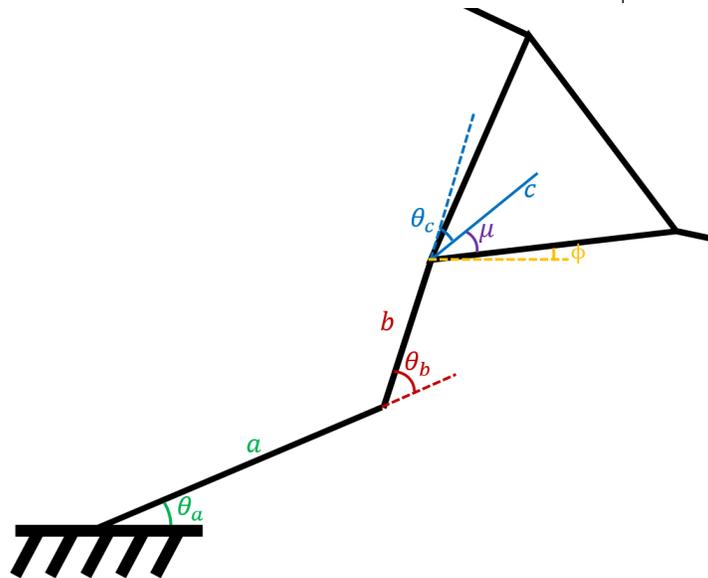


Ilustración 27 Variables del robot paralelo delta

En la anterior ilustración se muestran los eslabones a, b y c, con los respectivos ángulos θ_a , θ_b y θ_c , cabe mencionar que existe un ángulo adicional que llamamos ϕ , que relaciona a la horizontal del sistema inercial del robot con la arista inferior del delta.

De lo anterior podemos mencionar que $\theta_a + \theta_b + \theta_c$ resulta $\phi + \mu_i$ donde μ_i cambia para cada cadena en 120° , siendo $\mu_1 = 30^\circ$, $\mu_2 = 150^\circ$ y $\mu_3 = 270^\circ$, de modo que μ_i es el i-ésimo ángulo medido en la junta rotacional y, entre la arista inferior (la arista de estudio) y el eslabón c, de cada cadena cinemática.

Posteriormente y de forma similar al manipulador serial, se definen las transformaciones de cada sistema visto desde el sistema inmediato anterior, por tanto, haciendo uso nuevamente de las ecuaciones Ecuación 1 y Ecuación 2:

$$\mathbf{T}_{OP} = \mathbf{T}_{XYZ}(x_P, y_P, 0)$$

Ecuación 20 Transformación de P visto desde O

$$\mathbf{T}_{PA} = \mathbf{Q}_Z(\theta_a) \mathbf{T}_X(a)$$

Ecuación 21 Transformación de A visto desde P

$$\mathbf{T}_{AB} = \mathbf{Q}_Z(\theta_a) \mathbf{Q}_Z(\theta_b) \mathbf{T}_X(b)$$

Ecuación 22 Transformación de B visto desde A

$$\mathbf{T}_{BC} = \mathbf{Q}_Z(\theta_a) \mathbf{Q}_Z(\theta_b) \mathbf{Q}_Z(\theta_c) \mathbf{T}_X(c)$$

Ecuación 23 Transformación de C visto desde B

$$\mathbf{T}_{OA} = \mathbf{T}_{OP} + \mathbf{T}_{PA}$$

Ecuación 24 Transformación de A visto desde O

$$\mathbf{T}_{OB} = \mathbf{T}_{OA} + \mathbf{T}_{AB}$$

Ecuación 25 Transformación de B visto desde O

$$\mathbf{T}_{OC} = \mathbf{T}_{OB} + \mathbf{T}_{BC}$$

Ecuación 26 Transformación de C visto desde O

Finalmente, las ecuaciones que definen al punto del origen del sistema C, son:

$$X_C = x_P + a \cdot \cos(\theta_a) + b \cdot \cos(\theta_a + \theta_b) + c \cdot \cos(\mu_i + \phi)$$

$$Y_C = y_P + a \cdot \sin(\theta_a) + b \cdot \sin(\theta_a + \theta_b) + c \cdot \sin(\mu_i + \phi)$$

Ecuación 27 Punto del efector final del robot paralelo, note el cambio de variable de $\mu_i + \phi$

Note que el valor de Z se omite dado que el robot paralelo tiene su espacio de trabajo en el plano XY.

Con lo anterior, se define la cinemática directa del paralelo delta, que es el punto de partida de la cinemática inversa, la que a continuación se desarrolla con base en lo estudiado por el Dr. Patricio Martínez Zamudio [20]:

Partiendo de la Ecuación 27, se separan términos:

$$b \cdot \cos(\theta_a + \theta_b) = X_C - x_P - a \cdot \cos(\theta_a) - c \cdot \cos(\mu_i + \phi)$$

$$b \cdot \sin(\theta_a + \theta_b) = Y_C - y_P - a \cdot \sin(\theta_a) - c \cdot \sin(\mu_i + \phi)$$

Ecuación 28 Separación de términos en ambos lados de la ecuación para X y Y

Posteriormente se elevan al cuadrado ambos términos de la ecuación, para emplear la identidad de la suma del cuadrado de seno y coseno de ángulos:

$$b^2 \cdot (\cos(\theta_a + \theta_b))^2 = (X_C - x_P - a \cdot \cos(\theta_a) - c \cdot \cos(\mu_i + \phi))^2$$

$$b^2 \cdot (\sin(\theta_a + \theta_b))^2 = (Y_C - y_P - a \cdot \sin(\theta_a) - c \cdot \sin(\mu_i + \phi))^2$$

Ecuación 29 Ambos lados de la ecuación al cuadrado

$$b^2 = (X_C - x_P - a \cdot \cos(\theta_a) - c \cdot \cos(\mu_i + \phi))^2 + (Y_C - y_P - a \cdot \sin(\theta_a) - c \cdot \sin(\mu_i + \phi))^2$$

Ecuación 30 Suma de cuadrados de la ecuación anterior

Recordando que $\phi + \mu_i$ es una constante, la única incógnita en la ecuación anterior es θ_a , por lo que se hace uso de una identidad trigonométrica para el seno, coseno y tangente de dicho ángulo:

$$\sin(\theta_a) = \frac{2t}{1+t^2} \quad \cos(\theta_a) = \frac{1-t^2}{1+t^2} \quad t = \tan\left(\frac{\theta_a}{2}\right)$$

Ecuación 31 Identidades trigonométricas

$$\left(X_C - x_P - a \cdot \frac{1-t^2}{1+t^2} - c \cdot \cos(\mu_i + \phi)\right)^2 + \left(Y_C - y_P - a \cdot \frac{2t}{1+t^2} - c \cdot \sin(\mu_i + \phi)\right)^2 - b^2 = 0$$

Ecuación 32 Polinomio igualado a 0

Se obtiene la anterior ecuación de segundo grado en función de t, para sustituirla por la tercera identidad trigonométrica, y se despeja finalmente a θ_a , por ser una ecuación de segundo grado, se tienen dos soluciones para θ_a de la cadena 1.

$$\theta_a = 2 \tan^{-1}(\pm t)$$

Ecuación 33 Solución para θ_a

Para determinar θ_b , se sustituye el valor de θ_a en cualquier igualdad de la Ecuación 28 que tomando a las abscisas resulta:

Análisis Cinemático

$$\theta_b = \cos^{-1} \left(\left(X_C - x_P - a \cdot \cos(\theta_a) - c \cdot \cos(\mu_i + \phi) \right) \cdot \frac{1}{b} \right) - \theta_a$$

Ecuación 34 Solución para θ_b

Dada la simetría del robot, se repite el procedimiento anterior para cada una de las cadenas restantes, con lo que en total suman 8 posibles soluciones para el robot, las cuales se computan mediante un software de cálculo Wolfram Mathematica tanto para la solución como para la simulación.

Con esto se termina el estudio de la cinemática del robot paralelo, y a continuación se presenta el estudio del móvil omnidireccional. Posteriormente la forma en que se propone unir estos tres sistemas.

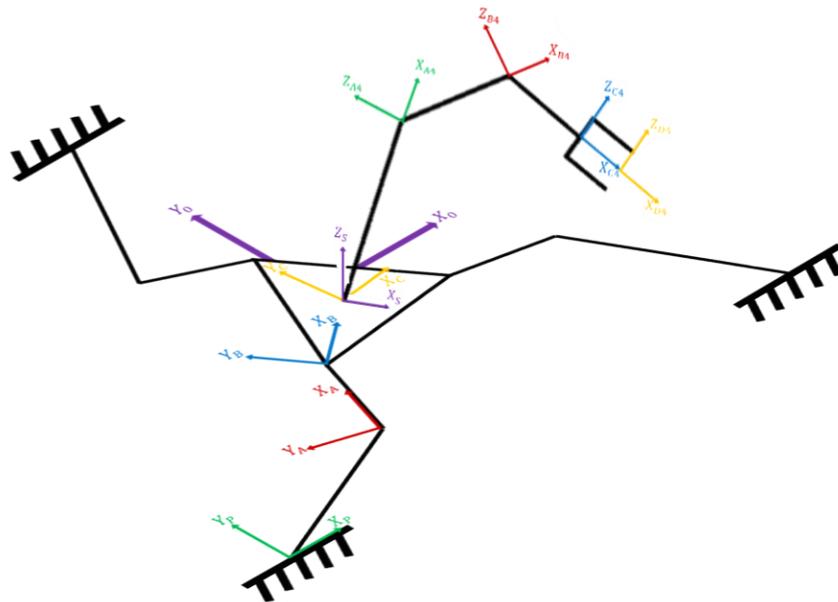


Ilustración 28 Sistemas de referencia del manipulador serie paralelo

Cinemática del Robot Móvil Omnidireccional

El análisis cinemático de todo robot móvil depende en gran parte de la geometría de las ruedas que lo conforman y de su localización en la plataforma, características especiales que en conjunto dotan al robot de un movimiento específico. Es por ello que para realizar dicho análisis cinemático es necesario definir distintos marcos coordenados que vinculen a cada una de las ruedas con el marco de referencia de la plataforma móvil (en este caso se trata del centro del robot) y así poder propagar la velocidad del centro de la plataforma a cada una de las cuatro ruedas; empleando el método de propagación de velocidades de vínculo a vínculo propuesto John J. Craig, el cual se detalla más adelante.

Es importante mencionar que para dicho modelado cinemático el autor, el Dr. Víctor Javier González Villela responsable del grupo de investigación MRG (Mechatronics Research Group) de la Facultad de Ingeniería; consideró una plataforma móvil como un cuerpo rígido no deformable; cuyas ruedas no tienen deslizamientos y cuyo desplazamiento se realiza sobre planos horizontales.

Para poder referir la posición del robot móvil en el ambiente inteligente¹, que es delimitado por el rango de visión de la cámara, se define de manera arbitraria un marco coordenado global $\{x_0, y_0\}$, el marco $\{x_1, y_1\}$ correspondiente a al centro de la plataforma (punto P), el marco $\{x_2, y_2\}$ que se encuentra alineado al eje de giro de la rueda motorizada y finalmente el marco $\{x_3, y_3\}$, asignado al eje de giro de la corona de rodillos que envuelven a la rueda motorizada. Quedando así finalmente el sistema del robot móvil descrito por los siguientes parámetros que se muestran en la siguiente ilustración.

d: coordenada de en x_1 de la posición de la rueda

b: coordenada en y_1 de la posición de la rueda

α : Ángulo de la llanta respecto al sistema de referencia $\{x_1, y_1\}$

γ : Ángulo del vector velocidad de salida de cada rodillo de las ruedas respecto al sistema de referencia $\{x_2, y_2\}$

¹ Entiéndase por ambiente inteligente al espacio de trabajo cuyos elementos tienen propiedades individuales (forma, tamaño, ubicación espacial, entre otras) que comunican a otros elementos del mismo espacio, y a la vez, conocen las propiedades de dichos otros elementos. Todo ello con el fin de crear un entorno de cooperación y colaboración entre dichos elementos. Estos elementos pueden ser robots, actuadores, ventanas, puntos de acceso Wifi, y muchas otras cosas más.

Análisis Cinemático

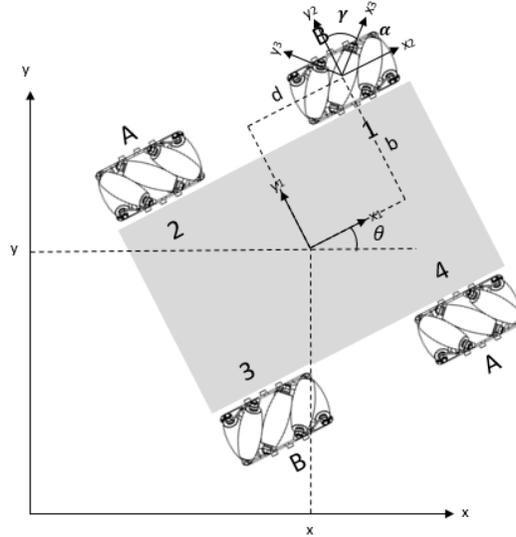


Ilustración 29 Sistemas de referencia de la plataforma móvil omnidireccional

El siguiente paso consiste en declarar el vector postura del robot, el cual queda definido por los parámetros $\{x, y\}$ que señala la posición del punto P en el sistema inercial y θ el ángulo de orientación con respecto del eje horizontal "x" inercial.

$$\xi = (X \ Y \ \theta)^T$$

Ecuación 35 Vector Postura del Robot

La rotación del marco coordenado de P $\{x_1, y_1\}$ puede ser representado por una matriz de rotación respecto del eje z, como se muestra a continuación.

$$R(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Ecuación 36 Matriz de rotación respecto al eje Z

De acuerdo con John J. Craig en su libro *Introduction to Robotics, Mechanics and Control* el estudio del movimiento de cada eslabón, siempre hace uso de un marco de referencia $\{0\}$ del cual se parte para realizar los cálculos pertinentes que indican el movimiento relativo del eslabón respecto al anterior a éste.

El método de propagación de velocidades de eslabón a eslabón que explica que la velocidad del eslabón siguiente (v_{i+1}) es la velocidad del anterior (v_i) más una componente de la velocidad que le haya agregado es eslabón (i+1); convirtiéndolo en una manera efectiva para el cálculo de las velocidades lineales y angulares de una cadena de eslabones unidos y referidos a una base;

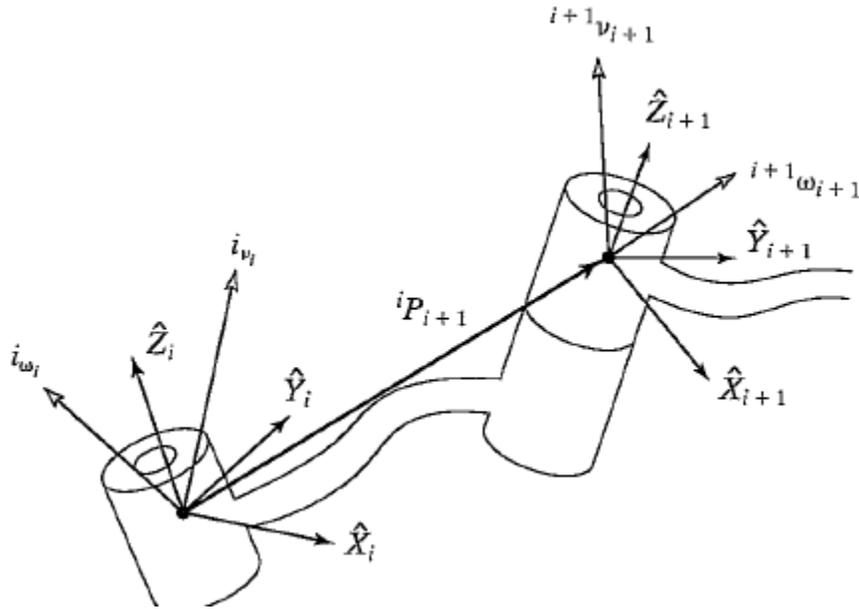


Ilustración 30 Vectores de velocidades de eslabones adyacentes

La velocidad angular del eslabón siguiente se calcula de la siguiente manera

$${}^{i+1}\omega_{i+1} = {}^{i+1}R^i \omega_i + \dot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1}.$$

Ecuación 37 Velocidad angular de un eslabón visto desde sí mismo

Donde:

- ${}^{i+1}R^i$: matriz de rotación sistema del eslabón i al marco del eslabón $i+1$
- ${}^i\omega$: velocidad angular del eslabón anterior i
- θ'_{i+1} : velocidad articular del sistema del eslabón $i+1$
- $\theta'_{i+1} * \hat{Z}_{i+1}$ que expresada de manera vectorial resulta ser: $\theta'_{i+1} * \hat{Z}_{i+1} = \begin{bmatrix} 0 \\ 0 \\ \theta'_{i+1} \end{bmatrix}$

Situación que se repite de manera similar en la velocidad lineal del origen del sistema del eslabón v_{i+1} , ya que es la misma que la del sistema anterior v_i más una nueva componente causada por la velocidad rotacional del eslabón o brazo de palanca ${}^{i+1}P$ (longitud del eslabón $i+1$).

$${}^{i+1}v_{i+1} = {}^{i+1}R^i ({}^i v_i + {}^i \omega_i \times {}^i P_{i+1}).$$

Ecuación 38 Velocidad lineal de un eslabón visto desde sí mismo

Una vez explicado lo anterior, se emplea el método de propagación de velocidades, partiendo del marco de referencia de P $\{x_1, y_1\}$ hasta propagarse al marco de las ruedas $\{x_2, y_2\}$, $\{x_3, y_3\}$. Para poder realizar esto, es necesario trasladar y rotar cada marco de referencia visto desde su marco anterior, tal y como se muestra a continuación

Análisis Cinemático

$${}^0P_1 = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \quad {}^0R_1 = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ecuación 39 Traslación y rotación del sistema 0 al sistema 1 del robot móvil

$${}^1P_2 = \begin{bmatrix} d \\ b \\ 0 \end{bmatrix} \quad {}^1R_2 = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ecuación 40 Traslación y rotación del sistema 1 al sistema 2 del robot móvil

$${}^2P_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad {}^2R_3 = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ecuación 41 Traslación y rotación del sistema 2 al sistema 3 del robot móvil

Para el caso de las ruedas Mecanum, estos son los parámetros que deben tomarse en cuenta

\mathbf{v}_R : Velocidad lineal de la rueda

\mathbf{v}_r : Velocidad lineal del rodillo de la rueda

$\dot{\phi}_r$: Velocidad angular del rodillo

$\dot{\phi}_R$: Velocidad angular de la rueda

r_r : Radio del rodillo

r_R : Radio de la rueda mecanum

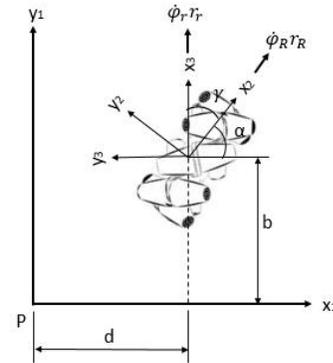


Ilustración 31 Sistemas de referencia de las ruedas del robot móvil

Finalmente propagando las velocidades de sistema a sistema se obtiene las siguientes restricciones cinemáticas:

$$\begin{pmatrix} \cos(\gamma + \alpha + \theta) & \sin(\gamma + \alpha) & d \cdot \sin(\gamma + \alpha + \theta) - b \cdot \cos(\gamma + \alpha) \\ -\sin(\gamma + \alpha + \theta) & \cos(\gamma + \alpha) & b \cdot \sin(\gamma + \alpha + \theta) - d \cdot \cos(\gamma + \alpha) \end{pmatrix} \cdot \begin{pmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{\theta} \end{pmatrix} - \begin{pmatrix} r_R \cdot \cos(\gamma) & 1 \\ -r_R \cdot \sin(\gamma) & 0 \end{pmatrix} \cdot \begin{pmatrix} \dot{\phi}_R \\ \dot{v}_r \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Ecuación 42 Matriz de restricciones cinemáticas

Cuyo vector de coordenadas generalizadas es:

$$q' = \begin{pmatrix} x' \\ y' \\ \theta' \\ \phi'_1 \\ \phi'_2 \\ \phi'_3 \\ \phi'_4 \end{pmatrix}$$

Ecuación 43 Vector de coordenadas generalizadas

Dicho vector de coordenadas generalizadas contiene las velocidades angulares de cada rueda, las cuales quedan definidas por la siguiente ecuación; expresión matemática que fue programada en Matlab.

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{r_R} & -\frac{1}{r_R} & -\frac{b+d}{r_R} \\ \frac{1}{r_R} & \frac{1}{r_R} & -\frac{b+d}{r_R} \\ \frac{1}{r_R} & -\frac{1}{r_R} & \frac{b+d}{r_R} \\ \frac{1}{r_R} & \frac{1}{r_R} & \frac{b+d}{r_R} \end{bmatrix} * \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}$$

Ecuación 44 Modelo matemático final del robot móvil omnidireccional

Implementación

A continuación, se describen los sistemas de manera general, que se utilizaron en el funcionamiento del robot híbrido, los cuales se muestran en la siguiente ilustración:

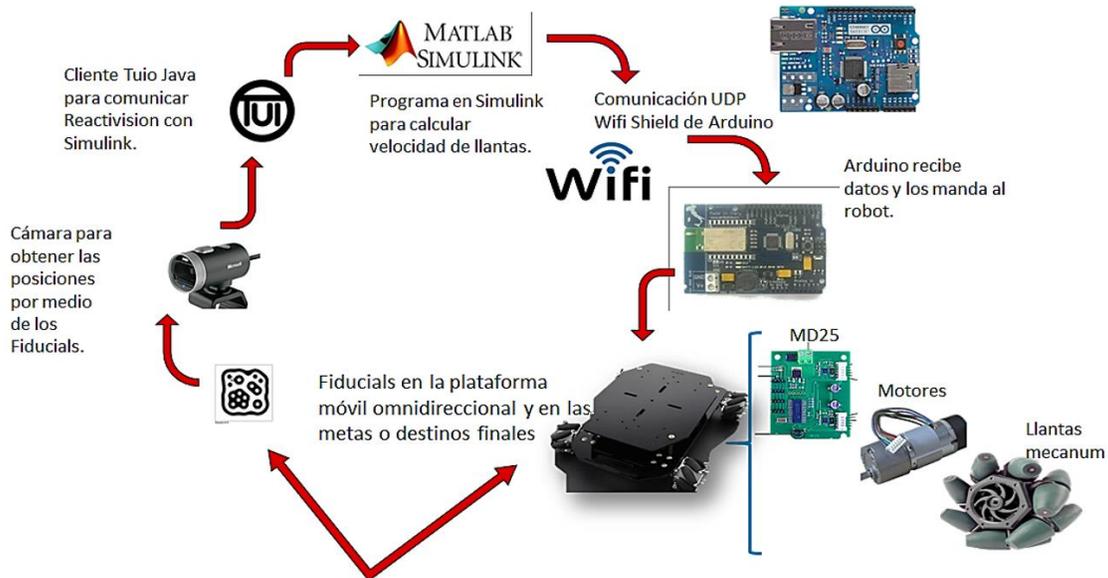


Ilustración 32 Mapa general del funcionamiento del sistema

Visión Artificial

El sistema aquí estudiado, funciona primeramente mediante la implementación de la visión artificial, pues permite adquirir datos en tiempo real, es decir, ubicaciones y orientaciones de los objetos en estudio, lo cual se logra gracias a la implementación de cámaras, marcadores y software de medición de dichos marcadores llamados fiduciales, a continuación una muestra de estos:

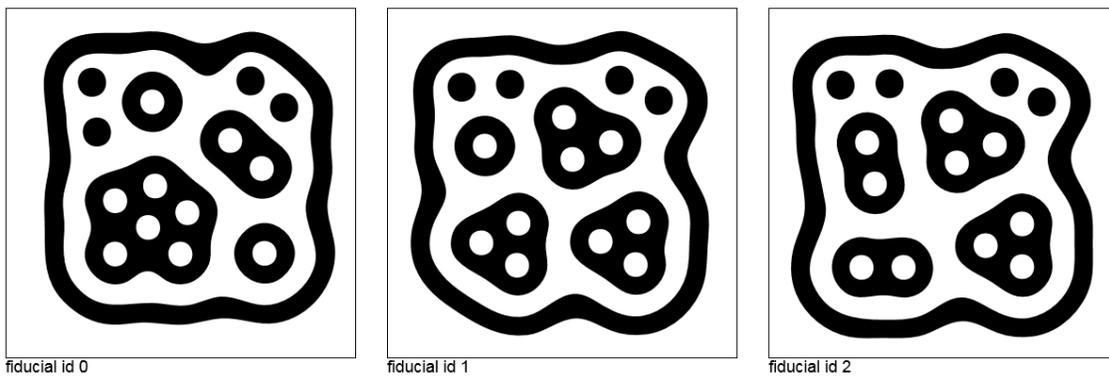


Ilustración 33 Marcadores fiducial 0, 1 y 2 respectivamente

ReacTIVision

Posteriormente, se hace uso del programa ReacTIVision, para poder leer las posiciones y orientaciones de los fiducial utilizados, enviar dichos datos al programa implementado en Matlab mediante un programa que enlaza ambas plataformas, llamado cliente TUIO, y realizar algunas calibraciones del espacio de trabajo.



Ilustración 34 Imagen generada mediante la cámara y el programa ReacTIVision

El espacio de trabajo mostrado en la ilustración anterior, cuenta con 2.20 x 1.60 metros, con la cámara a 2.2 metros de altura, cabe mencionar también que la cámara posee un ancho y alto de fotograma de 800x600 píxeles.

Espacio de Trabajo TUIO

El programa mencionado es el enlace entre los dos softwares Matlab y ReacTIVision, es el puente por donde se envía el flujo de datos de entrada, mediante el protocolo UDP, en inglés, User Datagram Protocol.

Este espacio de trabajo, es muy utilizado en aplicaciones referidas a pantallas táctiles, objetos en movimiento y seguimiento de trayectorias, como es nuestro caso. Se trata de un protocolo abierto, que describe de forma abstracta el estado de los objetos de estudio y genera la comunicación para que un cliente reciba y decodifique la información para su posterior uso.



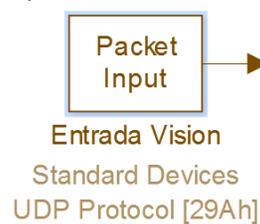
Ilustración 35 Logotipo y dirección electrónica del protocolo

Procesamiento de Datos en Matlab

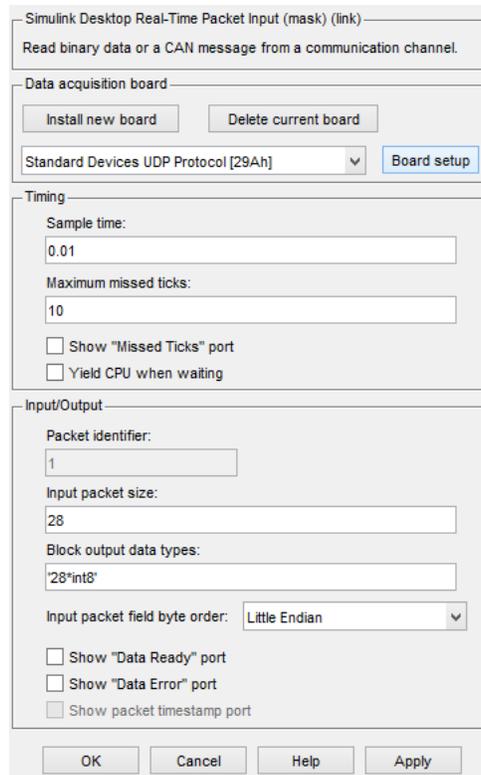
El programa principal que procesa el flujo de información del sistema y da forma al comportamiento que este tiene, se ejecuta en MatLab. Lugar donde se genera los valores de velocidades y posiciones que cada motor debe de cumplir, con base en la información de entrada de posiciones y orientaciones de los objetos en el ambiente inteligente.

Recepción de Datos

Como se menciona anteriormente, los datos de entrada que necesita Matlab para empezar a ejecutarse, son básicamente las orientaciones y posiciones de los fiducials 1, 2 y 3 asociados a la posición actual del robot, el objetivo y a un elemento pasivo respectivamente. En este apartado de recepción de datos se emplea el bloque *Packet Input* de *Matlab Simulink* el cual contiene el buffer de datos con información correspondiente a los tres fiducials empleados; datos que llegan vía Wifi mediante el protocolo de comunicación UDP en tiempo real.



Implementación



Local UDP port:	666
Remote address:	127.0.0.1
Remote UDP port:	12000

Ilustración 36 Configuración del bloque Packet Input

Módulo de Visión

Antes de realizar los ajustes de visión de Matlab es necesario haber logrado que la lente de la cámara estuviese lo más paralela posible al piso, para evitar deformaciones del espacio observado siendo conveniente utilizar una plomada para la correcta calibración de la cámara y la malla de calibración que nos proporciona ReactIVision similar a la que se muestra en la siguiente imagen.



Ilustración 37 Marco coordenado de la webcam y malla de calibración de ReactIVision

Conversión de píxeles a centímetros

De manera general ver Ilustración 38, aquí se hace el acondicionamiento de los datos de entrada para su procesamiento, separando la información de cada fiducial en posición y orientación $\{x_i, y_i, a_i, r\}$; cabe destacar que a la salida de este bloque los ángulos de orientación están dados en radianes.

Sin embargo, si se hace un zoom a este módulo de visión se puede observar que en su interior existen dos subsistemas que hacen dos ajustes previos, para que nuestra salida de información (en posición y orientación) esté dada en centímetros y radianes respectivamente.

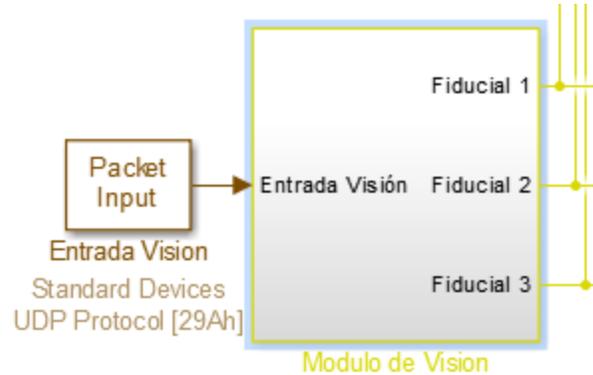


Ilustración 38 Módulo de Visión

El primer bloque llamado “3 Fiducials 28 datos de entrada” recibe los datos de entrada de los tres fiducials enviados por el cliente TUIO, datos que son separados y asociados con su respectiva $\{x_i, y_i, \theta_i\}$, a la salida de este primer bloque, y los ángulos de orientación ya están en radianes; posteriormente en “Vision_3fid” se hace una conversión de píxeles a centímetros de los datos $\{x, y\}$ de la siguiente manera. Se escala la imagen transmitida por la cámara, cuyos valores oscilan entre 0 y 100 siendo 0 el origen de nuestro marco coordenado de referencia inercial y 100 al punto final máximo alcanzado a visualizar por la cámara. Las ecuaciones para la conversión a centímetros de $\{x, y\}$ contemplan la distancia absoluta l que va desde $\{0,0\}$ hasta un $\{X_{máx}, 0\}$, y de manera similar a a como la distancia absoluta que va desde el origen hasta un $\{0, Y_{máx}\}$.

Donde

$$l = 110 [cm] \text{ y } a = 80[cm]$$

$$x_s = \frac{x}{100} * (-2l) + l$$

$$y_s = \frac{y}{100} * (2a) - a$$

Ecuación 45 Conversión de píxeles a centímetros de $\{x,y\}$

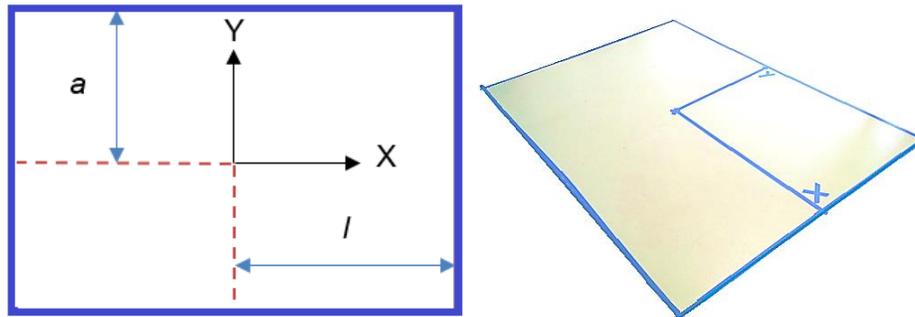


Ilustración 39 Ambiente inteligente

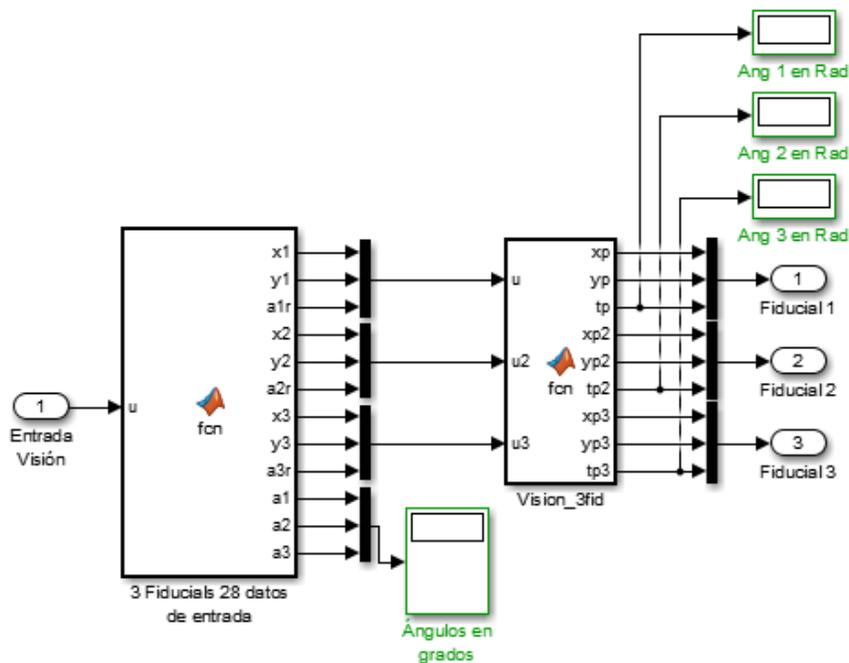


Ilustración 40 Sub-bloques del módulo de visión

Además, es en este bloque donde se realiza el ajuste de fiduciales que a continuación se explica.

Ajuste de fiduciales

Como se explica anteriormente, el análisis cinemático de la plataforma móvil parte del centro de la misma; sin embargo, dada la configuración del robot en cuestión no es posible colocar el fiducial justo en el centro de la plataforma razón por la cual se tiene que colocar una placa trasera a manera de extensión sobre la que se coloca el fiducial; modificación física que enseguida obliga a compensar de alguna forma este desplazamiento y rotación del punto P,

Implementación

situación que puede ser ejemplificada con la siguiente imagen del manipulador omnidireccional de la Ing. Irene Hernández Calderón.

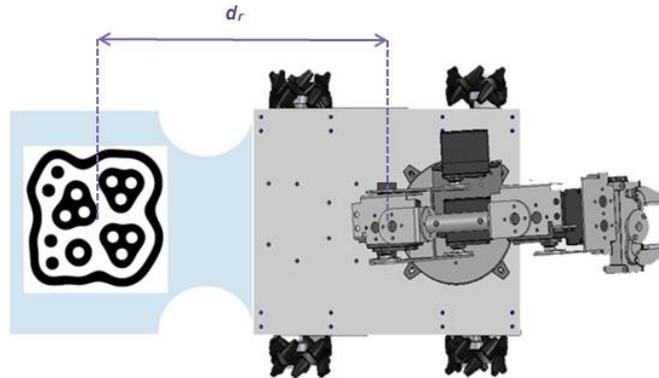


Ilustración 41 Distancia del centro del fiducial al centro de la plataforma móvil

Al igual que el robot, el objetivo de éste también es colocado en una mesa cuyo fiducial de identificación es desplazado y rotado como se ilustra en la siguiente imagen.

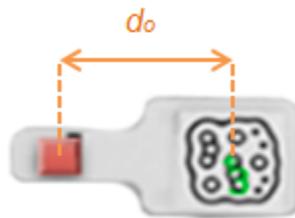


Ilustración 42 Distancia del centro del fiducial al objeto a manipular

La rotación sobre el eje Z y el traslado sobre el eje X se puede conocer empleando la matriz de rotación del P respecto de Z; multiplicado a su vez por un vector columna que indica el desplazamiento de una distancia dr sobre el eje X:

$$H_{MMO} = \begin{bmatrix} \cos\theta & -\text{sen}\theta & 0 \\ \text{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} dr \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} d_r \cos\theta \\ d_r \text{sen}\theta \\ 0 \end{bmatrix}$$

Ecuación 46 Transformación sobre el Eje X

Resultando la siguiente expresión:

$$\begin{aligned} x_{MMO_{real}} &= x_{MMO} + d_r \cos\theta_{MMO} \\ y_{MMO_{real}} &= y_{MMO} + d_r \text{sen}\theta_{MMO} \end{aligned}$$

Ecuación 47 Resultados de las transformaciones de X y Y para el móvil

De manera similar para la mesa objetivo se obtiene:

$$\begin{aligned} x_{Goal_{real}} &= x_t - d_o \cos\theta_{Goal} \\ y_{Goal_{real}} &= y_t - d_o \text{sen}\theta_{Goal} \end{aligned}$$

Ecuación 48 Resultados de las transformaciones de X y Y para el objetivo

Ecuaciones que son programadas en el bloque de "Vision_3fid" finalizando así los ajustes pertinentes de visión para el procesamiento de datos para los siguientes bloques de análisis matemáticos.

Definición de un Objetivo

Bloque en el que únicamente se definen las posiciones actuales y finales del robot; siendo el punto de partida el asociado al fiducial F1 que se encuentra fijo a la plataforma móvil y el punto meta al fiducial F2 con la mesa donde se ubica el objeto a tomar; calculando mediante el teorema de Pitágoras la distancia que existe entre estos dos fiduciales. Lo cual resulta en $PosAct = F1$ y posición deseada es $PosDes = F2$; cuya distancia entre ellos se calcula como:

$$DF1F2 = \sqrt{(x_2 - x_1)^2 + (Y_2 - Y_1)^2}$$

Ecuación 49 Distancia entre fiducial 1 a fiducial 2 (robot y objetivo)

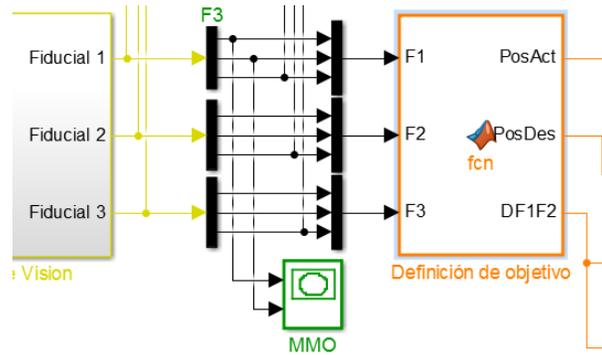


Ilustración 43 Bloque de asignación del objetivo

Campos potenciales

La teoría de campos potenciales, surge de la analogía de la teoría electroestática la cual estudia la interacción entre las cargas eléctricas cuyos efectos aparecen en forma de atracciones y repulsiones entre los cuerpos que las poseen, dependiendo si son cargas de signos contrarios o iguales. En el caso de la planeación de la trayectoria de un robot móvil, se toma en cuenta la existencia de una fuerza de atracción sobre la plataforma móvil si se trata de una meta, y una de repulsión en caso de ser un obstáculo.

Implementación

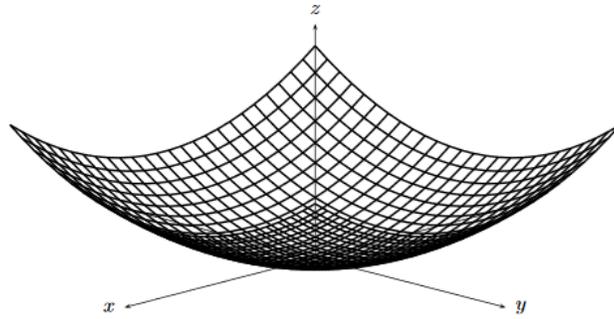


Ilustración 44 Campo potencial parabólico centrada en el origen asociado al objetivo

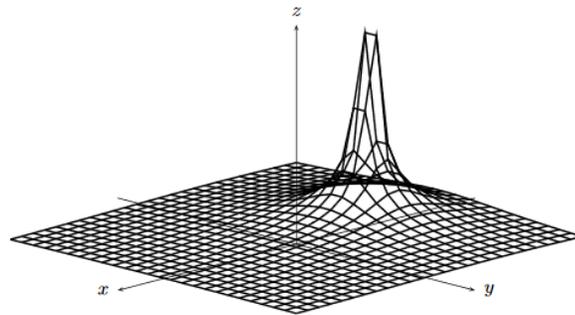


Ilustración 45 Campo potencial parabólico de un obstáculo

Cabe señalar que el tipo de planificación de trayectoria por campos potenciales artificiales es del tipo reactiva, es decir, que es capaz actuar de manera autónoma redireccionando su avance si la ubicación de metas y obstáculos cambian en el ambiente inteligente; ya que la transmisión de información se hace en tiempo real a partir de un campo de fuerzas.

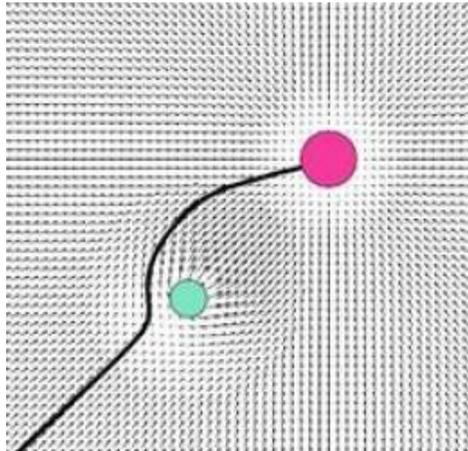


Ilustración 46 Ejemplo de un campo potencial artificial representado por un obstáculo y la meta

Para aplicar este método de planeación de trayectorias, el trabajo del Dr. Víctor González Villela titulado *Evadiendo obstáculos con robots móviles*, explica que estos métodos de campos potenciales constituidos por las fuerzas vectoriales de repulsión y atracción no son reales y no existen en el medio ambiente real del robot, éstas son añadidas a los modelos

Implementación

matemáticos del robot para generar los comandos de aceleración en las ruedas del robot, son función de la distancia a la que se encuentra el robot de la meta (para la fuerza atractiva) y de la distancia a la que se encuentra de los obstáculos (fuerza repulsiva), es por eso que estos métodos se han hecho llamar Campos Potenciales (Véase [24]); el cual posee la ventaja de ser aplicado conociendo únicamente el modelo cinemático del robot en cuestión.

Una vez establecida la relación de fiducial (robot, objetivo y elemento pasivo), el siguiente paso es alimentar con esta información al bloque de *Campos potenciales*, es decir, con la posición actual y deseada del móvil.

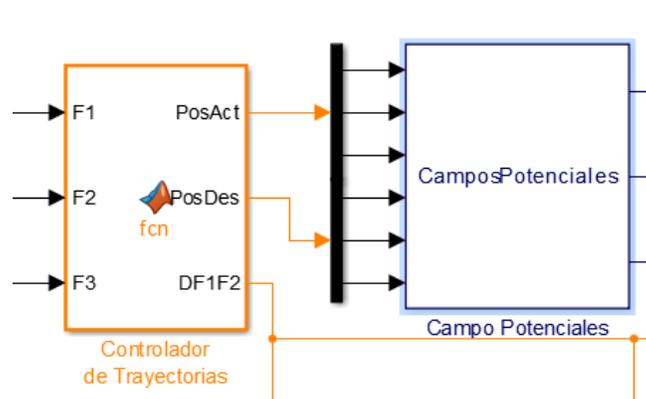


Ilustración 47 Bloque de Campos Potenciales

Esta imagen muestra los parámetros que son tomados en cuenta en el bloque de campos potenciales:

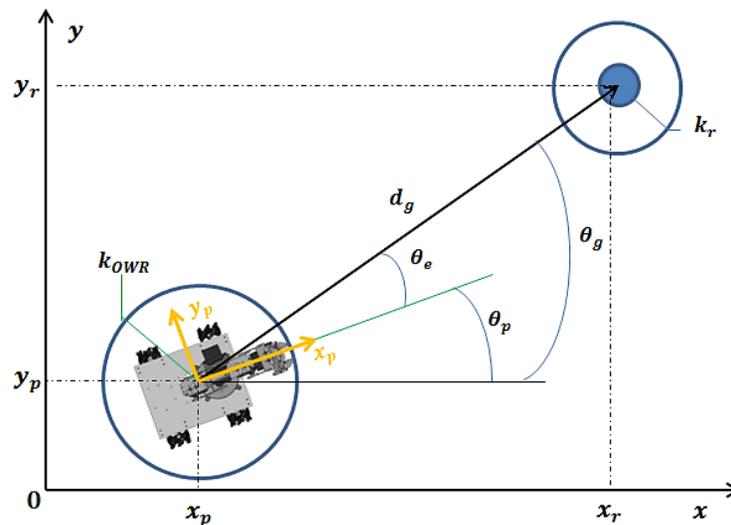


Ilustración 48 Parámetros del método de campos potenciales

d_g : Vector de atracción de la plataforma móvil hacia el objetivo, el cual se calcula a partir de los puntos iniciales y finales $P_i = (x_i, y_i)$, $P_f = (x_f, y_f)$

Implementación

$$d_g = \sqrt{(x_f - x_i)^2 + (y_f - y_i)^2}$$

Ecuación 50

θ_g : Ángulo del vector de atracción que tiene respecto de eje X del marco coordenado inercial, determinado por la siguiente ecuación.

$$\theta_g = \tan^{-1} \left(\frac{y_f - y_i}{x_f - x_i} \right)$$

Ecuación 51

θ_p : Ángulo de orientación del móvil respecto de la horizontal

θ_e : Ángulo de error que hay entre θ_g y θ_p determinado por:

$$\theta_e = \theta_g - \theta_p$$

Ecuación 52

$v_{m\acute{a}x}$: Velocidad máxima con la que puede desplazarse; delimitada en un porcentaje conveniente para una mejor ejecución de la tarea que se detalla en el código generado en Matlab

k_r : Radio del área de evasión

k_{OWR} : Área de seguridad del robot

De lo anterior se deduce que la magnitud velocidad de avance del móvil es proporcional a la distancia que separa al robot de la meta; lo que se traduce en que la velocidad de avance máxima permitida será cuando éste lejos de la meta e irá disminuyendo conforme se vaya acercando al k_r o radio del área de evasión.

Por otro lado, para dotar al robot móvil de la orientación determinada por el ángulo θ_g del vector de atracción se requiere de una velocidad angular ω que pueda orientarlo haciendo que el ángulo de error tienda a cero grados ($\theta_e \rightarrow 0^\circ$). Dicha velocidad angular se determina mediante la ecuación:

$$\omega = \omega_{m\acute{a}x} * \sin(\theta_e)$$

Ecuación 53

Ecuaciones auxiliares empleadas:

$$\omega = \omega_{m\acute{a}x} * \sin(-\theta_p)$$

Ecuación 54

$$\omega = \omega_{m\acute{a}x} * \sin\left(\theta_e - \frac{\pi}{2}\right)$$

Ecuación 55

Implementación

Donde $\omega_{m\acute{a}x}$ es la velocidad angular maxima que puede ser enviada por Matlab, la cual se alcanza cuando θ_e es $\pm 90^\circ$ y la mınima cuando ya ha alcanzado los cero grados. Algunas de las ecuaciones

Las siguientes ecuaciones corresponden a las velocidades lineales de P en el eje x y y respectivamente.

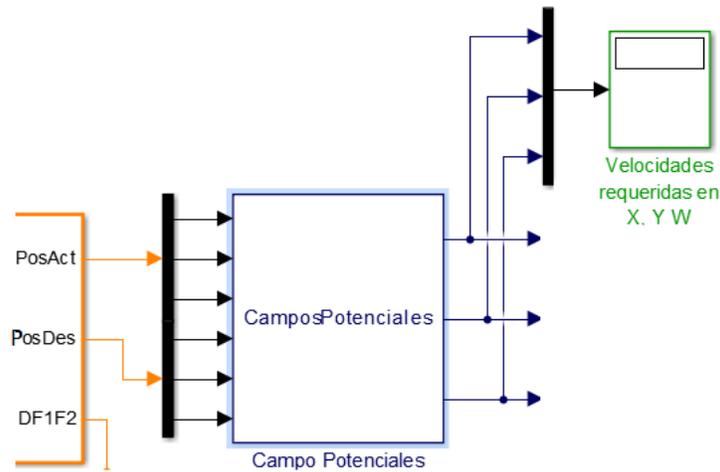
$$v_{xp} = \begin{cases} v_{xmax} * \cos(\theta_e) & \text{si } |d_g| > k_p + K_{OWR} \\ \frac{v_{xmax}}{k_r} * d_g \cos(\theta_e) & \text{si } |d_g| < k_p + K_{OWR} \\ 0 & \text{si } |d_g| \leq K_{OWR} \end{cases}$$

Ecuaci3n 56 Velocidad sobre el eje X

$$v_{yp} = \begin{cases} v_{ymax} * \sin(\theta_e) & \text{si } |d_g| > k_p + K_{OWR} \\ \frac{v_{ymax}}{k_r} * d_g \sin(\theta_e) & \text{si } |d_g| < k_p + K_{OWR} \\ 0 & \text{si } |d_g| \leq K_{OWR} \end{cases}$$

Ecuaci3n 57 Velocidad sobre el eje Y

Finalmente obteniendo como salida de este bloque las velocidades lineales y angulares (v_{xp}, v_{yp}, ω)



Ilustraci3n 49 Salida de bloque de Campos Potenciales

Cinemática de plataforma móvil

El bloque que contiene la cinemática de la plataforma móvil contiene el modelo cinemático final obtenido anteriormente, en el cual finalmente se obtienen las velocidades angulares [rad/s] que serán enviadas a cada rueda en función de las velocidades v_{xp}, v_{yp}, ω

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{r_R} & -\frac{1}{r_R} & -\frac{b+d}{r_R} \\ \frac{1}{r_R} & \frac{1}{r_R} & -\frac{b+d}{r_R} \\ \frac{1}{r_R} & -\frac{1}{r_R} & \frac{b+d}{r_R} \\ \frac{1}{r_R} & \frac{1}{r_R} & \frac{b+d}{r_R} \end{bmatrix} * \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}$$

Ilustración 50 Modelo que permite asignar las velocidades angulares a cada rueda

Aquí intervienen datos como el radio de las ruedas, la coordenada en x_1 de la posición de la rueda (d) y coordenada en y_1 de la posición de la rueda (b)

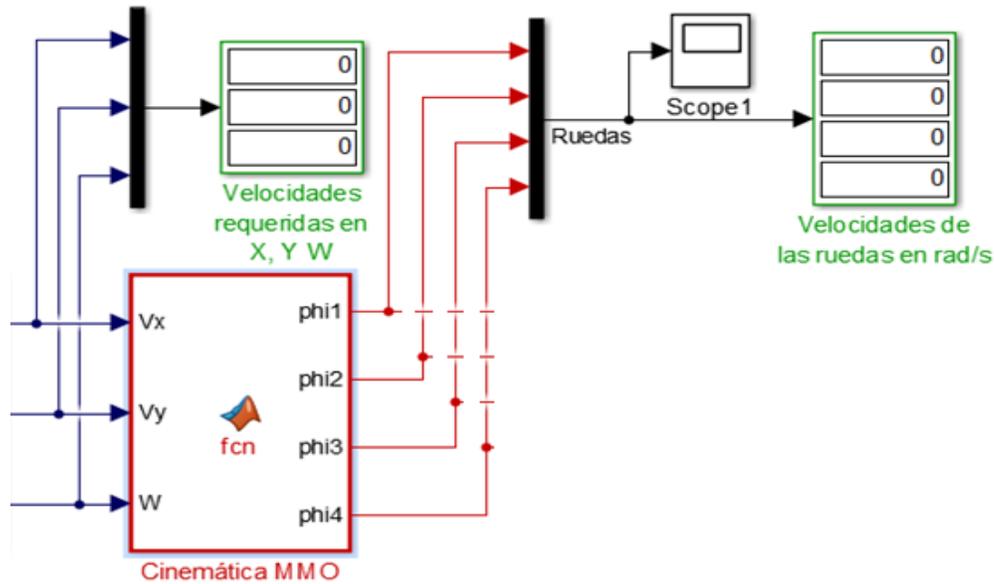


Ilustración 51 Bloque de del modelo cinemático

Compensaciones

Bloque en el cual se declaran algunas condicionales para asegurar que la plataforma móvil se detenga cuando ya ha llegado al radio del área de evasión evitando de esta manera que la invada. En resumidas palabras dice que si la distancia actual entre el móvil y la meta (DF1F2) es menor o igual que la distancia de acople, entonces las velocidades angulares de las cuatro ruedas son iguales a cero en caso contrario, serán iguales a las calculadas en el bloque de *Cinemática MMO*.

Implementación

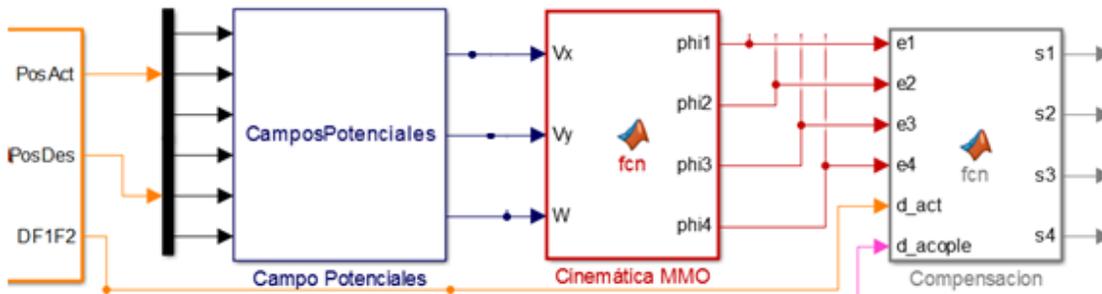


Ilustración 52 Bloque de Compensaciones

Procesamiento de Datos de Salida

Únicamente se realiza una conversión de los datos enviados por el bloque anterior a valores enteros sin signo, es decir, los valores de las velocidades angulares que son enviados a la tarjeta MD25 se encuentren en enteros entre 0 a 255; siendo estos límites las velocidades máximas en sentidos de giro positivo o negativo, asimismo en este bloque se hace el corrimiento del 0 al 128, tal y como lo ilustra la siguiente imagen.

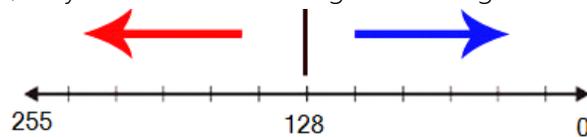


Ilustración 53 Ajuste de corrimiento del cero, datos positivos y enteros

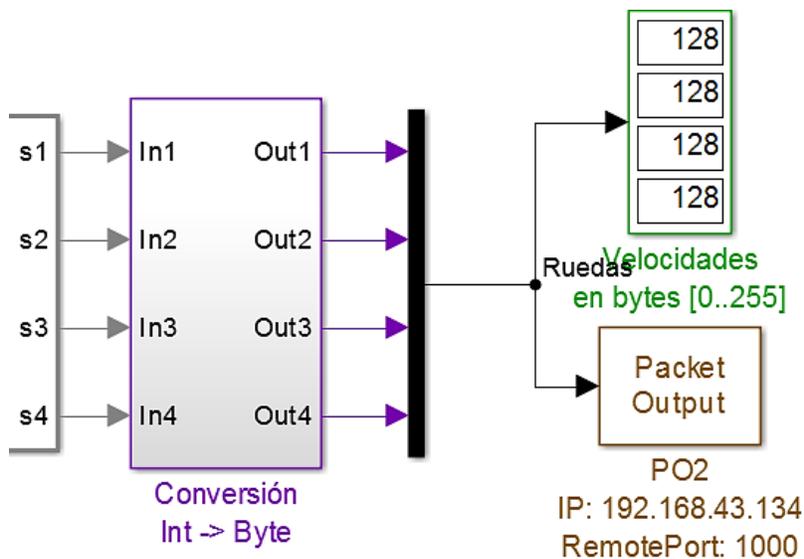


Ilustración 54 Bloque de conversión

Implementación

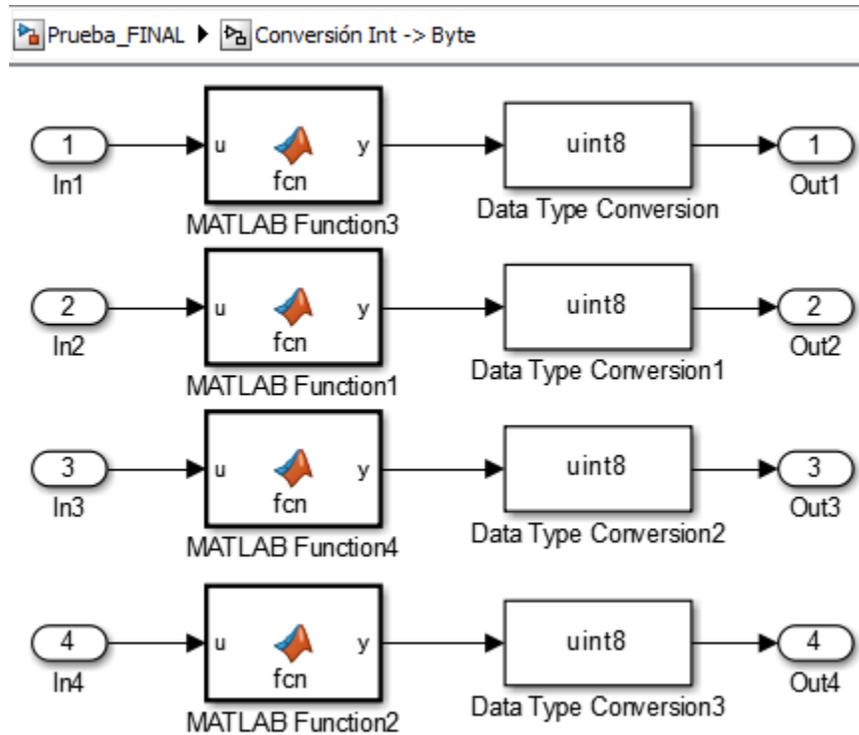


Ilustración 55 Conversión a enteros sin signo de un byte

Envío de Datos

De manera similar al bloque packet input para la recepción de datos, el bloque packet output realiza el envío de datos de las velocidades angulares en este caso; ya que en el segundo bloque packet output empleado en el programa de Matlab Simulink son enviados los valores de los grados correspondientes al manipulador serie-paralelo.

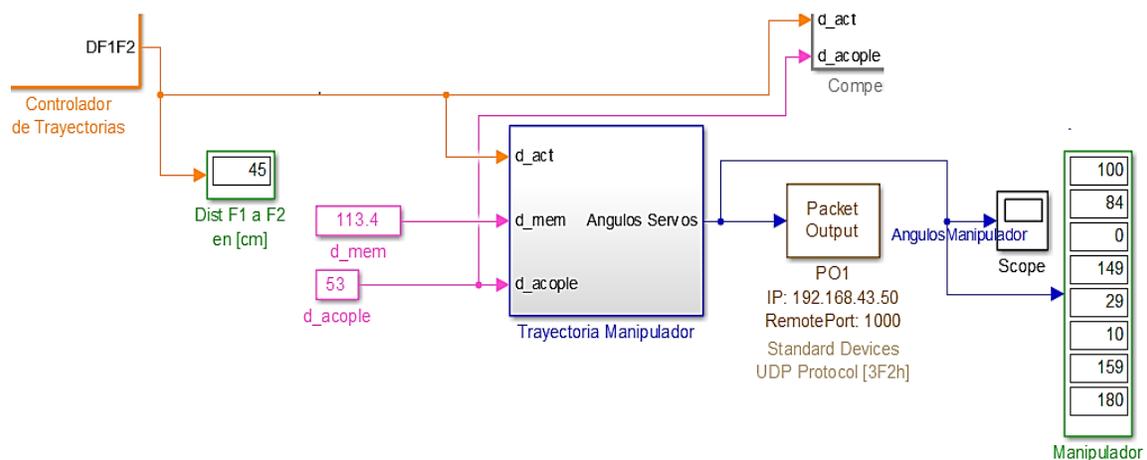
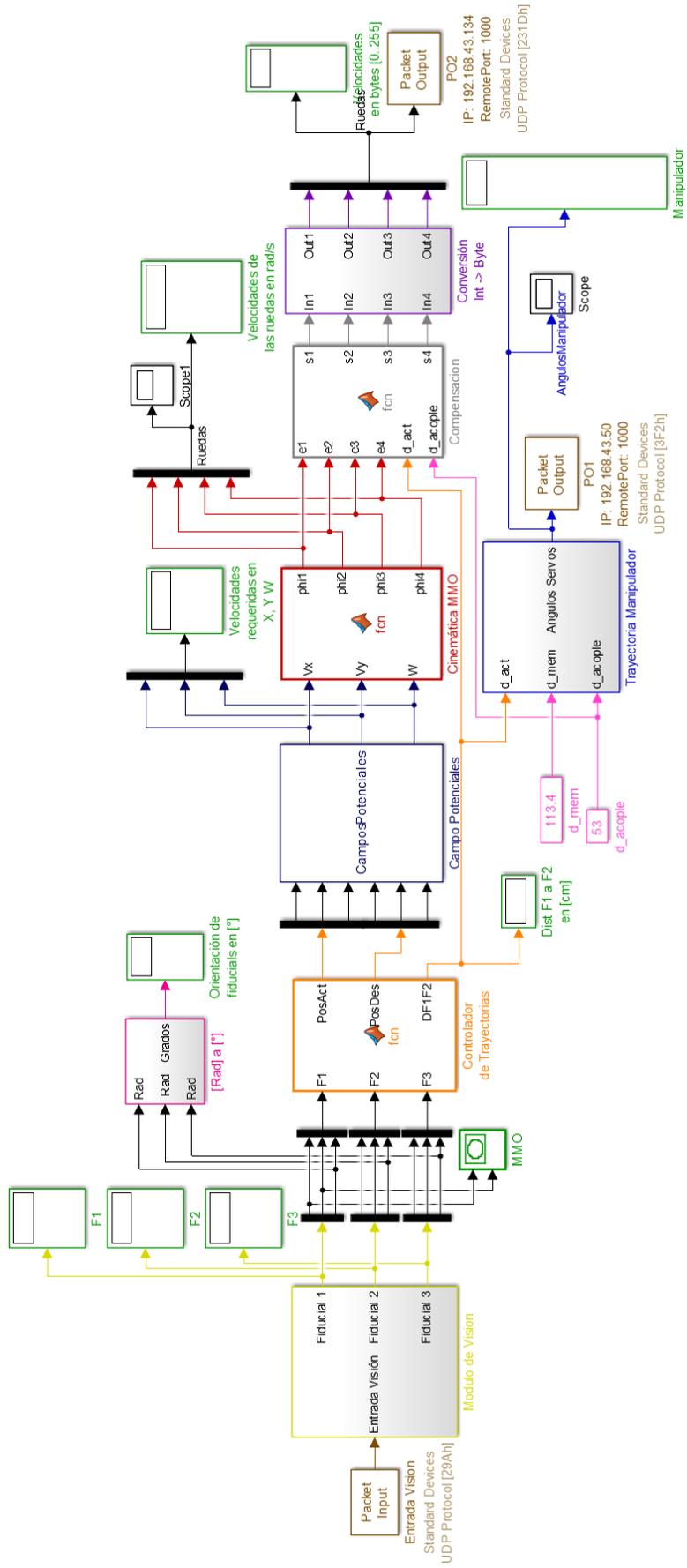


Ilustración 56 Envío de datos de manipulador serie-paralelo

Implementación



Conexión wifi

Para completar la comunicación entre los sistemas de procesamiento y ejecución de movimientos, se implementó el uso de wifi. Mediante direcciones IP y puertos, además del ya mencionado protocolo UDP, se logró construir un canal de comunicación que consistía en un módem inalámbrico al cual se conectó la computadora (que ejecutaba el programa principal de Matlab), y los microcontroladores Arduino, con ello se vinculó la dirección IP de cada Shield Wifi, para poder enviar los datos de forma independiente y simultánea. En el siguiente diagrama se muestra la conexión realizada.

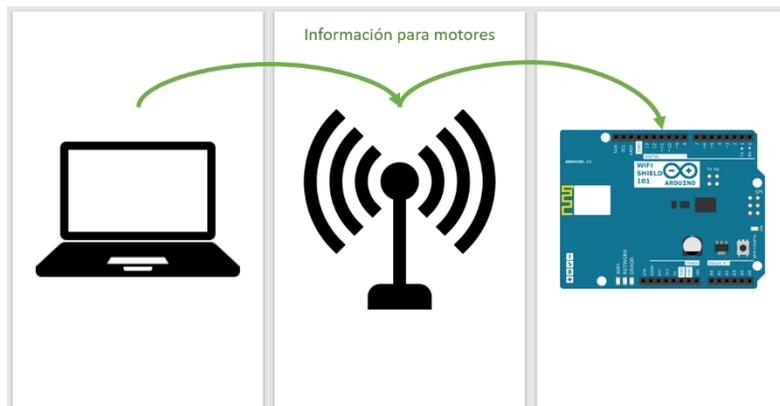


Ilustración 58 Comunicación inalámbrica empleada

Cabe mencionar que esta forma de comunicación, puede ser aplicada de varias formas, dado que la antena wifi en la Ilustración 58, representa una gran cantidad de dispositivos, como módems, o en nuestro caso, un teléfono inteligente; se aplicó lo anterior debido a que se buscaba no tener interferencia, retardo o algún otro inconveniente en la comunicación. Dado que, a nuestro punto de acceso, solo se conectan tres dispositivos, el flujo de información, se transmitía sin problema alguno.

Función del shield wifi de Arduino

Para recibir la información de la computadora se implementó el Shield Wifi de Arduino debido a su facilidad de comunicación, implementación y uso. Consiste en un módulo de comunicación diseñado para los microcontroladores Arduino, el cual se monta encima de estos para desempeñar su tarea. Su programación se realiza en el mismo IDE de Arduino, no sin antes implementar las correspondientes librerías y ajustes necesarios.

La función desempeñada por este módulo consiste en lo siguiente:

- Enlazar el microcontrolador al punto de conexión Wifi.

Implementación

- Recibir los datos de los motores (Servomotores y DC) en forma de buffer.
- Enviar a Arduino dicha información para su posterior ejecución.

Para más información se recomienda la lectura de la sección Descripción del hardware o la sección de Apéndices para revisión del código implementado.

Función de Arduino

Arduino es un microcontrolador de fácil programación y uso, que por dichas características, fue implementado en el presente estudio. Cuenta con un IDE (Entorno de Desarrollo Integrado por sus siglas en inglés) que permite la administración y programación del microcontrolador. El mismo, cuenta con diferentes líneas de entrada y salidas, analógicas y digitales, que permiten el control y la comunicación con diferentes tipos de sistemas electrónicos, electromecánicos y de comunicación.



```
sketch_jan22a Arduino 1.6.9
Archivo Editar Programa Herramientas Ayuda
sketch_jan22a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

Ilustración 59 Ejemplo de IDE de Arduino v1.6.9

El microcontrolador implementado, es un Arduino Mega 2560, seleccionado por la velocidad de procesamiento, la cantidad de entradas y salidas disponibles entre otras cosas.

Nuestro microcontrolador desempeña las tareas que a continuación se enlistan:

- Establecer comunicación para recibir la información de los motores como se muestra en la a Ilustración 58, con la computadora que ejecuta el programa principal, mediante el punto de acceso y el Shield wifi.
- Leer el paquete de datos mediante un buffer recibido del Shield Wifi.
- Decodificar y procesar dicha información para su posterior uso.
- Con base en los datos recibidos, ejecutar los movimientos para cada motor en particular.

Como se puede apreciar, nuestro sistema contiene y realiza todos los cálculos en el programa de Matlab, destinando un procesador más poderoso, como lo es el de una computadora personal, a realizar una gran cantidad de cálculos simultáneos de cada uno de los 13 diferentes motores, en lugar de destinar el cómputo a los microcontroladores; es decir, nuestros

Implementación

microcontroladores no procesan las matemáticas que rigen el comportamiento de nuestro sistema, sino que reciben y ejecutan la información que les es enviada.

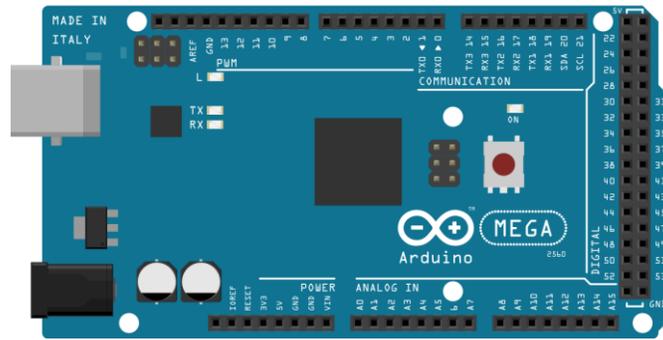


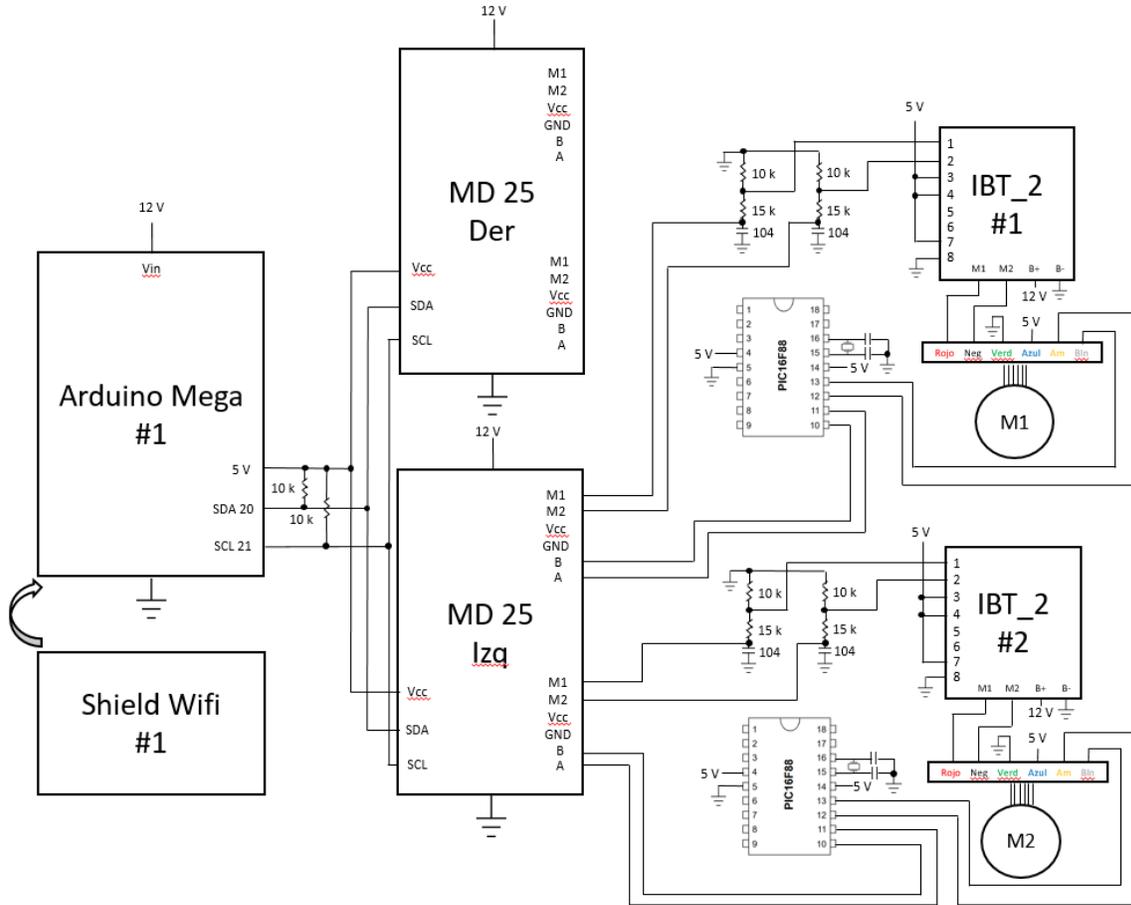
Ilustración 60 Ilustración de Arduino utilizado, imagen creada con Fritzing²

Descripción del hardware

En esta sección se describen a detalle los elementos electrónicos y mecánicos implementados, se relatan las funciones que cada uno de estos desempeñan, y el impacto que tienen en el modelo funcional final. Para ello se anexan los siguientes esquemas generales de conexiones donde se incluyen los elementos implementados, comenzando con la plataforma móvil omnidireccional para finalizar con el manipulador híbrido:

² Según su página de internet en fritzing.org: "Fritzing is an open-source hardware initiative that makes electronics accessible as a creative material for anyone."

Implementación



Implementación

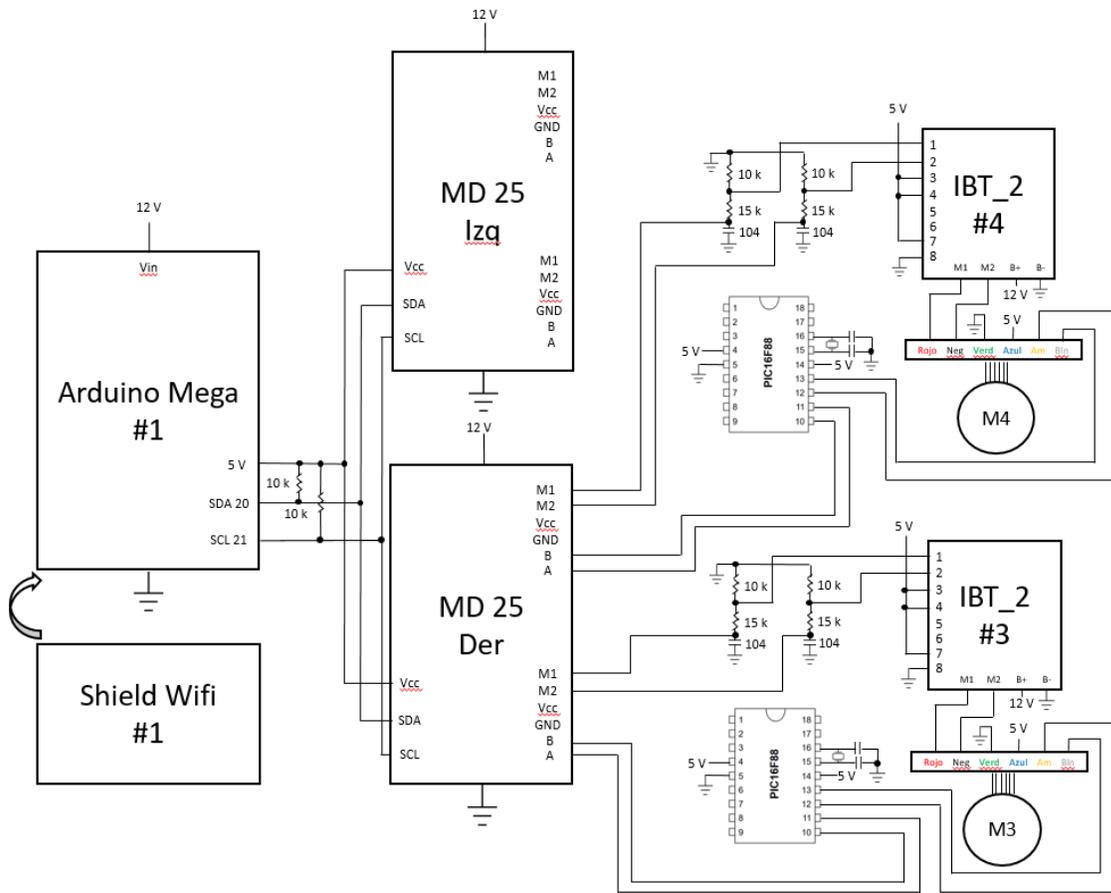


Ilustración 61 Diagrama de conexiones de la plataforma móvil omnidireccional

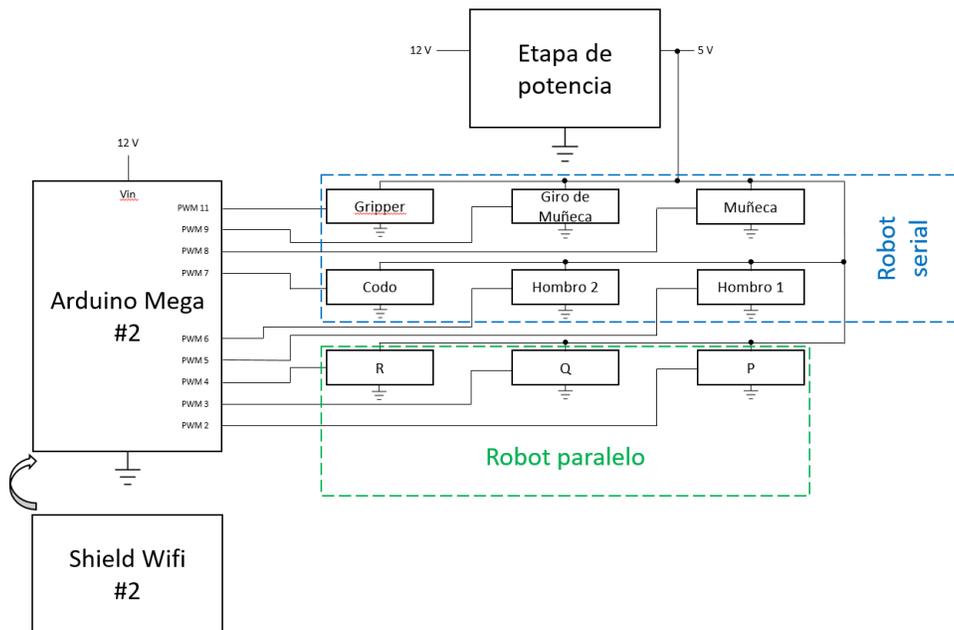


Ilustración 62 Diagrama de conexiones del manipulador serie paralelo

Implementación

Arduino Mega 2560

Como se ha mencionado anteriormente se decidió utilizar este microcontrolador por sus características que presenta, la cantidad de entradas y salidas que maneja y la facilidad de uso. Contiene un procesador ATmega2560 de la marca Atmel, que cuenta con diferentes características como convertidores analógico-digital, lectura y escritura de señales digitales y analógicas, capacidad de comunicación I2C y serial, generación de señales PWM, entre otras cosas. Algunas otras características técnicas se muestran a continuación:

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Ilustración 63 Características técnicas de Arduino Mega 2560³

A continuación se muestra una ilustración del microcontrolador utilizado en el presente proyecto:



Ilustración 64 Ejemplo de Arduino Mega 2560 utilizado en este trabajo⁴

³ Fuente: Arduino.cc

⁴ Para más información acerca de este microcontrolador se sugiere visitar <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>

Implementación

Cabe recalcar que la función de este microcontrolador utilizado es recibir la información de los motores DC y servos, provenientes del programa de Matlab, mediante el Shield Wifi, y enviar a dichos motores las velocidades y posiciones requeridas. Por tanto, no ejecuta ningún cálculo matemático complejo, como la cinemática o algún otro, como se indica en la sección Función de Arduino.

Shield Wifi de Arduino

El módulo implementado para lograr la conexión con la computadora y los microcontroladores fue el Shield Wifi de Arduino, el cual permite una conexión sencilla y eficiente. Dado que es un dispositivo específicamente diseñado para los microcontroladores Arduino. Como características se tiene que opera con 5 V, su conexión se establece vía red 802.11 b/g, soporta los tipos de encriptación WEP y WPA2, cuenta con una ranura para tarjeta de almacenamiento de datos micro SD y cuenta con led indicadores de estados, L9 en amarillo, LINK en verde que indica que la conexión se ha establecido, ERROR, en rojo que indica que hubo un error en la comunicación a la red, DATA en azul, que mientras se encienda indica que hay un flujo de información.

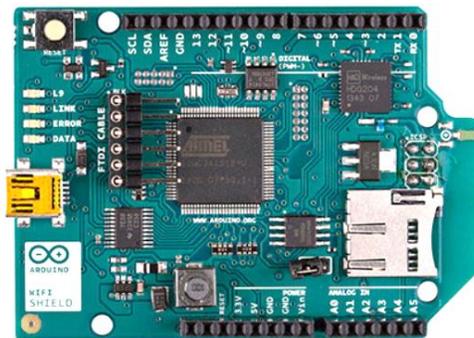


Ilustración 65 Shield Wifi de Arduino

Fuente de alimentación

La fuente de alimentación implementada, se trata de una fuente de computadora, dado que se contaba con una gran cantidad de motores, los cuales demandaban una considerable corriente, por ejemplo, como se puede observar en la sección Motores 37D 50:1, se especifica que dichos motores, demandan cada uno 4.5A en paro, lo cual suma 18A para comenzar a mover el móvil. Por otro lado, los servomotores utilizados, tienen un consumo de corriente de 2.3A mientras se encuentren bloqueados.

Por tales motivos, se decidió emplear en lugar de baterías recargables, una fuente de alimentación.

Tarjeta MD25

La tarjeta MD25, es una tarjeta que controla la velocidad de giro de los motores de DC, mediante un PID (conocido también como, control proporcional integrativo derivativo), para mantener la velocidad deseada, cuenta con dos tipos de comunicación, I2C empleado en este estudio y comunicación serial a diferentes velocidades de transmisión. Entre sus más destacables características las tarjetas MD25 cuentan con:

- Lectura de encoders para determinar distancia y dirección recorrida.
- Control independiente o combinado de los dos motores que puede manejar.
- Lectura de corriente que fluye por los motores.
- Regulación a 5V para los componentes electrónicos internos.
- Control PID para la regulación de la velocidad.

Se muestra a continuación una tarjeta MD25, con sus respectivas conexiones y puertos:

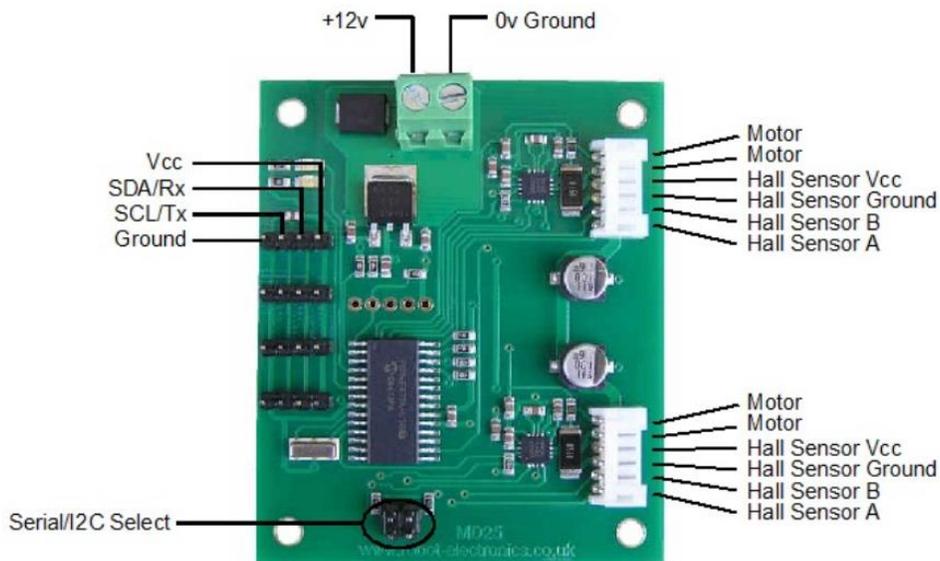


Ilustración 66 Tarjeta MD25 utilizada para el control de dos motores DC ⁵

Dado que las tarjetas MD25, está diseñadas para los motores EMG 30, pues dichos motores alcanzan a consumir 2.8A bloqueados y sus encoders de cuadratura, generan 360 pulsos por revolución, se han implementado algunas modificaciones que nos permitieron no solo usar nuestros motores 37D, sino utilizar cualquier otro motor DC con Encoder que se desee.

Para utilizar dichas tarjetas, como se ha mencionado en comunicación I2C, se debe de iniciar un ciclo de transmisión el cual es como sigue: iniciar comunicación I2C con la dirección de la

⁵ Imagen obtenida de la página oficial de la empresa diseñadora de estas tarjetas <http://www.robot-electronics.co.uk/htm/md25tech.htm>

respectiva tarjeta⁶, acceder al registro, enviar la velocidad deseada, cerrar la comunicación I2C y comenzar nuevamente el ciclo de comunicación.

Cabe mencionar que la tarjeta de control, cuenta con distintos modos de operación, los cuales en la comunicación I2C, definen el tipo de dato que debe ser enviado a las MD25:

- 0: Este modo indica que las velocidades de giro deben ser un byte sin signo desde 0 reversa total, 128 paro hasta 255 avance total.
- 1: Este modo modifica las velocidades de forma que el byte de información contiene un signo, -128 reversa total, 0 paro 127 avance total.
- 2: Con esta configuración, el registro Speed1 controla ambos motores con la velocidad que este tenga, y el registro Speed2, indica el sentido de giro, con el arreglo del byte de velocidad como se tiene en el modo 0.
- 3: Se tiene un control similar al modo 2, con la única diferencia que el byte de velocidades, contiene un signo, es decir, recorre desde -128 hasta 127.

La forma implementada, dada la configuración de nuestro sistema es utilizar los motores de forma independiente y recorriendo el byte de velocidades desde 0 hasta 255.

Es importante mencionar que estas tarjetas son muy delicadas en su uso, por lo que se recomienda ampliamente el uso de resistencias de 10 kohms, como se muestra en la Ilustración 61 Diagrama de conexiones de la plataforma móvil omnidireccional, entre las MD25 y el microcontrolador que envía las velocidades.

Motores 37D 50:1 con encoder

Los motores utilizados para la plataforma móvil omnidireccional, son los motores 37D 50:1, de la marca Pololu, estos motores entre sus características tienen:

- Voltaje de operación de 6 a 12V ⁷.
- Corriente en bloqueo de 5A.
- Corriente sin carga a 12V, 300mA.
- Velocidad sin carga a 12V, 200rpm.
- Par máximo 12 kg cm.
- Relación de engranes 50:1.
- Masa 220g.
- Diámetro de la flecha 6mm.

⁶ Dicha dirección para nuestro sistema fue 0x58 y 0x59 para las MD25 (aunque pueden ser modificadas), como se aprecia en la línea 12 y 16 del código presentado en la sección Código de la Plataforma Móvil Omnidireccional.

⁷ El voltaje utilizado para alimentar a estos motores fue de 12V.

Implementación

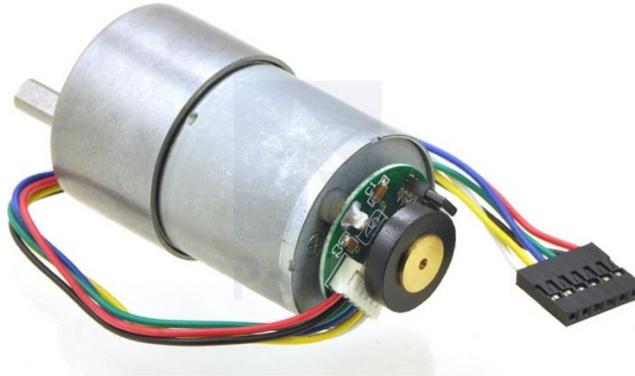


Ilustración 67 Motores 37D 50:1 De la marca Pololu ⁸

Una característica fundamental que tienen estos motores es su encoder, el cual genera señales de la misma amplitud con la que es alimentado el sensor, a diferencia de otros que generan señales TTL con una alimentación mayor a 5V.

Otra característica muy importante es que por cada revolución, se generan 3200 pulsos, por tanto, para poder volver compatible, este motor con la tarjeta MD25, se debe hacer una reducción en su frecuencia. Haciendo los cálculos, entre los motores EMG30 para los que está diseñada la MD25 y estos, existe una relación de pulsos de 8.888. Tal contratiempo se solucionó con el uso del PIC16F88, cuya función es la de disminuir la frecuencia con la que estos pulsos son generados, al implementar un contador que reduce la frecuencia a razón de 9, es decir, cada 9 pulsos entrantes, hay uno de salida⁹. Es por esta modificación que el control PID de la MD25, puede ser implementado en cualquier motor con encoder.

Etapa de reducción de amplitud de PWM

Dado que se tiene la capacidad de implementar a la tarjeta MD25, motores que no está diseñados para esta, se hace uso de una etapa de potencia externa, en este caso el módulo IBT 2, el cual requiere una señal PWM de 5V de amplitud. La MD25 proporciona una amplitud para su PWM de 12V, por este motivo se emplea un circuito de divisor de voltaje como se presenta a continuación:

⁸ Las conexiones mostradas en la Ilustración 67 Motores 37D 50:1 De la marca Pololu, son rojo y negro, para la alimentación del motor, verde, tierra del encoder, azul alimentación del encoder, amarillo señal A del encoder, blanco, señal B del encoder.

⁹ Con más detalle se explica el funcionamiento de la reducción de la frecuencia en la sección Implementación del PIC16F88

Implementación

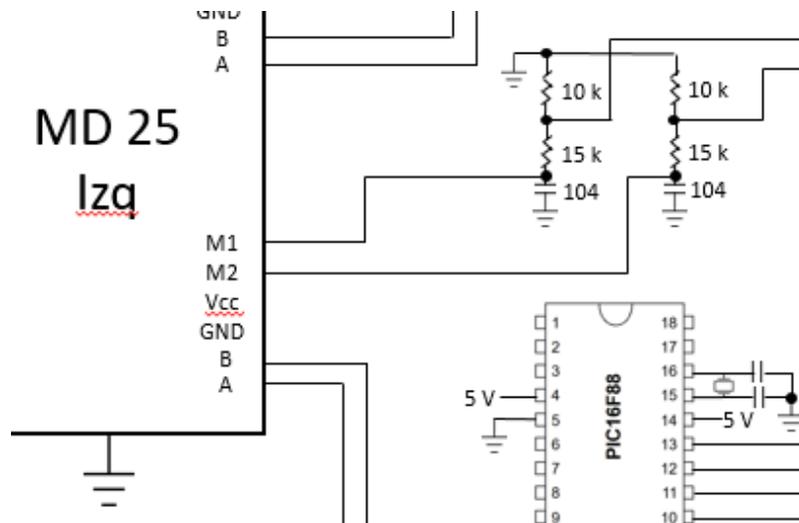


Ilustración 68 Etapa de reducción de PWM

Como se muestra en la figura, la MD25 genera las señales de 12V de amplitud, entran a la reducción y salen hacia la etapa de potencia IBT 2. Se añade un capacitor extra para disminuir el ruido en la salida, cabe mencionar que con gran éxito.

Según la aplicación de las leyes de Kirchoff en relación a la tensión, se tiene:

$$V_{out} = V_{in} \frac{R_{10}}{R_{10} + R_{15}} = 12 [V] \frac{10 [k\Omega]}{10 [k\Omega] + 15 [k\Omega]} = 4.8 [V]$$

Ecuación 58 Modelo matemático de la división de voltaje

Aunque no se obtiene un voltaje exacto de reducción, es suficiente para el funcionamiento de la etapa de potencia IBT 2, con esto se logra una compatibilidad en los sistemas que permite la implementación de muchos tipos de motores.

Etapa de potencia IBT 2

Dado que la sola tarjeta MD25, no suministra la energía necesaria para mover los motores, se ha implementado esta etapa de potencia.

Funciona mediante transistores de efecto de campo metal-óxido-semiconductor o por sus siglas en inglés MOSFET, que forman un puente H.

Soportan de 12 a 36V y 43A con picos de 60A, son alimentados con un voltaje de entrada en sus terminales B+ y B-, y generan un PWM en las salidas M1 y M2. Necesitan una entrada de PWM de 5V de amplitud y cuentan con la capacidad de leer la corriente en los motores. Para el procesamiento de datos necesita una alimentación extra de 5V, con lo que se energiza el microcontrolador interno.

Implementación

Con lo robusto de esta potencia, representa una excelente opción para suministrar de energía a una buena cantidad de motores.

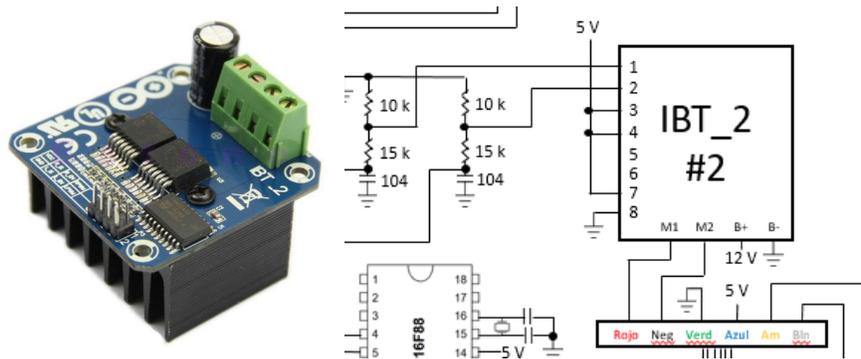


Ilustración 69 Módulo de potencia y diagrama de conexión

Servomotores

Se utilizaron los servomotores para dotar de movimiento al manipulador paralelo-serial. Para ello, fueron necesarios 9 servomotores, considerando 3 servomotores para el robot paralelo, que fueron llamados P, Q y R, y para el robot serial se utilizaron 6, dos para el hombro, uno para el codo, muñeca, el giro de la muñeca y para abrir y cerrar el efector final.



Ilustración 70 Servomotor 1501 MG implementado

Los motores P, Q, R y 2 para los hombros tienen las siguientes características:

- Son motores de alto par, 17 kg cm.
- Tienen engranes de bronce.
- Son alimentados a 5V.
- En bloqueo consumen 2.3A y en movimiento con carga 400mA.

Para el resto de las juntas actuadas, se utilizaron servomotores de engranes plásticos pues se considera tener menor carga en las juntas más cercanas al efector final, como la muñeca o su giro¹⁰.

¹⁰ Con dicho giro de la muñeca, se hace referencia al sistema D, del manipulador serial.

Implementación

Etapa de potencia de los servomotores

Dado que se tiene un consumo considerable de energía para la etapa del manipulador, considerando 5 servomotores de 2.3A se acumulan 11.5A solamente para estos, considerando el resto de los motores, la corriente necesaria, ronda entre 13 y 15A, es por ello que se ha implementado una etapa de potencia para el suministro energético del manipulador:

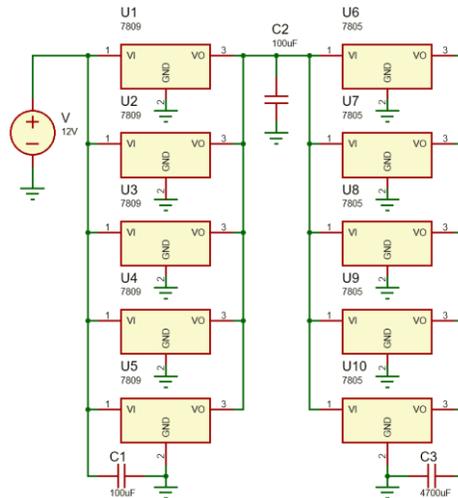


Ilustración 71 Etapa de potencia para los servomotores

Esta etapa de potencia consta de una etapa de regulación de 12 a 9V, con los reguladores LM7809, posteriormente la salida de dichos 7809, resulta la entrada de los reguladores LM7805 a 5V, se han añadido algunos capacitores para la reducción del ruido que pueda generarse. Este circuito, proporciona la potencia necesaria para mover (dependiendo de nuestra rutina) 8 o 9 servomotores simultáneamente¹¹.

Implementación del PIC16F88

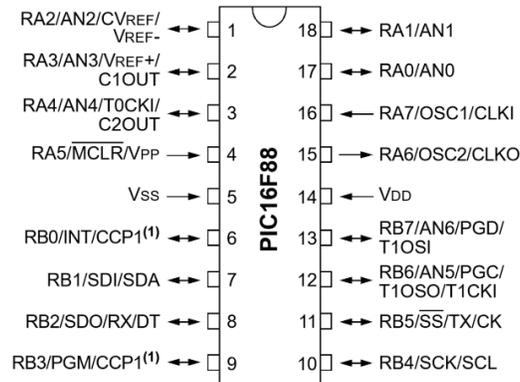
Características del PIC16F88

El PIC16F88, es un microcontrolador de la marca Microchip de uso general que presenta entre sus características las siguientes:

- 3 módulos timer.
- 2 Comparadores.
- Convertidores analógico digitales.
- Entradas analógicas y digitales.
- Módulos PWM.
- Capacidad de usar osciladores desde 31kHz hasta 20MHz.
- Manejo de interrupciones por cambios de estado, por flanco y otras.

¹¹ Esta etapa de potencia es la mostrada en la Ilustración 62 Diagrama de conexiones del manipulador serie paralelo.

Implementación



Ecuación 59 Diagrama de pines del PIC16F88

Dado el hecho de ser un microcontrolador, confiable y simple de programar, se eligió para la tarea de reducir la frecuencia de los pulsos del encoder. Con ello, se ha realizado el código en lenguaje C, con el uso del IDE MPLab X, proporcionado por Microchip.

Funcionamiento del PIC16F88

Como se ha mencionado, gracias a la modificación que utilizamos, la utilización de la tarjeta de control MD25, puede aplicarse a cualquier tipo de encoder y motor, simplemente volviendo compatibles los encoders que recibe la tarjeta, esto es generando 360 pulsos por revolución. Esto fue logrado gracias a un PIC16F88 (uno para cada motor DC), el cual recibía la información que generaban los encoders de los motores y los traducía a 360 pulsos aproximadamente. Utilizando un contador de 36 estados, se logra clasificar cada opción con base en la información del estado actual y el anterior para generar una salida, utilizando el siguiente diagrama:

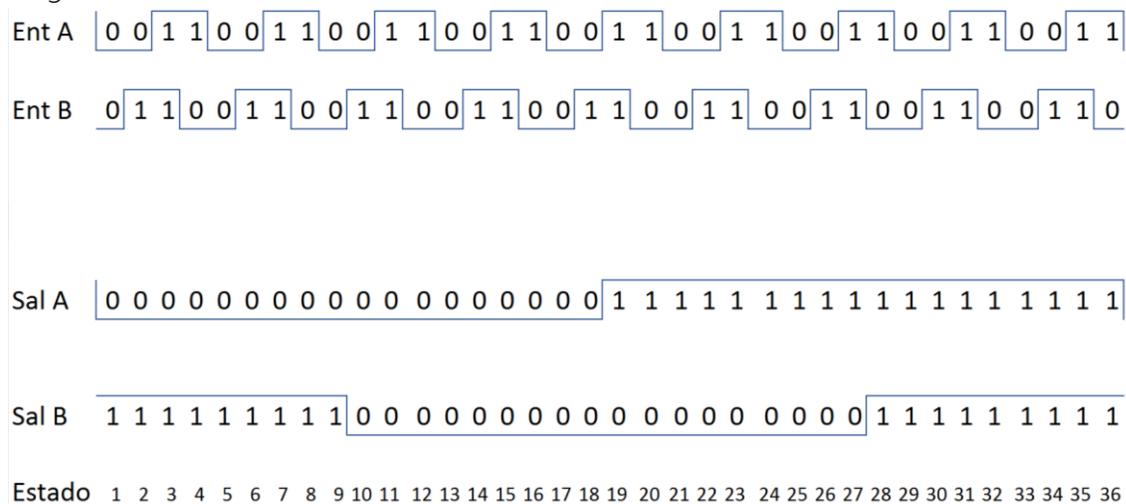


Ilustración 72 Información de entrada y salida del PIC16F88 para la reducción de frecuencia.

Implementación

Se han contabilizado 36 estados, pues 4 estados son necesarios para describir a un encoder, y 9 es el factor de reducción como se muestra a continuación:

$$\frac{3200}{360} = \frac{80}{9} = 8.888 \approx 9$$

Ecuación 60 Factor de reducción en los pulsos por revolución de los encoders

El funcionamiento del reductor de frecuencia es como sigue: el código está diseñado para leer las señales de los sensores A y B, y con ello, puesto que es un número binario de dos dígitos, puede caer en 1 de 4 posibilidades, escogiendo arbitrariamente se asigna a los estados 1, 2, 3 o 4. Dichos estados están asignados a ciertas salidas, como se muestra en la Ilustración 72.

En este punto el PIC entra en un bucle infinito esperando por las interrupciones por cambio de estado que generan las entradas, y con la información recibida, se decide si el estado se incrementa o disminuye, siempre teniendo en cuenta que el estado puede ser un entero entre 1 y 36. Con este incremento y decremento de los estados, se recorre en todos y cada uno, generando así la salida que corresponde a cada estado.

Por ejemplo, se inicia el programa, y se leen los puertos de las señales A y B de los encoders y las señales A y B son ambas 0, se decide entonces que al estado en que se debe "ir" es el estado 1, al cual corresponden las salidas de A y B respectivamente de 0 y 1, entonces se entra en un bucle infinito que espera otra interrupción. Sea esa interrupción un cambio en A, es decir la actual señal del encoder es 10, esto ocasiona que se recorra al estado 36, cuyas salidas son 11 respectivamente, el siguiente cambio recorrerá el contador al estado 35 o 1, los cuales tienen sus salidas determinadas, y con ello se ejecuta una reducción en la frecuencia.

Aunque no se ha reducido en una proporción exacta, se ha alcanzado una aproximación aceptable pues la tarjeta de control considera esa leve discrepancia como una perturbación externa que continuamente corrige¹².

¹² Para una descripción más detallada véase la sección Código Principal 'Encoder4.c'

Coordinación de Movimientos

Dimensiones del robot

De las variables definidas en la sección Análisis Cinemático, se enlistan las siguientes dimensiones de los eslabones:

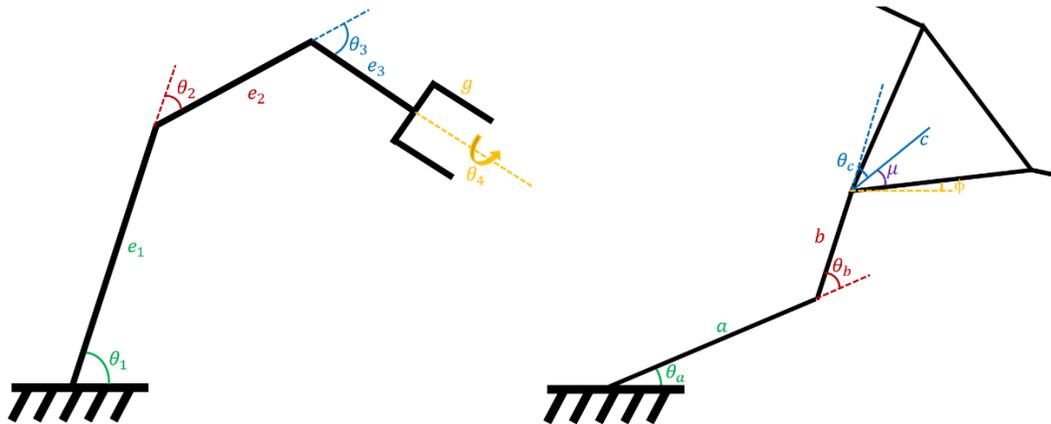


Ilustración 73 Dimensiones del manipulador híbrido

Robot	Eslabón	Dimensión [mm]
Serial	e1	92
	e2	148
	e3	67
	g	45
Paralelo	a	127
	b	112
	c	55

Tabla 4 Tabla de dimensiones de los eslabones

Con ello, se realizaron los cálculos cinemáticos que a continuación se presentan en la sección siguiente.

Dado que los objetivos del presente estudio no contemplan el diseño del modelo funcional, se realizaron los correspondientes acoples y ensambles mediante un trabajo técnico, y se muestran las siguientes imágenes:

Coordinación de Movimientos

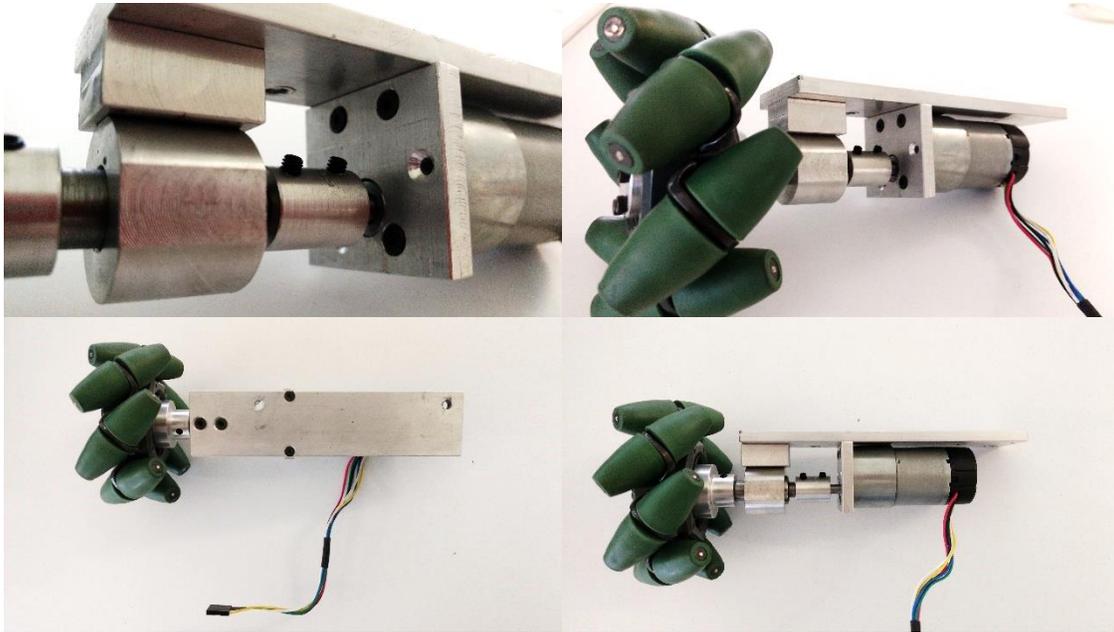


Ilustración 74 Acoples de la rueda al eje y motor, unidos a la base del paralelo

Con ello las medidas de las ruedas que se tienen son las que se exponen a continuación:

Robot Móvil	Diámetro de Ruedas	Distancia Entre Ejes	Distancia Entre Ruedas
[mm]	100	470	375

Tabla 5 Dimensiones del robot móvil¹³

¹³ Se recomienda la revisión de la sección Análisis Cinemático para ver los diagramas del móvil que ahí se presentan.

Cinemática inversa del manipulador paralelo-serie

Tomando en cuenta que la tarea que se ha programado a realizar es el transporte de materiales, se ha programado una rutina en la que se acerca el robot al objetivo y simultáneamente, el manipulador se posiciona para tomarlo, de modo que cuando ya ha llegado el robot a su objetivo, este, simplemente acciona el efector final. En este punto, el manipulador regresa a una posición inicial y finaliza su tarea. El siguiente diagrama muestra la tarea realizada por el manipulador:

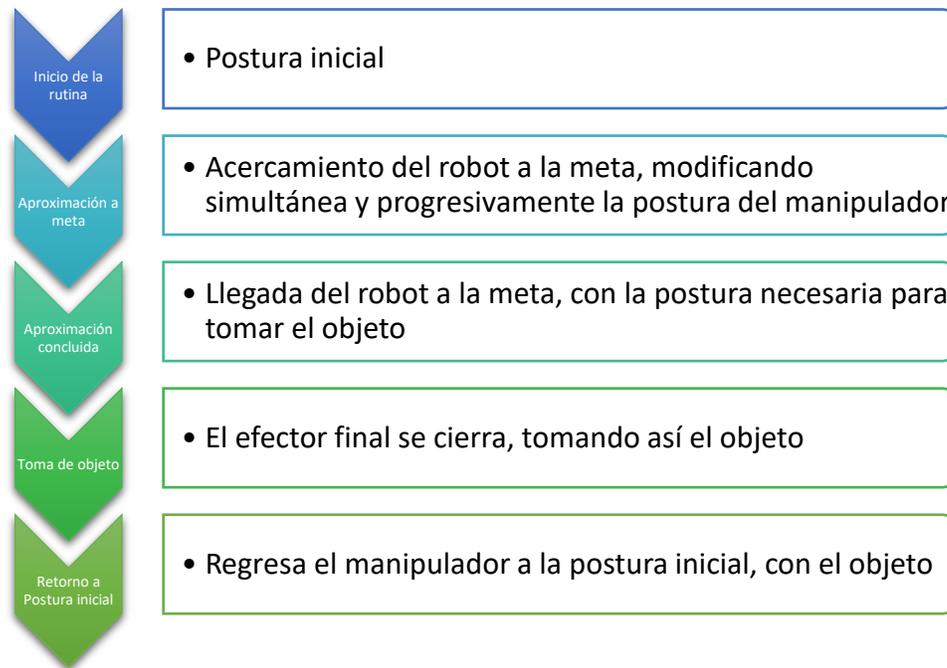


Ilustración 75 Tarea del manipulador paralelo-serie

De este modo, se logra simular un movimiento natural al tomar el objeto, pues se realiza una modificación en la postura del manipulador al mismo tiempo que este se acerca al objeto. A continuación, se muestran los valores de las variables articulares en los puntos iniciales y finales de la rutina:

Coordinación de Movimientos

Robot		Variable	Inicial [°]	Final [°]
Serial		Hombro	150	75
		Codo	30	15
		Muñeca	10	70
		Giro de Muñeca	160	65
		Coordenada	(3.23, 0, 19.4)	(26.88, 0, 3.29)
Paralelo		θ_{a1}	100	150
		θ_{b1}	103.9432	94.2154
		θ_{a2}	85	72
		θ_{b2}	66.642	114.969
		θ_{a3}	0	55
		θ_{b3}	130.63	5.01
		Coordenada	(0, 0, 0)	(0, -12, 0)

Tabla 6 Valores de las variables al inicio y fin de la rutina ¹⁴

Cabe mencionar que los valores para los ángulos θ_b , corresponden a las soluciones seleccionadas que se tienen para las posibles configuraciones y combinaciones codo arriba o codo debajo del robot paralelo. Por cuestiones mecánicas como rozamientos o interferencia entre otros eslabones, se han escogido los ángulos aquí mostrados, véase la Tabla 6.

Descripción del movimiento

Inicio de la rutina

Conociendo los puntos finales e iniciales de nuestro manipulador, es decir, la postura con la que inicia y termina la rutina, se ha desarrollado la siguiente cinemática inversa:

El algoritmo utilizado para ello, tiene como argumento, a la distancia que separa el objetivo de la meta, o sea, la postura varía en función de la separación al objetivo, además de algunas otras constantes como, la distancia que inicialmente los separaba, y la distancia final deseada. Con estos datos comienza a ejecutarse la rutina.

¹⁴ Se debe mencionar que las coordenadas iniciales y finales, son respecto a los sistemas locales (y no a un sistema general) del robot serial y paralelo respectivamente, dado que sirven como referencia.

Aproximación a la meta

Dicho algoritmo, consta de polinomios de tercer grado, uno para cada servomotor, excepto el efector final dado que solo tiene dos estados, abierto y cerrado.

Para la construcción de dichos polinomios, se ha fraccionado la distancia de separación en 256 partes, con el objeto de volver el movimiento natural y fluido, y sin importar la separación inicial, se divide automáticamente en 256 puntos. Lo anterior se muestra en la ecuación:

$$Puntos = 256 \frac{d_{memorizada} - d_{actual}}{d_{memorizada} - d_{acople}}$$

Ecuación 61 Ecuación Generadora de Puntos

La ecuación anterior, tiene como resultados a números reales desde 0 hasta 256, iniciando con 0 y termina en 256, dado que al inicio la distancia memorizada (la distancia inicial de separación guardada en memoria) es igual a la distancia actual, y al final de la ejecución de la rutina, la distancia actual, es la distancia de acople, el cociente de la Ecuación 61 tiene valores entre 0 y 1.

En este momento, se utiliza la variable "Puntos" de la ecuación anterior abreviada con la variable x, se utiliza como argumento de los siguientes polinomios, uno por cada servomotor, de cuyos resultados corresponden los ángulos de cada servomotor en las gráficas siguientes:

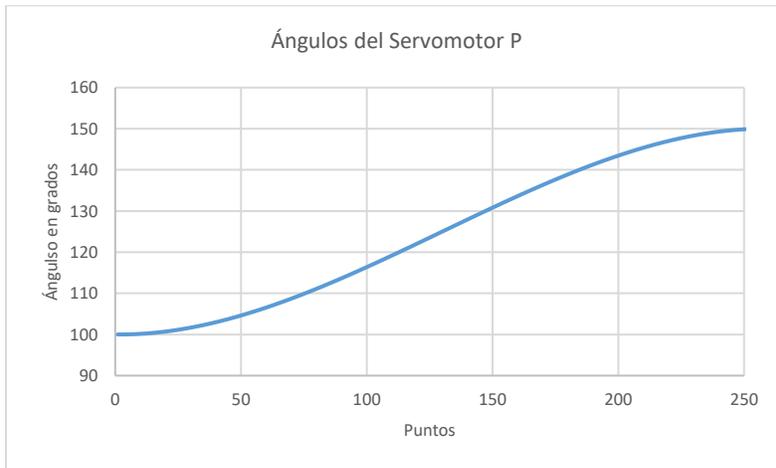


Ilustración 76 Comportamiento del servomotor P en el trayecto de acercamiento a la meta

$$P = - 0.000005960464478x^3 + 0.002306699752808x^2 - 0.004595518112184x + 100.002294778817$$

Ecuación 62 Polinomio del servomotor P

Coordinación de Movimientos

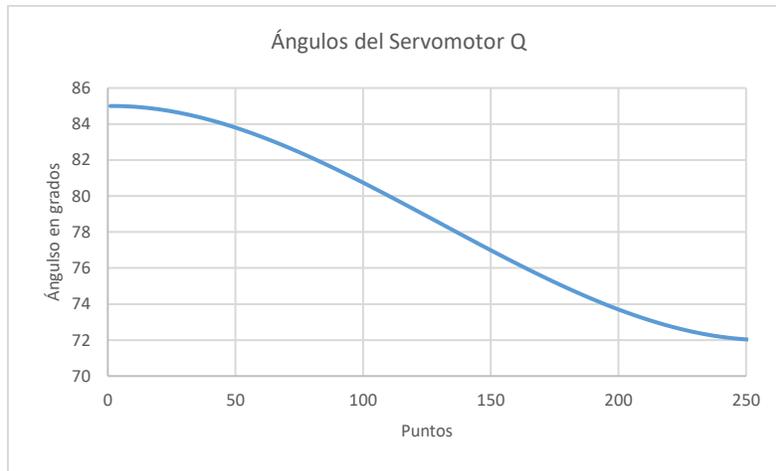


Ilustración 77 Comportamiento del servomotor Q en el trayecto de acercamiento a la meta

$$Q = 0.000001549720764x^3 - 0.000599741935730x^2 + 0.001194834709195x + 84.9994035575012$$

Ecuación 63 Polinomio del servomotor Q



Ilustración 78 Comportamiento del servomotor R en el trayecto de acercamiento a la meta

$$R = -0.000006556510925x^3 + 0.002537369728088x^2 - 0.005055069923401x + 0.002524256703509$$

Ecuación 64 Polinomio del servomotor R

Coordinación de Movimientos

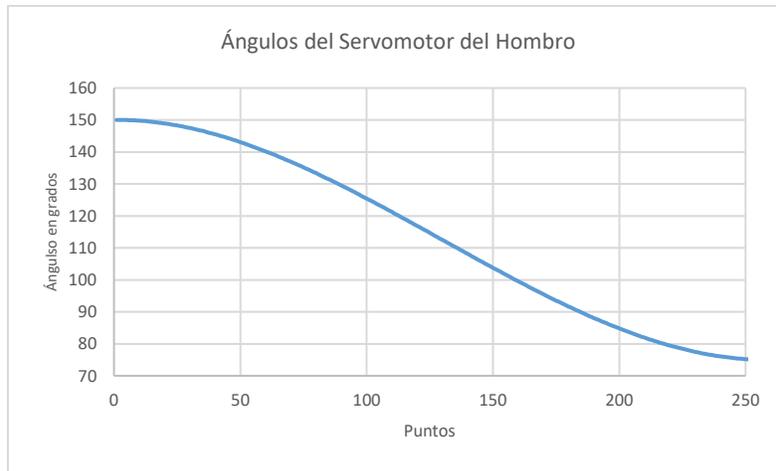


Ilustración 79 Comportamiento del servomotor del hombro en el trayecto de acercamiento a la meta

$$\text{Hombro} = 0.000008940696716x^3 - 0.003460049629212x^2 + 0.006893277168303x + 149.996557831758$$

Ecuación 65 Polinomio de los servomotores de los hombros

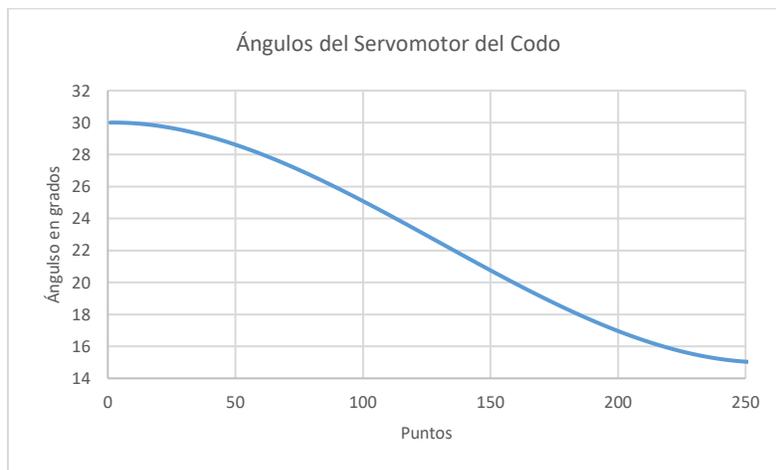


Ilustración 80 Comportamiento del servomotor del codo en el trayecto de acercamiento a la meta

$$\text{Codo} = 0.000001788139343x^3 - 0.000692009925842x^2 + 0.001378655433651x + 29.9993115663517$$

Ecuación 66 Polinomio del servomotor del Codo

Coordinación de Movimientos

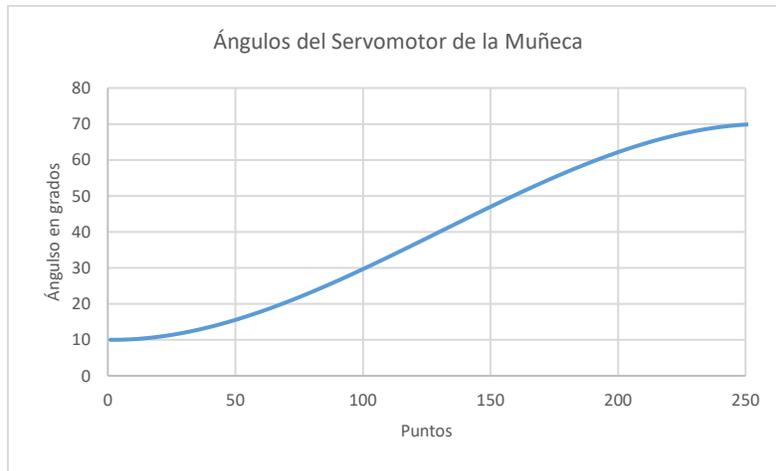


Ilustración 81 Comportamiento del servomotor de la muñeca en el trayecto de acercamiento a la meta

$$\text{Muñeca} = -0.000007152557373x^3 + 0.002768039703369x^2 - 0.005514621734605x + 10.0027537345854$$

Ecuación 67 Polinomio del servomotor de la Muñeca

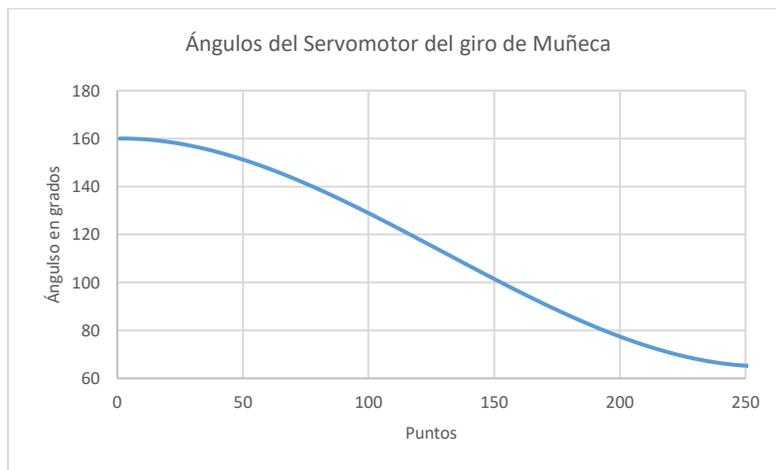


Ilustración 82 Comportamiento del servomotor del giro de la muñeca en el trayecto de acercamiento a la meta

$$g\text{Muñeca} = 0.000011324852507x^3 - 0.004382729530334x^2 + 0.008731484413190x + 159.99563992023$$

Ecuación 68 Polinomio del servomotor del giro de la Muñeca

Como se puede apreciar, cada uno de los comportamientos responde a un perfil cúbico de velocidades. Se analizaron distintas opciones de perfiles de velocidad, resultando en un perfil cúbico como la mejor opción, dado que presenta la menor aceleración en los motores, lo que conlleva a menores gastos de energía, menor desgaste en los actuadores y movimientos menos bruscos que en perfiles tales como, sinusoidal, de segundo, cuarto o quinto grado.

Además, sabiendo que el comportamiento parte de un comportamiento discreto, al utilizar una regresión polinómica, se vuelve continuo el comportamiento.

Aproximación concluida y toma del objeto

Con lo anterior, se logra posicionar al móvil a la distancia adecuada para tomar el objeto y a la vez, teniendo ya una postura para hacerlo. Entonces, se ejecuta un movimiento de cerrar las pinzas del efector final, tomando el objeto, para regresar a la postura inicial del robot, y finalizando con ello la tarea.

El movimiento de retorno a la postura inicial, se logra utilizando los mismos polinomios utilizados alcanzar la postura de tomado del objeto, pero iniciando un conteo en sentido contrario.

Retorno a la postura Inicial

Dado que el movimiento programado para adquirir la postura del manipulador para tomar el objeto, funcionan con polinomios (mostrados a partir de la Ecuación 62) cuyo argumento debe incrementarse con números reales, partiendo del cero y llegando hasta el 256, para ejecutar una rutina contraria, se debe recorrer el argumento decrementando desde el punto 256 hasta el 0, es decir en orden inverso.

Recordando lo mencionado en la sección Procesamiento de Datos en Matlab, se tiene una condicional en la que se decide el momento en el que se ejecuta el retorno, esta toma 5 segundos en ejecutarse.

El bloque de Matlab que procesa esta información se encuentra en el sistema Trayectoria del Manipulador, subsistema Pick-Pick Back.

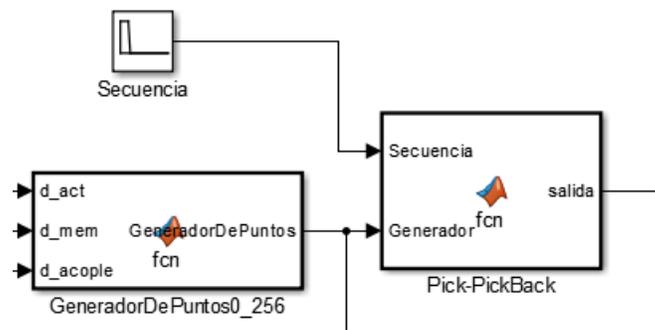


Ilustración 83 Subsistema generador del movimiento de retorno a la postura inicial

Simulaciones, pruebas y resultados

Lo que se pretende en este capítulo es mostrar los resultados obtenidos en la ejecución de la tarea, por ende, se anexan imágenes, fotografías y simulaciones realizadas.

La trayectoria programada para el manipulador paralelo serie se exhibe a continuación mediante las siguientes imágenes, estas muestran los puntos por los que se posiciona el efector final, respectivamente son las ejecuciones en 0, 25, 50, 75 y 100% de la tarea realizada:

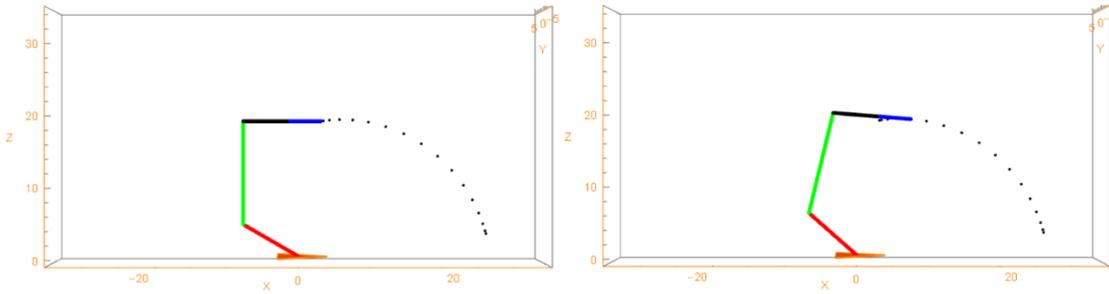


Ilustración 84 0 y 25% de la tarea realizada

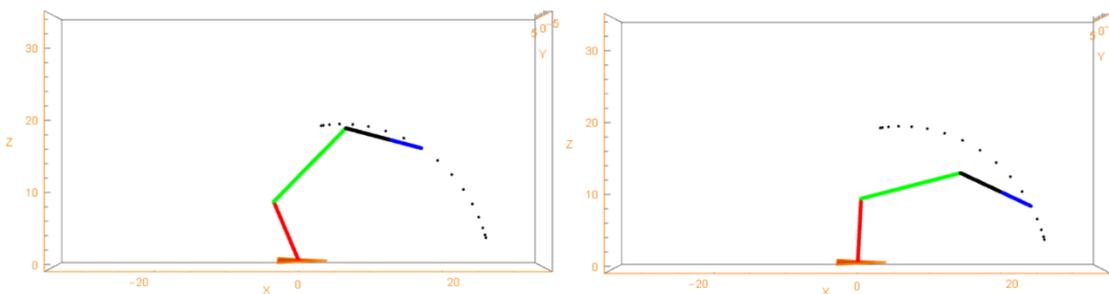


Ilustración 85 50 y 75% de la tarea realizada

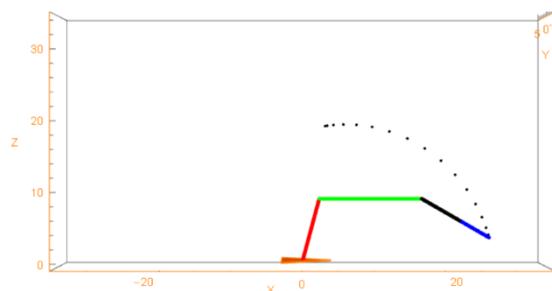


Ilustración 86 100% de la tarea realizada

Además de lo anterior, se anexa la trayectoria ejecutada por el manipulador serial, combinando los movimientos definidos por los polinomios de cada uno de sus motores, cabe mencionar que la siguiente ilustración muestra el movimiento independiente del robot serial, sin el paralelo.

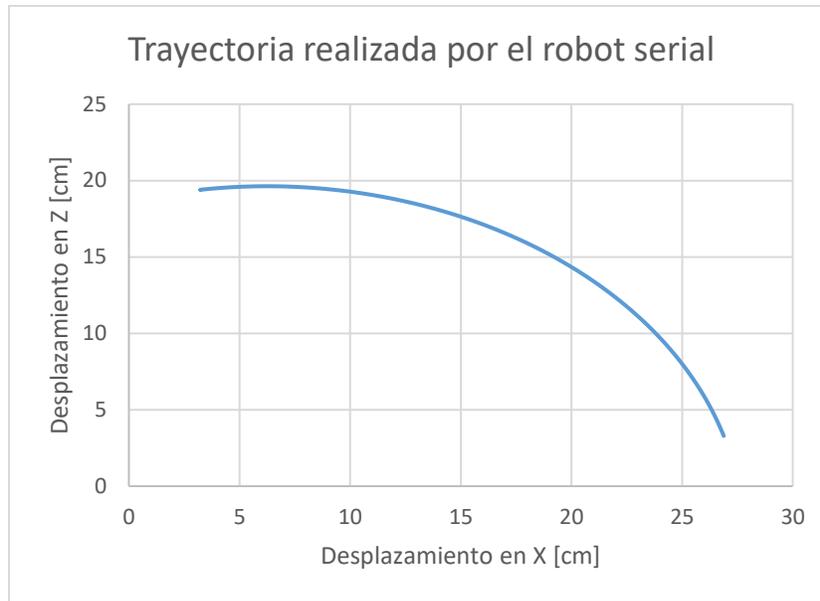


Ilustración 87 Trayectoria ejecutada por el robot serie en el plano XZ¹⁵

Se debe recordar que, el robot serial, posee movimiento en el plano únicamente, prescindiendo por su particular arreglo de un giro en su base, dado que este es proporcionado por el robot paralelo.

Respectivamente con este, se muestra a continuación la simulación efectuada para probar la cinemática del mismo:

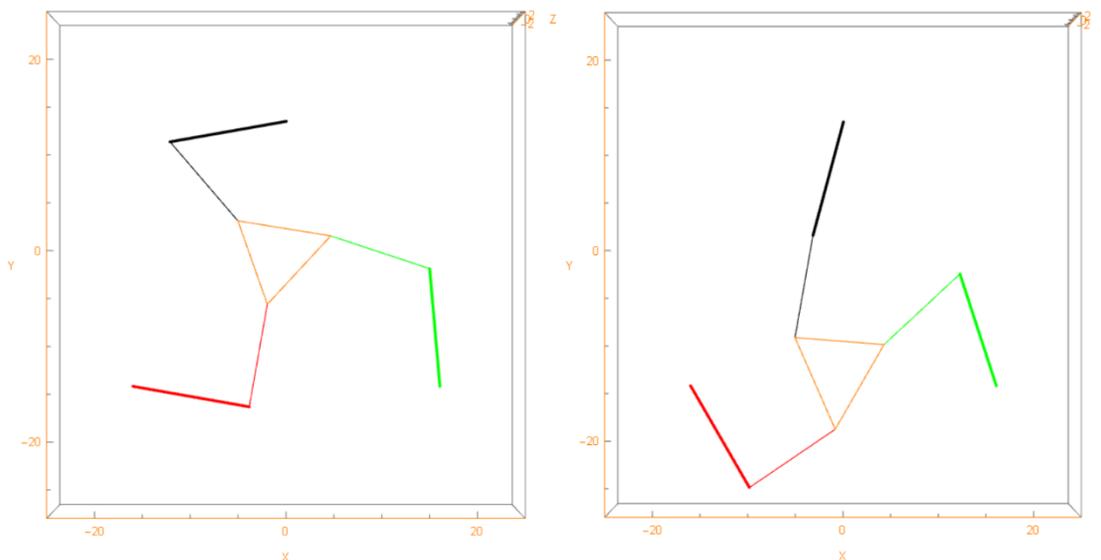


Ilustración 88 Punto inicial y final en la simulación del robot paralelo delta

¹⁵ Se recuerda al lector, que dicho sistema XZ corresponde al sistema local de la base del robot serial y no a un marco de referencia general

Lo anterior se muestra también en las siguientes fotografías del manipulador paralelo serie, en donde se exhibe un movimiento completo y coordinado entre la parte serial y paralela:

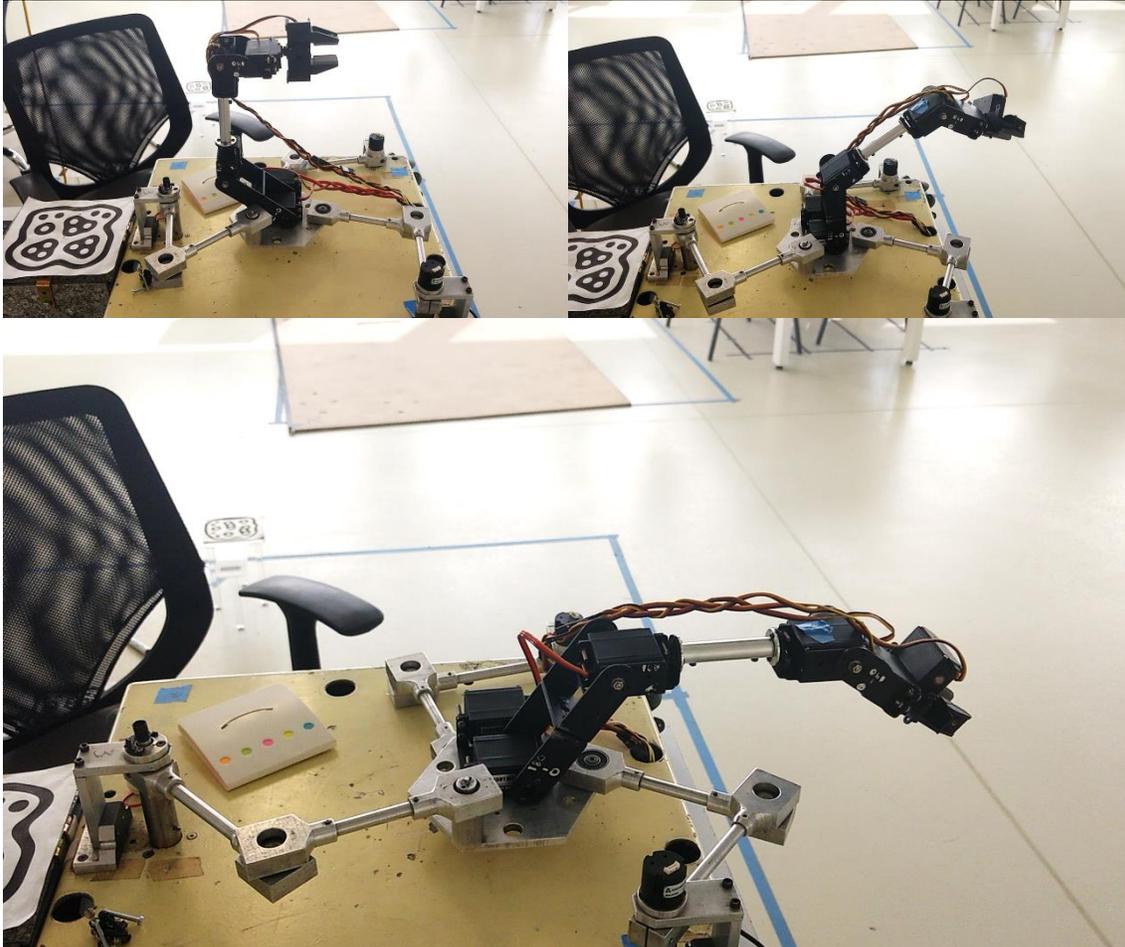
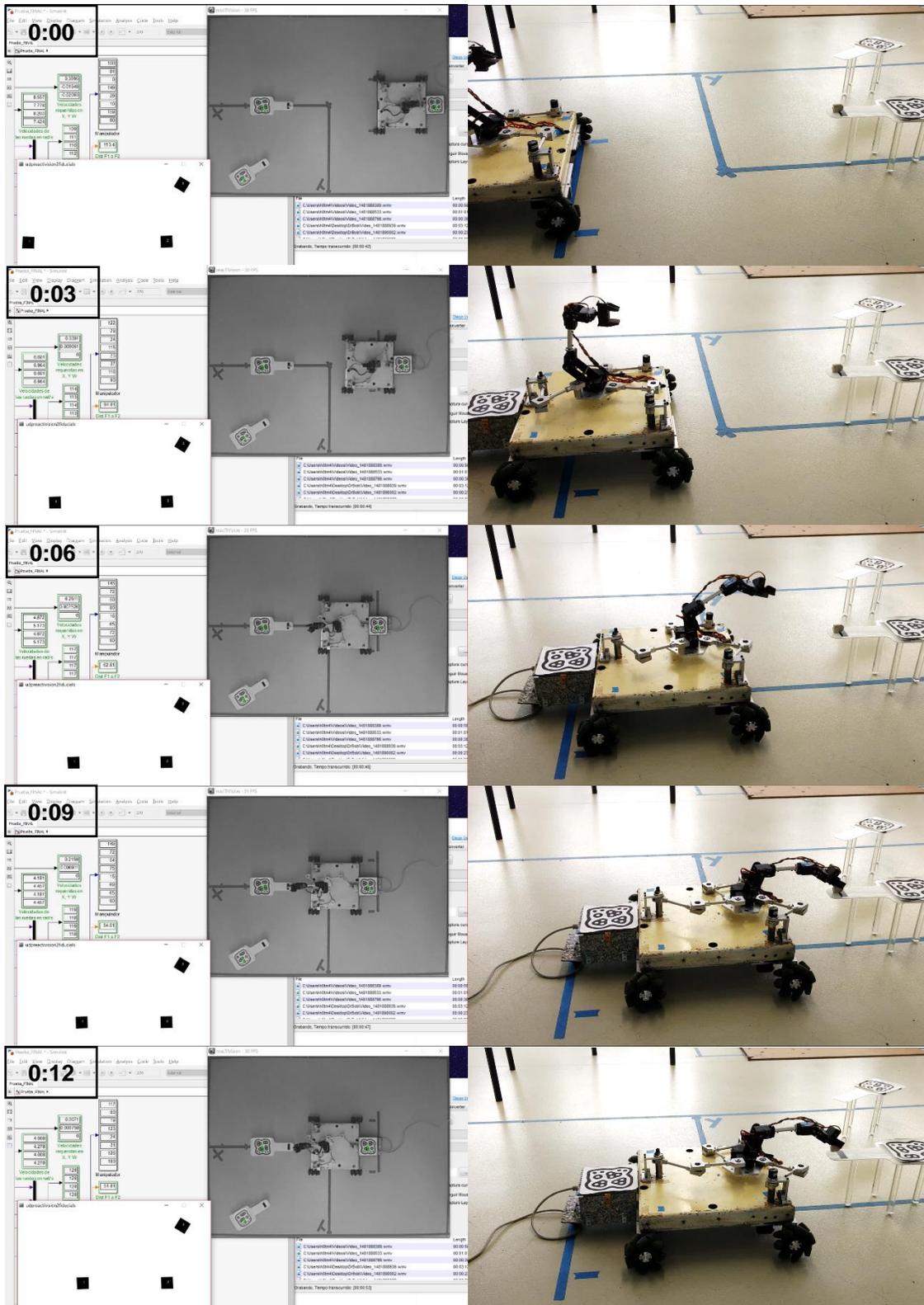


Ilustración 89 Ejecución de rutina de tomar objeto al (respectivamente) 0, 50 y 100% ¹⁶

A continuación, se presentan las imágenes donde se aprecia la ejecución de la tarea mediante fotografías tomadas de videos cada 3 segundos aproximadamente, esto es, desde que comienza la tarea hasta que se acaba la rutina:

¹⁶ Se pide al lector observar el giro en la muñeca del manipulador serial en su posición inicial y final, en estas fotografías, ya que en las simulaciones no era posible apreciarse, el cual permite una postura adecuada para tomar un objeto. En la primera fotografía (izquierda) el efector final tiene una posición vertical, que cambia paulatinamente hasta tomar la posición final que es horizontal, como se muestra en la última fotografía (abajo).

Simulaciones, pruebas y resultados



Simulaciones, pruebas y resultados

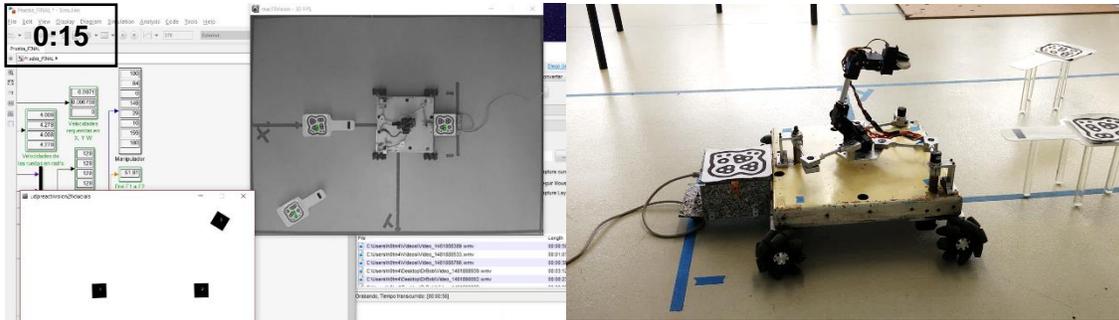


Ilustración 90 Secuencia de ejecución de la tarea, imágenes tomadas de videos realizados simultáneamente

Respecto a la simulación con la que se ejecutó este movimiento, se presentan las siguientes imágenes:

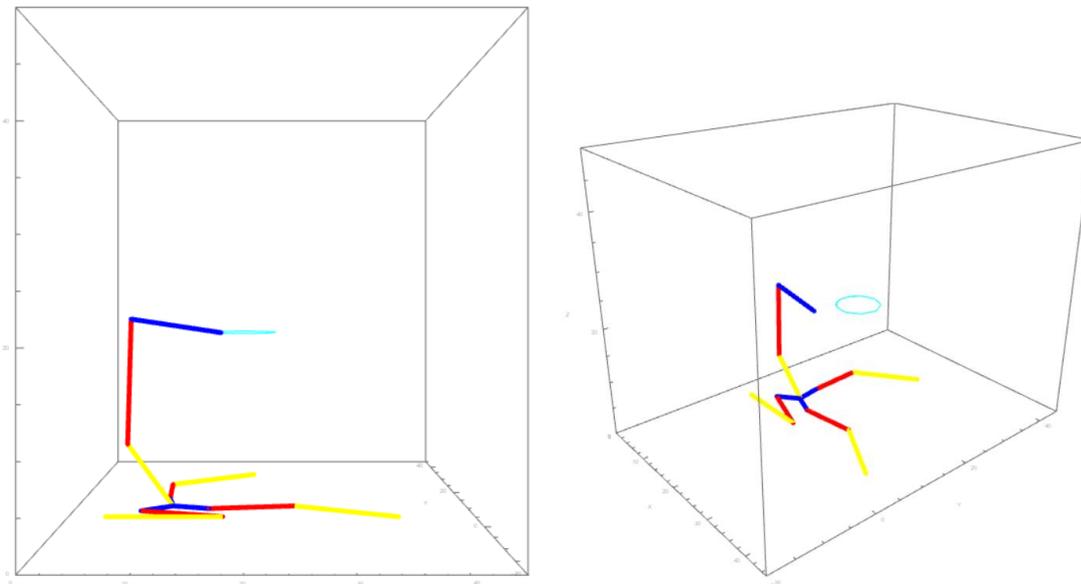


Ilustración 91 Simulaciones del movimiento del manipulador híbrido realizando una trayectoria de prueba

Se hace la mención de que se cuentan con más fotografías que muestran el movimiento del robot, desde diferentes ángulos y cámaras simultáneamente en la sección Imágenes Adicionales de la Tarea a Realizar.

Comentarios finales

Discusión

Cabe mencionar que, a lo largo de la realización de este trabajo, se han hecho notar las siguientes cosas respecto a aspectos técnicos:

Con el arreglo propuesto, se puede implementar a cualquier tipo de motor el control PID de la tarjeta MD25, dado que se puede implementar a cualquier motor, el microcontrolador reductor de frecuencia cumpliendo con ello la simple condición de que se escalen los pulsos del motor en cuestión para que existan 360 pulsos por revolución.

Respecto al robot paralelo serie, no se define por una regla de comportamiento, sino que las limitaciones físicas restringen su movimiento, ya que por ejemplo, el cambio en los ángulos cercanos al límite de giro de los servos es poco confiable, o la interferencia entre los eslabones del robot serial y del paralelo imposibilitan algunas soluciones generadas en la cinemática inversa, dado que por ser un elemento físico, tiene limitaciones que restringen su amplia gama de movimientos.

Se ha observado también que los servomotores necesitan un retardo en el programa para que funcionen adecuadamente, a diferencia de los motores de DC. Esto se debe a que los servomotores trabajan en el campo de las posiciones y no el de las velocidades, y además, se necesita de cierto tiempo para poder tomar el ángulo deseado, por la velocidad de giro particular que cada uno de los servomotores tiene.

Puesto que se implementan una gran cantidad de motores, se necesita de una gran cantidad de energía para manejar adecuadamente el robot, alrededor de 20A, por lo que se necesitó de la etapa de potencia implementada, para los motores de DC y los servomotores.

Debido a que la visión de la cámara tiene una forma cónica en lugar de una pirámide con base plana, se tiene un mejor desempeño de la visión artificial, en las zonas cercanas al centro del ambiente inteligente, en comparación a la periferia debido a la distorsión de la imagen.

Conclusiones

Igualmente, se tienen los siguientes comentarios respecto a la cuestión teórica desarrollada y generada. Por un lado, la separación en los puntos de la trayectoria en la Ilustración 86 señala que la postura del manipulador híbrido varía en función de la velocidad la cual no es constante, sino como se muestra anteriormente, cúbica. Eligiendo este perfil, por presentar la menor aceleración en los actuadores del manipulador, lo que implica un menor desgaste, fricción y demanda energética en ellos, en comparación a perfiles polinómicos de mayor grado, trigonométricos u otros. Además, que esta función cúbica provoca que el manipulador cuente

con una posición en cada momento. Información que, se tiene mostrada en la interfaz de usuario de Matlab.

El giro de la muñeca es importante dado que permite una postura adecuada para tomar un objeto; este giro cambia de una posición vertical en un momento inicial, y una postura horizontal, en el momento de tomar el objeto; tal y como se muestra en la Ilustración 89.

La unión entre diferentes sistemas de referencia (posición y velocidad), se logró al determinar la distancia final en la que se acoplaría el robot. Dicha coordinación se realizó con base en las restricciones físicas del manipulador, dado que no se logran todos los movimientos matemáticamente posibles. Por ello, se propuso esta primera solución véase Ilustración 88. También se notó que la solución óptima no depende del tiempo, sino de la posición. Esto se confirma con los comportamientos naturales que se realizan en sistemas biológicos, por ejemplo, al cazar, un águila no toma en cuenta el tiempo que le tomará llegar hasta su presa, sino la distancia, igualmente una persona al atrapar un balón en el aire, se concentra en la decreciente distancia que separa su mano del balón, y no el tiempo que requerirá tomarlo. Igualmente, nuestra solución está basada en ello, en distancia y no en tiempo.

Debido a la configuración empleada, el espacio de trabajo del manipulador, fue ampliado mediante la combinación de un robot paralelo en la base, dotando al ahora manipulador híbrido, de un giro en su base respecto a la vertical. Es también necesario mencionar que se aportaron las bases del modelo cinemático de un sistema totalmente nuevo, el cual tiene un sinfín de aplicaciones y configuraciones aún no exploradas.

Finalmente, para efectuar el desplazamiento en el ambiente inteligente, no fue necesario dotar al robot híbrido de sensores para la adquisición de datos de posición y orientación; ya que se implementó un sistema de visión artificial, repercutiendo de manera positiva en la velocidad de procesamiento de datos.

Trabajo a futuro

Profundizar en las ventajas cinemáticas que el robot paralelo puede aportar en la compensación del ángulo de error provocado por la plataforma móvil, además de la posibilidad de manipular objetos localizados en los costados del móvil y no únicamente en su frente; para ello se sugiere modificar las dimensiones o la disposición de los eslabones del manipulador con el fin de evitar o disminuir las restricciones mecánicas. Y finalmente realizar el modelado dinámico del sistema, el cual puede derivar en otra propuesta de coordinación de movimientos para el transporte de materiales; así como la elaboración de los planos mecánicos por computadora para desarrollar un estudio más detallado del comportamiento mecánico del robot.

Apéndices

Código de los Reductores de Frecuencia

Bits de Configuración 'Config.h'

```

Encoder4.c x Config.h x
Source History
1 // PIC16F88 Configuration Bit Settings
2 // 'c' source line config statements
3 // CONFIG1
4 #pragma config FOSC = HS // Oscillator Selection bits (HS oscillator)
5 #pragma config WDTE = OFF // Watchdog Timer Enable bit (WDT disabled)
6 #pragma config FWRT = OFF // Power-up Timer Enable bit (PWRT disabled)
7 #pragma config MCLR = ON // RA5/MCLR/VPP Pin Function Select bit (RA5/MCLR/VPP pin function is MCLR)
8 #pragma config BOREN = OFF // Brown-out Reset Enable bit (BOR disabled)
9 #pragma config LVP = OFF // Low-Voltage Programming Enable bit (RB3 is digital I/O, HV on MCLR must be used for programming)
10 #pragma config CPD = OFF // Data EE Memory Code Protection bit (Code protection off)
11 #pragma config WRT = OFF // Flash Program Memory Write Enable bits (Write protection off)
12 #pragma config CCPMX = RB0 // CCP1 Pin Selection bit (CCP1 function on RB0)
13 #pragma config CP = OFF // Flash Program Memory Code Protection bit (Code protection off)
14 // CONFIG2
15 #pragma config FCMEN = OFF // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock Monitor disabled)
16 #pragma config IESO = OFF // Internal External Switchover bit (Internal External Switchover mode disabled)
17 // #pragma config statements should precede project file includes.
18 // Use project enums instead of #define for ON and OFF.
19 // #include <xc.h>

```

Código Principal 'Encoder4.c'

```

Encoder4.c x Config.h x
Source History
1 /*
2  * File: Encoder4.c
3  * Author: Annie
4  *
5  * Created on 2 de septiembre de 2016, 17:00
6  */
7 #include <xc.h>
8 #include "Config.h"
9 #include <stdio.h>
10 #include <stdlib.h>
11 #define XTAL_FREQ 2000000
12
13 int Ac, Bc, Ah, Bh, estado;
14
15 void main(void) {
16     //Configuraciones de puertos e interrupciones
17     ANSEL = 0; // I/O digitales
18     TRISB = 0b11000000; // Entradas y salidas del PORTB
19     PORTB = 0; // Limpia el PORTB
20     INTCON = 0b10001000; // Habilita interrupciones generales y
21     OPTION_REG = 0b01111111; // por cambios de estado en <RB7:RB4>
22
23     __delay_ms(400);
24     Ah = PORTBbits.RB7;
25     Bh = PORTBbits.RB6;
26
27     //definimos el punto de partida
28     switch(2 * Ah + Bh){
29         case 0:
30             estado = 1;

```

```
31         break;
32
33     case 1:
34         estado = 2;
35         break;
36
37     case 3:
38         estado = 3;
39         break;
40
41     case 2:
42         estado = 4;
43         break;
44     }
45
46     while(1);
47
48     return;
49 }
50
51 void interrupt CambioDeEstado(){
52     Ac = PORTBbits.RB7;
53     Bc = PORTBbits.RB6;
54
55     if(Ah != Ac){
56         switch(2 * Bh + Ac){
57             case 0:
58                 estado ++;
59                 break;
60
```

```
61     case 1:
62         estado --;
63         break;
64
65     case 2:
66         estado --;
67         break;
68
69     case 3:
70         estado ++;
71         break;
72     }
73 }else{
74     switch(2 * Ah + Bc){
75         case 0:
76             estado --;
77             break;
78
79         case 1:
80             estado ++;
81             break;
82
83         case 2:
84             estado ++;
85             break;
86
87         case 3:
88             estado --;
89             break;
90     }
```

```

91     }
92     if(estado == 0){
93         estado = 36;
94     }
95     if (estado == 37){
96         estado = 1;
97     }
98     if(estado >= 1 && estado <= 9){
99         PORTBbits.RB5 = 0;
100        PORTBbits.RB4 = 1;
101    }
102
103    if(estado >= 10 && estado <= 18){
104        PORTBbits.RB5 = 0;
105        PORTBbits.RB4 = 0;
106    }
107
108    if(estado >= 19 && estado <= 27){
109        PORTBbits.RB5 = 1;
110        PORTBbits.RB4 = 0;
111    }
112
113    if(estado >= 28 && estado <= 36){
114        PORTBbits.RB5 = 1;
115        PORTBbits.RB4 = 1;
116    }
117    Ah = Ac; Bh = Bc;
118    INTCON = 0b10001000;
119    return;
120 }

```

Código de la Plataforma Móvil Omnidireccional

```

1 /*Código realizado por Annie y Beto :3
2  con la ayuda de Pablo, Arturo, Ignacio, Irene, Jesús, etc, etc, etc
3  Título: PlataformaMóvil
4  Tesis de Annie y Beto
5  Febrero 2017
6  * */
7 //Liberias wifi
8 #include <WiFi.h>
9 #include <Wire.h>
10 #include <WiFiUdp.h>
11
12 #define i2cAddressT1 0x58 //dirección de la MD25 T1
13 #define RvelmIT1 0x00 //dirección del registro de velocidad D1 T1
14 #define RvelmDT1 0x01 //dirección del registro de velocidad I1 T1
15
16 #define i2cAddressT2 0x59 //dirección de la MD25 T2
17 #define RvelmIT2 0x00 //dirección del registro de velocidad D2 T2
18 #define RvelmDT2 0x01 //dirección del registro de velocidad I2 T2
19
20 char ssid[]="Ana"; //nombre de la red

```

Apéndices

```
21 char pass[]="AnnieAnnie!";          //contraseña de la red
22 int status = WL_IDLE_STATUS;
23
24 int R3=128, R4=128, R1=128, R2=128; //Velocidades iniciadas en "0"
25
26 int localPort=1000;                 // Puerto remoto del packet output 1000
27 char packetBuffer[255];             // buffer para contener el paquete entrante
28 WiFiUDP Udp;
29
30 void setup(){
31   Wire.begin();//Inicio de puerto serial
32   Serial.begin(115200);
33   while(!Serial){// Espera por el puerto serial para conectar.
34   }
35
36   // Checa la presencia del shield:
37   if (WiFi.status() == WL_NO_SHIELD){
38     Serial.println("Sin Wifi shield");
39     //Mientras no haya Shield no continua
40     while (true);
41   }
42
43   //Mientras el shield no esté conectado
44   while (status !=WL_CONNECTED){
45     // Intenta
46     status = WiFi.begin(ssid, pass);
47     //Esperando conexión
48     delay (1000);
49     //Conectando...
50     printWifiStatus();
51     //Inicia comunicación UDP por el puerto local
52     Udp.begin(localPort);
53   }
54 }
55
56 void loop()
57 {
58   //Si hay datos disponibles, lee el paquete
59   int packetSize=Udp.parsePacket(); // Checa presencia del paquete UDP y reporta el tamaño
60                                     //debe ser llamado antes de leer el buffer ( UDP.read())
61   if (packetSize){
62     //tamaño del Udp packet len =255
63     int len=Udp.read(packetBuffer,255);
64     if(len>0) {packetBuffer[len]=0;}
65
66     R1=convertir(packetBuffer[0]);
67     R2=convertir(packetBuffer[1]);
68     R3=convertir(packetBuffer[2]);
69     R4=convertir(packetBuffer[3]);
70
71     //-----Transmisión de velocidades a la MD25-----
72     //Tarjeta1 llantaDerecha x058
73     digitalWrite(13,HIGH);
74     Wire.beginTransmission(i2cAddressT1);
75     Wire.write(RvelMDT1);
76     Wire.write(R3);
77     Wire.endTransmission();
78     //Tarjeta1 llantaIzquierda x058
79     Wire.beginTransmission(i2cAddressT1);
80     Wire.write(RvelIT1);
```

Apéndices

```
81     Wire.write(R4);
82     Wire.endTransmission();
83     digitalWrite(13,LOW);
84
85     //Tarjeta2 llantaDerecha x059
86     digitalWrite(13,HIGH);
87     Wire.beginTransmission(i2cAddressT2);
88     Wire.write(RvelmDT2);
89     Wire.write(R1);
90     Wire.endTransmission();
91     //Tarjeta2 llantaIzquierda x059
92     Wire.beginTransmission(i2cAddressT2);
93     Wire.write(RvelmIT2);
94     Wire.write(R2);
95     Wire.endTransmission();
96     digitalWrite(13,LOW);
97 }
98 }
99
100 void printWifiStatus(){
96     digitalWrite(13,LOW);
97 }
98 }
99
100 void printWifiStatus(){
101     IPAddress ip = WiFi.localIP();
102     long rssi=WiFi.RSSI();
103 }
104
105 int convertir(int dato){
106     //Recorriendo el cero a 128
107     if(dato<0){
108         int negdato=0;
109         negdato=dato*(-1);
110         dato=negdato+((dato+128)*2);
111     }
112     return dato;
113 }
114
115
```

Código del Manipulador

```
Manipulador$
1 /*Código realizado por Annie y Beto :3
2   con la ayuda de Pablo, Arturo, Ignacio, Irene, Jesús, etc, etc, etc
3   Título: Manipulador
4   Tesis de Annie y Beto
5   Febrero 2017
6   * */
7 //Liberias wifi
8 #include <WiFi.h>
9 #include <WiFiUdp.h>
10 #include <Servo.h>
11
12 char ssid[] = "Ana";           //nombre de la red
13 char pass[] = "AnnieAnnie!";  //contraseña de la red
14 int status = WL_IDLE_STATUS;
15
16 int gP, gQ, gR, gH, gC, gM, ggM, gG; //Posición angular del robot
17 int retardo = 40;
18
19 int localPort = 1000;         // Puerto remoto del packet output 1000
20 char packetBuffer[255];      // buffer para contener el paquete entrante
```

Apéndices

```
Manipulador
21 WiFiUDP Udp;
22 Servo P, Q, R, Hombro1, Hombro2, Codo, Muneca, gMuneca, Gripper;
23
24 void setup() {
25     // Wire.begin();//Inicio de puerto serial
26     Serial.begin(115200);//Nos gusta la velocidad
27     while (!Serial) { // Espera por el puerto serial para conectar.
28     }
29
30     // Checa la presencia del shield:
31     if (WiFi.status() == WL_NO_SHIELD) {
32         Serial.println("Sin Wifi shield");
33         //Mientras no haya Shield no continua
34         while (true);
35     }
36
37     //Mientras el shield no esté conectado
38     while (status != WL_CONNECTED) {
39         // Intenta
40         status = WiFi.begin(ssid, pass);
41         //Esperando conexión
42         delay (500);
43         //Conectando...
44         printWifiStatus();
45         //Inicia comunicación UDP por el puerto local
46         Udp.begin(localPort);
47     }
48     P.attach(2);
49     Q.attach(3);
50     R.attach(4);
51     Hombro1.attach(5);
52     Hombro2.attach(6);
53     Codo.attach(7);
54     Muneca.attach(8);
55     gMuneca.attach(9);
56     Gripper.attach(11);
57
58     ManipuladorEnHome();
59     delay(3000);
60 }
```

Apéndices

```
Manipulador
61
62 void loop()
63 {
64     //Si hay datos disponibles, lee el paquete
65     int packetSize = Udp.parsePacket(); // Checa presencia del paquete UDP
66     //y reporta el tamaño debe ser llamado antes de leer el buffer ( UDP.read())
67     if (packetSize) {
68         //tamaño del Udp packet len =255
69         int len = Udp.read(packetBuffer, 255);
70         if (len > 0) {
71             packetBuffer[len] = 0;
72         }
73
74         //Lectura de información de grados
75         gP = convertir(packetBuffer[0]);
76         gQ = convertir(packetBuffer[1]);
77         gR = convertir(packetBuffer[2]);
78         gH = convertir(packetBuffer[3]);
79         gC = convertir(packetBuffer[4]);
80         gM = convertir(packetBuffer[5]);
81
82         ggM = convertir(packetBuffer[6]);
83         gG = convertir(packetBuffer[7]);
84
85         P.write(gP);
86         Q.write(gQ);
87         R.write(gR);
88         Hombro1.write(gH);
89         Hombro2.write(gH);
90         Codo.write(gC);
91         Muneca.write(gM);
92         gMuneca.write(ggM);
93         Gripper.write(gG);
94
95         //Retardo para evitar el parkinson del robot 0.0?
96         delay(retardo);
97     }
98
99 void printWifiStatus() {
100     IPAddress ip = WiFi.localIP();
Guardado.
```

Apéndices

```
Manipulador
101 long rssi = WiFi.RSSI();
102 }
103
104 int convertir(int dato) {
105     //Recorriendo el cero a 128
106     if (dato < 0) {
107         int negdato = 0;
108         negdato = dato * (-1);
109         dato = negdato + ((dato + 128) * 2);
110     }
111     return dato;
112 }
113
114 boolean ManipuladorEnHome() {
115     //Posicion inicial
116     P.write(100);
117     Q.write(85);
118     R.write(0);
119     Hombro1.write(150);
120     Hombro2.write(150);
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }

Guardado.
Manipulador
109     dato = negdato + ((dato + 128) * 2);
110 }
111 return dato;
112 }
113
114 boolean ManipuladorEnHome() {
115     //Posicion inicial
116     P.write(100);
117     Q.write(85);
118     R.write(0);
119     Hombro1.write(150);
120     Hombro2.write(150);
121     Codo.write(30);
122     Muneca.write(10);
123     gMuneca.write(160);
124     Gripper.write(180);
125
126     //Retardo para evitar el parkinson del robot O.o?
127     delay(retardo);
128 }
```

Código Principal de Wolfram Mathematica

CINEMÁTICA INVERSA DEL ROBOT PARALELO 3 RRR POR SUSTITUCIÓN POLIMOMIAL

Funciones

```

In[1]= Qx[φx] := {{1, 0, 0, 0}, {0, Cos[φx], -Sin[φx], 0}, {0, Sin[φx], Cos[φx], 0}, {0, 0, 0, 1}}
           |coseno |seno |seno |coseno
Qy[φy] := {{Cos[φy], 0, Sin[φy], 0}, {0, 1, 0, 0}, {-Sin[φy], 0, Cos[φy], 0}, {0, 0, 0, 1}}
           |coseno |seno |seno |coseno
Qz[φz] := {{Cos[φz], -Sin[φz], 0, 0}, {Sin[φz], Cos[φz], 0, 0}, {0, 0, 1, 0}, {0, 0, 0, 1}}
           |coseno |seno |seno |coseno

Tx[x_] := {{1, 0, 0, x}, {0, 1, 0, 0}, {0, 0, 1, 0}, {0, 0, 0, 1}}
Ty[y_] := {{1, 0, 0, 0}, {0, 1, 0, y}, {0, 0, 1, 0}, {0, 0, 0, 1}}
Tz[z_] := {{1, 0, 0, 0}, {0, 1, 0, 0}, {0, 0, 1, z}, {0, 0, 0, 1}}
Txyz[x_, y_, z_] := {{1, 0, 0, x}, {0, 1, 0, y}, {0, 0, 1, z}, {0, 0, 0, 1}}
n = {0, 0, 0, 1};
T3D[p_] := {p[[1]], p[[2]], p[[3]]};

Grafica[tabla_, rojo_, verde_, azul_, x_, y_] := ListPlot[tabla,
           |representación de lista
ImageSize → 800, Joined → True, PlotStyle → {AbsoluteThickness[6], RGBColor[rojo, verde, azul]},
           |tamaño de imagen |unido |ver... |estilo de repr... |grosor absoluto |color RGB
BaseStyle → {24, FontFamily → "Arial"}, Frame → True,
           |estilo base |familia de tipo de letra |marco |verdadero
FrameLabel → {x, y}, GridLines → Automatic, PlotRange → All];
           |etiqueta de marco |parrilla de li... |automático |rango de re... |todo

```

CINEMÁTICA DIRECTA POR SUMA VECTORIAL DEL ROBOT PARALELO 3RRR

CADENA I CINEMÁTICA DIRECTA

```

In[11]= ROP = Txyz[Xp, Yp, 0];
Ra1 = Qz[θa1].Tx[a1];
Rb1 = Qz[θa1].Qz[θb1].Tx[b1];
(*Que el vector Rc1 es un vector medido desde el sistema local del punto B
También es posible medir el ángulo desde el sistema inercial*)
Rc1 = Qz[φ + μ1].Tx[c1];

In[15]= (*POR SUMA DE VECTORES*)
ROA1 = ROP + Ra1;
ROA1.n // MatrixForm // Simplify;
           |forma de matriz |simplifica
ROB1 = ROP + Ra1 + Rb1;
ROB1.n // MatrixForm // Simplify;
           |forma de matriz |simplifica
ROC1 = ROP + Ra1 + Rb1 + Rc1;
ROC1.n // MatrixForm // Simplify
           |forma de matriz |simplifica

```

```

Out[20]/MatrixForm=

$$\begin{pmatrix} Xp + c1 \cos[\mu1 + \phi] - a1 \cos[\theta a11i[6]] + b1 \cos[\theta a11i[6] + \theta b11i[6]] \\ Yp - c1 \sin[\mu1 + \phi] + a1 \sin[\theta a11i[6]] + b1 \sin[\theta a11i[6] + \theta b11i[6]] \\ 0 \\ 4 \end{pmatrix}$$

CINMÁTICA INVERSA POR REDUCCION POLINOMIAL CADENA I
In[21]= (*Obtenemos la ecuación de posición del robot igualada a cero*)
POC = Txyz[Xc, Yc, Zc];
Ec = (ROP + Ra1 + Rb1 + Rc1 - POC) .n // Simplify;
|simplifica

Ec // MatrixForm // Simplify
|forma de matriz |simplifica

Out[23]/MatrixForm=

$$\begin{pmatrix} -Xc - Xp + c1 \cos[\mu1 + \phi] + a1 \cos[\theta a11i[6]] + b1 \cos[\theta a11i[6] + \theta b11i[6]] \\ -Yc - Yp + c1 \sin[\mu1 + \phi] - a1 \sin[\theta a11i[6]] + b1 \sin[\theta a11i[6] + \theta b11i[6]] \\ -Zc \\ 3 \end{pmatrix}$$

In[24]= (*Separ la ecuación general para elevar al cuadrado*)
EcuI = (ROP + Ra1 - POC + Rc1) .n;
EcuI // MatrixForm // Simplify
|forma de matriz |simplifica

EcuD = (Rb1) .n;
EcuD // MatrixForm // Simplify
|forma de matriz |simplifica

Out[25]/MatrixForm=

$$(-Xc - Xp + c1 \cos[\mu1 + \phi] - a1 \cos[\theta a11i[6]])$$

Out[25]/MatrixForm=

$$\begin{pmatrix} -Xc - Xp + c1 \cos[\mu1 + \phi] - a1 \cos[\theta a11i[6]] \\ -Yc - Yp + c1 \sin[\mu1 + \phi] + a1 \sin[\theta a11i[6]] \\ -Zc \\ 2 \end{pmatrix}$$

Out[27]/MatrixForm=

$$\begin{pmatrix} b1 \cos[\theta a11i[6] - \theta b11i[6]] \\ b1 \sin[\theta a11i[6] + \theta b11i[6]] \\ 0 \\ 1 \end{pmatrix}$$

In[28]= (*Elevamos al cuadrado ambos lados*)
(*EcI=EcD*)
EcI = EcuI[[1]]^2 + EcuI[[2]]^2 // Simplify
|simplifica

EcD = EcuD[[1]]^2 + EcuD[[2]]^2 // Simplify
|simplifica

Out[28]=

$$(-Xc + Xp + c1 \cos[\mu1 + \phi] + a1 \cos[\theta a11i[6]])^2 + (-Yc + Yp + c1 \sin[\mu1 + \phi] - a1 \sin[\theta a11i[6]])^2$$

Out[29]=

$$b1^2$$

In[30]= (*EcI-Ecd=0*)
In[31]= EI = Expand[EcI - EcD] // Simplify
|expande factores |simplifica

Out[31]=

$$a1^2 - b1^2 + c1^2 + Xc^2 - 2 Xc Xp + Xp^2 + Yc^2 - 2 Yc Yp + Yp^2 - 2 c1 (Xc - Xp) \cos[\mu1 + \phi] + 2 a1 c1 \cos[\mu1 + \phi - \theta a11i[6]] - 2 a1 Xc \cos[\theta a11i[6]] + 2 a1 Xp \cos[\theta a11i[6]] -$$


```

Apéndices

```

Out[31]=
a12 - b12 + c12 + Xc2 - 2 Xc Xp - Xp2 + Yc2 - 2 Yc Yp + Yp2 - 2 c1 (Xc - Xp) Cos[μ1 + φ] +
2 a1 c1 Cos[μ1 + φ - θa11i[6]] - 2 a1 Xc Cos[θa11i[6]] + 2 a1 Xp Cos[θa11i[6]] -
2 c1 Yc Sin[μ1 + φ] + 2 c1 Yp Sin[μ1 + φ] - 2 a1 Yc Sin[θa11i[6]] + 2 a1 Yp Sin[θa11i[6]]

In[32]= EcuA = EcI - EcD /. Sin[θa1] →  $\frac{2 \text{tal}}{1 + \text{tal}^2}$  /. Cos[θa1] →  $\frac{1 - \text{tal}^2}{1 + \text{tal}^2}$  // Simplify
[seno] [coseno] [simplifica]

Out[32]=
-b12 +  $\left(\frac{a1 - a1 \text{tal}^2}{1 + \text{tal}^2} - Xc + Xp + c1 \text{Cos}[\mu1 + \phi]\right)^2 + \left(\frac{2 a1 \text{tal}}{1 + \text{tal}^2} - Yc + Yp + c1 \text{Sin}[\mu1 + \phi]\right)^2$ 

In[33]= Solal = Solve[EcuA == 0, tal] // Simplify
[resuelve] [simplifica]

Out[33]=
{{tal →
-  $\left(\left(-2 a1 Yc + 2 a1 Yp + 2 a1 c1 \text{Sin}[\mu1 + \phi] + \frac{1}{2} \sqrt{(16 a1^2 (-Yc + Yp + c1 \text{Sin}[\mu1 + \phi])^2 - 4 (a1^2 - b1^2 + c1^2 + 2 a1 Xc + Xc^2 - 2 a1 Xp - 2 Xc Xp + Xp^2 + Yc^2 - 2 Yc Yp + Yp^2 - 2 c1 (a1 + Xc - Xp) \text{Cos}[\mu1 + \phi] - 2 c1 (Yc - Yp) \text{Sin}[\mu1 + \phi]) (a1^2 - b1^2 + c1^2 - 2 a1 Xc + Xc^2 + 2 a1 Xp - 2 Xc Xp + Xp^2 + Yc^2 - 2 Yc Yp + Yp^2 + 2 c1 (a1 - Xc + Xp) \text{Cos}[\mu1 + \phi] - 2 c1 (Yc - Yp) \text{Sin}[\mu1 + \phi])}\right)\right) /$ 
 $(a1^2 - b1^2 + c1^2 + 2 a1 Xc + Xc^2 - 2 a1 Xp - 2 Xc Xp + Xp^2 + Yc^2 - 2 Yc Yp + Yp^2 - 2 c1 (a1 + Xc - Xp) \text{Cos}[\mu1 + \phi] - 2 c1 (Yc - Yp) \text{Sin}[\mu1 + \phi])\right)}$ ,
{tal →  $2 a1 Yc - 2 a1 Yp - 2 a1 c1 \text{Sin}[\mu1 + \phi] + \frac{1}{2} \sqrt{(16 a1^2 (-Yc + Yp + c1 \text{Sin}[\mu1 + \phi])^2 - 4 (a1^2 - b1^2 + c1^2 + 2 a1 Xc + Xc^2 - 2 a1 Xp - 2 Xc Xp + Xp^2 + Yc^2 - 2 Yc Yp + Yp^2 - 2 c1 (a1 + Xc - Xp) \text{Cos}[\mu1 + \phi] - 2 c1 (Yc - Yp) \text{Sin}[\mu1 + \phi]) (a1^2 - b1^2 + c1^2 - 2 a1 Xc + Xc^2 + 2 a1 Xp - 2 Xc Xp + Xp^2 + Yc^2 - 2 Yc Yp + Yp^2 + 2 c1 (a1 - Xc + Xp) \text{Cos}[\mu1 + \phi] - 2 c1 (Yc - Yp) \text{Sin}[\mu1 + \phi])}\right)\right) /$ 
 $(a1^2 - b1^2 + c1^2 + 2 a1 Xc + Xc^2 - 2 a1 Xp - 2 Xc Xp + Xp^2 + Yc^2 - 2 Yc Yp + Yp^2 - 2 c1 (a1 + Xc - Xp) \text{Cos}[\mu1 + \phi] - 2 c1 (Yc - Yp) \text{Sin}[\mu1 + \phi])\right)}$ }}

In[34]= θa11 = 2 ArcTan[tal] /. Solal[[1]]
[arco tangente]

Out[34]=
-2 ArcTan  $\left[ \left( -2 a1 Yc + 2 a1 Yp + 2 a1 c1 \text{Sin}[\mu1 + \phi] + \frac{1}{2} \sqrt{(16 a1^2 (-Yc + Yp + c1 \text{Sin}[\mu1 + \phi])^2 - 4 (a1^2 - b1^2 + c1^2 + 2 a1 Xc + Xc^2 - 2 a1 Xp - 2 Xc Xp + Xp^2 + Yc^2 - 2 Yc Yp + Yp^2 - 2 c1 (a1 + Xc - Xp) \text{Cos}[\mu1 + \phi] - 2 c1 (Yc - Yp) \text{Sin}[\mu1 + \phi]) (a1^2 - b1^2 + c1^2 - 2 a1 Xc + Xc^2 + 2 a1 Xp - 2 Xc Xp + Xp^2 + Yc^2 - 2 Yc Yp + Yp^2 + 2 c1 (a1 - Xc + Xp) \text{Cos}[\mu1 + \phi] - 2 c1 (Yc - Yp) \text{Sin}[\mu1 + \phi])}\right)\right) /$ 
 $(a1^2 - b1^2 + c1^2 + 2 a1 Xc + Xc^2 - 2 a1 Xp - 2 Xc Xp + Xp^2 + Yc^2 - 2 Yc Yp + Yp^2 - 2 c1 (a1 + Xc - Xp) \text{Cos}[\mu1 + \phi] - 2 c1 (Yc - Yp) \text{Sin}[\mu1 + \phi])\right]$ 

In[35]= θa12 = 2 ArcTan[tal] /. Solal[[2]]
[arco tangente]

Out[35]=

```

Apéndices

Out[35]=

$$2 \operatorname{ArcTan} \left[\left(2 a_1 Y_c - 2 a_1 Y_p - 2 a_1 c_1 \sin[\mu_1 + \phi] + \frac{1}{2} \sqrt{16 a_1^2 (-Y_c + Y_p + c_1 \sin[\mu_1 + \phi])^2 - 4 (a_1^2 - b_1^2 + c_1^2 + 2 a_1 X_c + X_c^2 - 2 a_1 X_p - 2 X_c X_p - X_p^2 + Y_c^2 - 2 Y_c Y_p + Y_p^2 - 2 c_1 (a_1 + X_c - X_p) \cos[\mu_1 + \phi] - 2 c_1 (Y_c - Y_p) \sin[\mu_1 + \phi]) (a_1^2 - b_1^2 + c_1^2 - 2 a_1 X_c + X_c^2 + 2 a_1 X_p - 2 X_c X_p + X_p^2 + Y_c^2 - 2 Y_c Y_p + Y_p^2 + 2 c_1 (a_1 - X_c + X_p) \cos[\mu_1 + \phi] - 2 c_1 (Y_c - Y_p) \sin[\mu_1 + \phi])} \right) \right] / \left(a_1^2 - b_1^2 + c_1^2 + 2 a_1 X_c + X_c^2 - 2 a_1 X_p - 2 X_c X_p - X_p^2 + Y_c^2 - 2 Y_c Y_p + Y_p^2 - 2 c_1 (a_1 + X_c - X_p) \cos[\mu_1 + \phi] - 2 c_1 (Y_c - Y_p) \sin[\mu_1 + \phi] \right)$$

In[36]= **Ec[1]**

Out[36]=

$$-X_c + X_p + c_1 \cos[\mu_1 + \phi] + a_1 \cos[\theta_{a11i}[6]] + b_1 \cos[\theta_{a11i}[6] + \theta_{b11i}[6]]$$

In[37]= (*Para encontrar el ángulo θ_{b1} *)

Csa1b1 = Solve[Ec[1] == 0, Cos[$\theta_{a1} + \theta_{b1}$]] // Flatten
|resuelve |coseno |aplana

Sna1b1 = Solve[Ec[2] == 0, Sin[$\theta_{a1} + \theta_{b1}$]] // Flatten
|resuelve |seno |aplana

Out[37]=

$$\left\{ \cos[\theta_{a11i}[6] + \theta_{b11i}[6]] \rightarrow \frac{X_c - X_p - c_1 \cos[\mu_1 + \phi] - a_1 \cos[\theta_{a11i}[6]]}{b_1} \right\}$$

Out[38]=

$$\left\{ \sin[\theta_{a11i}[6] + \theta_{b11i}[6]] \rightarrow \frac{Y_c - Y_p - c_1 \sin[\mu_1 + \phi] - a_1 \sin[\theta_{a11i}[6]]}{b_1} \right\}$$

In[39]= **$\theta_{b11} = (\operatorname{ArcTan}[\cos[\theta_{a1} + \theta_{b1}] /. Csa1b1, \sin[\theta_{a1} + \theta_{b1}] /. Sna1b1]) - \theta_{a11}$**

Apéndices

```

θb12 = (ArcTan[Cos[θa1 + θb1] /. Csa1b1, Sin[θa1 + θb1] /. Sna1b1]) - θa12
      [arco t... [coseno]          [seno]
Out[39]=
2 ArcTan[
  ( -2 a1 Yc - 2 a1 Yp + 2 a1 c1 Sin[μ1 + φ] +  $\frac{1}{2} \sqrt{(16 a1^2 (-Yc + Yp + c1 Sin[\mu1 + \phi])^2 - 4 (a1^2 - b1^2 + c1^2 + 2 a1 Xc + Xc^2 - 2 a1 Xp - 2 Xc Xp + Xp^2 + Yc^2 - 2 Yc Yp + Yp^2 - 2 c1 (a1 + Xc - Xp) Cos[\mu1 + \phi] - 2 c1 (Yc - Yp) Sin[\mu1 - \phi]) (a1^2 - b1^2 + c1^2 - 2 a1 Xc + Xc^2 + 2 a1 Xp - 2 Xc Xp + Xp^2 + Yc^2 - 2 Yc Yp + Yp^2 + 2 c1 (a1 - Xc - Xp) Cos[\mu1 + \phi] - 2 c1 (Yc - Yp) Sin[\mu1 + \phi])}$  ) /
  ( a1^2 - b1^2 + c1^2 + 2 a1 Xc + Xc^2 - 2 a1 Xp - 2 Xc Xp + Xp^2 + Yc^2 - 2 Yc Yp + Yp^2 - 2 c1 (a1 + Xc - Xp) Cos[μ1 + φ] - 2 c1 (Yc - Yp) Sin[μ1 + φ] ) ) +
  ArcTan[  $\frac{Xc - Xp - c1 Cos[\mu1 + \phi] - a1 Cos[\theta a11i[6]]}{b1}$  ,
   $\frac{Yc - Yp - c1 Sin[\mu1 + \phi] - a1 Sin[\theta a11i[6]]}{b1}$  ]
Out[40]=
-2 ArcTan[
  ( 2 a1 Yc - 2 a1 Yp - 2 a1 c1 Sin[μ1 + φ] +  $\frac{1}{2} \sqrt{(16 a1^2 (-Yc + Yp + c1 Sin[\mu1 + \phi])^2 - 4 (a1^2 - b1^2 + c1^2 + 2 a1 Xc + Xc^2 - 2 a1 Xp - 2 Xc Xp + Xp^2 + Yc^2 - 2 Yc Yp + Yp^2 - 2 c1 (a1 + Xc - Xp) Cos[\mu1 + \phi] - 2 c1 (Yc - Yp) Sin[\mu1 - \phi]) (a1^2 - b1^2 + c1^2 - 2 a1 Xc + Xc^2 + 2 a1 Xp - 2 Xc Xp + Xp^2 + Yc^2 - 2 Yc Yp + Yp^2 + 2 c1 (a1 - Xc - Xp) Cos[\mu1 + \phi] - 2 c1 (Yc - Yp) Sin[\mu1 + \phi])}$  ) /
  ( a1^2 - b1^2 + c1^2 + 2 a1 Xc + Xc^2 - 2 a1 Xp - 2 Xc Xp + Xp^2 + Yc^2 - 2 Yc Yp + Yp^2 - 2 c1 (a1 + Xc - Xp) Cos[μ1 + φ] - 2 c1 (Yc - Yp) Sin[μ1 + φ] ) ) +
  ArcTan[  $\frac{Xc - Xp - c1 Cos[\mu1 + \phi] - a1 Cos[\theta a11i[6]]}{b1}$  ,
   $\frac{Yc - Yp - c1 Sin[\mu1 + \phi] - a1 Sin[\theta a11i[6]]}{b1}$  ]

```

CADENA 2

```

In[41]= R0Q = Txyz[Xq, Yq, 0];
Ra2 = Qz[θa2].Tx[a2];
Rb2 = Qz[θa2].Qz[θb2].Tx[b2];
(*Que el vector Rc1 es un vector medido desde el sistema local del punto B
También es posible medir el ángulo desde el sistema inercial*)
Rc2 = Qz[φ + μ2].Tx[c2];
In[45]= (*POR SUMA DE VECTORES*)
R0A2 = R0Q + Ra2;
R0A2.n // MatrixForm // Simplify;
      [forma de matriz [simplifica]
R0B2 = R0Q + Ra2 + Rb2;
R0B2.n // MatrixForm // Simplify;
      [forma de matriz [simplifica]
R0C2 = R0Q + Ra2 + Rb2 + Rc2;
R0C2.n // MatrixForm // Simplify;
      [forma de matriz [simplifica]

```

```

CINEMÁTICA INVERSA POR REDUCCION POLINOMIAL CADENA2
In[52]= (*Obtenemos la ecuación de posición del robot igualada a cero*)
      POC = Txyz[Xc, Yc, Zc];
      Ec = (R0Q + Ra2 + Rb2 + Rc2 - POC) . n // Simplify;
           [simplifica]

      Ec // MatrixForm // Simplify
           [forma de matriz [simplifica]

Out[54]/MatrixForm=
      (-Xc - Xq + c2 Cos[μ2 + φ] - a2 Cos[θa21i[6]] + b2 Cos[θa21i[6] + θb21i[6]]
      -Yc - Yq + c2 Sin[μ2 + φ] + a2 Sin[θa21i[6]] + b2 Sin[θa21i[6] + θb21i[6]]
      -Zc
      3

In[55]= (*Separ la ecuación general para elevar al cuadrado*)
      EcuI = (R0Q + Ra2 - POC + Rc2) . n;
      EcuI // MatrixForm // Simplify
           [forma de matriz [simplifica]

      EcuD = (Rb2) . n;
      EcuD // MatrixForm // Simplify
           [forma de matriz [simplifica]

Out[56]/MatrixForm=
      (-Xc - Xq + c2 Cos[μ2 + φ] + a2 Cos[θa21i[6]]
      -Yc - Yq + c2 Sin[μ2 + φ] + a2 Sin[θa21i[6]]
      -Zc
      2

Out[58]/MatrixForm=
      (b2 Cos[θa21i[6] + θb21i[6]])

Out[58]/MatrixForm=
      (b2 Cos[θa21i[6] + θb21i[6]]
      b2 Sin[θa21i[6] + θb21i[6]]
      0
      1

In[59]= (*Elevamos al cuadrado ambos lados*)
      (*EcuI=EcuD*)
      EcuI = EcuI[[1]]^2 + EcuI[[2]]^2 // Simplify
           [simplifica]

      EcuD = EcuD[[1]]^2 + EcuD[[2]]^2 // Simplify
           [simplifica]

Out[59]=
      (-Xc + Xq + c2 Cos[μ2 + φ] + a2 Cos[θa21i[6]])^2 + (-Yc + Yq + c2 Sin[μ2 + φ] + a2 Sin[θa21i[6]])^2

Out[60]=
      b2^2

In[61]= EI = Expand[EcuI - EcuD] // Simplify
           [expande factores [simplifica]

Out[61]=
      a2^2 - b2^2 + c2^2 + Xc^2 - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 - 2 c2 (Xc - Xq) Cos[μ2 + φ] +
      2 a2 c2 Cos[μ2 + φ - θa21i[6]] - 2 a2 Xc Cos[θa21i[6]] + 2 a2 Xq Cos[θa21i[6]] -
      2 c2 Yc Sin[μ2 + φ] + 2 c2 Yq Sin[μ2 + φ] - 2 a2 Yc Sin[θa21i[6]] + 2 a2 Yq Sin[θa21i[6]]

In[62]= EcuA = EcuI - EcuD /. Sin[θa2] →  $\frac{2 \tan \theta}{1 + \tan^2 \theta}$  /. Cos[θa2] →  $\frac{1 - \tan^2 \theta}{1 + \tan^2 \theta}$  // Simplify
           [seno [coseno [simplifica]

Out[62]=

```

Apéndices

```

-2 ArcTan[[-2 a2 Yc - 2 a2 Yq + 2 a2 c2 Sin[μ2 + φ] +
  1/2 √(16 a2^2 (-Yc + Yq + c2 Sin[μ2 + φ])^2 - 4 (a2^2 - b2^2 + c2^2 + 2 a2 Xc + Xc^2 - 2 a2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq +
    Yq^2 - 2 c2 (a2 + Xc - Xq) Cos[μ2 + φ] - 2 c2 (Yc - Yq) Sin[μ2 + φ]) (a2^2 - b2^2 + c2^2 - 2 a2 Xc + Xc^2 + 2 a2 Xq -
    2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 + 2 c2 (a2 - Xc + Xq) Cos[μ2 + φ] - 2 c2 (Yc - Yq) Sin[μ2 + φ])] /
  (a2^2 - b2^2 + c2^2 + 2 a2 Xc + Xc^2 - 2 a2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 - 2 c2 (a2 + Xc - Xq) Cos[μ2 + φ] -
    2 c2 (Yc - Yq) Sin[μ2 + φ])]
In[65]= θa22 = 2 ArcTan[ta2] /. Sola2[[2]]
      |arco tangente
Out[65]=
2 ArcTan[ (2 a2 Yc - 2 a2 Yq - 2 a2 c2 Sin[μ2 + φ] +
  1/2 √(16 a2^2 (-Yc + Yq + c2 Sin[μ2 + φ])^2 - 4 (a2^2 - b2^2 + c2^2 + 2 a2 Xc + Xc^2 - 2 a2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq +
    Yq^2 - 2 c2 (a2 + Xc - Xq) Cos[μ2 + φ] - 2 c2 (Yc - Yq) Sin[μ2 + φ]) (a2^2 - b2^2 + c2^2 - 2 a2 Xc + Xc^2 + 2 a2 Xq -
    2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 + 2 c2 (a2 - Xc + Xq) Cos[μ2 + φ] - 2 c2 (Yc - Yq) Sin[μ2 + φ])) /
  (a2^2 - b2^2 + c2^2 + 2 a2 Xc + Xc^2 - 2 a2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 - 2 c2 (a2 + Xc - Xq) Cos[μ2 + φ] -
    2 c2 (Yc - Yq) Sin[μ2 + φ])]
In[66]= (*Para encontrar el ángulo θb1*)
Csa2b2 = Solve[Ec[[1]] == 0, Cos[θa2 + θb2]] // Flatten
      |resuelve |coseno |aplana
Sna2b2 = Solve[Ec[[2]] == 0, Sin[θa2 + θb2]] // Flatten
      |resuelve |seno |aplana
-b2^2 + (a2 - a2 ta2^2 / (1 + ta2^2) - Xc + Xq + c2 Cos[μ2 + φ])^2 + (2 a2 ta2 / (1 + ta2^2) - Yc + Yq + c2 Sin[μ2 + φ])^2
In[63]= Sola2 = Solve[Ecua == 0, ta2] // Simplify
      |resuelve |simplifica
Out[63]=
{{ta2 →
  - ( (-2 a2 Yc + 2 a2 Yq + 2 a2 c2 Sin[μ2 + φ] + 1/2 √(16 a2^2 (-Yc + Yq + c2 Sin[μ2 + φ])^2 - 4 (a2^2 - b2^2 + c2^2 + 2 a2 Xc +
    Xc^2 - 2 a2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 - 2 c2 (a2 + Xc - Xq) Cos[μ2 + φ] - 2 c2
    (Yc - Yq) Sin[μ2 + φ]) (a2^2 - b2^2 + c2^2 - 2 a2 Xc + Xc^2 + 2 a2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2
    Yc Yq + Yq^2 + 2 c2 (a2 - Xc + Xq) Cos[μ2 + φ] - 2 c2 (Yc - Yq) Sin[μ2 + φ])) /
    (a2^2 - b2^2 + c2^2 + 2 a2 Xc + Xc^2 - 2 a2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 -
    2 c2 (a2 + Xc - Xq) Cos[μ2 + φ] - 2 c2 (Yc - Yq) Sin[μ2 + φ]))},
  {ta2 → (2 a2 Yc - 2 a2 Yq - 2 a2 c2 Sin[μ2 + φ] + 1/2 √(16 a2^2 (-Yc + Yq + c2 Sin[μ2 + φ])^2 -
    4 (a2^2 - b2^2 + c2^2 + 2 a2 Xc + Xc^2 - 2 a2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 -
    2 c2 (a2 + Xc - Xq) Cos[μ2 + φ] - 2 c2 (Yc - Yq) Sin[μ2 + φ]) (a2^2 - b2^2 + c2^2 - 2 a2 Xc + Xc^2 + 2 a2 Xq -
    2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 + 2 c2 (a2 - Xc + Xq) Cos[μ2 + φ] - 2 c2 (Yc - Yq) Sin[μ2 + φ])) /
    (a2^2 - b2^2 + c2^2 + 2 a2 Xc + Xc^2 - 2 a2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 - 2 c2 (a2 + Xc - Xq) Cos[μ2 + φ] -
    2 c2 (Yc - Yq) Sin[μ2 + φ]))}
In[64]= θa21 = 2 ArcTan[ta2] /. Sola2[[1]]

```

Apéndices

```

-2 ArcTan[[-2 a2 Yc - 2 a2 Yq + 2 a2 c2 Sin[μ2 + φ] +
  1/2 √(16 a2^2 (-Yc + Yq + c2 Sin[μ2 + φ])^2 - 4 (a2^2 - b2^2 + c2^2 + 2 a2 Xc + Xc^2 - 2 a2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq +
    Yq^2 - 2 c2 (a2 + Xc - Xq) Cos[μ2 + φ] - 2 c2 (Yc - Yq) Sin[μ2 + φ]) (a2^2 - b2^2 + c2^2 - 2 a2 Xc + Xc^2 + 2 a2 Xq -
    2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 + 2 c2 (a2 - Xc + Xq) Cos[μ2 + φ] - 2 c2 (Yc - Yq) Sin[μ2 + φ])] /
(a2^2 - b2^2 + c2^2 + 2 a2 Xc + Xc^2 - 2 a2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 - 2 c2 (a2 + Xc - Xq) Cos[μ2 + φ] -
  2 c2 (Yc - Yq) Sin[μ2 + φ])]
In[65]= θa22 = 2 ArcTan[ta2] /. Sola2[[2]]
          |arco tangente
Out[65]= 2 ArcTan[ (2 a2 Yc - 2 a2 Yq - 2 a2 c2 Sin[μ2 + φ] +
  1/2 √(16 a2^2 (-Yc + Yq + c2 Sin[μ2 + φ])^2 - 4 (a2^2 - b2^2 + c2^2 + 2 a2 Xc + Xc^2 - 2 a2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq +
    Yq^2 - 2 c2 (a2 + Xc - Xq) Cos[μ2 + φ] - 2 c2 (Yc - Yq) Sin[μ2 + φ]) (a2^2 - b2^2 + c2^2 - 2 a2 Xc + Xc^2 + 2 a2 Xq -
    2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 + 2 c2 (a2 - Xc + Xq) Cos[μ2 + φ] - 2 c2 (Yc - Yq) Sin[μ2 + φ])) /
(a2^2 - b2^2 + c2^2 + 2 a2 Xc + Xc^2 - 2 a2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 - 2 c2 (a2 + Xc - Xq) Cos[μ2 + φ] -
  2 c2 (Yc - Yq) Sin[μ2 + φ])]
In[66]= (*Para encontrar el ángulo θb1*)
Csa2b2 = Solve[Ec[[1]] == 0, Cos[θa2 + θb2]] // Flatten
          |resuelve |coseno |aplana
Out[66]= Csa2b2 = Solve[Ec[[1]] == 0, Sin[θa2 + θb2]] // Flatten
          |resuelve |seno |aplana
          {Cos[θa21i[[6]] + θb21i[[6]]] -> (Xc - Xq - c2 Cos[μ2 + φ] - a2 Cos[θa21i[[6]]) / b2}
Out[67]= {Sin[θa21i[[6]] + θb21i[[6]]] -> (Yc - Yq - c2 Sin[μ2 + φ] - a2 Sin[θa21i[[6]]) / b2}
In[68]= θb21 = (ArcTan[Cos[θa2 + θb2] /. Csa2b2, Sin[θa2 + θb2] /. Sna2b2]) - θa21
          |arco t... |coseno |seno
          θb22 = (ArcTan[Cos[θa2 + θb2] /. Csa2b2, Sin[θa2 + θb2] /. Sna2b2]) - θa22
          |arco t... |coseno |seno
Out[68]= 2 ArcTan[
  (-2 a2 Yc + 2 a2 Yq + 2 a2 c2 Sin[μ2 + φ] + 1/2 √(16 a2^2 (-Yc + Yq + c2 Sin[μ2 + φ])^2 - 4 (a2^2 - b2^2 + c2^2 + 2 a2 Xc + Xc^2 -
    2 a2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 - 2 c2 (a2 + Xc - Xq) Cos[μ2 + φ] - 2 c2 (Yc - Yq) Sin[μ2 + φ])
    (a2^2 - b2^2 + c2^2 - 2 a2 Xc + Xc^2 + 2 a2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 +
    2 c2 (a2 - Xc + Xq) Cos[μ2 + φ] - 2 c2 (Yc - Yq) Sin[μ2 + φ])) /
(a2^2 - b2^2 + c2^2 + 2 a2 Xc + Xc^2 - 2 a2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 - 2 c2 (a2 + Xc - Xq) Cos[μ2 + φ] -
  2 c2 (Yc - Yq) Sin[μ2 + φ])] +
  ArcTan[ (Xc - Xq - c2 Cos[μ2 + φ] - a2 Cos[θa21i[[6]]) / b2,
    (Yc - Yq - c2 Sin[μ2 + φ] - a2 Sin[θa21i[[6]]) / b2 ]

```

Out[69]=

$$-2 \operatorname{ArcTan} \left[\frac{\left(2 a^2 Yc - 2 a^2 Yq - 2 a^2 c^2 \sin[\mu 2 + \phi] + \frac{1}{2} \sqrt{\left(16 a^2^2 (-Yc + Yq + c^2 \sin[\mu 2 + \phi])^2 - 4 (a^2^2 - b^2^2 + c^2^2 + 2 a^2 Xc + Xc^2 - 2 a^2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 - 2 c^2 (a^2 + Xc - Xq) \cos[\mu 2 + \phi] - 2 c^2 (Yc - Yq) \sin[\mu 2 + \phi] \right) (a^2^2 - b^2^2 + c^2^2 - 2 a^2 Xc + Xc^2 + 2 a^2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 + 2 c^2 (a^2 - Xc - Xq) \cos[\mu 2 + \phi] - 2 c^2 (Yc - Yq) \sin[\mu 2 + \phi]) \right)}{\left(a^2^2 - b^2^2 + c^2^2 + 2 a^2 Xc + Xc^2 - 2 a^2 Xq - 2 Xc Xq + Xq^2 + Yc^2 - 2 Yc Yq + Yq^2 - 2 c^2 (a^2 + Xc - Xq) \cos[\mu 2 + \phi] - 2 c^2 (Yc - Yq) \sin[\mu 2 + \phi] \right) + \operatorname{ArcTan} \left[\frac{Xc - Xq - c^2 \cos[\mu 2 + \phi] - a^2 \cos[\theta a 21 i [6]]}{b^2}, \frac{Yc - Yq - c^2 \sin[\mu 2 + \phi] - a^2 \sin[\theta a 21 i [6]]}{b^2} \right] \right]$$

CADENA 3

```
In[70]= R0R = Txyz[Xr, Yr, 0];
Ra3 = Qz[θa3].Tx[a3];
Rb3 = Qz[θa3].Qz[θb3].Tx[b3];
(*Que el vector Rcl es un vector medido desde el sistema local del punto B
También es posible medir el ángulo desde el sistema inercial*)
Rc3 = Qz[φ + μ3].Tx[c3];
In[74]= (*POR SUMA DE VECTORES*)
R0A3 = R0R + Ra3;
R0A3.n // MatrixForm // Simplify;
[forma de matriz |simplifica]
R0B3 = R0R + Ra3 + Rb3;
R0B3.n // MatrixForm // Simplify;
[forma de matriz |simplifica]
R0C3 = R0R + Ra3 + Rb3 + Rc3;
R0C3.n // MatrixForm // Simplify
[forma de matriz |simplifica]
```

Out[79]//MatrixForm=

$$\begin{pmatrix} Xr + c3 \cos[\mu 3 + \phi] + a3 \cos[\theta a 31 i [6]] + b3 \cos[\theta a 31 i [6] + \theta b 31 i [6]] \\ Yr + c3 \sin[\mu 3 + \phi] + a3 \sin[\theta a 31 i [6]] + b3 \sin[\theta a 31 i [6] + \theta b 31 i [6]] \\ 0 \\ 4 \end{pmatrix}$$

CINMÁTICA INVERSA POR REDUCCION POLINOMIAL CADENA3

```
In[80]= (*Obtenemos la ecuación de posición del robot igualada a cero*)
POC = Txyz[Xc, Yc, Zc];
Ec = (R0R + Ra3 + Rb3 + Rc3 - POC).n // Simplify;
[simplifica]
Ec // MatrixForm // Simplify
[forma de matriz |simplifica]
```

Out[82]//MatrixForm=

$$\begin{pmatrix} -Xc - Xr + c3 \cos[\mu 3 + \phi] + a3 \cos[\theta a 31 i [6]] + b3 \cos[\theta a 31 i [6] + \theta b 31 i [6]] \\ -Yc - Yr + c3 \sin[\mu 3 + \phi] + a3 \sin[\theta a 31 i [6]] + b3 \sin[\theta a 31 i [6] + \theta b 31 i [6]] \\ -Zc \\ 3 \end{pmatrix}$$

```
In[83]= (*Separ la ecuación general para elevar al cuadrado*)
```

Apéndices

```

EcuI = (R0R + Ra3 - POC + Rc3) .n;
EcuI // MatrixForm // Simplify
|forma de matriz |simplifica

EcuD = (Rb3) .n;
EcuD // MatrixForm // Simplify
|forma de matriz |simplifica

Out[84]/MatrixForm=

$$\begin{pmatrix} -Xc + Xr + c3 \cos[\mu3 + \phi] + a3 \cos[\theta a31i[6]] \\ -Yc + Yr + c3 \sin[\mu3 + \phi] + a3 \sin[\theta a31i[6]] \\ -Zc \\ 2 \end{pmatrix}$$


Out[86]/MatrixForm=

$$\begin{pmatrix} b3 \cos[\theta a31i[6] + \theta b31i[6]] \\ b3 \sin[\theta a31i[6] + \theta b31i[6]] \\ 0 \\ 1 \end{pmatrix}$$


In[87]= (*Elevamos al cuadrado ambos lados*)
(*EcuI=EcuD*)
EcI = EcuI[[1]]^2 + EcuI[[2]]^2 // Simplify
|simplifica

EcD = EcuD[[1]]^2 + EcuD[[2]]^2 // Simplify
|simplifica

Out[87]=

$$(-Xc + Xr + c3 \cos[\mu3 + \phi] + a3 \cos[\theta a31i[6]])^2 + (-Yc + Yr + c3 \sin[\mu3 + \phi] + a3 \sin[\theta a31i[6]])^2$$


Out[88]=

$$b3^2$$


In[89]= EI = Expand[EcuI - EcD] // Simplify
|expande factores |simplifica

Out[89]=

$$a3^2 - b3^2 + c3^2 + Xc^2 - 2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr + Yr^2 - 2 c3 (Xc - Xr) \cos[\mu3 + \phi] + 2 a3 c3 \cos[\mu3 + \phi - \theta a31i[6]] - 2 a3 Xc \cos[\theta a31i[6]] + 2 a3 Xr \cos[\theta a31i[6]] - 2 c3 Yc \sin[\mu3 + \phi] + 2 c3 Yr \sin[\mu3 + \phi] - 2 a3 Yc \sin[\theta a31i[6]] + 2 a3 Yr \sin[\theta a31i[6]]$$


In[90]= EcuA = EcI - EcD /. Sin[theta3] ->  $\frac{2 ta3}{1 + ta3^2}$  /. Cos[theta3] ->  $\frac{1 - ta3^2}{1 + ta3^2}$  // Simplify
|seno |coseno |simplifica

Out[90]=

$$-b3^2 + \left( \frac{a3 - a3 ta3^2}{1 + ta3^2} - Xc + Xr + c3 \cos[\mu3 + \phi] \right)^2 + \left( \frac{2 a3 ta3}{1 + ta3^2} - Yc + Yr + c3 \sin[\mu3 + \phi] \right)^2$$


In[91]= Sola3 = Solve[EcuA == 0, ta3] // Simplify
|resuelve |simplifica

Out[91]=

$$\left\{ \left\{ ta3 \rightarrow \right. \right. \\ \left. \left. - \left( \left( -2 a3 Yc + 2 a3 Yr + 2 a3 c3 \sin[\mu3 + \phi] + \frac{1}{2} \sqrt{16 a3^2 (-Yc + Yr + c3 \sin[\mu3 + \phi])^2 - 4 (a3^2 - b3^2 + c3^2 + 2 a3 Xc + Xc^2 - 2 a3 Xr - 2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr + Yr^2 - 2 c3 (a3 + Xc - Xr) \cos[\mu3 + \phi] - 2 c3 (Yc - Yr) \sin[\mu3 + \phi]) (a3^2 - b3^2 + c3^2 - 2 a3 Xc + Xc^2 + 2 a3 Xr - 2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr + Yr^2 + 2 c3 (a3 - Xc + Xr) \cos[\mu3 + \phi] - 2 c3 (Yc - Yr) \sin[\mu3 + \phi])} \right) \right) \right. \\ \left. (a3^2 - b3^2 + c3^2 + 2 a3 Xc + Xc^2 - 2 a3 Xr - 2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr + Yr^2 - 2 c3 (a3 + Xc - Xr) \cos[\mu3 + \phi] - 2 c3 (Yc - Yr) \sin[\mu3 + \phi]) \right\} \right\},$$


```

Apéndices

```

{ta3 -> (2 a3 Yc - 2 a3 Yr - 2 a3 c3 Sin[mu3 + phi] + 1/2 sqrt(16 a3^2 (-Yc + Yr + c3 Sin[mu3 + phi])^2 -
4 (a3^2 - b3^2 + c3^2 + 2 a3 Xc - Xc^2 - 2 a3 Xr - 2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr + Yr^2 -
2 c3 (a3 + Xc - Xr) Cos[mu3 + phi] - 2 c3 (Yc - Yr) Sin[mu3 + phi]) (a3^2 - b3^2 + c3^2 - 2 a3 Xc + Xc^2 + 2 a3 Xr -
2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr + Yr^2 + 2 c3 (a3 - Xc + Xr) Cos[mu3 + phi] - 2 c3 (Yc - Yr) Sin[mu3 + phi])))/
(a3^2 - b3^2 + c3^2 + 2 a3 Xc + Xc^2 - 2 a3 Xr - 2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr + Yr^2 - 2 c3 (a3 + Xc - Xr) Cos[mu3 + phi] -
2 c3 (Yc - Yr) Sin[mu3 + phi]))}
In[92]= theta31 = 2 ArcTan[ta3] /. Sola3[1]
          |arco tangente
Out[92]=
-2 ArcTan[(-2 a3 Yc + 2 a3 Yr + 2 a3 c3 Sin[mu3 + phi] +
1/2 sqrt(16 a3^2 (-Yc + Yr + c3 Sin[mu3 + phi])^2 - 4 (a3^2 - b3^2 + c3^2 + 2 a3 Xc + Xc^2 - 2 a3 Xr - 2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr +
Yr^2 - 2 c3 (a3 + Xc - Xr) Cos[mu3 + phi] - 2 c3 (Yc - Yr) Sin[mu3 + phi]) (a3^2 - b3^2 + c3^2 - 2 a3 Xc + Xc^2 + 2 a3 Xr -
2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr + Yr^2 + 2 c3 (a3 - Xc + Xr) Cos[mu3 + phi] - 2 c3 (Yc - Yr) Sin[mu3 + phi])))/
(a3^2 - b3^2 + c3^2 + 2 a3 Xc + Xc^2 - 2 a3 Xr - 2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr + Yr^2 - 2 c3 (a3 + Xc - Xr) Cos[mu3 + phi] -
2 c3 (Yc - Yr) Sin[mu3 + phi]))]
In[93]= theta32 = 2 ArcTan[ta3] /. Sola3[2]
          |arco tangente
Out[93]=
2 ArcTan[(2 a3 Yc - 2 a3 Yr - 2 a3 c3 Sin[mu3 + phi] +
1/2 sqrt(16 a3^2 (-Yc + Yr + c3 Sin[mu3 + phi])^2 - 4 (a3^2 - b3^2 + c3^2 + 2 a3 Xc + Xc^2 - 2 a3 Xr - 2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr +
Yr^2 - 2 c3 (a3 + Xc - Xr) Cos[mu3 + phi] - 2 c3 (Yc - Yr) Sin[mu3 + phi]) (a3^2 - b3^2 + c3^2 - 2 a3 Xc + Xc^2 + 2 a3 Xr -
2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr + Yr^2 + 2 c3 (a3 - Xc + Xr) Cos[mu3 + phi] - 2 c3 (Yc - Yr) Sin[mu3 + phi])))/
(a3^2 - b3^2 + c3^2 + 2 a3 Xc + Xc^2 - 2 a3 Xr - 2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr + Yr^2 - 2 c3 (a3 + Xc - Xr) Cos[mu3 + phi] -
2 c3 (Yc - Yr) Sin[mu3 + phi]))]
In[94]= (*Para encontrar el ángulo theta1*)
Csa3b3 = Solve[Ec[1] == 0, Cos[theta3 + theta3]] // Flatten
          |resuelve |coseno |aplana
Sna3b3 = Solve[Ec[2] == 0, Sin[theta3 + theta3]] // Flatten
          |resuelve |seno |aplana
Out[94]=
{Cos[theta31i[6]] + theta31i[6] -> (Xc - Xr - c3 Cos[mu3 + phi] - a3 Cos[theta31i[6]])/b3}
Out[95]=
{Sin[theta31i[6]] + theta31i[6] -> (Yc - Yr - c3 Sin[mu3 + phi] - a3 Sin[theta31i[6]])/b3}
In[96]= theta31 = (ArcTan[Cos[theta3 + theta3] /. Csa3b3, Sin[theta3 + theta3] /. Sna3b3]) - theta31
          |arco t... |coseno |seno
theta32 = (ArcTan[Cos[theta3 + theta3] /. Csa3b3, Sin[theta3 + theta3] /. Sna3b3]) - theta32
          |arco t... |coseno |seno
Out[96]=
2 ArcTan[

```

Apéndices

```

Out[97]=
ArcTan[
  (-2 a3 Yc + 2 a3 Yr + 2 a3 c3 Sin[μ3 + φ] + 1/2 √(16 a3^2 (-Yc + Yr + c3 Sin[μ3 + φ])^2 - 4 (a3^2 - b3^2 + c3^2 + 2 a3 Xc - Xc^2 -
    2 a3 Xr - 2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr + Yr^2 - 2 c3 (a3 + Xc - Xr) Cos[μ3 + φ] - 2 c3 (Yc - Yr) Sin[μ3 + φ])
    (a3^2 - b3^2 + c3^2 - 2 a3 Xc + Xc^2 + 2 a3 Xr - 2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr + Yr^2 +
    2 c3 (a3 - Xc - Xr) Cos[μ3 + φ] - 2 c3 (Yc - Yr) Sin[μ3 + φ])))/
  (a3^2 - b3^2 + c3^2 + 2 a3 Xc + Xc^2 - 2 a3 Xr - 2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr + Yr^2 - 2 c3 (a3 + Xc - Xr) Cos[μ3 + φ] -
    2 c3 (Yc - Yr) Sin[μ3 + φ]) +
  ArcTan[
    (Xc - Xr - c3 Cos[μ3 + φ] - a3 Cos[θa31i[6]])/b3,
    (Yc - Yr - c3 Sin[μ3 + φ] - a3 Sin[θa31i[6]])/b3 ]
]
In[99]=
-2 ArcTan[
  (2 a3 Yc - 2 a3 Yr - 2 a3 c3 Sin[μ3 + φ] + 1/2 √(16 a3^2 (-Yc + Yr + c3 Sin[μ3 + φ])^2 - 4 (a3^2 - b3^2 + c3^2 + 2 a3 Xc + Xc^2 -
    2 a3 Xr - 2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr + Yr^2 - 2 c3 (a3 + Xc - Xr) Cos[μ3 + φ] - 2 c3 (Yc - Yr) Sin[μ3 + φ])
    (a3^2 - b3^2 + c3^2 - 2 a3 Xc + Xc^2 + 2 a3 Xr - 2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr + Yr^2 +
    2 c3 (a3 - Xc + Xr) Cos[μ3 + φ] - 2 c3 (Yc - Yr) Sin[μ3 + φ])))/
  (a3^2 - b3^2 + c3^2 + 2 a3 Xc + Xc^2 - 2 a3 Xr - 2 Xc Xr + Xr^2 + Yc^2 - 2 Yc Yr + Yr^2 - 2 c3 (a3 + Xc - Xr) Cos[μ3 + φ] -
    2 c3 (Yc - Yr) Sin[μ3 + φ]) +
  ArcTan[
    (Xc - Xr - c3 Cos[μ3 + φ] - a3 Cos[θa31i[6]])/b3,
    (Yc - Yr - c3 Sin[μ3 + φ] - a3 Sin[θa31i[6]])/b3 ]
]
In[100]=
Cadena 4 serial
R0S = Txyz[Xc, Yc, 0];
Ra4 = QY[θa4].Tx[a4];
Rb4 = QY[θa4].QY[θb4].Tx[b4];
Rc4 = QY[θa4].QY[θb4].QY[θc4].Tx[c4];
Rd4 = QY[θa4].QY[θb4].QY[θc4].QX[θd4].Tx[d4];
ROA4 = R0S + Ra4;
ROB4 = R0S + Ra4 + Rb4;
ROC4 = R0S + Ra4 + Rb4 + Rc4;
ROD4 = R0S + Ra4 + Rb4 + Rc4 + Rd4;

ROD4.n // MatrixForm // FullSimplify
[forma de matriz |simplifica completamente]
Out[109]/MatrixForm=
(
  Xc + a4 Cos[θa41i[6]] + b4 Cos[θa41i[6]] + θb41i[6] + (c4 + d4) Cos[θc4 + θa41i[6]] + θb41i[6]
  Yc
  -a4 Sin[θa41i[6]] - b4 Sin[θa41i[6]] - θb41i[6] - (c4 + d4) Sin[θc4 + θa41i[6]] - θb41i[6]
  5
)

```

Apéndices

Cinematica inversa cadena 4 serial

```

In[110]=
(*Obtenemos la ecuación de posición del robot igualada a cero*)
POE = Txyz[Xf, Yf, Zf];
Ec = (R0S + Ra4 + Rb4 + Rc4 + Rd4 - POE).n // Simplify;
                                     |simplifica

Ec // MatrixForm // Simplify;
    |forma de matriz |simplifica
Ec4 = Ec /. {a4 + b4 + c4 -> psi};

In[114]=
(*Separ la ecuación general para elevar al cuadrado*)
EcuI = (R0S + Ra4 - POE + Rc4 + Rd4) // FullSimplify;
                                     |simplifica completamente

EcuIC = EcuI.n /. {a4 + b4 + c4 -> psi};
EcuIC // MatrixForm // Simplify
    |forma de matriz |simplifica

EcuD = (Rb4).n;
EcuD // MatrixForm // Simplify
    |forma de matriz |simplifica

Out[116]/MatrixForm=

$$\begin{pmatrix} Xc - Xf + (c4 + d4) \cos[\psi] + a4 \cos[\theta_{a41i}[6]] & Yc - Yf \\ -Zf - (c4 + d4) \sin[\psi] - a4 \sin[\theta_{a41i}[6]] & \\ b4 \cos[\theta_{a41i}[6]} + \theta_{b41i}[6] & \\ 0 & \\ -b4 \sin[\theta_{a41i}[6]} + \theta_{b41i}[6] & \\ 1 & \end{pmatrix}$$


In[119]=
(*Elevamos al cuadrado ambos lados*)
(*EcuI=EcuD*)
EcIC = EcuIC[[1]]^2 + EcuIC[[3]]^2 // Simplify
                                     |simplifica
EcD = EcuD[[1]]^2 + EcuD[[3]]^2 // Simplify
                                     |simplifica

Out[119]=
(Xc - Xf + (c4 + d4) Cos[psi] + a4 Cos[theta_{a41i}[6]])^2 + (Zf + (c4 + d4) Sin[psi] + a4 Sin[theta_{a41i}[6]])^2

Out[120]=
b4^2

In[121]=
EI = Expand[EcIC - EcD] // Simplify
    |expande factores |simplifica

Out[121]=
a4^2 - b4^2 + c4^2 + 2 c4 d4 + d4^2 + Xc^2 - 2 Xc Xf + Xf^2 + Zf^2 +
2 (c4 + d4) (Xc - Xf) Cos[psi] + 2 a4 (c4 + d4) Cos[psi - theta_{a41i}[6]] + 2 a4 Xc Cos[theta_{a41i}[6]] -
2 a4 Xf Cos[theta_{a41i}[6]] + 2 c4 Zf Sin[psi] + 2 d4 Zf Sin[psi] - 2 a4 Zf Sin[theta_{a41i}[6]]

In[122]=
Ecu4 = EcIC - EcD /. Sin[theta_{a4}] -> \frac{2 ta4}{1 + ta4^2} /. Cos[theta_{a4}] -> \frac{1 - ta4^2}{1 + ta4^2} // Simplify
    |seno |coseno |simplifica

```

Apéndices

```

-b4^2 + (a4 - a4 ta4^2 / (1 + ta4^2) + Xc - Xf + (c4 + d4) Cos[psi])^2 + (2 a4 ta4 / (1 + ta4^2) + Zf + (c4 + d4) Sin[psi])^2
In[123]=
In[124]=
Sola4 = Solve[Ecu4 == 0, ta4] // Simplify
      |resuelve      |simplifica
Out[124]=
{{ta4 ->
  -((2 a4 Zf + 2 a4 c4 Sin[psi] - 2 a4 d4 Sin[psi] + 1/2 sqrt(16 a4^2 (Zf + (c4 + d4) Sin[psi])^2 - 4 (a4^2 - b4^2 + c4^2 - 2 c4 d4 +
    d4^2 + 2 a4 Xc + Xc^2 - 2 a4 Xf - 2 Xc Xf + Xf^2 + Zf^2 + 2 (c4 + d4) (a4 + Xc - Xf) Cos[psi] + 2
    (c4 + d4) Zf Sin[psi]) (a4^2 - b4^2 + c4^2 + 2 c4 d4 + d4^2 - 2 a4 Xc + Xc^2 + 2 a4 Xf - 2
    Xc Xf + Xf^2 + Zf^2 - 2 (c4 + d4) (a4 - Xc - Xf) Cos[psi] + 2 (c4 + d4) Zf Sin[psi]))) /
  (a4^2 - b4^2 + c4^2 - 2 c4 d4 + d4^2 - 2 a4 Xc + Xc^2 - 2 a4 Xf - 2 Xc Xf + Xf^2 + Zf^2 -
  2 (c4 + d4) (a4 - Xc + Xf) Cos[psi] + 2 (c4 + d4) Zf Sin[psi]))},
{ta4 -> (-2 a4 Zf - 2 a4 c4 Sin[psi] - 2 a4 d4 Sin[psi] +
  1/2 sqrt(16 a4^2 (Zf + (c4 + d4) Sin[psi])^2 - 4 (a4^2 - b4^2 + c4^2 - 2 c4 d4 + d4^2 + 2 a4 Xc + Xc^2 - 2 a4 Xf - 2 Xc Xf + Xf^2 +
  Zf^2 + 2 (c4 + d4) (a4 + Xc - Xf) Cos[psi] + 2 (c4 + d4) Zf Sin[psi]) (a4^2 - b4^2 + c4^2 + 2 c4 d4 + d4^2 - 2 a4 Xc +
  Xc^2 + 2 a4 Xf - 2 Xc Xf + Xf^2 + Zf^2 - 2 (c4 + d4) (a4 - Xc + Xf) Cos[psi] + 2 (c4 + d4) Zf Sin[psi]))) /
  (a4^2 - b4^2 + c4^2 - 2 c4 d4 + d4^2 - 2 a4 Xc + Xc^2 + 2 a4 Xf - 2 Xc Xf + Xf^2 + Zf^2 -
  2 (c4 + d4) (a4 - Xc + Xf) Cos[psi] + 2 (c4 + d4) Zf Sin[psi]))}}
In[125]=
oa41 = 2 ArcTan[ta4] /. Sola4[[1]]
      |arco tangente
Out[125]=
-2 ArcTan[2 a4 Zf + 2 a4 c4 Sin[psi] + 2 a4 d4 Sin[psi] +
  1/2 sqrt(16 a4^2 (Zf + (c4 + d4) Sin[psi])^2 - 4 (a4^2 - b4^2 + c4^2 + 2 c4 d4 + d4^2 + 2 a4 Xc + Xc^2 - 2 a4 Xf - 2 Xc Xf + Xf^2 +
  Zf^2 + 2 (c4 + d4) (a4 + Xc - Xf) Cos[psi] + 2 (c4 + d4) Zf Sin[psi]) (a4^2 - b4^2 + c4^2 + 2 c4 d4 + d4^2 - 2 a4 Xc +
  Xc^2 + 2 a4 Xf - 2 Xc Xf + Xf^2 + Zf^2 - 2 (c4 + d4) (a4 - Xc + Xf) Cos[psi] + 2 (c4 + d4) Zf Sin[psi])))] /
  (a4^2 - b4^2 + c4^2 - 2 c4 d4 + d4^2 - 2 a4 Xc + Xc^2 - 2 a4 Xf - 2 Xc Xf + Xf^2 + Zf^2 -
  2 (c4 + d4) (a4 - Xc + Xf) Cos[psi] + 2 (c4 + d4) Zf Sin[psi])]
In[126]=
oa42 = 2 ArcTan[ta4] /. Sola4[[2]]
      |arco tangente
Out[126]=
2 ArcTan[-2 a4 Zf - 2 a4 c4 Sin[psi] - 2 a4 d4 Sin[psi] +
  1/2 sqrt(16 a4^2 (Zf + (c4 + d4) Sin[psi])^2 - 4 (a4^2 - b4^2 + c4^2 + 2 c4 d4 + d4^2 + 2 a4 Xc + Xc^2 - 2 a4 Xf - 2 Xc Xf + Xf^2 +
  Zf^2 + 2 (c4 + d4) (a4 + Xc - Xf) Cos[psi] + 2 (c4 + d4) Zf Sin[psi]) (a4^2 - b4^2 + c4^2 + 2 c4 d4 + d4^2 - 2 a4 Xc +
  Xc^2 + 2 a4 Xf - 2 Xc Xf + Xf^2 + Zf^2 - 2 (c4 + d4) (a4 - Xc + Xf) Cos[psi] + 2 (c4 + d4) Zf Sin[psi])))] /
  (a4^2 - b4^2 + c4^2 - 2 c4 d4 + d4^2 - 2 a4 Xc + Xc^2 + 2 a4 Xf - 2 Xc Xf + Xf^2 + Zf^2 -
  2 (c4 + d4) (a4 - Xc + Xf) Cos[psi] + 2 (c4 + d4) Zf Sin[psi])]

```

Apéndices

```

Xc2 + 2 a4 Xf - 2 Xc Xf - Xf2 + Zf2 - 2 (c4 + d4) (a4 - Xc + Xf) Cos[ψ] + 2 (c4 + d4) Zf Sin[ψ])]] /
(a42 - b42 + c42 + 2 c4 d4 + d42 - 2 a4 Xc + Xc2 - 2 a4 Xf - 2 Xc Xf + Xf2 + Zf2 -
2 (c4 + d4) (a4 - Xc + Xf) Cos[ψ] + 2 (c4 + d4) Zf Sin[ψ])]]
In[127]=
Ec4[[1]
Out[127]=
Xc - Xf + c4 Cos[ψ] + d4 Cos[ψ] + a4 Cos[θa41i[6]] + b4 Cos[θa41i[6]] + θb41i[6]]
In[128]=
(*Para encontrar el ángulo θb1*)
Csa4b4 = Solve[Ec4[[1]] == 0, Cos[θa4 + θb4]] // Flatten
|resuelve |coseno |aplana
Sna4b4 = Solve[Ec4[[3]] == 0, Sin[θa4 + θb4]] // Flatten
|resuelve |seno |aplana
Out[128]=
{Cos[θa41i[6]] + θb41i[6]] ->  $\frac{-Xc + Xf - c4 \cos[\psi] - d4 \cos[\psi] - a4 \cos[\theta a41i[6]]}{b4}$ }
Out[129]=
{Sin[θa41i[6]] + θb41i[6]] ->  $\frac{-Zf - c4 \sin[\psi] - d4 \sin[\psi] - a4 \sin[\theta a41i[6]]}{b4}$ }
In[130]=
θb41 = (ArcTan[Cos[θa4 + θb4] /. Csa4b4, Sin[θa4 + θb4] /. Sna4b4]) - θa41
|arco t... |coseno |seno
θb42 = (ArcTan[Cos[θa4 + θb4] /. Csa4b4, Sin[θa4 + θb4] /. Sna4b4]) - θa42
|arco t... |coseno |seno
Out[130]=
2 ArcTan[ $\left[ \frac{2 a4 Zf + 2 a4 c4 \sin[\psi] + 2 a4 d4 \sin[\psi] + \frac{1}{2} \sqrt{(16 a4^2 (Zf + (c4 + d4) \sin[\psi])^2 - 4 (a4^2 - b4^2 + c4^2 + 2 c4 d4 + d4^2 + 2 a4 Xc + Xc^2 - 2 a4 Xf - 2 Xc Xf + Xf^2 + Zf^2 + 2 (c4 + d4) (a4 + Xc - Xf) \cos[\psi] + 2 (c4 + d4) Zf \sin[\psi]) (a4^2 - b4^2 + c4^2 + 2 c4 d4 + d4^2 - 2 a4 Xc + Xc^2 + 2 a4 Xf - 2 Xc Xf + Xf^2 + Zf^2 - 2 (c4 + d4) (a4 - Xc + Xf) \cos[\psi] + 2 (c4 + d4) Zf \sin[\psi])}}{2 (c4 + d4) (a4 - Xc + Xf) \cos[\psi] + 2 (c4 + d4) Zf \sin[\psi]}} \right] +$ 
ArcTan[ $\frac{-Xc + Xf - c4 \cos[\psi] - d4 \cos[\psi] - a4 \cos[\theta a41i[6]]}{b4}, \frac{-Zf - c4 \sin[\psi] - d4 \sin[\psi] - a4 \sin[\theta a41i[6]]}{b4}$ ]
Out[131]=
-2 ArcTan[ $\left[ \frac{-2 a4 Zf - 2 a4 c4 \sin[\psi] - 2 a4 d4 \sin[\psi] + \frac{1}{2} \sqrt{(16 a4^2 (Zf + (c4 + d4) \sin[\psi])^2 - 4 (a4^2 - b4^2 + c4^2 + 2 c4 d4 + d4^2 + 2 a4 Xc + Xc^2 - 2 a4 Xf - 2 Xc Xf + Xf^2 + Zf^2 + 2 (c4 + d4) (a4 + Xc - Xf) \cos[\psi] + 2 (c4 + d4) Zf \sin[\psi]) (a4^2 - b4^2 + c4^2 + 2 c4 d4 + d4^2 - 2 a4 Xc + Xc^2 + 2 a4 Xf - 2 Xc Xf + Xf^2 + Zf^2 - 2 (c4 + d4) (a4 - Xc + Xf) \cos[\psi] + 2 (c4 + d4) Zf \sin[\psi])}}{2 (c4 + d4) (a4 - Xc + Xf) \cos[\psi] + 2 (c4 + d4) Zf \sin[\psi]}} \right] +$ 
(a42 - b42 + c42 + 2 c4 d4 + d42 - 2 a4 Xc + Xc2 + 2 a4 Xf - 2 Xc Xf + Xf2 - Zf2 -
2 (c4 + d4) (a4 - Xc + Xf) Cos[ψ] + 2 (c4 + d4) Zf Sin[ψ])]] +

```

$$\text{ArcTan}\left[\frac{-Xc + Xf - c4 \cos[\psi] - d4 \cos[\psi] - a4 \cos[\theta a41i[6]]}{b4}, \frac{-Zf - c4 \sin[\psi] - d4 \sin[\psi] - a4 \sin[\theta a41i[6]]}{b4}\right]$$

DATOS

```
In[132]=
pi = {15, 15, 0};
pf = {23, 23, 0};

a1 = 13;
b1 = 10;
c1 = 4.5;
a2 = 13;
b2 = 10;
c2 = 4.5;
a3 = 13;
b3 = 10;
c3 = 4.5;

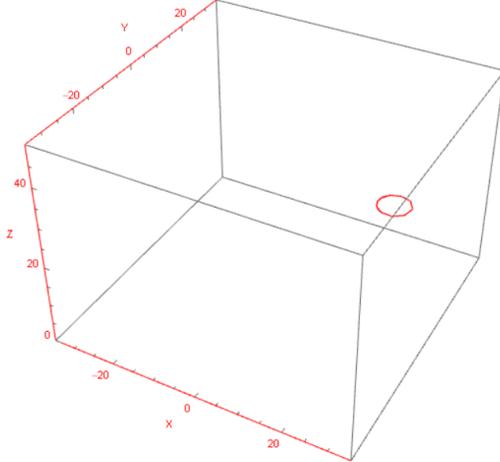
a4 = 9.5;
b4 = 14.5;
c4 = 6.5;
d4 = 4;
Xp = 4;
Yp = 5.5;
Xq = 36.5;
Yq = 5.5;
Xr = 20;
Yr = 34;

psi = 10 °;
theta4 = 5 °;
phi = 10 °;
mu1 = 30 °;
mu2 = 150 °;
mu3 = 270 °;
tf = 20;
```

TRAYECTORIA

```
In[160]=
For[t = 0, t <= tf, t += 1,
  para cada
    p = pi + (10 * (t/20)^3 - 15 * (t/20)^4 + 6 * (t/20)^5) * (pf - pi);
    (*Función para la trayectoria S=f(x,y,z)*)
    S[t] = {19 + 4 * Cos[p[[1]], 19 + 4 * Sin[p[[2]], 20};
    ];
  tablaXYZ = Table[{S[t][[1]], S[t][[2]], S[t][[3]], {t, 0, tf, 1}};
  tray = Line[tablaXYZ];
Graphics3D[
  gráfico 3D
  {
    {AbsoluteThickness[2], RGBColor[1, 0, 0], tray}
  },
  Axes -> True, AxesLabel -> {"X", "Y", "Z"},
  AxesStyle -> Directive[Red, 18],
  PlotRange -> {{-35, 35}, {-35, 35}, {0, 50}},
  ImageSize -> 850
]
```

Out[163]=



VALORES DE LAS VARIABLES ARTICULARES

In[164]=

```
For[t = 0, t ≤ tf, t += 1,
  para cada
```

```

Xc = 0.7 * S[t][[1]];
Yc = 0.7 * S[t][[2]];
(*Cadena 1 solución 1*)
θa11i[t] = θa11;
θa1 = θa11i[t];
θb11i[t] = θb11;
θb1 = θb11i[t];
(*Cadena 1 solución 2*)
θa12i[t] = θa12;
θa1 = θa12i[t];
θb12i[t] = θb12;
θb1 = θb12i[t];

(*Cadena 2 solución 1*)
θa21i[t] = θa21;
θa2 = θa21i[t];
θb21i[t] = θb21;
θb2 = θb21i[t];
(*Cadena 2 solución 2*)
θa22i[t] = θa22;
```

Apéndices

```

    theta22i[t] = theta22;
    theta2 = theta22i[t];
    theta_b22i[t] = theta_b22;
    theta_b2 = theta_b22i[t];

    (*Cadena 3 solución 1*)
    theta31i[t] = theta31;
    theta3 = theta31i[t];
    theta_b31i[t] = theta_b31;
    theta_b3 = theta_b31i[t];
    (*Cadena 3 solución 2*)
    theta32i[t] = theta32;
    theta3 = theta32i[t];
    theta_b32i[t] = theta_b32;
    theta_b3 = theta_b32i[t];

    Xf = S[t][[1]];
    Yf = S[t][[2]];
    Zf = S[t][[3]];

    (*Cadena 4 solución 1*)
    theta41i[t] = theta41;
    theta4 = theta41i[t];
    (*Cadena 4 solución 1*)
    theta41i[t] = theta41;
    theta4 = theta41i[t];
    theta_b41i[t] = theta_b41;
    theta_b4 = theta_b41i[t];
    (*Cadena 4 solución 2*)
    theta42i[t] = theta42;
    theta4 = theta42i[t];
    theta_b42i[t] = theta_b42;
    theta_b4 = theta_b42i[t];

    theta_c4 = -(theta4 + theta_b4) + psi
]

In[165]=
(*Xf//N
  Yf//N
  Zf//N
  Xc//N
  Yc//N*)
psi
? theta41i
? theta_b41i
```

SIMULACIÓN

```

In[168]= Animate[
  |anima
  |
  |   $\theta_{a1} = \theta_{a11i}[t];$ 
  |   $\theta_{b1} = \theta_{b11i}[t];$ 
  |
  |   $\theta_{a2} = \theta_{a21i}[t];$ 
  |   $\theta_{b2} = \theta_{b21i}[t];$ 
  |
  |   $\theta_{a3} = \theta_{a31i}[t];$ 
  |   $\theta_{b3} = \theta_{b31i}[t];$ 
  |
  |  (*Solución del serial, simulación*)
  |
  |   $\theta_{a4} = \theta_{a41i}[t];$ 
  |   $\theta_{b4} = \theta_{b41i}[t];$ 
  |
  |  Ce = {0, 0, 0};
  |  Eslabon11 = Line[{T3D[R0P.n], T3D[R0A1.n]};
  |  |línea
  |  Eslabon12 = Line[{T3D[R0A1.n], T3D[R0B1.n]};
  |  |línea
  |  Eslabon13 = Line[{T3D[R0B1.n], T3D[R0C1.n]};
  |  Eslabon21 = Line[{T3D[R0Q.n], T3D[R0A2.n]};
  |  |línea
  |  Eslabon22 = Line[{T3D[R0A2.n], T3D[R0B2.n]};
  |  |línea
  |  Eslabon23 = Line[{T3D[R0B2.n], T3D[R0C2.n]};
  |  |línea
  |
  |  Eslabon31 = Line[{T3D[R0R.n], T3D[R0A3.n]};
  |  |línea
  |  Eslabon32 = Line[{T3D[R0A3.n], T3D[R0B3.n]};
  |  |línea
  |  Eslabon33 = Line[{T3D[R0B3.n], T3D[R0C3.n]};
  |  |línea
  |
  |  Eslabon4 = Line[{(0.7 * S[t][[1]], 0.7 * S[t][[2]], 0), T3D[R0A4.n]};
  |  |línea
  |  Eslabon5 = Line[{T3D[R0A4.n], T3D[R0B4.n]};
  |  |línea
  |  Eslabon6 = Line[{T3D[R0B4.n], T3D[R0C4.n]};
  |  |línea
  |  Eslabon7 = Line[{T3D[R0C4.n], T3D[R0D4.n]};
  |  |línea

```

```

Graphics3D[
|gráfico 3D
{
  {AbsoluteThickness[1], RGBColor[0, 1, 1], tray},
  |grosor absoluto |color RGB
  {AbsoluteThickness[8], RGBColor[1, 1, 0], Eslabon11},
  |grosor absoluto |color RGB
  {AbsoluteThickness[8], RGBColor[1, 0, 0], Eslabon12},
  |grosor absoluto |color RGB
  {AbsoluteThickness[8], RGBColor[0, 0, 1], Eslabon13},
  |grosor absoluto |color RGB

  {AbsoluteThickness[8], RGBColor[1, 1, 0], Eslabon21},
  |grosor absoluto |color RGB
  {AbsoluteThickness[8], RGBColor[1, 0, 0], Eslabon22},
  |grosor absoluto |color RGB
  {AbsoluteThickness[8], RGBColor[0, 0, 1], Eslabon23},
  |grosor absoluto |color RGB

  {AbsoluteThickness[8], RGBColor[1, 1, 0], Eslabon31},
  |grosor absoluto |color RGB
  {AbsoluteThickness[8], RGBColor[1, 0, 0], Eslabon32},
  |grosor absoluto |color RGB
  {AbsoluteThickness[8], RGBColor[0, 0, 1], Eslabon33},
  |grosor absoluto |color RGB
  {AbsoluteThickness[8], RGBColor[1, 1, 0], Eslabon4},
  |grosor absoluto |color RGB
  {AbsoluteThickness[8], RGBColor[1, 0, 0], Eslabon5},
  |grosor absoluto |color RGB
  {AbsoluteThickness[8], RGBColor[0, 0, 1], Eslabon6},
  |grosor absoluto |color RGB
  {AbsoluteThickness[8], RGBColor[0, 0, 1], Eslabon7}
  |grosor absoluto |color RGB

},
Axes → True, AxesLabel → {"X", "Y", "Z"},
|ejes |ver... |etiqueta de ejes
ImageSize → 900,
|tamaño de imagen
PlotRange → {{0, 45}, {-20, 45}, {0, 50}}
|rango de representación
], {t, 0, tf, 1}
]

```

Código para Manipulación de Coordenadas y Posturas del Manipulador

Manipulador Paralelo Serie de Annie y Beto

```

M[1]= Qz[e_] :=  $\begin{pmatrix} \text{Cos}[e] & -\text{Sin}[e] & 0 & 0 \\ \text{Sin}[e] & \text{Cos}[e] & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ ; (*Matrices de rotación*)

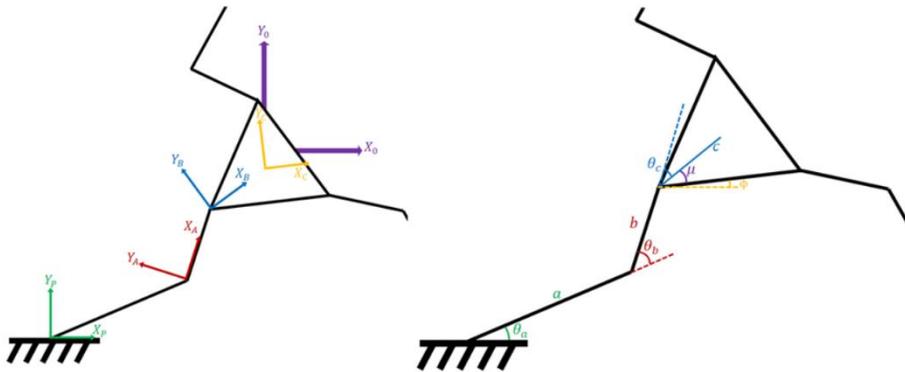
Qy[e_] :=  $\begin{pmatrix} \text{Cos}[e] & 0 & \text{Sin}[e] & 0 \\ 0 & 1 & 0 & 0 \\ -\text{Sin}[e] & 0 & \text{Cos}[e] & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ ;

Qx[e_] :=  $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \text{Cos}[e] & -\text{Sin}[e] & 0 \\ 0 & \text{Sin}[e] & \text{Cos}[e] & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ ;

Txyz[x_, y_, z_] :=  $\begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$ ; (*Matrices de traslación*)

Tx[x_] :=  $\begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ ;

VectorizarX[M_] := M[[1, 4]]; (*Método para extraer elementos de una matriz de 4x4*)
VectorizarY[M_] := M[[2, 4]]; (*en X, Y y Z*)
VectorizarZ[M_] := M[[3, 4]];
VectorizarM[M_] := {M[[1, 4]], M[[2, 4]], M[[3, 4]]};
(*)
    
```



REGLA DE LA MANO DERECHA:
 * Pulgar, apunta a la parte positiva de tu eje de giro SIEMPRE!
 * Si los 4 dedos restantes, cierran el puño, es un giro positivo.
 * Si los 4 dedos restantes, abren el puño, es un giro negativo.
 *)

CD Solución Paralelo (cadena I)

```

M[10]= (*
Empleando el método de suma vectorial
    
```

Apéndices

```

h[10]= (*
Empleando el método de suma vectorial
Donde los eslabones ai, bi, ci; se escriben con minúsculas,
igual que los ángulos θai, θbi, θci, para i = 1, 2, 3.
Vectores inician con minúscula.

    Crece en sentido horario
*)
v0P = Txyz[Xp, Yp, 0];
vA1 = Qz[90 ° - θa1].Tx[a];
vB1 = Qz[90 ° - θa1].Qz[θb1].Tx[b];
vC1 = Qz[φ + μ1].Tx[c];
v0C1 = v0P + vA1 + vB1 + vC1;
(*Ecuación del punto c del centro del robot paralelo visto desde el centro del móvil*)
v0A1 = v0P + vA1;
v0B1 = v0P + vA1 + vB1;

(*Cadena 2 Crece en sentido horario*)
v0Q = Txyz[Xq, Yq, 0];
vA2 = Qz[π - θa2].Tx[a];
vB2 = Qz[π - θa2].Qz[θb2].Tx[b];
vC2 = Qz[φ + μ2].Tx[c];
v0C2 = v0Q + vA2 + vB2 + vC2;
(*Ecuación del punto c del centro del robot paralelo visto desde el centro del móvil*)
v0A2 = v0Q + vA2;
v0B2 = v0Q + vA2 + vB2;

(*Cadena 3 Crece en sentido antihorario*)
v0R = Txyz[Xr, Yr, 0];
vA3 = Qz[π + θa3].Tx[a];
vB3 = Qz[π + θa3].Qz[θb3].Tx[b];
vC3 = Qz[φ + μ3].Tx[c];
v0C3 = v0R + vA3 + vB3 + vC3;
(*Ecuación del punto c del centro del robot paralelo visto desde el centro del móvil*)
v0A3 = v0R + vA3;
v0B3 = v0R + vA3 + vB3;

VectorizarM[v0C1] // FullSimplify // MatrixForm
|simplifica completame... |forma de matriz
VectorizarM[v0C2] // FullSimplify // MatrixForm
|simplifica completame... |forma de matriz
VectorizarM[v0C3] // FullSimplify // MatrixForm
|simplifica completame... |forma de matriz


$$\begin{pmatrix} Xp - c \cos[\mu_1 + \phi] + a \sin[\theta_{a1}] + b \sin[\theta_{a1} - \theta_{b1}] \\ Yp + a \cos[\theta_{a1}] + b \cos[\theta_{a1} - \theta_{b1}] + c \sin[\mu_1 + \phi] \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} Xq - a \cos[\theta_{a2}] - b \cos[\theta_{a2} - \theta_{b2}] + c \cos[\mu_2 + \phi] \\ Yq + a \sin[\theta_{a2}] + b \sin[\theta_{a2} - \theta_{b2}] + c \sin[\mu_2 + \phi] \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} Xr - a \cos[\theta_{a3}] - b \cos[\theta_{a3} + \theta_{b3}] + c \cos[\mu_3 + \phi] \\ Yr - a \sin[\theta_{a3}] - b \sin[\theta_{a3} + \theta_{b3}] + c \sin[\mu_3 + \phi] \\ 0 \end{pmatrix}$$


```

Datos Robot

```

In[34]:=
EslabonA1 = Line[{VectorizarM[v0P], VectorizarM[v0A1]};
           |línea
EslabonA2 = Line[{VectorizarM[v0Q], VectorizarM[v0A2]};
           |línea
EslabonA3 = Line[{VectorizarM[v0R], VectorizarM[v0A3]};
           |línea

EslabonB1 = Line[{VectorizarM[v0A1], VectorizarM[v0B1]};
           |línea
EslabonB2 = Line[{VectorizarM[v0A2], VectorizarM[v0B2]};
           |línea
EslabonB3 = Line[{VectorizarM[v0A3], VectorizarM[v0B3]};
           |línea

EslabonC1 = Line[{VectorizarM[v0B1], VectorizarM[v0C1]};
           |línea
EslabonC2 = Line[{VectorizarM[v0B2], VectorizarM[v0C2]};
           |línea
EslabonC3 = Line[{VectorizarM[v0B3], VectorizarM[v0C3]};
           |línea

Triangulo1 = Line[{VectorizarM[v0B1], VectorizarM[v0B2]};
           |línea
Triangulo2 = Line[{VectorizarM[v0B2], VectorizarM[v0B3]};
           |línea
Triangulo3 = Line[{VectorizarM[v0B3], VectorizarM[v0B1]};
           |línea
a = 12.7;  μ1 = 30 °;
b = 11.2;  μ2 = 150 °;
c = 5.5;   μ3 = 270 °;

Xp = -16.5;  Xq = 16.5;   Xr = 0;
Yp = -14.5;  Yq = -14.5;  Yr = 14;

φ = 60 °;

θα1 = 100 °;  θβ1 = 90 °;
θα2 = 85 °;   θβ2 = 66.642 °;
θα3 = 10 °;   θβ3 = 120.63 °;

```

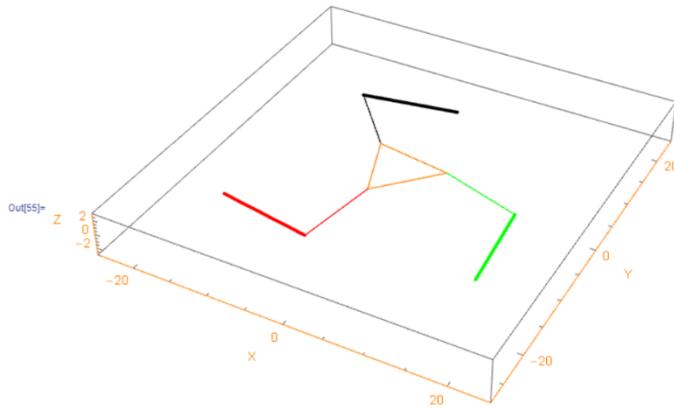
Dibujo

```

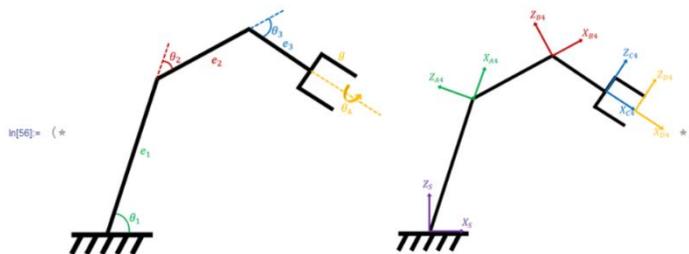
In[55]:= Graphics3D[
  |gráfico 3D
  {
    {AbsoluteThickness[3], RGBColor[1, 0, 0], EslabonA1},
    |grosor absoluto |color RGB
    {AbsoluteThickness[3], RGBColor[0, 1, 0], EslabonA2},
    |grosor absoluto |color RGB
    {AbsoluteThickness[3], RGBColor[0, 0, 0], EslabonA3},
    |grosor absoluto |color RGB

    {AbsoluteThickness[1], RGBColor[1, 0, 0], EslabonB1},
    |grosor absoluto |color RGB
    {AbsoluteThickness[1], RGBColor[0, 1, 0], EslabonB2},
    {AbsoluteThickness[1], RGBColor[0, 0, 0], EslabonB3},
    |grosor absoluto |color RGB
    {AbsoluteThickness[1], RGBColor[1, 0.5, 0], Triangulo1},
    |grosor absoluto |color RGB
    {AbsoluteThickness[1], RGBColor[1, 0.5, 0], Triangulo2},
    |grosor absoluto |color RGB
    {AbsoluteThickness[1], RGBColor[1, 0.5, 0], Triangulo3}
    |grosor absoluto |color RGB
  },
  Axes → True,
  |ejes |verdadero
  AxesLabel → {"X", "Y", "Z"},
  |etiqueta de ejes
  AxesStyle → Directive[Orange, 12],
  |estilo de ejes |directiva |naranja
  PlotRange → {{-25, 25}, {-28, 25}, {-3, 3}},
  |rango de representación
  ImageSize → Large
  |tamaño de ima... |grande
]

```



Serial



```
(*Desde último sistema del paralelo (sistema c) el sistema de la base del manipulador serial,
se denomina S, y respecto al anterior, se tiene un giro horario de 90°*)
vCS = (*Qz[-Pi/2].*)Txyz[0, 0, 0];
vA4 = Qy[-theta1].Tx[e1];
vB4 = Qy[-theta1].Qy[-theta2 + pi/2].Tx[e2];
(*Haciendo uso de la restricción cinemática, vemos que la suma de los ángulos theta, resulta phi*)
vC4 = Qy[-theta1].Qy[-theta2 + pi/2].Qy[-theta3 + pi/2 + 10°].Tx[e3];
vD4 = Qy[-theta1].Qy[-theta2 + pi/2].Qy[-theta3 + pi/2 + 10°].Qx[theta4].Tx[g];

vCA4 = vCS + vA4;
vCB4 = vCS + vA4 + vB4;
vCC4 = vCS + vA4 + vB4 + vC4;
vCD4 = vCS + vA4 + vB4 + vC4 + vD4;
vCD4 // MatrixForm // FullSimplify
{theta4} Cos[theta1 + theta2] - Sin[theta1] - 2 Cos[frac{theta4}{2}]^2 Sin[10° - theta1 - theta2 - theta3] e1 Cos[theta1] - (e3 + g) Cos[10° - theta1 - theta2 - theta3] + e2 Sin[theta1 + theta2]
{theta4} 1 - Cos[theta1] - Cos[10° - theta1 - theta2 - theta3] (1 + Cos[theta4]) + Sin[theta1 + theta2] - e2 Cos[theta1 + theta2] + e1 Sin[theta1] + (e3 + g) Sin[10° - theta1 - theta2 - theta3]
0
5

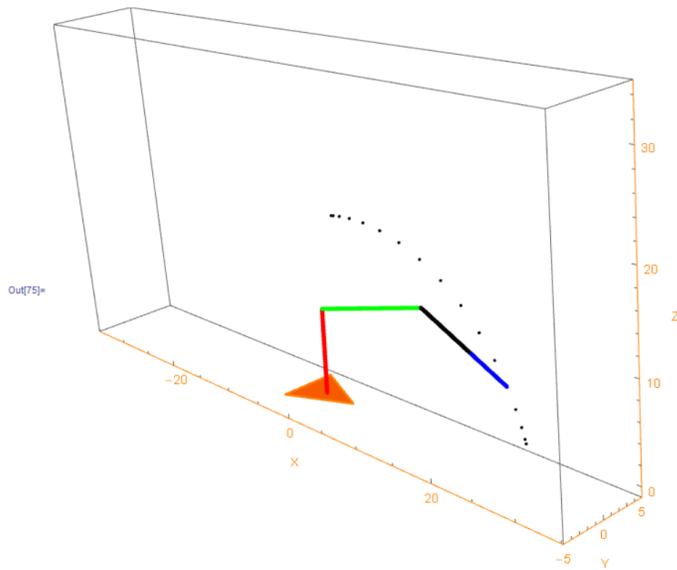
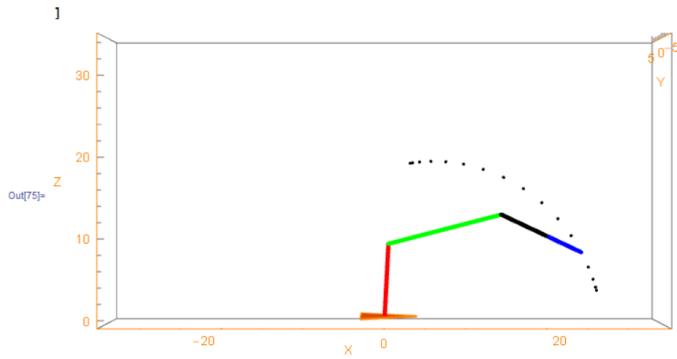
In[66]:= EslabonE1 = Line[{VectorizarM[vCS], VectorizarM[vCA4]}];
EslabonE2 = Line[{VectorizarM[vCA4], VectorizarM[vCB4]}];
EslabonE3 = Line[{VectorizarM[vCB4], VectorizarM[vCC4]}];
EslabonG = Line[{VectorizarM[vCC4], VectorizarM[vCD4]}];
BaseParalelo = Triangle[{{4, 0, 0}, {-3, 3, 0}, {-3, -3, 0}}];

e1 = 9.2; theta1 = 87.05004751°;
e2 = 14.8; theta2 = 17.4100095°;
e3 = 6.7; theta3 = 60.35996199°;
g = 4.5; theta4 = 0;

In[75]:= Graphics3D[
{
{AbsoluteThickness[4], RGBColor[1, 0, 0], EslabonE1},
{AbsoluteThickness[4], RGBColor[0, 1, 0], EslabonE2},
{AbsoluteThickness[4], RGBColor[0, 0, 0], EslabonE3},
{AbsoluteThickness[4], RGBColor[0, 0, 1], EslabonG},
}
```

Apéndices

```
(EdgeForm[{Thick, Orange}], Orange, BaseParalelo),
  {forma de borde |grueso |naranja |naranja
Point[{3.233909622, 0, 19.40020947}], Point[{3.522760878, 0, 19.44300458}],
  |punto |punto
Point[{4.424107472, 0, 19.54813644}],
  |punto
Point[{5.863742133, 0, 19.62866004}], Point[{7.760054939, 0, 19.57406539}],
  |punto |punto
  Point[{10.01313512, 0, 19.27268008}],
  |punto
Point[{12.50059779, 0, 18.63253738}], Point[{15.08182087, 0, 17.59919163}],
  |punto |punto
  Point[{17.60985772, 0, 16.16792325}],
  |punto
Point[{19.94810816, 0, 14.38803956}], Point[{21.98741975, 0, 12.35845022}],
  |punto |punto
  Point[{23.65909611, 0, 10.21579204}],
  |punto
Point[{24.94035896, 0, 8.118256015}], Point[{25.8507071, 0, 6.229180634}], Point[{26.43959151, 0, 4.704058183}],
  |punto |punto |punto
Point[{26.76716514, 0, 3.682961425}], Point[{26.8802246, 0, 3.287953196}]
  |punto |punto
),
Axes → True,
  |ejes |verdadero
AxesLabel → {"X", "Y", "Z"},
  |etiqueta de ejes
AxesStyle → Directive[Orange, 12],
  |estilo de ejes |directiva |naranja
PlotRange → {{-35.2, 35.2}, {-5, 5}, {-1, 35.2}},
  |punto de vista |frente
ImageSize → Large
  |tamaño de imagen |grande
```



Imágenes Adicionales de la Tarea a Realizar

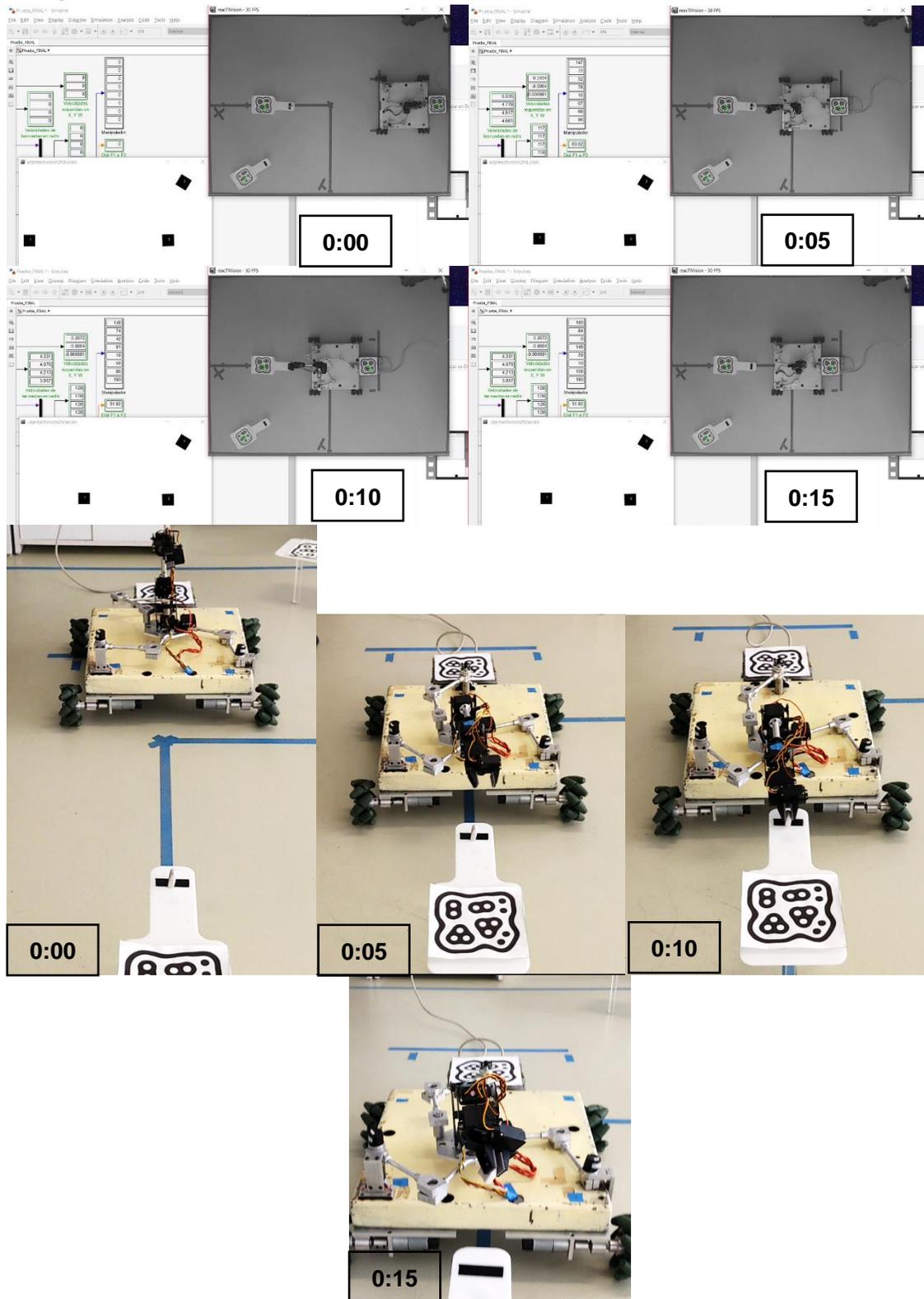


Ilustración 92 Ejecución de la tarea vista desde distinto ángulo

Apéndices

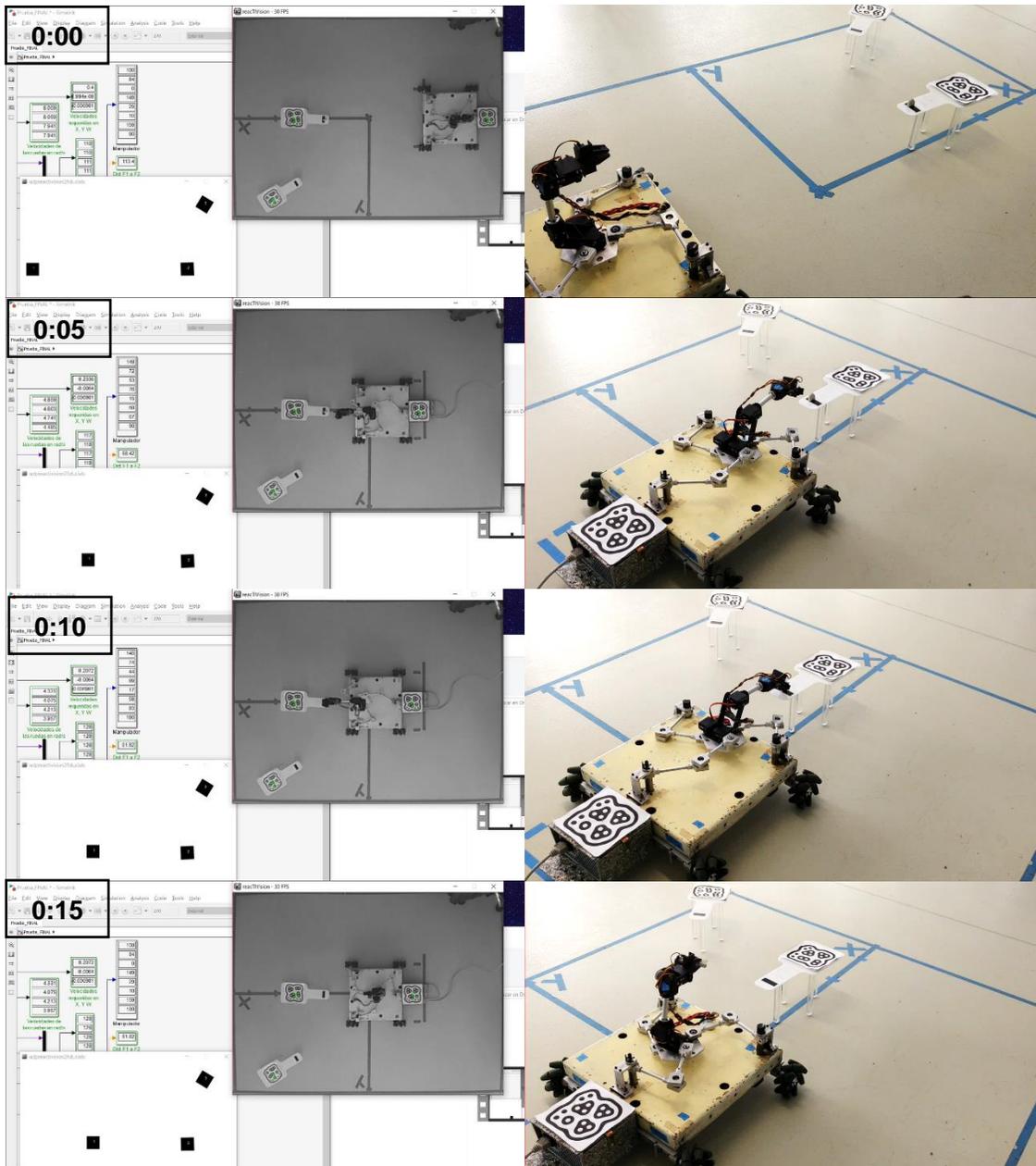


Ilustración 93 Ejecución de la tarea, distinto ángulo, nótese la secuencia generada

Con las presentes imágenes se muestra la ejecución generada de la tarea, desde distintos ángulos y en pruebas diferentes. Como se puede apreciar, se hace énfasis en las tomas, a ciertos ángulos y enfoques, teniendo imágenes completas en ciertas ocasiones y en otras, acercamientos, igualmente, se muestra siempre la interfaz de usuario utilizada.

Índice de Ecuaciones

Ecuación 1	Matrices de rotación sobre los ejes X, Y y Z respectivamente.....	21
Ecuación 2	Matriz de traslación.....	22
Ecuación 3	Transformación de A visto desde O.....	23
Ecuación 4	Transformación de B visto desde A.....	23
Ecuación 5	Transformación de C visto desde B.....	23
Ecuación 6	Transformación de D visto desde C.....	23
Ecuación 7	Sistema B visto desde el sistema inercial.....	23
Ecuación 8	Sistema C visto desde el sistema inercial.....	23
Ecuación 9	Sistema D visto desde el sistema inercial.....	23
Ecuación 10	Coordenada en X del efector final desde la base del manipulador.....	24
Ecuación 11	Coordenada en Y del efector final desde la base del manipulador.....	24
Ecuación 12	Coordenada en Z del efector final desde la base del manipulador.....	24
Ecuación 13	Punto del efector final del robot serial.....	24
Ecuación 14	Punto del efector final empleando la restricción cinemática.....	24
Ecuación 15	Despeje auxiliar para la solución de la cinemática inversa.....	24
Ecuación 16	Ecuación elevada al cuadrado para implementar las identidades trigonométricas.....	25
Ecuación 17	Identidades trigonométricas implementadas en la cinemática inversa.....	25
Ecuación 18	Polinomio cuadrático de solución de θ_1	25
Ecuación 19	Solución para θ_2	25
Ecuación 20	Transformación de P visto desde O.....	27
Ecuación 21	Transformación de A visto desde P.....	27
Ecuación 22	Transformación de B visto desde A.....	27
Ecuación 23	Transformación de C visto desde B.....	27
Ecuación 24	Transformación de A visto desde O.....	27
Ecuación 25	Transformación de B visto desde O.....	27
Ecuación 26	Transformación de C visto desde O.....	27
Ecuación 27	Punto del efector final del robot paralelo, note el cambio de variable de $\mu + \varphi$	27
Ecuación 28	Separación de términos en ambos lados de la ecuación para X y Y.....	28
Ecuación 29	Ambos lados de la ecuación al cuadrado.....	28
Ecuación 30	Suma de cuadrados de la ecuación anterior.....	28
Ecuación 31	Identidades trigonométricas.....	28

Apéndices

Ecuación 32 Polinomio igualado a 0	28
Ecuación 33 Solución para θ_a	28
Ecuación 34 Solución para θ_b	29
Ecuación 35 Vector Postura del Robot	31
Ecuación 36 Matriz de rotación respecto al eje Z.....	31
Ecuación 37 Velocidad angular de un eslabón visto desde sí mismo.....	32
Ecuación 38 Velocidad lineal de un eslabón visto desde sí mismo	32
Ecuación 39 Traslación y rotación del sistema 0 al sistema 1 del robot móvil.....	33
Ecuación 40 Traslación y rotación del sistema 1 al sistema 2 del robot móvil.....	33
Ecuación 41 Traslación y rotación del sistema 2 al sistema 3 del robot móvil.....	33
Ecuación 42 Matriz de restricciones cinemáticas	33
Ecuación 43 Vector de coordenadas generalizadas.....	33
Ecuación 44 Modelo matemático final del robot móvil omnidireccional.....	34
Ecuación 45 Conversión de pixeles a centímetros de $\{x,y\}$	39
Ecuación 46 Transformación sobre el Eje X.....	41
Ecuación 47 Resultados de las transformaciones de X y Y para el móvil.....	41
Ecuación 48 Resultados de las transformaciones de X y Y para el objetivo	41
Ecuación 49 Distancia entre fiducial 1 a fiducial 2 (robot y objetivo)	42
Ecuación 50	45
Ecuación 51.....	45
Ecuación 52	45
Ecuación 53	45
Ecuación 54.....	45
Ecuación 55	45
Ecuación 56 Velocidad sobre el eje X	46
Ecuación 57 Velocidad sobre el eje Y.....	46
Ecuación 58 Modelo matemático de la división de voltaje	61
Ecuación 59 Diagrama de pines del PIC16F88	64
Ecuación 60 Factor de reducción en los pulsos por revolución de los encoders.....	65
Ecuación 61 Ecuación Generadora de Puntos.....	70
Ecuación 62 Polinomio del servomotor P	70
Ecuación 63 Polinomio del servomotor Q	71
Ecuación 64 Polinomio del servomotor R.....	71
Ecuación 65 Polinomio de los servomotores de los hombros.....	72
Ecuación 66 Polinomio del servomotor del Codo	72

Ecuación 67 Polinomio del servomotor de la Muñeca	73
Ecuación 68 Polinomio del servomotor del giro de la Muñeca.....	73

Índice de Tablas

Tabla 1 Tipos de articulaciones [5], [7], [3]	4
Tabla 2 Tipos de configuración	5
Tabla 3 Ejemplos de robots híbridos	9
Tabla 4 Tabla de dimensiones de los eslabones	66
Tabla 5 Dimensiones del robot móvil	67
Tabla 6 Valores de las variables al inicio y fin de la rutina	69

Índice de Imágenes

Ilustración 1 Esquema general de un robot [3]	2
Ilustración 2 Clasificación de los robots, propuesta desde la tesis de Cárdenas Verdugo [4]	2
Ilustración 3 Posiblemente el primer robot paralelo patentado en 1931 (US Patent No. 1,789,680) [8]	6
Ilustración 4 Plataforma Gough en 1954 (Proc. IMechE, 1965-66) [8]	6
Ilustración 5 Ejemplos de distintos robots paralelos planares con actuadores prismáticos y giratorios	7
Ilustración 6 Ejemplos de robots paralelo espacial	7
Ilustración 7 Clasificación de ruedas	10
Ilustración 8 Rueda fija [12]	11
Ilustración 9 Rueda orientable centrada [12]	11
Ilustración 10 Rueda castora [12]	11
Ilustración 11 Ruedas suecas omnidireccionales [12]	12
Ilustración 12 Rueda Mecanum	12
Ilustración 13 Triciclo clásico [16]	13
Ilustración 14 Configuración diferencial [16]	13
Ilustración 15 Configuración Ackerman	14
Ilustración 16 Locomoción síncrona [12]	14
Ilustración 17 Robot diferencial no holonómico [14]	15
Ilustración 18 Esquema de robot cosechador de cerezas [22]	16

Apéndices

Ilustración 19 Robot híbrido manufacturado para la tesis del Dr. Patricio Martínez Zamudio [20]	17
Ilustración 20 Modelo en CAD del estudio de un robot manipulador híbrido móvil [21]	18
Ilustración 21 Modelo en CAD del modelo funcional para la tesis de Ignacio Carlos Cruz López y Juan de Dios Flores Méndez [20]	18
Ilustración 22 Modelo funcional final de robot hexápodo [4]	19
Ilustración 23 Plataforma móvil omnidireccional utilizada en la tesis [14]	19
Ilustración 24 Sistemas de referencia del manipulador serial	22
Ilustración 25 Definición del manipulador serial con base en sus variables	23
Ilustración 26 Sistemas de referencia del robot paralelo	26
Ilustración 27 Variables del robot paralelo delta	26
Ilustración 28 Sistemas de referencia del manipulador serie paralelo	29
Ilustración 29 Sistemas de referencia de la plataforma móvil omnidireccional	31
Ilustración 30 Vectores de velocidades de eslabones adyacentes	32
Ilustración 31 Sistemas de referencia de las ruedas del robot móvil	33
Ilustración 32 Mapa general del funcionamiento del sistema	35
Ilustración 33 Marcadores fiducial 0, 1 y 2 respectivamente	35
Ilustración 34 Imagen generada mediante la cámara y el programa ReacTIVision	36
Ilustración 35 Logotipo y dirección electrónica del protocolo	36
Ilustración 36 Configuración del bloque Packet Input	38
Ilustración 37 Marco coordinado de la webcam y malla de calibración de ReacTIVision	38
Ilustración 38 Módulo de Visión	39
Ilustración 39 Ambiente inteligente	40
Ilustración 40 Sub-bloques del módulo de visión	40
Ilustración 41 Distancia del centro del fiducial al centro de la plataforma móvil	41
Ilustración 42 Distancia del centro del fiducial al objeto a manipular	41
Ilustración 43 Bloque de asignación del objetivo	42
Ilustración 44 Campo potencial parabólico centrada en el origen asociado al objetivo	43
Ilustración 45 Campo potencial parabólico de un obstáculo	43
Ilustración 46 Ejemplo de un campo potencial artificial representado por un obstáculo y la meta	43
Ilustración 47 Bloque de Campos Potenciales	44
Ilustración 48 Parámetros del método de campos potenciales	44
Ilustración 49 Salida de bloque de Campos Potenciales	46
Ilustración 50 Modelo que permite asignar las velocidades angulares a cada rueda	47

Apéndices

Ilustración 51 Bloque de del modelo cinemático	47
Ilustración 52 Bloque de Compensaciones	48
Ilustración 53 Ajuste de corrimiento del cero, datos positivos y enteros	48
Ilustración 54 Bloque de conversión	48
Ilustración 55 Conversión a enteros sin signo de un byte	49
Ilustración 56 Envío de datos de manipulador serie-paralelo	49
Ilustración 57 Programa completo de procesamiento de datos en Matlab Simulink	51
Ilustración 58 Comunicación inalámbrica empleada	51
Ilustración 59 Ejemplo de IDE de Arduino v1.6.9	52
Ilustración 60 Ilustración de Arduino utilizado, imagen creada con Fritzing©	53
Ilustración 61 Diagrama de conexiones de la plataforma móvil omnidireccional	55
Ilustración 62 Diagrama de conexiones del manipulador serie paralelo	55
Ilustración 63 Características técnicas de Arduino Mega 2560	56
Ilustración 64 Ejemplo de Arduino Mega 2560 utilizado en este trabajo	56
Ilustración 65 Shield Wifi de Arduino	57
Ilustración 66 Tarjeta MD25 utilizada para el control de dos motores DC	58
Ilustración 67 Motores 37D 50:1 De la marca Pololu	60
Ilustración 68 Etapa de reducción de PWM	61
Ilustración 69 Módulo de potencia y diagrama de conexión	62
Ilustración 70 Servomotor 1501 MG implementado	62
Ilustración 71 Etapa de potencia para los servomotores	63
Ilustración 72 Información de entrada y salida del PIC16F88 para la reducción de frecuencia.	64
Ilustración 73 Dimensiones del manipulador híbrido	66
Ilustración 74 Acoples de la rueda al eje y motor, unidos a la base del paralelo	67
Ilustración 75 Tarea del manipulador paralelo-serie	68
Ilustración 76 Comportamiento del servomotor P en el trayecto de acercamiento a la meta	70
Ilustración 77 Comportamiento del servomotor Q en el trayecto de acercamiento a la meta	71
Ilustración 78 Comportamiento del servomotor R en el trayecto de acercamiento a la meta	71
Ilustración 79 Comportamiento del servomotor del hombro en el trayecto de acercamiento a la meta	72
Ilustración 80 Comportamiento del servomotor del hombro en el trayecto de acercamiento a la meta	72
Ilustración 81 Comportamiento del servomotor de la muñeca en el trayecto de acercamiento a la meta	73

Apéndices

Ilustración 82 Comportamiento del servomotor del giro de la muñeca en el trayecto de acercamiento a la meta	73
Ilustración 83 Subsistema generador del movimiento de retorno a la postura inicial	74
Ilustración 84 0 y 25% de la tarea realizada	75
Ilustración 85 50 y 75% de la tarea realizada	75
Ilustración 86 100% de la tarea realizada	75
Ilustración 87 Trayectoria ejecutada por el robot serie en el plano XZ	76
Ilustración 88 Punto inicial y final en la simulación del robot paralelo delta	76
Ilustración 89 Ejecución de rutina de tomar objeto al (respectivamente) 0, 50 y 100%	77
Ilustración 90 Secuencia de ejecución de la tarea, imágenes tomadas de videos realizados simultáneamente	79
Ilustración 91 Simulaciones del movimiento del manipulador híbrido realizando una trayectoria de prueba	79
Ilustración 92 Ejecución de la tarea vista desde distinto ángulo	117
Ilustración 93 Ejecución de la tarea, distinto ángulo, nótese la secuencia generada	118

Referencias

- [1] A. Gamero, «La Piedra de Sísifo,» 2 Abril 2014. [En línea]. Available: <http://lapiedradesisifo.com/2014/04/02/los-alucinantes-inventos-de-her%C3%B3n-en-el-siglo-i-d-c/>. [Último acceso: 24 Julio 2016].
- [2] J. L. Molina Marticorena, «Profesor Molina,» 21 Julio 2016. [En línea]. Available: <http://www.profesormolina.com.ar/tecnologia/robotica/historia.htm>.
- [3] A. O. Baturone, de *Manipuladores y robots móviles*, Barcelona España, Marcombo Boixareu Editores, 2001, p. 594.
- [4] M. Á. Cárdenas Verdugo, *Diseño, construcción y control de un robot hexápodo*, México: UNAM, 2011.
- [5] A. Barrientos, L. F. Peñin, C. Balaguer y R. Aracil, *Fundamentos de robótica*, Mc Graw Hill, 1997.
- [6] R. Sánchez y U. d. Huelva, «Conceptos y definiciones en cinemática,» Septiembre 2008. [En línea]. Available: <http://www.uhu.es/rafael.sanchez/ingenieriamaquinas/carpetaapuntes.htm/Apuntes%20Tema%201.pdf>. [Último acceso: 4 Diciembre 2016].
- [7] H. I. M. Castillo, «Movilidad del Mecanismo Esférico Tipo 4R,» *Acta Universitaria*, vol. 13, nº 2, pp. 1-11, Mayo 2003.
- [8] I. Bonev, «ParalleMIC, The True Origins of Parallel Robots,» [En línea]. Available: <http://www.parallemic.org/Reviews/Review007.html>. [Último acceso: 4 Diciembre 2016].
- [9] M. Inc, «What is a parallel robot?,» 2016. [En línea]. Available: <http://www.mecademic.com/What-is-a-parallel-robot.html>. [Último acceso: 18 Septiembre 2016].
- [10] V. R. B. Sotelo, J. R. G. Sánchez y R. S. Ortigoza, «Robots Móviles: Evolución y Estado del Arte Polibits,» 2007. [En línea]. Available: <http://www.redalyc.org/articulo.oa?id=402640448003>. [Último acceso: 17 Septiembre 2016].
- [11] I. R. N. Roland Siegwart, «Locomotion,» de *Introduction to Autonomous Mobile Robots*, London, England, The MIT Press, 2004, p. 336.
- [12] «Sistemas de locomoción de robots móviles,» [En línea]. Available: http://www.esi2.us.es/~vivas/ayr2iaei/LOC_MOV.pdf. [Último acceso: 17 Septiembre 2016].

Referencias

- [13] V. T. Mora, «Centro universitario UAEM Zumpango,» Septiembre 2015. [En línea]. Available: <https://ri.uaemex.mx/bitstream/handle/20.500.11799/35197/secme-21956.pdf?sequence=1>. [Último acceso: 4 Diciembre 2016].
- [14] A. E. Suárez Arriaga y A. M. Sánchez Balpuesta, *Plataforma Móvil Omnidireccional de 4 Llantas Suecas (Mecanum) en configuración AB*, CD.MX: UNAM, 2015.
- [15] R. S. Ortigoza, R. G. Sánchez, R. B. Sotelo, M. A. M. Vilchis, V. M. H. Guzmán y G. S. Ortigoza, «Una panorámica de los robots móviles. Télématique,» 2007. [En línea]. Available: <http://publicaciones.urbe.edu/index.php/telematique/article/view/833/2039>. [Último acceso: 17 Septiembre 2016].
- [16] T. Bräunl, «Driving robots,» de *Embedded Robotics*, Berlín Alemania, Springer, 2006, pp. 97-454.
- [17] R. G. Sánchez, «Universidad de Almería, Robótica Móvil,» [En línea]. Available: http://www.ual.es/personal/rgonzalez/papers/Robotica_movil.pdf. [Último acceso: 17 Septiembre 2016].
- [18] W. Mark y A. Haruhiko, «International Conference on Robotics and Automation. Desing of a Holonomic Omnidirectional Vehicle,» Nice France Mayo 1992. [En línea]. Available: <http://ftp.math.fu-berlin.de/pub/Rojas/omniwheel/00220328.pdf>. [Último acceso: 18 Septiembre 2016].
- [19] K. Tanigaki, T. Fujiura, A. Akase y J. Imagawa, «Cherry-Harvesting Robot,» de *Computers and Electronics in Agriculture*, Osaka, Japón, 2008, pp. 65-72.
- [20] P. Martínez Zamudio, *Análisis Cinemático de un Manipulador Paralelo Híbrido Tipo Delta*, México: UNAM, 2009.
- [21] S. A. A. Moosavian, A. Pourreza y K. Alipour, *Kinematics and Dynamics of a Hybrid Serial-Parallel Mobile Robot*, Teherán, Irán, 2009.
- [22] I. C. Cruz López y J. D. D. Flores Méndez, *Robot Manipulador Paralelo Híbrido de 7 GDL: Cinemática, Dinámica y Seguimiento de una trayectoria*, México: UNAM, 2013.
- [23] J. J. Craig, *Introduction to Robotics Mechanics and Control*, New Jersey: Pearson Prentice Hall, 2005.
- [24] V. J. González Villela, «Evadiendo Obstáculos con Robots Móviles,» *Revista Digital Universitaria*, vol. 6, nº 1, p. 9, 2005.
- [25] I. Hernández, *Manipulador móvil omnidireccional, su coordinación de movimientos en ambientes inteligentes*, CD.MX: UNAM, 2016.