

SEP

SECRETARÍA DE
EDUCACIÓN PÚBLICA



INSTITUTO TECNOLÓGICO DE CD GUZMAN

TESIS

TEMA:

**Uso de plataformas para la Simulación y Estudio de Robots
Móviles con Brazo Manipulador**

PARA OBTENER EL GRADO:

MAESTRO EN INGENIERIA ELECTRONICA

PRESENTA:

ING. ERIKA BERENICE LÁZARO FLORES

DIRECTOR DE TESIS:

DR. JAIME JALOMO CUEVAS

CODIRECTOR DE TESIS:

DR. JOSÉ ABEL CHOCOTECO CAMPOS

Cd. Guzmán, Jalisco, México, 2020



*El mejor placer de la vida es hacer las cosas
que la gente dice que no podemos hacer. **Walter Bagehot.***

*Cuando quieres realmente una cosa, todo el Universo conspira para ayudarte a
conseguirla. **Pablo Coelho.***



Agradecimientos

En primer lugar, doy gracias a Dios por la oportunidad de poder cursar la maestría y dado la fuerza para seguir adelante hasta poder concluir y obtener el título y sobre todo por superar todas las adversidades durante ese tiempo.

Agradezco a mi esposo Javier Dimas por todo el apoyo incondicional que tuvo hacia mí para poder concluir mis estudios. A mis hijos que fueron el motor para seguir adelante. A mi madre que siempre estuvo ahí y no dejarme sola. A todos y cada uno de los integrantes de la familia que estuvieron presente durante esta etapa de mi carrera profesional.

Agradezco a el CONACYT, al Tecnológico Nacional de México que, a través del Instituto Tecnológico de Ciudad Guzmán que me dieron las herramientas necesarias, así como el apoyo económico para terminar la maestría.

Mi más sincero agradecimiento a el Dr. Jaime Jalomo Cuevas por haber sido el director de tesis, al Dr. José Abel Chocoteco ya que junto con mi director me guiaron e instruyeron para el presente proyecto de tesis.

A el Dr. Sergio Sandoval y al Dr. Humberto Bracamontes (+) les agradezco haber mostrado el mejor interés y contribuir a la culminación de los estudios de maestría.



Resumen

El campo de los robots manipuladores ha experimentado un alto desarrollo desde la década de los setenta, y la denominada robótica móvil, que ha cobrado una importancia creciente durante los años ochenta y noventa.

En la actualidad en México, los sistemas para almacenar y clasificar objetos en los almacenes son operados por personal con una experiencia y capacitación limitadas, los sistemas robóticos que se utilizan son escasos y limitados, por lo tanto la baja respuesta dinámica o no óptima de los sistemas robóticos conduce a tiempos muertos y a un consumo elevado de energía que a su vez generarán problemas ambientales, es por esa razón que se propone realizar una investigación utilizando controles inteligentes no lineales de los robots móviles con brazo manipulador para incrementar los índices de respuesta dinámica.

La presente tesis propone la simulación y el estudio del movimiento, utilizando a la vez tres plataformas experimentales, del objeto de estudio es el Kuka youBot, un robot móvil con brazo manipulador. Para posteriormente pasar a probar diferentes tipos de controles, tales como: PID y entre otros tipos; a nivel de software, con el propósito de hacer un controlador en físico, que mejore la eficacia e incrementar los índices de respuesta dinámica de este tipo de sistema robótico.



Abstract

The field of manipulator robots has experienced a high development since the seventies, and the so-called mobile robotics, which has gained increasing importance during the eighties and nineties.

Currently in Mexico, the systems for storing and classifying objects in the warehouses are operated by personnel with limited experience and training, the robotic systems that are used are scarce and limited, therefore the low dynamic or non-optimal response of the robotic systems leads to downtime and high energy consumption that in turn will generate environmental problems, it is for this reason that it is proposed to carry out an investigation using non-linear intelligent controls of mobile robots with manipulator arms to increase dynamic response rates.

This thesis proposes the simulation and study of movement, using three experimental platforms at the same time, the object of study is the Kuka youBot, a mobile robot with a manipulator arm. To later go on to test different types of controls, such as: PID, and among other types; at the software level, with the purpose of making a physical controller, which improves the efficiency and increases the dynamic response rates of this type of robotic system.



Índice

Capítulo 1.....	1
1.1 Introducción.....	1
1.2 Antecedentes.....	4
1.3 Metodología.....	10
1.3.1 Investigación documental.....	10
1.3.2 Adquisición de materiales.....	11
1.3.3 Implementación y desarrollo del proyecto.....	11
1.3.4 Redacción completa del proyecto de Tesis.....	11
Capítulo 2.....	12
2 Fundamentación Teórica.....	12
2.1 Robots Móviles.....	12
2.2 Robots Móviles Omnidireccionales.....	14
2.3 Tipos de Ruedas.....	15
2.3.1 Rueda Universal.....	15
2.3.2 Rueda Mecanum.....	15
2.3.3 Modos de Locomoción.....	16
2.4 Robot Móvil Manipulador.....	18
2.4.1 Método Denavit-Hartenberg.....	18
2.4.2 Cinemática Inversa.....	22
2.4.3 Robot plano de dos enlaces.....	24
2.4.4 Robot móvil (WMR) omnidireccional con las cuatro ruedas Mecanum.....	27
2.5 Modelado MM (Manipulador Móvil).....	32
2.5.2 Modelo General Cinemático.....	32
2.5.3 Modelado General Dinámico.....	35
Capítulo 3.....	36
3 Introducción a los Controladores para Manipuladores Móviles.....	36
3.4 Estructura General de Control.....	36
3.5 Control Proporcional – Derivativo de Posición con Control de Corriente.....	37
3.6 Control Proporcional – Derivativo de Posición Simplificado.....	38
3.7 Control por Torque Precalculado.....	39
3.8 Control por Campos Potenciales.....	40
Capítulo 4.....	43



4	Herramientas Virtuales	43
4.1	MATLAB ® (MATrix LABoratory)	43
4.1.1	Simulink ®.....	43
4.1.2	Simscape Multibody™	44
4.1.3	V- Realm Editor.....	45
4.2	SOLIDWORKS.....	46
4.2.1	Módulos de SolidWorks.....	47
4.2.2	Estudio de Movimiento	48
4.3	Blender	50
4.4	Integración con LabVIEW.....	51
4.4.1	Toolkits y módulos de LabVIEW	53
Capítulo 5.....		55
5	KUKA youBot	55
5.1	Descripción general.....	55
5.2	Modelo cinemático de la plataforma.....	57
5.3	Modelo cinemático del manipulador	60
5.4	Dinámica del robot.....	63
5.4.1	Descripción de la plataforma	63
Capítulo 6.....		71
6	Plataforma Experimental de Simulación	71
6.1	Modelado en Blender.....	71
6.2	Modelado V- Realm (VRML) – Simulink	75
6.3	SolidWorks- Simscape Multibody	76
6.3.1	Procedimiento de enlace de SolidWorks-Simulink (MATLAB)	76
Capítulo 7.....		80
7.	Resultados de la Simulación	80
7.1	Conclusiones.....	81
7.2	Trabajos Futuros.....	81
Referencias Bibliográficas.....		83
Anexos		85
Definiciones y Abreviaturas		90

Lista de Figuras

Fig.1 El robot móvil con ruedas Shakey de SRI que está equipado con lógica integrada, una cámara, un sensor de telémetro y un detector de impacto.	5
Fig.2 El Stanford Cart.	6
Fig.3 El robot explorador Dante II.	6
Fig.4 El rover robótico Sojourner de la NASA.	6
Fig.5 A) Seekur (350 kg, dimensions 1.4 m31.3 m31.1 m), (B) Pioneer 3D-X, and (C) PowerBot.	7
Fig. 8 TALÓN	8
Fig. 9 TEOdor	8
Fig. 10 TeleMAX	8
Fig. 11 Komodo	8
Fig. 12 Konabot	9
Fig. 13 VALI I VALI II	9
Fig. 14 Hilare 2 bis	9
Fig. 15 Little Helper	9
Fig. 16 Ejemplos de sistemas de locomoción.	13
Fig. 17 Manipulador móvil: robot móvil + brazo robótico	13
Fig. 18 Robots Móviles Omnidireccionales	14
Fig. 19 Robot Omnidireccional Uranus con ruedas suecas o mecanum.	14
Fig. 20 Rueda Universal Omnidireccional. Diseños de la rueda universal.	15
Fig. 21 Rueda Mecanum Omnidireccional	16
Fig. 22 Rueda Móvil Omnidireccional. Configuración estándar.	17
Fig. 23 Modos de locomoción de un robot móvil omnidireccional. Modos básicos de locomoción.	17
Fig. 24 Modos de locomoción en el plano 2-D.	18
Fig. 25 Un enlace robótico entre las articulaciones.	20
Fig. 26 Un enlace robótico entre las articulaciones.	20
Fig. 27 Representación de la posición y orientación del efector final por la matriz 4×4 T .	21
Fig. 28 Los cinco tipos de brazos robóticos industriales.	22



Fig. 29 (A) Robot plano de dos grados de libertad y (B) geometría para encontrar el modelo cinemático inverso (codo hacia abajo, codo hacia arriba).	24
Fig. 30 Robot plano de dos enlaces.	26
Fig. 31 WMR de cuatro ruedas mecanum (A) Geometría cinemática (B) Un WMR real de cuatro ruedas mecanum.	28
Fig. 32 (A) El mecanum WMR de cuatro ruedas y las fuerzas que actúan sobre él y (B) un prototipo experimental de WMR mecanum de cuatro ruedas.	31
Fig. 33 Características Geométricas de un MM con plataforma de transmisión diferencial.	32
Fig. 34 Modelo general para control de robot manipulador	36
Fig.35 Bloques de estrategia de posición PD con control de corriente	38
Fig. 36 Diagrama de bloques de control PD	38
Fig. 37 Diagrama de bloques de control de torque precalculado	39
Fig. 38 Alcance del objetivo con control de campos potenciales	41
Fig. 39 Ventana principal de Simulink – Matlab.	44
Fig.40 Detección y actuación de articulaciones	45
Fig. 41 Ventana principal de V-Real Builder	46
Fig. 42 Área de trabajo de SolidWorks	47
Fig. 43 Interfaz de SolidWorks Motion	49
Fig. 44 Tipos de estudio de movimiento	50
Fig. 45 Modelando en Blender. Unir y separar objetos	51
Fig. 56 Panel frontal de LabVIEW	52
Fig. 57 Ejemplo de diagrama de bloques en LabVIEW	52
Fig. 58 KUKA youBot	56
Fig. 59 Motor de la fábrica maxon motor.	56
Fig. 60 El software de control del robot dispone de una serie de interfaces abiertas	57
Fig. 61 Medidas de la plataforma móvil	58
Fig.62 Rueda Mecanum de KUKA youBot.	59
Fig. 63 Plataforma móvil. Movimiento.	59
Fig. 64 Parámetros articulares de la parte del manipulador	60
Fig. 65 Posición de los marcos de referencia para cada una de las articulaciones.	60
Fig. 66 Las posiciones relativas de las juntas (mm) del modelo 3D. Kuka youBot	62
Fig. 67 Representación esquemática del KUKA youBot	63



Fig. 68 La base youBot con ruedas Mecanum	64
Fig 69 Modelado del Kuka youBot en Blender	71
Fig 70 Kuka YouBot en BGE	71
Fig 71. Entorno de la simulación	72
Fig 72 Parte de la secuencia de la simulación	72
Fig 73 El gripper toma el objeto	73
Fig 74 Parte final de la secuencia de la simulación de movimiento. Deja el objeto en otro lugar	73
Fig 75 Lógica realizada en el BGE de Blender	74
Fig. 76 a) Lógica de simulación, control del Kuka YouBot por teclado	74
Fig. 77 Simulink- Entrando al sistema "mundo" VRML-Builder	75
Fig. 78 "Mundo" VRML. Sistema físico YouBot	76
Fig. 79 Entorno y sistema físico YouBot en SolidWorks	77
Fig. 80 Sistema físico en SimMechanics parte móvil	77
Fig. 81 Sistema físico SimMultibody del MM	78
Fig. 82 Kuka youBot sin controles desde Simulink de Matlab	79



Lista de Tablas

Tabla A1. Posición de los marcos de referencia del manipulador	62
Tabla A2. Parámetros de Denavit-Hartenberg para el manipulador de youBot	62
Tabla A3. platform Kuka youBot characteristics	88
Tabla A4. platform Kuka youBot characteristics, part 2	88
Tabla A5. Kuka youBot arm characteristics	89
Tabla A6. Kuka youBot, gripper data	89

Capítulo 1

1.1 Introducción

En la actualidad hemos llegado a un uso habitual de autómatas o robots en entornos principalmente industriales. La rápida implantación de los robots se ha visto propiciada por su facilidad para adaptarse a tareas repetitivas en entornos estructurados. Sin embargo, en el país los sistemas robóticos que se utilizan son escasos y limitados (montacargas habitual) por lo tanto, la baja respuesta dinámica o no óptima de los sistemas robóticos conduce a tiempos muertos y a un consumo elevado de energía que a su vez generarán problemas ambientales, es por esa razón que se propone realizar una investigación utilizando controles inteligentes no lineales en un sistema robótico autónomo en específico de los robots móviles con brazo manipulador para incrementar los índices de respuesta dinámica.

Los índices de respuesta dinámica de los robots móviles con brazo manipulador se pueden optimizar utilizando controles inteligentes no lineales. Los índices de respuesta dinámica son: Tiempo de asentamiento (cuando la salida alcanza su valor final y se queda ahí un tiempo determinado o mínimo, sin error en estado permanente y ninguna componente oscilatoria entre instantes); tiempo de retardo (el valor de retardo se elige de manera que asegure una ley de control implementada sin el esquema predictor- observador no sea capaz de resolver el problema); y al máximo sobre impulso (tiempo en el cual la respuesta alcanza el primer pico del sobrepaso, el tiempo pico es inversamente proporcional a la frecuencia natural amortiguada).

Todos los en los robots manipuladores como en la robótica móvil existen puntos de interés común: el modelado cinemático, el modelado dinámico, el control (arquitecturas, algoritmos...), la planificación, el reconocimiento del entorno, etc.

Así pues, la presente tesis surge ante la necesidad de dar respuesta a las cuestiones cinemáticas (modelado, control, ...) y dinámicas de los robots móviles con brazo manipulador a través del estudio de eficiencia por técnicas no lineales y de inteligencia artificial.



La mayoría de los robots manipuladores son brazos articulados generalmente se modelan, desde el punto de vista cinemático, con matrices de transformación homogénea entre sistemas de coordenadas. En el modelado dinámico se manejan distintas formulaciones y métodos: LaGrange-Euler, Newton-Euler, etc.

Para mejorar sus prestaciones se investiga en técnicas para identificar los modelos dinámicos eficientemente y en métodos de control de articulaciones que compensan no-linealidades y acoplamientos, así como en optimización dinámica y control adaptativo para distintas condiciones de trabajo.

Un aspecto importante que se debe considerar es que los modelos cinemáticos dependen de los ángulos de las ruedas con articulación de dirección, por lo que hay valores especiales de estos ángulos para los que se produce la singularidad del modelo, que implica deslizamiento o pérdida de movilidad en el vehículo. Para el modelado dinámico diversos autores los han abordado con deslizamiento.

Para el control del robot móvil algunos autores han planteado métodos geométricos y otros la utilización de herramientas de la teoría de control clásica: aproximación lineal; linealización exacta; controlabilidad; estabilidad por Lyapunov; control adaptativo; control predictivo, etc.

En otra cuestión, es seguimiento de línea es una aplicación muy habitual en robótica, utilizándose cables o sensores ópticos. Se ha ido incrementando el uso de los sistemas de visión para esta aplicación con los últimos avances tecnológicos; tiene ventajas respecto a otros métodos con la desventaja que es costosa asociado al tratamiento de imagen: Umbralización, segmentación, detección de contornos, etc.

Sin embargo, debido a los elevados consumos de energía de los sistemas robóticos en este trabajo se propone realizar el estudio de eficiencia de los controles inteligentes no lineales de los robots móviles con brazo manipulador aplicados en las tareas de la clasificación y almacenamiento de artículos de un almacén.



El objetivo de la tesis a futuro es realizar el estudio comparativo de eficiencia de los controles inteligentes no lineales de los robots móviles con brazo manipulador para la clasificación de artículos en los almacenes. Para ello se inicia con la presente tesis con la utilización de diferentes plataformas para la simulación y estudio del robot móvil con brazo manipulador móvil.

El trabajo de tesis se divide en 7 capítulos, que se describen a continuación:

Capítulo 1. Se describe la Introducción y Antecedentes acerca de la robótica, en especial centrándose en los robots móviles con brazo manipulador, así como la metodología a utilizar.

Capítulo 2. En este capítulo se describe los fundamentos matemáticos necesarios para realizar el modelamiento cinemático y dinámico del robot móvil manipulador con brazo manipulador y una descripción de las características técnicas y de funcionamiento del objeto de estudio de la presente tesis.

Capítulo 3. Se presenta el objeto de estudio del Kuka youBot, la cinemática y dinámica del mismo, así como el modelado 3D.

Capítulo 4. Se encuentran las plataformas virtuales de simulación que se utilizaron para el modelado del robot Kuka youBot.

Capítulo 5. Resultados de Simulación. Describe que es lo que se realizó y hasta donde se llegó en cada una de las plataformas utilizada, también, que se podría hacer para sacar más provecho a las simulaciones.

Capítulo 6. Conclusiones y trabajos futuros. Son los resultados que se obtuvieron al utilizar las diferentes plataformas y cuáles son los trabajos y actividades que deben continuar para que se llegue al objetivo deseado.



1.2 Antecedentes

Los robots móviles son robots que pueden moverse de un lugar a otro de forma autónoma, es decir, sin la ayuda de operadores humanos externos. A diferencia de la mayoría de los robots industriales que solo pueden moverse en un espacio de trabajo específico, los robots móviles tienen la característica especial de moverse libremente dentro de un espacio de trabajo predefinido para lograr los objetivos deseados. Esta capacidad de movilidad los hace adecuados para un amplio repertorio de aplicaciones en entornos estructurados y no estructurados. Los robots móviles terrestres se distinguen en robots móviles con ruedas (WMR) y robots móviles con patas (LMR). Los robots móviles también incluyen vehículos aéreos no tripulados (UAV) y vehículos submarinos autónomos (AUV). Los WMR son muy populares porque son apropiados para aplicaciones típicas con complejidad mecánica y consumo de energía relativamente bajos. Los robots con patas son adecuados para tareas en entornos no estándar, escaleras, montones de escombros, etc. Normalmente, los sistemas con dos, tres, cuatro o seis patas son de interés general, pero también existen muchas otras posibilidades. Los robots de una sola pierna encuentran aplicaciones raras porque solo pueden moverse saltando. Los robots móviles también incluyen manipuladores móviles (robots con ruedas o patas equipados con uno o más manipuladores ligeros para realizar diversas tareas).

La historia de los robots se puede dividir en dos periodos generales [1,2]:

- Período antiguo y preindustrial;
- Período industrial y robosapien.
- ***Período antiguo y preindustrial***

El primer robot en la historia mundial (alrededor de 2500 3000 aC) es la criatura mecánica de la mitología griega llamada Talos (“Τάλως”) [3]. Este nombre se atribuye tanto a un ser humano (hijo de Pérdica, hermana de Dédalo) como a un ente artificial mecánico construido por Hefesto, por orden de Zeus, con cuerpo de

bronce y una sola vena desde el cuello hasta el tobillo, donde se encuentra en el clavo de cobre lo bloqueó. Alrededor del 350 a. C., el amigo de Platón Arquitas de Tarento construyó un pájaro mecánico (“paloma”) que era propulsado por vapor. Esto representa uno de los primeros estudios históricos

- **Período industrial y robosapien:**

Este período comienza en 1954 cuando George Devol, Jr. patentó su brazo robótico multiarticulado (el primer robot moderno). En 1956, junto con Joseph Engelberger fundaron la primera empresa de robots del mundo llamada Unimation (de Universal Automation).

1970: Este es el año de inicio de la robótica móvil. El robot móvil Shakey fue desarrollado en el Stanford Research Institute (hoy conocido como SRI Technology), controlado por algoritmos inteligentes que observan a través de sensores y reaccionan a sus propias acciones. Shakey se conoce como "la primera persona electrónica". El nombre Shakey se debe a su movimiento brusco (Fig. 1).



Fig.1 El robot móvil con ruedas Shakey de SRI que está equipado con lógica integrada, una cámara, un sensor de telémetro y un detector de impacto.

1979: El Stanford Cart, originalmente diseñado en 1970 como seguidor de línea, es reconstruido por Hans Moravec y equipado con una visión tridimensional más robusta que permite una mayor autonomía (Fig. 2). En un experimento, el carro de Stanford cruzó una habitación llena de sillas de forma autónoma utilizando una cámara de televisión que estaba tomando fotografías desde varios ángulos. Estas imágenes fueron procesadas por una computadora para analizar la distancia del carro a los obstáculos.

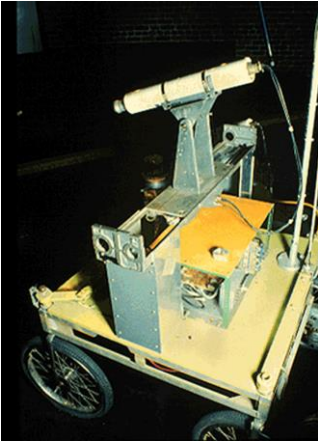


Fig.2 El Stanford Cart.

1990-1999: Durante esta década se produce el surgimiento de los “robots exploradores”. Estos robots iban donde el humano no visitaba antes o se consideraba demasiado arriesgado o inconveniente. Ejemplos de tales robots son Dante (1993) y Dante II (1994) que exploraron el Monte Erebrus en la Antártida y el Monte Spurr en Alaska (Fig. 3). Un ejemplo de misiones planetarias de la NASA destinadas a estudiar el clima y la geología del Planeta Rojo (Marte) es Path Finder. El Mars Observer aterrizaría cerca de las latitudes del polo sur, llevando dos instrumentos científicos y un módulo de aterrizaje (rover robótico) (Fig.4)



Fig.3 El robot explorador Dante II.

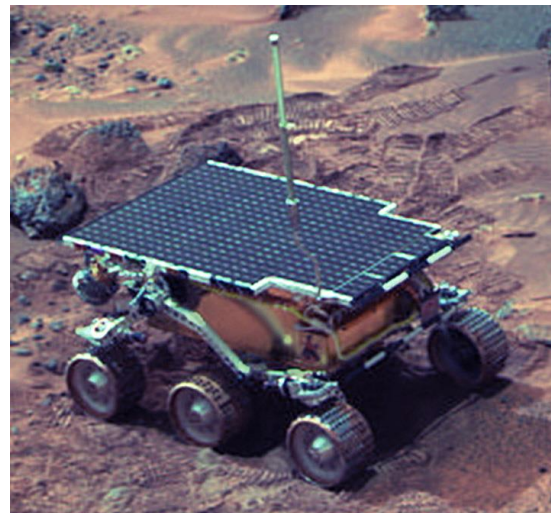


Fig.4 El rover robótico Sojourner de la NASA.

Las tres plataformas robóticas móviles actualmente disponibles comercialmente con fines de investigación son las siguientes (Fig. 5):

- Seekur: una gran plataforma robótica holonómica para todo clima para seguridad, inspección e investigación.
- Pioneer 3-DX: una plataforma totalmente programable equipada con codificadores de motores y 16 sondas ultrasónicas (frontales y traseras). Se utiliza para investigación y desarrollo rápido (localización, seguimiento, navegación, control, etc.).
- PowerBot: un robot de accionamiento diferencial de alta carga útil (hasta 100 kg) para investigación y creación rápida de prototipos en universidades e institutos de investigación.

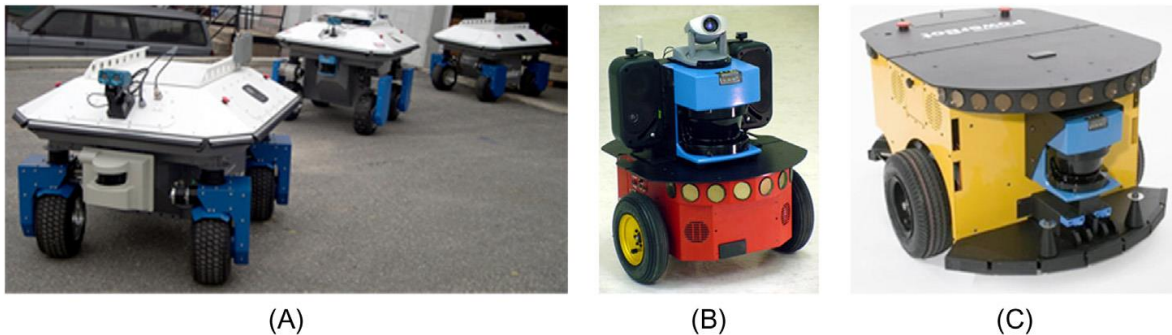


Fig.5 A) Seekur (350 kg, dimensions 1.4 m31.3 m31.1 m), (B) Pioneer 3D-X, and (C) PowerBot.

Las aplicaciones de la robótica móvil han tenido una evolución importante en diferentes campos, en especial en el campo militar. Los sistemas de robots construidos a partir de un brazo manipulador montado sobre una plataforma es el término que se le da a manipulador móvil, los más conocidos son los siguientes:

- *TALÓN*. Diseñado para efectuar desde misiones de reconocimiento, entrega de armas, detección de minas, inspecciones de vehículos y vigilancia en ruta, es muy versátil y ligero. El control es a través de *joystick*. [2].
- *TEOdor*. Es un sistema formado por dos robots el de eliminación de artefactos explosivos y un robot de observación, cuenta con sistema para

disposición de bombas; puede operar en condiciones de temperatura de -20°C a +60°C [3].



Fig. 8 TALÓN



Fig. 9 TEOdor

- *TeleMAX*. Se usa para operaciones que se llevan a cabo en espacios estrechos. Se controla mediante un dispositivo ergonómico de pulgar parecido a un joystick. [4].
- *Komodo*. Es un sistema robótico utilizado para la investigación. [5].

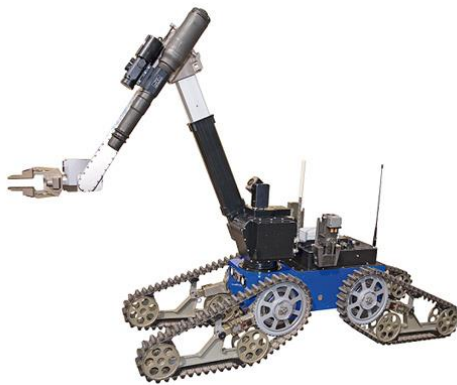


Fig. 10 TeleMAX



Fig. 11 Komodo

En el campo académico destacan los siguientes:

- *Konabot*. Para ser utilizado para desactivar artefactos explosivos y de forma remota, por internet [6].
- *VALI*. Vehículo Ligero Antiexplosivos, surge con la necesidad de contar con un robot móvil para evitar poner en riesgo la vida humana, desactivando explosivos y en acciones antiterroristas [7].
- *Hilare 2bis*. Se desarrolló el Hilare 2bis en Francia para experimentaciones multi-robot, y que actualmente está equipado con un brazo GT6A (6 grados de libertad) y, un remolque [8].
- *Little Helper*. Se puede integrar en diversas aplicaciones en entornos de producción existentes, ya que es capaz de servir a los equipos de producción habitual y llevar a cabo las operaciones relacionadas con el trabajo [9].



Fig. 12 Konabot



Fig. 13 VALI I VALI II

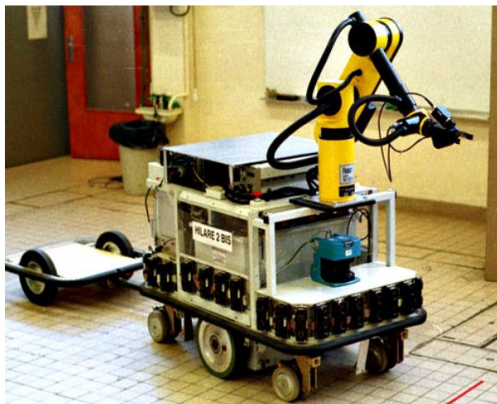


Fig. 14 Hilare 2 bis



Fig. 15 Little Helper



1.3 Metodología

Objetivo General

Construir e implementar un controlador físico para un robot móvil con brazo manipulador.

Objetivos Específicos

- Realizar las simulaciones en las diferentes plataformas.
- Estudiar el robot móvil con brazo manipulador y hacer pruebas a nivel software, incluyendo la implementación de controles.
- Construir el controlador físico y realizar pruebas de laboratorio.
- Implementar el controlador físico.

1.3.1 Investigación documental.

- Realizar investigación documental de los conceptos básicos de la robótica, estado del arte y benchmarking;
- que nos permita adquirir información útil sobre el contexto actual del control PID estándar y los controles inteligentes no lineales para robots móviles con brazo manipulador;
- centrada en la cinemática de los robots móviles con brazo manipulador y en los índices de respuesta dinámica;
- sobre el objeto de estudio, en este caso el robot Kuka youBot, su cinemática y dinámica.
- Elaboración del protocolo del anteproyecto, para posteriormente presentarlo para su aprobación.
- Realizar correcciones a protocolo de anteproyecto y elaborar el documento final.
- Realizar la investigación documental sobre herramientas de software, hardware y herramientas matemáticas.



1.3.2 Adquisición de materiales

- Una vez que se encuentre el tipo de control más adecuado se realizará la verificación de los materiales a necesitar para construir el controlador para su requisición.
- Buscar y cotizar alternativas de dispositivos electrónicos que te permitan poder continuar con el proyecto.

1.3.3 Implementación y desarrollo del proyecto

- Desarrollar el modelo matemático de robot móvil con brazo manipulador.
- Elaborar el entorno de simulación de robot móvil e implementar el control (a nivel software)
- Buscar alternativas de uso de plataformas.
- Probar diversos controles para el robot móvil Kuka youBot (nivel software).
- Recolectar datos de simulación y de las pruebas de los controles aplicados en la simulación del robot Kuka youBot.
- Realizar la comparación del de eficiencia entre el control PID estándar y un control inteligente.
- Construir el controlador.
- Realizar pruebas en físico en laboratorio.
- Realizar pruebas en físico en condiciones de campo.

1.3.4 Redacción completa del proyecto de Tesis

- Realizar la redacción del documento de Tesis conforme se avance en el proyecto.
- Reporte de resultados en artículos científicos
- Análisis y revisión del reporte para los artículos científicos para su envío.
- Realizar las correcciones que se indiquen de los artículos científicos.
- Realizar las correcciones necesarias al documento de reporte de Tesis.
- Entregar el reporte de Tesis corregido y hacer los trámites posteriores.



Capítulo 2

2 Fundamentación Teórica

2.1 Robots Móviles

Un robot móvil puede trabajar en condiciones en las cuales no siempre es posible que un operario humano pueda acceder, es decir, ambientes peligrosos. Su arquitectura está conformada por una combinación de subsistemas, son los siguientes:

- **Sensores:** Son los que se encargan de primero adquirir y luego muestrear las señales físicas de su propia estructura y de su entorno, tomando la información útil para su control.
- **Locomoción:** Movimiento del robot.
- **Control:** Analiza la información adquirida, calcula y envía las señales que dirigen los movimientos que debe realizar.
- **Comunicación:** Vigila el intercambio de señales entre los subsistemas.

La característica más significativa del robot móvil es su autonomía, es decir, mantenerse con cierta posición y orientación con la capacidad que tiene su controlador de ejecutar acciones que lo hacen poder mantenerse y que pueda interactuar con el entorno; el grado de autonomía dependen del conocimiento del entorno y su algoritmo de control, éste último, depende del tipo de configuración de su sistema de locomoción. La capacidad que tiene el robot móvil para realizar el seguimiento de una trayectoria (regulación o estabilización) y planificación de trayectoria (navegación en espacios desconocidos sólo con la información recopilada de los sensores; son los problemas de control de la robótica móvil.

Para el primer caso se necesita diseñar una ley de control que permita al robot desplazarse desde un punto inicial hasta otro final considerando como se mencionó su posición y orientación para garantizar la estabilidad, también considerando el caso de seguimiento de trayectorias es necesario diseñar una ley de control que permita al robot desplazarse desde un punto inicial hasta otro final, considerando

su posición y su orientación para garantizar la estabilidad, también considerar la velocidad en un tiempo y dividiendo la trayectoria en segmentos de línea recta o círculo, es decir, su cinemática. Para el segundo caso es tomar en cuenta las restricciones propias del robot (diseño, configuración, limitantes físicas) y el entorno que desarrollará su trayectoria y si puede encontrar obstáculos, que recalculé trayectoria sin alejarse de la original [3].

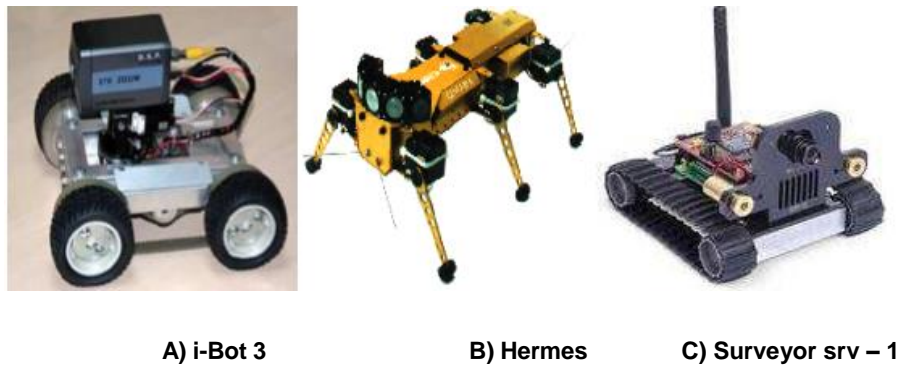


Fig. 16 Ejemplos de sistemas de locomoción.

Los robots móviles también incluyen los robots manipuladores móviles, ya sean acoplados sobre robots con ruedas, con patas o de orugas (tipos de locomoción), equipados con uno o más manipuladores ligeros para realizar varias tareas, de acuerdo al entorno que se desarrolla [10].



Fig. 17 Manipulador móvil: robot móvil + brazo robótico

2.2 Robots Móviles Omnidireccionales

Un robot omnidireccional posee máxima maniobrabilidad en el plano, puede moverse en cualquier dirección sin necesidad de rotar o reorientarse, a diferencia de otros tipos de robots que requieren girar y cambiar de dirección para llegar a un punto deseado.

Este tipo de robots a diferencia de otros que requieren girar o cambiar de dirección para llegar a un punto deseado sin necesidad de rotar o reorientarse, pueden ser construidos usando tres, cuatro o más ruedas omnidireccionales (Fig. 18). Los que usan tres ruedas son de tipo universal con un ángulo de 90° , con control, dirección simple y, estabilidad limitada; los de cuatro ruedas usan de tipo mecanum tienen mecánica y controles más complejos, pero, con mayor estabilidad y tracción (Fig. 19) [10].

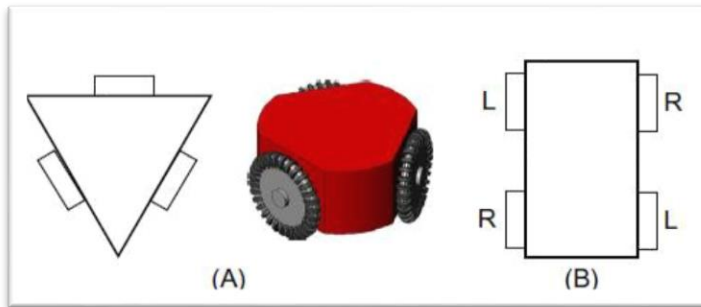


Fig. 18 Robots Móviles Omnidireccionales
(A) Caso de tres ruedas (B) Caso de cuatro ruedas.

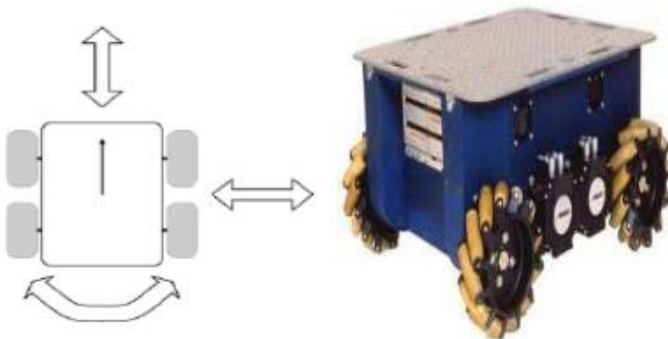


Fig. 19 Robot Omnidireccional Uranus con ruedas suecas o mecanum.

2.3 Tipos de Ruedas

2.3.1 Rueda Universal

Tiene una combinación de movimiento restringido y no restringido durante el giro. Contiene pequeños rodillos alrededor de su diámetro exterior que se montan perpendicular al eje (rotación) de la rueda; de esta manera la rueda puede rodar en la dirección paralela al eje, con rotación de una rueda normal [10].



Fig. 20 Rueda Universal Omnidireccional. Diseños de la rueda universal.

2.3.2 Rueda Mecanum

Fue inventada por el ingeniero sueco Bengt Ilon en 1973 durante su trabajo en la compañía sueca Mecanum AB, por esta razón, también se la conoce como rueda Ilon o rueda sueca [10].

Es similar a la rueda universal, excepto que los rodillos están montados en un ángulo α que no sea 90° , por lo general $\pm 45^\circ$ (Fig.21), (A) y 2.2 (B) muestra la rueda omnidireccional como se ve desde la parte inferior (a través de una mesa de cristal). La fuerza F producida por la rotación de la rueda actúa sobre el suelo a través del rodillo que tiene contacto con el suelo (se supone suficientemente plano y sin irregularidades) [10].

En este rodillo, la fuerza se descompone en una fuerza F_1 paralelo al eje del rodillo y una fuerza F_2 perpendicular al eje del rodillo. La fuerza perpendicular al eje del rodillo produce una pequeña rotación de los rodillos (velocidad v_r), pero

la fuerza paralela al eje del rodillo ejerce una fuerza sobre la rueda y por lo tanto sobre el vehículo resultando en la velocidad central v_h . La velocidad actual v_t del vehículo es la combinación de v_h y v_r [10].

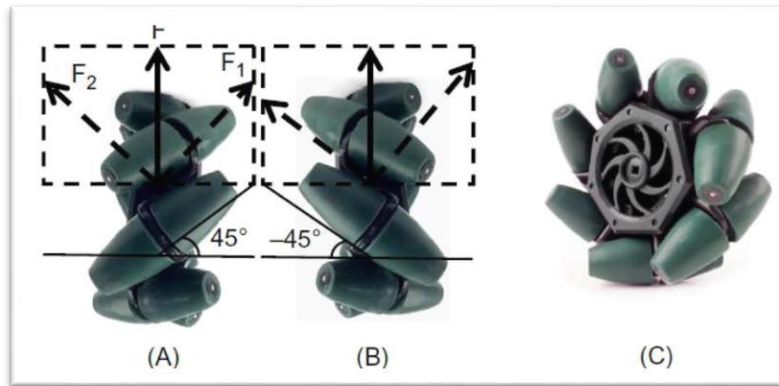


Fig. 21 Rueda Mecanum Omnidireccional

(A) Rueda Mecanum con $\alpha = 45^\circ$ (rueda izquierda), (B) Rueda Mecanum con $\alpha = -45^\circ$ (rueda derecha), y (C) rueda Mecanum actual.

2.3.3 Modos de Locomoción

Las cuatro ruedas en un robot omnidireccional tienen una distribución típica como se observa en la (Fig.22). De las cuatro ruedas dos de ellas son conocidas como ruedas zurdas (lado izquierdo L) y las otras dos ruedas diestras (lado derecho R). Las ruedas zurdas tienen un ángulo de rodillo $\alpha = 45^\circ$ y las diestras un ángulo $\alpha = -45^\circ$.

En la Figura 2.8 se muestra los seis movimientos básicos de un robot móvil omnidireccional de cuatro ruedas, etiquetados como: (A) movimiento adelante, (B) deslizamiento izquierdo, (C) giro (sentido horario), (D) de movimiento hacia atrás, (E) deslizamiento derecho y (F) giro (sentido anti-horario). Las flechas a la izquierda y el lado derecho del vehículo muestran la dirección de movimiento de las ruedas correspondientes [13].

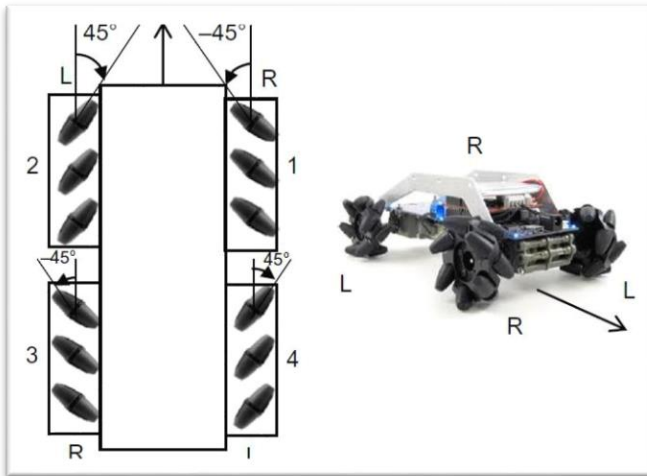


Fig. 22 Rueda Móvil Omnidireccional. Configuración estándar.

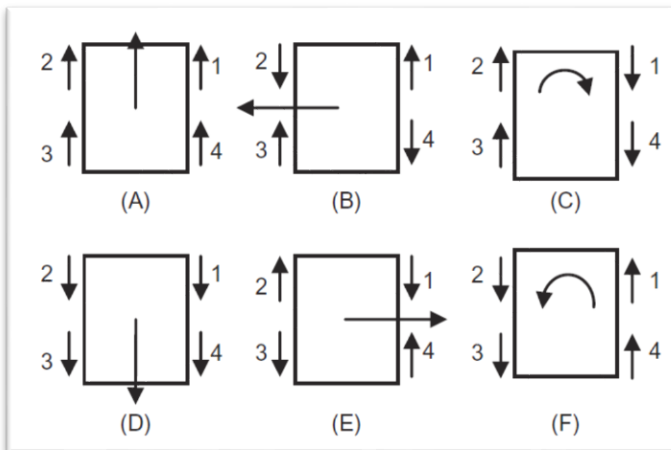


Fig. 23 Modos de locomoción de un robot móvil omnidireccional. Modos básicos de locomoción.

Las flechas sobre la plataforma del vehículo muestran las direcciones del movimiento del robot móvil con ruedas, es decir para la dirección hacia adelante del vehículo todas las ruedas deben moverse hacia adelante (como se muestra en la (Fig. 23) (A), para el deslizamiento a la izquierda las ruedas 1, 3 deberían moverse hacia adelante y las ruedas 2, 4 hacia atrás y su opuesto para el deslizamiento hacia la derecha.

Las locomociones mostradas en la (Fig. 24) se obtienen si todas las ruedas se mueven en la misma velocidad. Mediante variación de magnitud de velocidad de las ruedas se puede realizar movimientos en cualquier dirección en el plano 2-D. Se muestran unos pocos casos de locomoción en planos 2D [13].

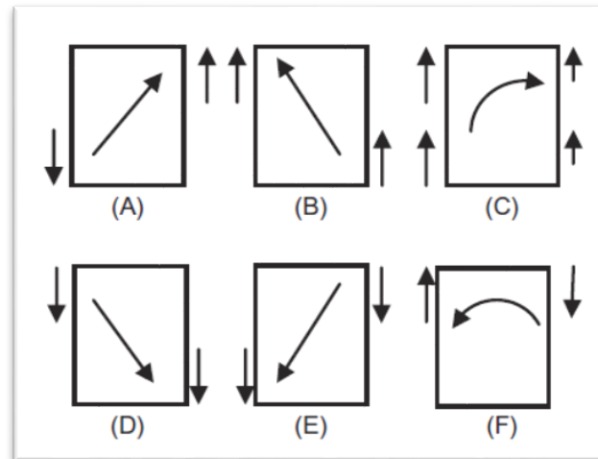


Fig. 24 Modos de locomoción en el plano 2-D.

2.4 Robot Móvil Manipulador

Los manipuladores móviles (MM) son sistemas robóticos que consisten en brazos articulados (manipuladores) montados en plataformas móviles holonómicas o no holonómicas. Aportan la destreza de los primeros y la ampliación del espacio de trabajo de los segundos. Por lo tanto, los MM son atractivos para muchas aplicaciones y hoy en día constituyen el cuerpo principal de los robots de servicio.

Uno de los problemas principales y más desafiantes en la investigación de MM es diseñar controladores precisos para todo el sistema. Debido a la fuerte interacción y acoplamiento del subsistema de la plataforma móvil y el(los) brazo(s) manipulador(es) montado(s) en la plataforma, se necesita una adecuada coordinación entre los respectivos controladores [1,10].

2.4.1 Método Denavit-Hartenberg

El método de Denavit Hartenberg proporciona un procedimiento sistemático para determinar la posición y orientación del efector final de un manipulador robótico de n articulaciones, es decir, para calcular X^0 .

Considere un vínculo i que se encuentra entre las articulaciones i e $i + 1$ de un robot como se muestra en la (Fig. 25). Cada vínculo se describe por la distancia



a_i entre los ejes (posiblemente no paralelos) i e $i + 1$ de las dos articulaciones, y el ángulo de rotación α_i del eje i al eje $i + 1$ con respecto a la normal común de los dos ejes. Cada articulación (prismática o rotatoria) es accionada por un motor (traslacional o rotacional) que produce el movimiento del eslabón i . En general, el brazo robótico tiene n articulaciones y $n + 1$ enlaces. La relación funcional del efector final y los desplazamientos de las articulaciones se pueden encontrar usando la convención de parámetros que se muestra en la (Fig. 26), llamada parámetros de Denavit Hartenberg.

Estos parámetros se refieren a la posición relativa de los marcos de coordenadas $O_{i-1}x_{i-1}y_{i-1}z_{i-1}$ y $O_ix_iz_i$ son las siguientes:

- La longitud a_i de la normal común $\Sigma_i O_i$.
- La distancia d_i entre el origen de O_{i-1} y el punto Σ_i .
- El ángulo α_i entre la articulación i (es decir, el eje z_{i-1}) y el eje z_i en la dirección positiva (en el sentido de las agujas del reloj).
- El ángulo θ_i entre el eje x_{i-1} y la normal común (es decir, la rotación sobre el eje z_{i-1}) en la dirección positiva.

En base a lo anterior, la transferencia del marco $O_{i-1}x_{i-1}y_{i-1}z_{i-1}$ al marco $O_ix_iz_i$ se puede realizar en cuatro pasos:

1. Rotación del marco $i - 1$ alrededor del eje z_{i-1} por un ángulo θ_i .
2. Traslación del marco $i - 1$ a lo largo del eje z_{i-1} por d_i .
3. Traslación del eje rotado x_{i-1} (que ahora coincide con x_i) a lo largo de la normal común por a_i .
4. Rotación alrededor de x_i por un ángulo α_i .

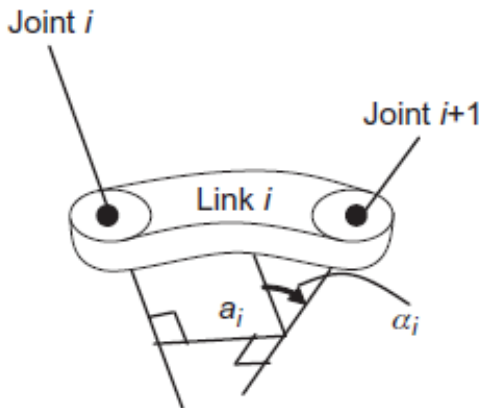


Fig. 25 Un enlace robótico entre las articulaciones

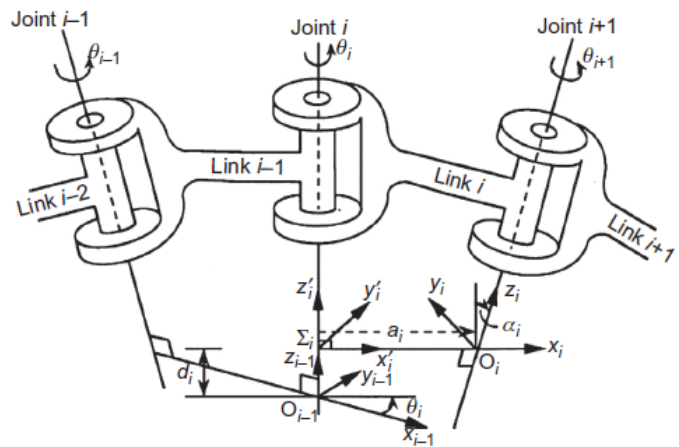


Fig. 26 Un enlace robótico entre las articulaciones

Denotando por A_i^* el resultado de los pasos 3 y 4 y por A_i^{i-1} el resultado de los pasos 1 y 2, el resultado general de los pasos 1 a 4 viene dado por:

$$A_i^{i-1} = A_i^* A_i^{i-1}$$

$$= \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

donde A_i^{i-1} da la posición y orientación del marco i con respecto al marco $i - 1$. Las primeras tres columnas de A_i^{i-1} contienen los cosenos directores de los ejes del marco i , mientras que la cuarta columna representa la posición del marco O_i .

En general, el desplazamiento de la articulación i se denota como q_i , donde:

$q_i = \theta_i$ para articulación giratoria,

$q_i = d_i$ para articulación prismática.

La posición y orientación del enlace i con respecto al enlace $i - 1$ es una función de q_i , es decir, $A_i^{i-1}(q_i)$.

La ecuación cinemática de un brazo robótico da la posición y orientación del último eslabón con respecto al marco de coordenadas de la base, y obviamente contiene todas las variables q_1, q_2, \dots, q_n generalizadas de las juntas. (Fig. 27) muestra gráficamente los marcos de coordenadas consecutivos desde la base hasta el efector final de la cadena cinemática robótica en serie.

La matriz T dada por:

$$T = A_1^0(q_1)A_2^1(q_2) \dots A_n^{n-1}(q_n)$$

representa la posición y orientación del efector final (que es el eslabón final con respecto a la base). Ahora es fácil determinar T para todos los tipos de robots (Fig. 28).

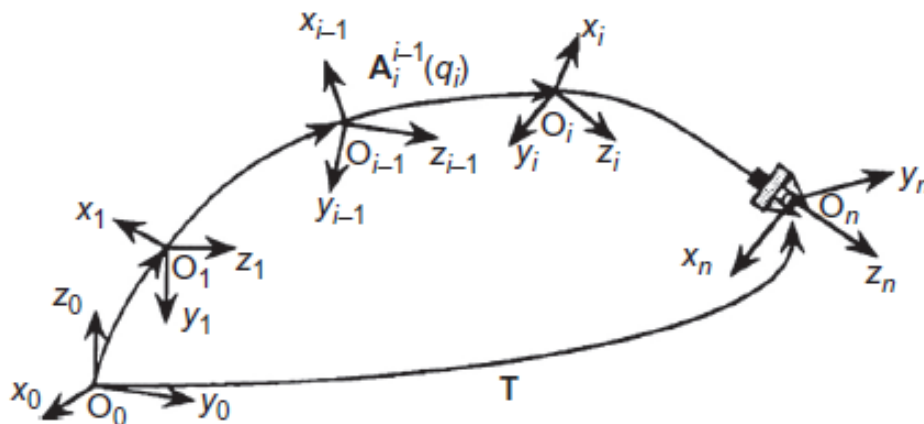


Fig. 27 Representación de la posición y orientación del efector final por la matriz 4x4 T .

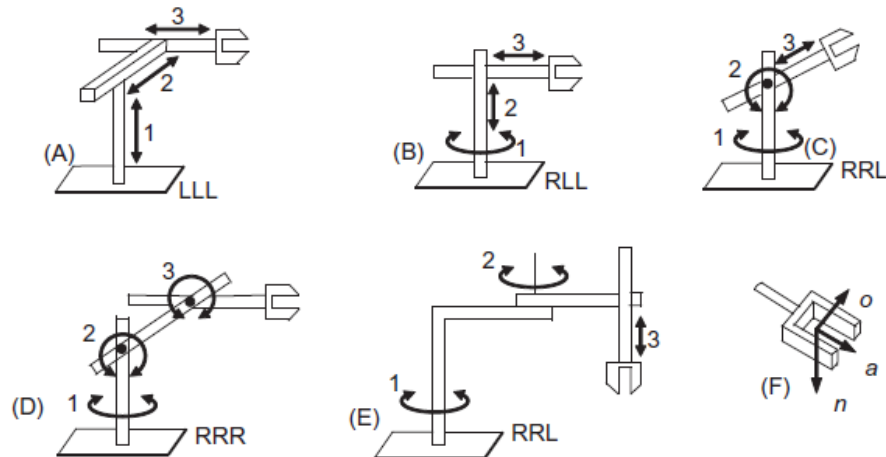


Fig. 28 Los cinco tipos de brazos robóticos industriales: (A) Cartesiano, (B) Cilíndrico, (C) Esférico (polar), (D) Articulado (antropomórfico), (E) Robot SCARA y (F) Marco de coordenadas del efector final. L denota movimiento lineal (de traslación) y R denota movimiento de rotación.

2.4.2 Cinemática Inversa

En el problema de cinemática directa estamos encontrando T conociendo los valores de q_1, q_2, \dots, q_n . En el problema de cinemática inversa hacemos lo contrario, $q_i (i = 1, 2, \dots, n)$. La cinemática directa se puede escribir en forma vectorial:

$$X = f(q)$$

donde X es el vector de seis dimensiones:

$$X = \begin{bmatrix} p \\ \psi \end{bmatrix}, p = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \psi = \begin{bmatrix} \psi_x \\ \psi_y \\ \psi_z \end{bmatrix}$$

de la posición p del efector final y la orientación ψ , f es una función vectorial de columna no lineal de seis dimensiones, y $q = q_1, q_2, \dots, q_n$.

Por lo tanto, la ecuación cinemática inversa es: $q = f^{-1}(X)$.



donde \mathbf{f}^{-1} denota la función inversa habitual de $\mathbf{f}(\cdot)$. Un método práctico sencillo para invertir la ecuación cinemática. Partimos de:

$$\mathbf{T} = \mathbf{A}_1^0(\mathbf{q}_1)\mathbf{A}_2^1(\mathbf{q}_2) \dots \mathbf{A}_6^5(\mathbf{q}_6)$$

y se obtiene la siguiente secuencia de ecuaciones:

$$\begin{aligned} (\mathbf{A}_1^0)^{-1}\mathbf{T} &= \mathbf{T}_6^1 \\ (\mathbf{A}_2^1)^{-1}(\mathbf{A}_1^0)^{-1}\mathbf{T} &= \mathbf{T}_6^2 \\ (\mathbf{A}_3^2)^{-1}(\mathbf{A}_2^1)^{-1}(\mathbf{A}_1^0)^{-1}\mathbf{T} &= \mathbf{T}_6^3 \\ (\mathbf{A}_4^3)^{-1}(\mathbf{A}_3^2)^{-1}(\mathbf{A}_2^1)^{-1}(\mathbf{A}_1^0)^{-1}\mathbf{T} &= \mathbf{T}_6^4 \\ (\mathbf{A}_5^4)^{-1}(\mathbf{A}_4^3)^{-1}(\mathbf{A}_3^2)^{-1}(\mathbf{A}_2^1)^{-1}(\mathbf{A}_1^0)^{-1}\mathbf{T} &= \mathbf{T}_6^5 \end{aligned}$$

Los elementos de los lados izquierdos de estas ecuaciones son funciones de los elementos de \mathbf{T} y las primeras $i - 1$ variables del robot. Los elementos de los lados derechos son constantes o funciones de las variables q_i, q_{i+1}, \dots, q_6 . De cada ecuación matricial obtenemos 12 ecuaciones, es decir, una ecuación para cada uno de los elementos de los cuatro vectores $\mathbf{n}, \mathbf{o}, \mathbf{a}$, y $\mathbf{p} = \mathbf{x}_0$. A partir de estas ecuaciones podemos determinar los valores de $q_i (i = 1, 2, \dots, 6)$ del robot.

Aunque la solución del problema de cinemática directa es única, no lo es para el problema de cinemática inversa, debido a la presencia de funciones trigonométricas. En algunos casos, la solución se puede encontrar analíticamente, pero, en general, la solución solo se puede encontrar de manera aproximada usando algún método numérico aproximado y la computadora. Además, si el robot tiene más de seis grados de libertad (es decir, si tenemos un robot redundante), hay infinitas soluciones para de $q_i (i = 1, 2, \dots, n; n > 6)$ que conducen a la misma posición y orientación del efector final.

2.4.3 Robot plano de dos enlaces

2.4.3.1 Cinemática

El modelo cinemático de este robot se puede encontrar mediante un cálculo trigonométrico simple, obteniendo el modelo cinemático directo:

$$x(\theta_1, \theta_2) = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2)$$

$$y(\theta_1, \theta_2) = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

es decir:

$$\mathbf{p} = \mathbf{f}(\boldsymbol{\theta}), \quad \mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$

Para derivar el modelo cinemático inverso $\boldsymbol{\theta} = \mathbf{f}^{-1}$ en la (Fig. 29 B) Usando la regla de cosenos, encontramos:

$$x^2 + y^2 = l_1^2 + l_2^2 - 2l_1l_2 \cos(180^\circ - \theta_2)$$

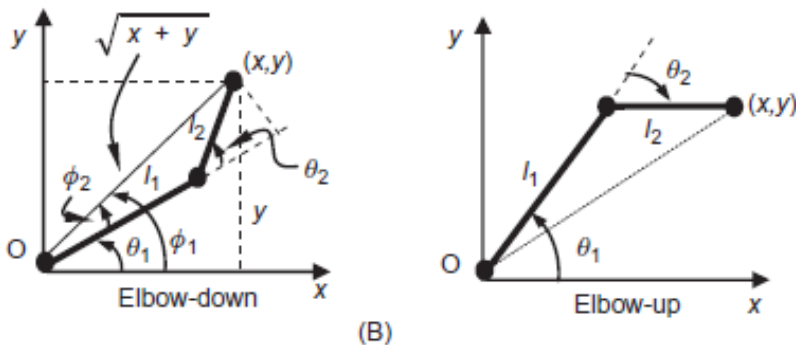
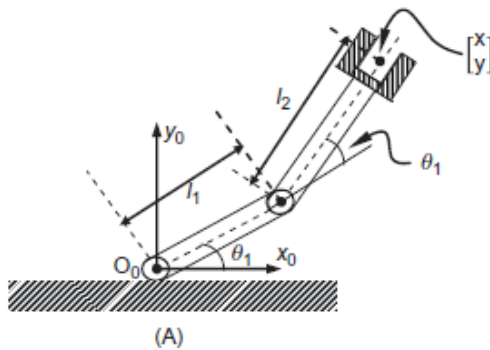


Fig. 29 (A) Robot plano de dos grados de libertad y (B) geometría para encontrar el modelo cinemático inverso (codo hacia abajo, codo hacia arriba).



a partir del cual se encuentra que el ángulo θ_2 es:

$$\theta_2 = \arccos[(x^2 + y^2 - l_1^2 + l_2^2)/2l_1l_2]$$

El ángulo θ_1 equivale a:

$$\theta_1 = \phi_1 - \phi_2$$

donde:

$$\tan \phi_1 = \frac{y}{x}, \quad \tan \phi_2 = \frac{l_2 \sin \theta_2}{l_1 + l_2 \cos \theta_2}$$

Por lo tanto:
$$\theta_1 = \arctan\left(\frac{y}{x}\right) - \arctan\left(\frac{l_2 \sin \theta_2}{l_1 + l_2 \cos \theta_2}\right)$$

En realidad, tenemos dos configuraciones que conducen a la misma posición \mathbf{p} del efector final, codo hacia abajo y codo hacia arriba, como se muestra en la (Fig. 29). Cuando $(x, y) = (0, 0)$, que se puede obtener si $l_1 = l_2$, la relación x/y no está definida [10].

2.4.3.2 Dinámica

Para derivar el modelo dinámico del robot aplicamos directamente el método de LaGrange. Considere la notación de la (Fig. 30)

Los símbolos θ_1 y θ_2 tienen el significado usual, m_1 y m_2 son las masas de los dos eslabones (concentradas en sus centros de gravedad), y l_1 y l_2 son las longitudes de los eslabones. El símbolo l_{ci} denota la distancia del centro de gravedad (COG) del i -ésimo eslabón al eje de la articulación i , y \tilde{I}_i denota el momento de inercia del eslabón i con respecto al eje que pasa por el COG de este eslabón y es perpendicular al plano xy (paralelo al eje z). Aquí, $q_1 = \theta_1$ y $q_2 = \theta_2$, y las energías cinemática y potencial de los eslabones 1 y 2 están dadas por:

$$K_1 = \frac{1}{2} m_1 l_{c1}^2 \dot{\theta}_1^2 + \frac{1}{2} \tilde{I}_1 \dot{\theta}_1^2, \quad P_1 = m g l_{c1} S_1$$

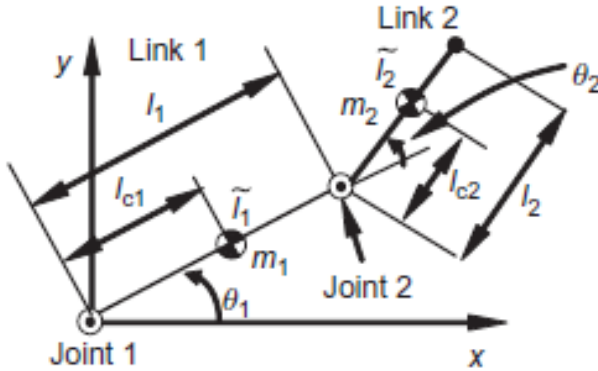


Fig. 30 Robot plano de dos enlaces.

$$K_2 = \frac{1}{2} m_2 \dot{s}_1^T \dot{s}_2 + \frac{1}{2} \tilde{I}_2 (\dot{\theta}_1^2, \dot{\theta}_1 \dot{\theta}_2), \quad P_2 = m g (l_1 S_1 + l_{c2} S_{12})$$

Usando la función Lagrangiana del robot $L = K_1 + K_2 - P_1 - P_2$:

$$D_{11} \ddot{\theta}_1 + D_{12} \ddot{\theta}_2 + h_{122} \dot{\theta}_2^2 + 2h_{112} \dot{\theta}_1 \dot{\theta}_2 + g_1 = \tau_1, \quad D_{21} \ddot{\theta}_1 + D_{22} \ddot{\theta}_2 + h_{211} \dot{\theta}_1^2 + g_2 = \tau_2$$

donde τ_1 y τ_2 son los pares externos aplicados a las articulaciones 1 y 2. El coeficiente D_{ii} es la inercia efectiva de la junta i , D_{ij} es la inercia de acoplamiento de las articulaciones (juntas) i y j , h_{jj} es el coeficiente de fuerza centrífuga, h_{jk} ($j \neq k$) es el coeficiente de aceleración de Coriolis de la articulación i debido a las velocidades de las articulaciones j y k , y g_i ($i = 1, 2$) representan los torques debidos a la gravedad. Las relaciones dinámicas se pueden escribir en la forma compacta estándar:

$$\mathbf{D}(\boldsymbol{\theta}) \ddot{\boldsymbol{\theta}} + \mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{g}(\boldsymbol{\theta}) = \boldsymbol{\tau}$$

$$\mathbf{D}(\boldsymbol{\theta}) = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}, \quad \mathbf{g}(\boldsymbol{\theta}) = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}, \quad \boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

$$\mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \text{col} \left[\sum_{j=1}^2 \sum_{k=1}^2 \left(\frac{\partial D_{ij}}{\partial \theta_k} - \frac{1}{2} \frac{\partial D_{jk}}{\partial \theta_i} \right) \dot{\theta}_j \dot{\theta}_k \right]$$

donde $\text{col}[h_i]$ denota un vector columna con elementos $h_i (i = 1, 2)$. En el caso especial, donde se supone que las masas de los eslabones m_1 y m_2 están concentradas al final de cada eslabón, tenemos $l_{c1} = l_1$ y $l_{c2} = l_2$. Es fácil verificar las propiedades descritas y la anti-simetría de $\mathbf{D} - 2\mathbf{C}$, donde K es la energía cinética total $K = K_1 + K_2$ del robot [10].

2.4.4 Robot móvil (WMR) omnidireccional con las cuatro ruedas Mecanum

2.4.4.1 Modelo Cinemático

Considere el WMR de 4 ruedas de la figura 2.14, donde las ruedas mecanum tienen un ángulo de giro de $\pm 45^\circ$ [11]. Aquí, tenemos marcos de coordenadas de cuatro ruedas O_{ci} ($i = 1, 2, 3, 4$). La velocidad angular \dot{q}_i de la rueda i tiene tres componentes:

1. $\dot{\theta}_{ix}$: velocidad de rotación alrededor del eje
2. $\dot{\theta}_{ir}$: velocidad de rotación del rodillo i
3. $\dot{\theta}_{iz}$: velocidad de giro de la rueda alrededor del punto de contacto

El vector de velocidad de la rueda $\mathbf{v}_{ci} = [\dot{x}_{ci}, \dot{y}_{ci}, \dot{\phi}_{ci}]^T$ en O_{ci} coordenadas viene dada por:

$$\begin{bmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \\ \dot{\phi}_{ci} \end{bmatrix} = \begin{bmatrix} 0 & r_i \sin \alpha_i & 0 \\ R_i & -r_i \cos \alpha_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_{ix} \\ \dot{\theta}_{ir} \\ \dot{\theta}_{iz} \end{bmatrix}$$

para $i = 1, 2, 3, 4$, donde R_i es el radio de la rueda, r_i es el radio del rodillo y α_i el ángulo del rodillo. El vector de velocidad $\dot{\mathbf{P}}_Q = [\dot{x}_Q, \dot{y}_Q, \dot{\phi}_Q]^T$ del robot en el marco de coordenadas Ox_Qy_Q es:

$$\dot{\mathbf{P}}_Q = \begin{bmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{\phi}_Q \end{bmatrix} = \begin{bmatrix} \cos \phi_{ci}^Q & -\sin \phi_{ci}^Q & d_{ciy}^Q \\ \sin \phi_{ci}^Q & \cos \phi_{ci}^Q & -d_{cix}^Q \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \\ \dot{\phi}_{ci} \end{bmatrix}$$

donde ϕ_{ci}^Q denota el ángulo de rotación (orientación) del marco O_{ci} con respecto a Qx_Qy_Q , y d_{cix}^Q, d_{ciy}^Q son las traslaciones de O_{ci} con respecto a Qx_Qy_Q .

$$\dot{\mathbf{P}}_Q = \mathbf{J}_i \dot{\mathbf{q}}_i \quad (i = 1, 2, 3, 4)$$

donde $\dot{\mathbf{q}}_i = [\dot{\theta}_{ix}, \dot{\theta}_{iy}, \dot{\theta}_{iz}]^T$, y,

$$\mathbf{J}_i = \begin{bmatrix} -R_i \sin \phi_{ci}^Q & r_i \sin(\phi_{ci}^Q + \alpha_i) & d_{ciy}^Q \\ R_i \cos \phi_{ci}^Q & -r_i \cos(\phi_{ci}^Q + \alpha_i) & -d_{cix}^Q \\ 0 & 0 & 1 \end{bmatrix}$$

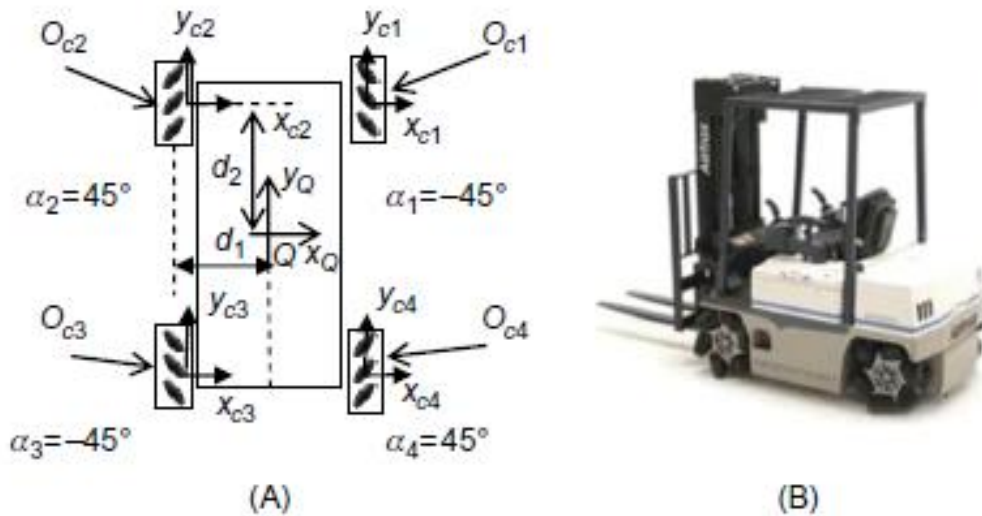


Fig. 31 WMR de cuatro ruedas mecanum (A) Geometría cinemática (B) Un WMR real de cuatro ruedas mecanum.

es la matriz Jacobiana de la rueda i , que es cuadrada e invertible. Si todas las ruedas son idénticas (excepto por la orientación de los rodillos), los parámetros cinemáticos del robot en la configuración que se muestra en la (Fig. 31) son:

$$\begin{aligned} \mathbf{R}_i &= \mathbf{R}, & r_i &= r, & \phi_{ci}^Q &= 0 \\ |d_{cix}^Q| &= d_1, & |d_{ciy}^Q| &= d_2 \\ \alpha_1 = \alpha_3 &= -45^\circ, & \alpha_2 = \alpha_4 &= 45^\circ \end{aligned}$$



Así, las matrices Jacobianas son:

$$J_1 = \begin{bmatrix} 0 & -r\sqrt{2}/2 & d_2 \\ R & -r\sqrt{2}/2 & d_1 \\ 0 & 0 & 1 \end{bmatrix}, \quad J_2 = \begin{bmatrix} 0 & r\sqrt{2}/2 & d_2 \\ R & -r\sqrt{2}/2 & -d_1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$J_3 = \begin{bmatrix} 0 & -r\sqrt{2}/2 & -d_2 \\ R & -r\sqrt{2}/2 & -d_1 \\ 0 & 0 & 1 \end{bmatrix}, \quad J_4 = \begin{bmatrix} 0 & r\sqrt{2}/2 & -d_2 \\ R & -r\sqrt{2}/2 & d_1 \\ 0 & 0 & 1 \end{bmatrix}$$

El movimiento del robot se produce por el movimiento simultáneo de todas las ruedas. En términos de $\dot{\theta}_{ix}$ (es decir, las velocidades angulares de las ruedas alrededor de sus ejes), el vector de velocidad $\dot{\mathbf{p}}_Q$ viene dado por:

$$\begin{bmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{\phi}_Q \end{bmatrix} = \frac{R}{4} \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{d_1 + d_2} & \frac{1}{d_1 + d_2} & \frac{1}{d_1 + d_2} & \frac{1}{d_1 + d_2} \end{bmatrix} \begin{bmatrix} \dot{\theta}_{1x} \\ \dot{\theta}_{2x} \\ \dot{\theta}_{3x} \\ \dot{\theta}_{4x} \end{bmatrix}$$

El vector de velocidad del robot $\dot{\mathbf{p}} = [\dot{x}, \dot{y}, \dot{\phi}]^T$ en el marco de coordenadas mundo se obtiene como:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{\phi}_Q \end{bmatrix}$$

donde ϕ es el ángulo de rotación del marco de coordenadas de la plataforma Qx_Qy_Q , z , alrededor del eje z que es ortogonal a Qx_Qy_Q . Inviertiendo las ecuaciones obtenemos el modelo cinemático inverso, que da las velocidades angulares $\dot{\theta}_{ix}$ ($i = 1, 2, 3, 4$) de las ruedas alrededor de sus ejes necesarios para obtener una velocidad deseada $[\dot{x}, \dot{y}, \dot{\phi}]^T$ del robot:



$$\begin{bmatrix} \dot{\theta}_{1x} \\ \dot{\theta}_{2x} \\ \dot{\theta}_{3x} \\ \dot{\theta}_{4x} \end{bmatrix} = \frac{1}{R} \begin{bmatrix} -1 & 1 & (d_1 + d_2) \\ 1 & 1 & -(d_1 + d_2) \\ -1 & 1 & -(d_1 + d_2) \\ 1 & 1 & (d_1 + d_2) \end{bmatrix} \begin{bmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{\phi}_Q \end{bmatrix}$$

$$\begin{bmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{\phi}_Q \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix}$$

2.4.4.2 Modelo Dinámico

Las fuerzas totales F_x y F_y que actúan sobre el robot en las direcciones x y y son:

$$F_x = (F_{x1} + F_{x2} + F_{x3} + F_{x4})$$

$$F_y = (F_{y1} + F_{y2} + F_{y3} + F_{y4})$$

donde F_{xi} , F_{yi} , ($i = 1, 2, 3, 4$) son las fuerzas que actúan sobre las ruedas a lo largo de los ejes x y y . En ausencia de movimiento de rotación separado, la dirección del movimiento se define por un ángulo δ donde:

$$\delta = \text{tg}^{-1}(F_y/F_x)$$

El par de torque τ que produce rotación pura es:

$$\tau = (F_{x1} - F_{x2} - F_{x3} + F_{x4})d_1 + (F_{y3} + F_{y4} - F_{y1} - F_{y2})d_2$$

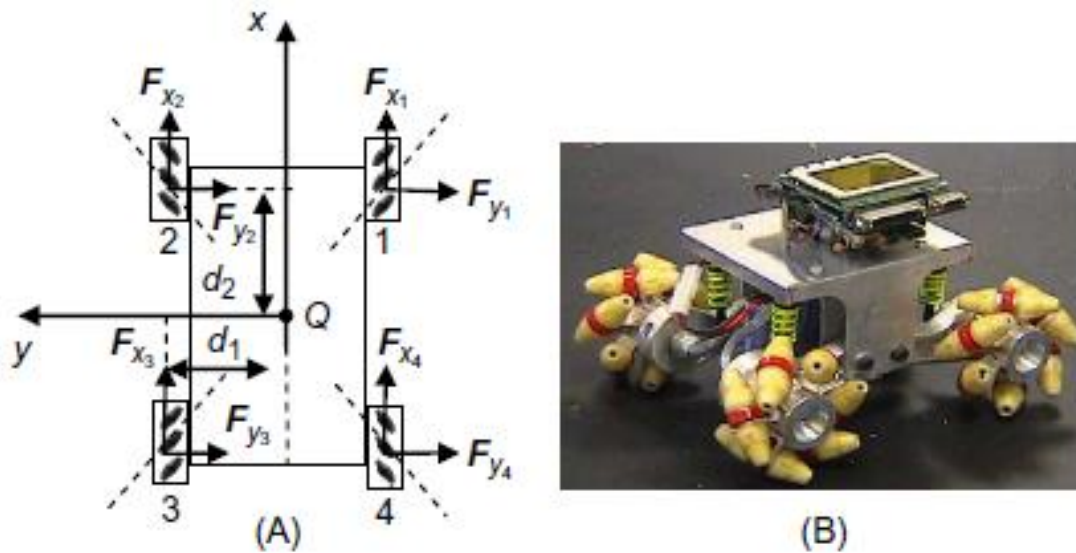


Fig. 32 (A) El mecanum WMR de cuatro ruedas y las fuerzas que actúan sobre él y (B) un prototipo experimental de WMR mecanum de cuatro ruedas.

Las ecuaciones de movimiento de Newton - Euler son:

$$m\ddot{x} = F_x - \beta_x \dot{x}$$

$$m\ddot{y} = F_y - \beta_y \dot{y}$$

$$I_Q \ddot{\phi} = \tau - \beta_z \dot{\phi}$$

donde β_x, β_y y β_z son los coeficientes de fricción lineal en el movimiento x, y , y ϕ, m, I_Q son la masa y el momento de inercia del robot. El robot puede alcanzar velocidades de estado ($\ddot{x} = 0, \ddot{y} = 0, \ddot{\phi} = 0$) $\dot{x}_{ss}, \dot{y}_{ss}$, y $\dot{\phi}_{ss}$, igual a:

$$\dot{x}_{ss} = \frac{F_x}{\beta_x}, \quad \dot{y}_{ss} = \frac{F_y}{\beta_y}, \quad \dot{\phi}_{ss} = \frac{\tau}{\beta_z}$$

2.5 Modelado MM (Manipulador Móvil)

Los modelos cinemáticos y dinámicos totales de un MM son complejos y fuertemente acoplados, combinando los modelos de la plataforma móvil y el manipulador robótico fijo [10, 11]. Hoy en día, los MM se utilizan con plataformas omnidireccionales, triciclos, similares a automóviles y de tracción diferencial que ofrecen la máxima maniobrabilidad.

2.5.2 Modelo General Cinemático

Considere el MM de la (Fig. 33) que tiene una plataforma de accionamiento diferencial y un manipulador robótico multienlace.

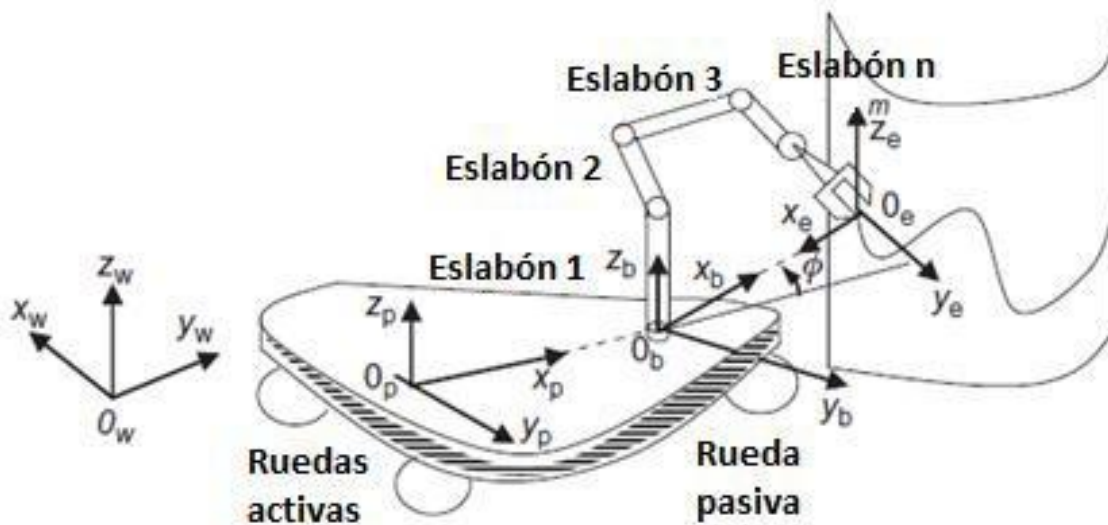


Fig. 33 Características Geométricas de un MM con plataforma de transmisión diferencial.

Se tiene los siguientes cuatro sistemas de coordenadas:

$O_w x_w y_w z_w$ es el global,

$O_p x_p y_p z_p$ el de la plataforma,

$O_b x_b y_b z_b$ es el sistema de coordenadas de la base del manipulador,

$O_e x_e y_e z_e$ es el sistema de coordenadas del efector final.



Entonces, la posición/orientación del efector final del manipulador con respecto a $O_w x_w y_w z_w$ está dada por:

$$\mathbf{T} = \mathbf{A}_p^w \mathbf{A}_b^p \mathbf{A}_e^b$$

Donde se obtiene:

\mathbf{A}_p^w es la matriz de transformación a partir de O_w a O_p ,

\mathbf{A}_b^p es la matriz de transformación a partir de O_p a O_b ,

\mathbf{A}_e^b es la matriz de transformación a partir de O_b a O_e .

En forma vectorial, el vector de posición/orientación del efector final \mathbf{x}_e^w en coordenadas mundiales tiene la forma:

$$\mathbf{x}_e^w = \mathbf{F}(\mathbf{q})$$

donde:

$$\mathbf{q} = [\mathbf{p}^T \boldsymbol{\theta}^T]^T$$

$$\mathbf{p} = [x, y, \phi]^T \text{ (plataforma)}$$

$$\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_{nm}]^T \text{ (manipulador)}$$

Por lo tanto, derivando con respecto al tiempo obtenemos:

$$\dot{\mathbf{x}}_e^w = \left(\frac{\partial \mathbf{F}}{\partial \mathbf{p}} \right) \dot{\mathbf{p}} + \left(\frac{\partial \mathbf{F}}{\partial \boldsymbol{\theta}} \right) \dot{\boldsymbol{\theta}}$$

$\dot{\mathbf{p}}$ está dada por el modelo cinemático de la plataforma [12]:

$$\dot{\mathbf{p}} = \mathbf{G}(\mathbf{p})\mathbf{u}_p, \quad \mathbf{u}_p \in \mathbb{R}^2$$

y $\dot{\boldsymbol{\theta}}$ por el modelo cinemático del manipulador. Suponga que el manipulador es de libre restricción, se describe:

$$\dot{\boldsymbol{\theta}} = \mathbf{u}_m$$



\mathbf{u}_m es el vector de comandos articulados de manipulados, se obtiene el modelo cinemático general del MM:

$$\dot{\mathbf{x}}_e^w(t) = \left(\frac{\partial \mathbf{F}}{\partial \mathbf{p}} \right) \mathbf{G}(\mathbf{p}) \mathbf{u}_p + \left(\frac{\partial \mathbf{F}}{\partial \boldsymbol{\theta}} \right) \mathbf{u}_m = \mathbf{J}(\mathbf{q}) \mathbf{u}(t)$$

donde:

$$\mathbf{u}(t) = [\mathbf{u}_p^T(t) \mathbf{u}_m^T(t)]^T \in R^{2+n_m}$$

$$\mathbf{J}(\mathbf{q}) = [\mathbf{J}_p(\mathbf{q}) \mathbf{G}(\mathbf{q}) : \mathbf{J}_m(\boldsymbol{\theta})]$$

$$\mathbf{J}_p(\mathbf{q}) = \frac{\partial \mathbf{F}}{\partial \mathbf{p}}$$

$$\mathbf{J}_m(\mathbf{q}) = \frac{\partial \mathbf{F}}{\partial \boldsymbol{\theta}}$$

Representa el modelo cinemático total del MM desde las entradas hasta las variables hasta el efector final. En realidad, en este caso, el sistema está sujeto a las restricciones no holonómica.

$$\mathbf{M}(\mathbf{p}) \dot{\mathbf{p}} = 0, \quad \mathbf{M}(\mathbf{p}) = [-\sin \phi, \cos \phi, 0, \dots, 0]$$

donde $\dot{\mathbf{p}}$ no se puede eliminar por integración. Por lo tanto, $\mathbf{J}(\mathbf{q})$ debe incluir una fila que represente esta restricción. Dado que la dimensionalidad $2 + n_m$ del vector de entrada de control $\mathbf{u}(t)$ es menor que el número total $3 + n_m$ de variables (grados de libertad) (variables) a controlar, el sistema siempre está subactuado.



2.5.3 Modelado General Dinámico

El modelo dinámico de LaGrange del MM viene dado por la ecuación.

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{M}^T(\mathbf{q}) \lambda = \mathbf{E}\boldsymbol{\tau}$$

donde $\mathbf{M}(\mathbf{q})$ viene dada por la $\mathbf{M}(\mathbf{p})\dot{\mathbf{p}} = 0$, la $\mathbf{M}(\mathbf{p}) = [-\sin \phi, \cos \phi, 0, \dots, 0]$.

Este modelo contiene dos partes, a saber:

Parte de la plataforma:

$$\mathbf{D}_p(\mathbf{q}_p, \mathbf{q}_m)\ddot{\mathbf{q}}_p + \mathbf{C}_p(\mathbf{q}_p, \mathbf{q}_m, \dot{\mathbf{q}}_p, \dot{\mathbf{q}}_m) = \mathbf{E}_p \boldsymbol{\tau}_p - \mathbf{M}^T(\mathbf{q}_p) \lambda - \mathbf{D}_p(\mathbf{q}_p, \mathbf{q}_m)\ddot{\mathbf{q}}_m$$

Parte del manipulador:

$$\mathbf{D}_m(\mathbf{q}_m)\ddot{\mathbf{q}}_m + \mathbf{C}_m(\mathbf{q}_p, \mathbf{q}_m; \dot{\mathbf{q}}_p) = \boldsymbol{\tau}_m - \mathbf{D}_m(\mathbf{q}_p, \mathbf{q}_m)\ddot{\mathbf{q}}_p$$

el subíndice p se refiere a la plataforma y m al manipulador, y los símbolos tienen el significado estándar. Aplicando la técnica que consiste en eliminar la restricción no holonómica $\mathbf{M}(\mathbf{q}_p)\dot{\mathbf{q}}_p = \mathbf{0}$, y obtenemos el modelo reducido (sin restricciones) de la forma:

$$\bar{\mathbf{D}}(\mathbf{q})\dot{\mathbf{v}} + \bar{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{v} = \bar{\mathbf{E}}\boldsymbol{\tau}$$

donde:

$$\mathbf{q} = [\mathbf{q}_p^T, \mathbf{q}_m^T]^T$$

y $\bar{\mathbf{D}}(\mathbf{q})$, $\bar{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})$ y $\bar{\mathbf{g}}(\mathbf{q})$ están dados por la combinación de los términos respectivos en las partes de la plataforma y el manipulador.

Capítulo 3

3 Introducción a los Controladores para Manipuladores Móviles

Los dos métodos de control más comunes utilizados en robótica son el controlador de derivadas proporcionales (PD) y el controlador de par predictivo (PTC). Estrictamente hablando, los servomotores utilizan un esquema de bucle de control anidado. Este método utiliza el lazo de control más interno para controlar directamente el par limitando la corriente suministrada al motor en proporción al error de velocidad. El lazo de control externo luego establece una referencia de velocidad proporcional al error de posición. En algunos casos, los actuadores electrónicos más sofisticados contienen más circuitos internos para regular la corriente [13].

3.4 Estructura General de Control

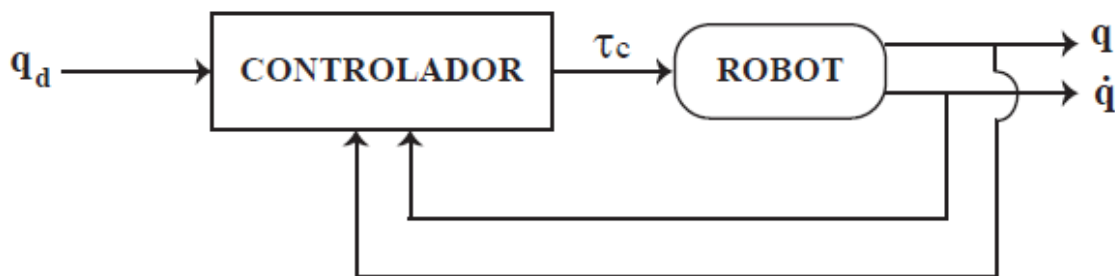


Fig. 34 Modelo general para control de robot manipulador

donde:

\mathbf{q}_d : Vector posición articular deseada.

\mathbf{q} : Vector posición articular medida.

$\dot{\mathbf{q}}$: Vector velocidad articular medida.

$\boldsymbol{\tau}$: Vector torque calculado aplicado al robot.

$\tilde{\mathbf{q}}$: Vector error de posición se define $\mathbf{q}_d - \mathbf{q}$.

$\dot{\tilde{\mathbf{q}}}$: Vector error de velocidad se define como $\dot{\mathbf{q}}_d - \dot{\mathbf{q}}$.

3.5 Control Proporcional – Derivativo de Posición con Control de Corriente

El diagrama de bloques de esta estrategia de control se muestra (Fig. 35), muestra el circuito interior de regulación de corriente del accionamiento electrónico (I), que es simplemente un circuito de lazo de control proporcional, aunque en realidad hay toda una lógica de conmutación de los transistores de potencia del variador electrónico. La corriente de referencia (I_d) se calcula a partir de la diferencia entre la velocidad de instrucción (\dot{q}_d) y la velocidad de ejecución (\dot{q}). A su vez, la velocidad deseada se calcula a partir del error de posición (\tilde{q}). Aunque este esquema se implementa en los servomotores más modernos que se fabrican en la actualidad, en aras de la simplicidad se puede suponer que el actuador electrónico tiene una función de transmisión uniforme (transferencia unitaria). Esta suposición es común porque los tiempos de respuesta de los variadores electrónicos son generalmente tres órdenes de magnitud más rápido que los sistemas mecánicos y con un buen control de corriente sin falsos positivos, suprimidos (sub-amortiguadas) y permanentes, el o los accionamientos electrónicos puede generar la corriente deseada, casi instantáneamente [13]. Por otro lado, para los accionamientos (actuadores) de imanes permanentes de CC, es posible establecer una relación directa entre el par y la corriente y, de nuevo, suponiendo simplemente que el par de una unidad es constante, la relación entre el par y la corriente es directa. Dado que la mayoría de los robots actuales utilizan transmisiones electrónicas muy rápidas con motores de CC, mencionar que de acuerdo al tipo de investigación se utilizan con un control de descarga parcial simple (sin control de corriente). Por otro lado, en algunos casos, se dispone de controladores móviles con actuadores electrónicos que no permiten controlar la corriente, sino que solo establecen la referencia de velocidad o posición en el caso de la base y solo las posiciones de referencia de proyección en el caso del brazo, aunque el fabricante indica en el manual la posibilidad de variar los umbrales de corriente. Dado que el equipo es muy costoso y modificar su propia electrónica es arriesgado y lleva más tiempo implementarlo, se decide convertir el comando de torque en comando de velocidad y/o posición de acuerdo con el modelo dinámico directo del robot.

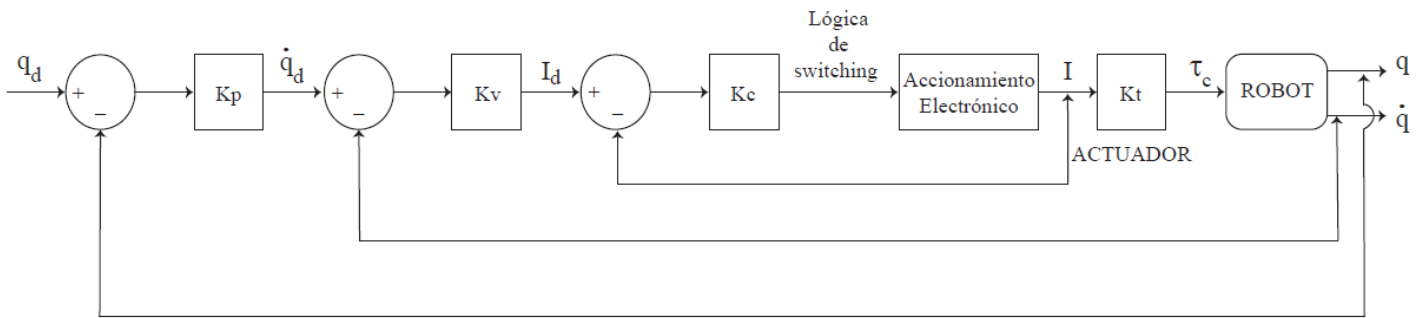


Fig.35 Bloques de estrategia de posición PD con control de corriente

3.6 Control Proporcional – Derivativo de Posición Simplificado

Es un controlador que tiene una ganancia vinculada al error de posición \tilde{q} y además una ganancia vinculada al error de velocidad $\dot{\tilde{q}}$, para crear una variable manipulada de corriente o torque, sin tener en cuenta explícitamente el control de corriente como se describió en el control anterior. El esquema de este control PD simplificado sin regulación de corriente se muestra en la (Fig.36) la cual corresponde a una utilización de la ecuación del control PD:

$$\tau_c = K_p \tilde{q} + K_v \dot{\tilde{q}}$$

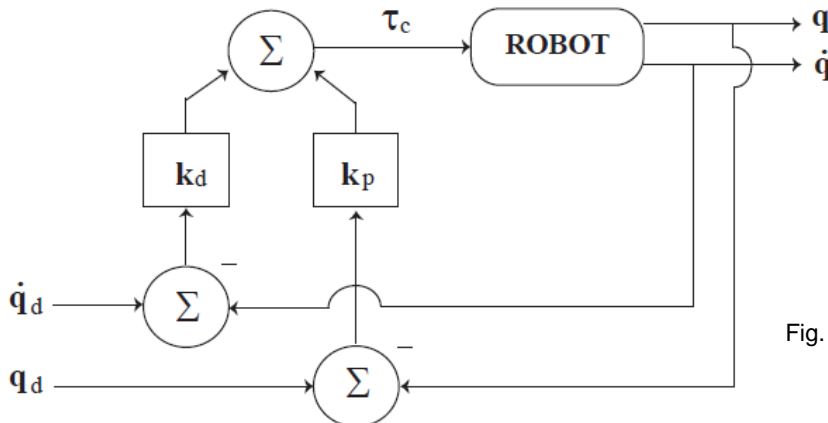


Fig. 36 Diagrama de bloques de control PD



3.7 Control por Torque Precalculado

Es un controlador con base en el modelo general para robots de N gdl. de la siguiente ecuación:

$$\tau = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})$$

La ecuación de este controlador explica un controlador no lineal el cual es obtenido de la retroalimentación lineal del modelo y que hace uso explícito del entendimiento de las matrices de este.

$$\tau_{ctp} = \mathbf{M}(\mathbf{q})[\ddot{\mathbf{q}}_d + \mathbf{K}_v\dot{\tilde{\mathbf{q}}} + \mathbf{K}_p\tilde{\mathbf{q}}] + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})$$

La ecuación de lazo cerrado se recibe sustituyendo la acción de control en la ecuación del modelo, dando como consecuencia:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})[\ddot{\mathbf{q}}_d + \mathbf{K}_v\dot{\tilde{\mathbf{q}}} + \mathbf{K}_p\tilde{\mathbf{q}}]$$

Debido a que $\mathbf{M}(\mathbf{q})$ es una matriz definida positiva e invertible [13], puede reducirse a:

$$\ddot{\mathbf{q}} + \mathbf{K}_v\dot{\tilde{\mathbf{q}}} + \mathbf{K}_p\tilde{\mathbf{q}} = \mathbf{0},$$

y en la práctica, aunque la anulación puede no ser perfecta gracias a la incertidumbre del modelo y perturbaciones existentes, el sistema puede aproximarse a un sistema lineal.

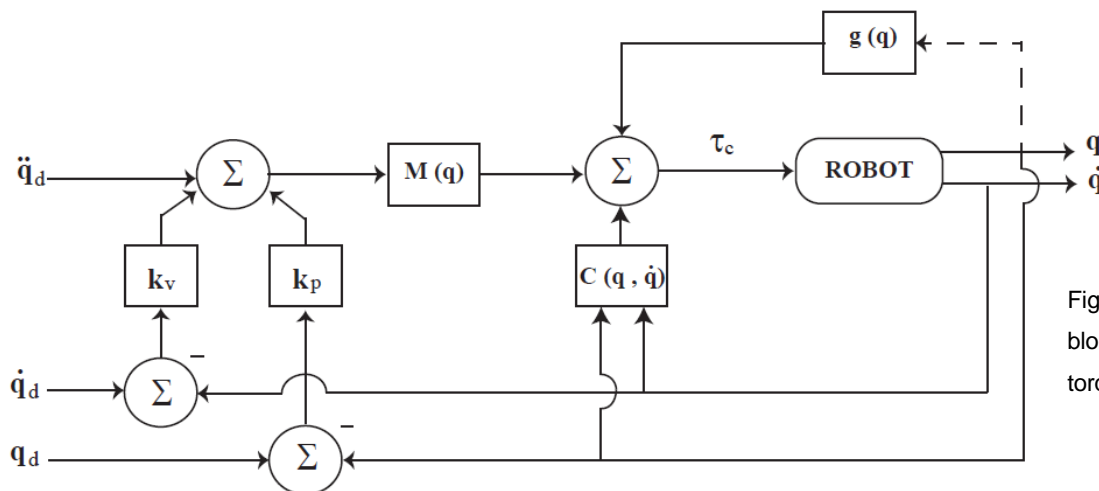


Fig. 37 Diagrama de bloques de control de torque precalculado



3.8 Control por Campos Potenciales

En la técnica de campos potenciales están contemplados la suma de fuerzas (vectores) de atracción (meta) y rechazo (obstáculos) en el sitio donde se haga la prueba (área de trabajo) como se explica en [14]. Es fundamental comprender que para desplazar el robot únicamente es elemental la orientación y los elementos de rapidez tanto en x y y , debido a que en esta situación no hay obstáculos.

Cuando se tiene las variables para mantener el control de, es viable ejercer una fórmula de control a la plataforma móvil, y de esta forma ver la habituación a las diversas trayectorias propuestas. El procedimiento de campos potenciales, puede aplicarse a partir de la perspectiva cinemático, cambiando los vectores de fuerza por los de distancia. El propósito al que se quiere llegar representa un vector de sentido que parte de la postura inicial del robot hacia dicho objetivo, siendo este el punto final. Para eludir colisiones se tienen que implantar superficies de estabilidad y velocidades estáticas y evadir las velocidades infinitas o bastante monumentales.

Se define un área de estabilidad en torno al objetivo y del robot, y una rapidez máxima como para el eje x , como para el eje y y una rapidez angular máxima. A medida que más cerca se encuentre el robot del punto objetivo irá reduciendo su rapidez hasta que los círculos sean tangenciales. Con base en lo anterior, obtendrá un conjunto de puntos que permitirán que el robot alcance el punto objetivo. Por lo tanto, se determina el vector de distancia desde la posición inicial del robot hasta el punto final (objetivo). Queda de la siguiente manera:

$$\mathbf{d}_g = (x_f - x_i, y_f - y_i)$$

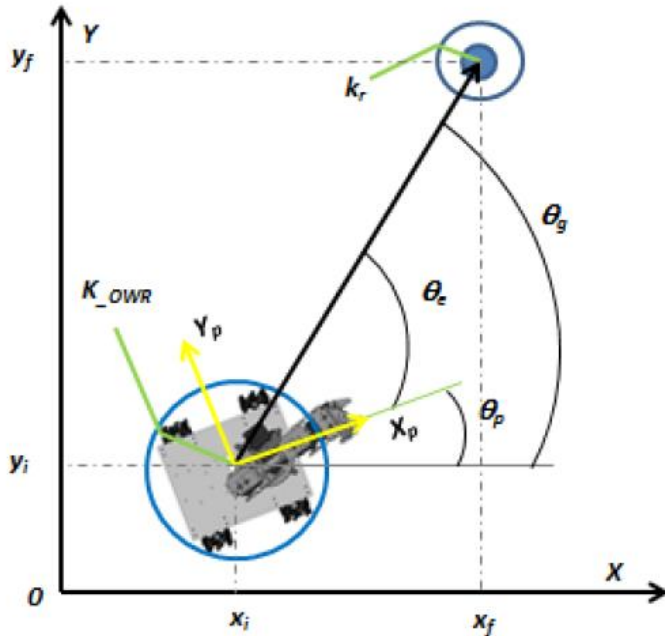


Fig. 38 Alcance del objetivo con control de campos potenciales

La magnitud de la distancia se obtiene por medio de:

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

El ángulo del objetivo o meta se encuentra:

$$\theta_g = \text{angtan} \left(\frac{y_f - y_i}{x_f - x_i} \right)$$

El ángulo de error lo obtenemos:

$$\theta_e = \theta_g - \theta_i$$

Se necesita una velocidad angular para que ayude a orientar por completo al robot y tenga la orientación deseada, θ_e vaya a cero, tenemos la siguientes ecuaciones:

$$w = w_{max} \sin \theta_e$$

ó

$$w = w_{max} \sin(-\theta_p)$$

$$w = w_{max} \sin\left(\theta_e - \frac{\pi}{2}\right)$$



Se tienen las siguientes ecuaciones tanto para su desplazamiento en el eje X , como en el eje Y , para la velocidad que debe tomar el móvil:

$$v_{xp} = \begin{cases} v_{xmax} \cdot \sin(\theta_e) & \text{si } |d_g| > k_r + k_{OWR} \\ \frac{v_{xmax}}{k_r + k_{OWR}} \cdot d_g \sin(\theta_e) & \text{si } |d_g| < k_r + k_{OWR} \\ 0 & \text{si } |d_g| \leq k_{OWR} \end{cases}$$

Las ecuaciones anteriores nos señalan a la velocidad que debe ir el robot, y está definida por:

d_g : el vector de atracción del robot al objetivo

θ_g : el ángulo de ese vector

θ_i : el ángulo de inclinación de robot:

θ_e : el error existente entre ambos ángulos

k_r : con radio, el área de seguridad (para evitar colisiones)

k_{OWR} : el área de seguridad del robot



Capítulo 4

4 Herramientas Virtuales

4.1 MATLAB® (MATrix LABoratory)

Las primordiales propiedades son: La manipulación de matrices, la representación de datos y funcionalidades, la utilización de algoritmos, la construcción de interfaces de cliente (GUI) y la comunicación con programas en otros idiomas y con otros dispositivos hardware. El paquete MATLAB dispone de 2 herramientas extras que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de cliente - GUI). Además, tienen la posibilidad de agrandar las habilidades de MATLAB con las cajas de herramientas (toolboxes); y las de Simulink con los paquetes de bloques (blocksets) [16,18].

4.1.1 Simulink®

Es un programa que funciona bajo la plataforma de Matlab y es un instrumento bastante eficaz para modelar, simular y examinar sistemas, tanto lineales como no lineales. En una interfaz gráfica (GUI) donde el cliente construye un diagrama de bloques que desarrollan métodos que hacen las operaciones matemáticas requeridas para la solución de un modelo; diseñar y examinar la mecánica de cuerpos firmes en 3D (tales como plataformas de vehículos y brazos manipuladores) y la dinámica de los actuadores (tales como sistemas mecatrónicos o fluidos).

Una vez que se usa MATLAB® y Simulink® conjuntamente, en verdad está combinando programación textual y gráfica para diseñar su sistema en un ámbito de simulación; de manera directa hay una cantidad enorme de algoritmos que ya se hallan en MATLAB, solo debería que integrar su código de MATLAB en un bloque de Simulink o un gráfico de Stateflow®; para producir conjuntos de datos de acceso que pongan en marcha la simulación y llevar a cabo una cantidad enorme de simulaciones en paralelo. Luego examinar y visualizar los datos en MATLAB [17].

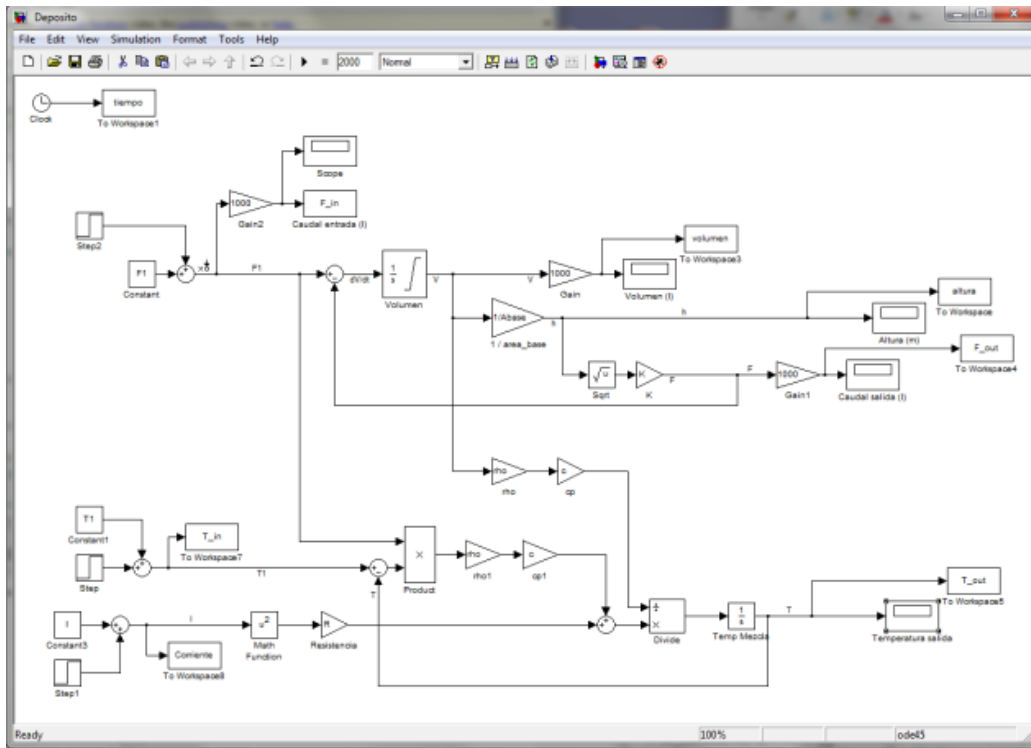


Fig. 39 Ventana principal de Simulink – Matlab.

4.1.2 Simscape Multibody™

Simscape Multibody™ (antes SimMechanics™) otorga un ámbito de simulación multicuerpo para 3D sistemas mecánicos, como robots, suspensiones de vehículos, maquinaria y ferrocarril de aterrizaje de aviones. Puede modelar utilizando los bloques que representan órganos, articulaciones, restricciones, recursos de fuerza y sensores de sistemas multicuerpo.

Multicuerpo Simscape ayuda a desarrollar sistemas de control y prueba de rendimiento de grado de sistema. Usted puede parametrizar sus modelos utilizando MATLAB® cambiantes y expresiones y diseño de sistemas de control para su sistema multicuerpo en Simulink®. Para llevar a cabo sus modelos en otros ambientes de simulación, incluyendo sistemas de hardware-in-the-loop (HIL), Simscape Multibody so la generación de código en C [16,18].

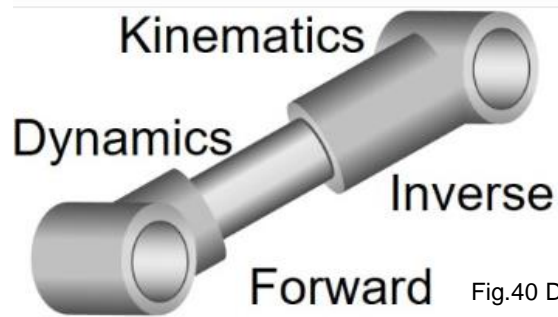


Fig.40 Detección y actuación de articulaciones

4.1.3 V- Realm Editor

Es un software que se utiliza para diseñar mundos virtuales y dibujar/importar objetos virtuales en 3D. Una vez que el usuario construye su mundo virtual, puede manipular los objetos y la escena virtual mediante comandos de MATLAB® y modelos de Simulink® para animar la escena. Las propiedades de los objetos 3D virtuales, como la traslación, la rotación, la escala y el color, se pueden cambiar mediante los comandos de MATLAB® y los modelos de Simulink®.

V-Realm Builder es un poderoso paquete de construcción tridimensional para la creación de objetos tridimensionales y "universos" para ser observados con el navegador V-Realm o cualquier otro navegador compatible con VRML 2.0. Es importante recordar que VRML 2.0 se desarrolló como una descripción de documento virtual amistoso de "red". V-Realm Builder que usa VRML ha sido elaborado para proporcionarnos herramientas para reducir la medida de los archivos y proveer un medio para modelar objetos más complicados con primitivos, sin tener que sobrecargar la red con archivos enormes.

V-Realm Builder otorga potentes habilidades de versión al tiempo que simplifica el proceso de construcción en 3D. No obstante, al automatizar el desarrollo de VRML y dar una retroalimentación visual inmediata, acelera extremadamente la obra de todo el mundo en 3D. La GUI de V-Realm Builder está diseñada especialmente para VRML e inclusive un inventor VRML con vivencia en la construcción de universos en 3D. Este control pone al V-Realm Builder en una clase propia [18].

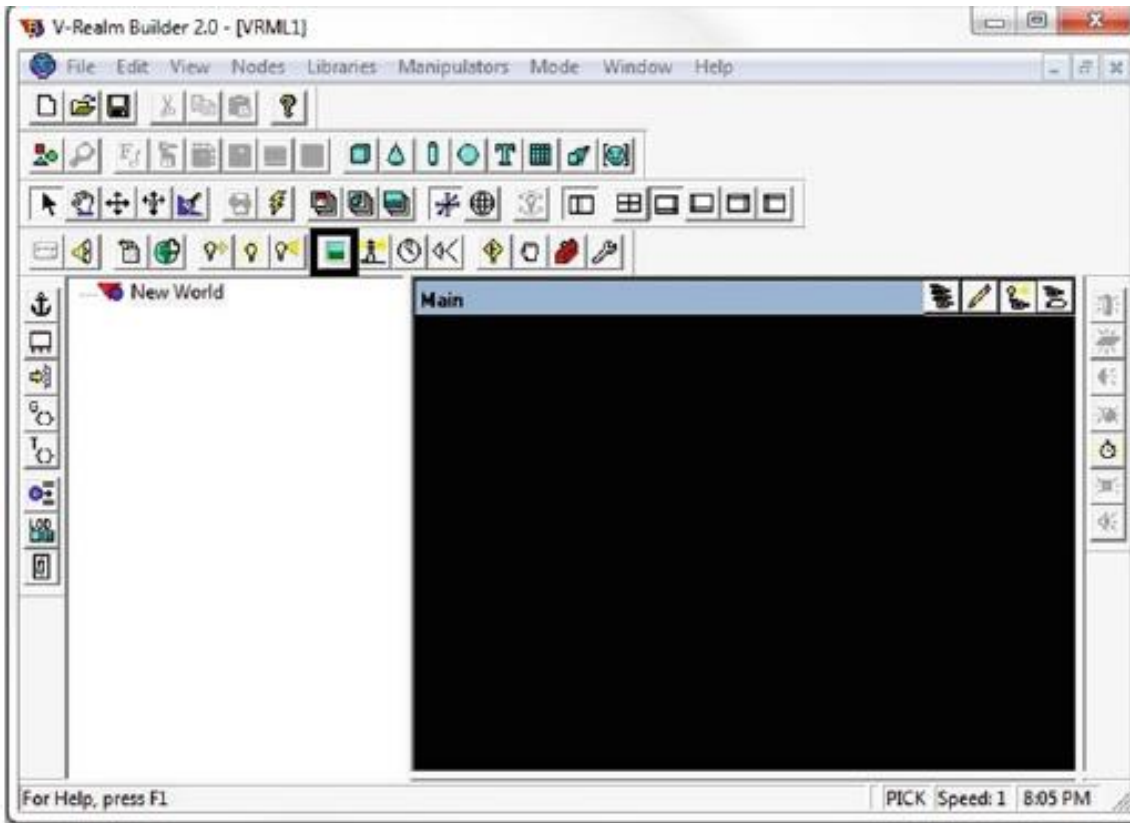


Fig. 41 Ventana principal de V-Real Builder

4.2 SOLIDWORKS

Es un programa de diseño asistido por computadora (CAD) para la creación de modelos mecánicos en 2D y 3D, actualmente desarrollado por SolidWorks Corp., una subsidiaria de Dassault Systèmes, S.A. (Suresnes, Francia), para Microsoft Windows [20]. El programa le permite modelar piezas y ensamblajes, extraer dibujos técnicos y otros tipos de información necesaria para la producción de ellos. Es un software que se basa en nuevas técnicas de modelado de sistemas, una amplia gama de herramientas mecánicas de CAD, validación de diseños, gestión de datos de productos, comunicación de diseños y productividad de CAD.

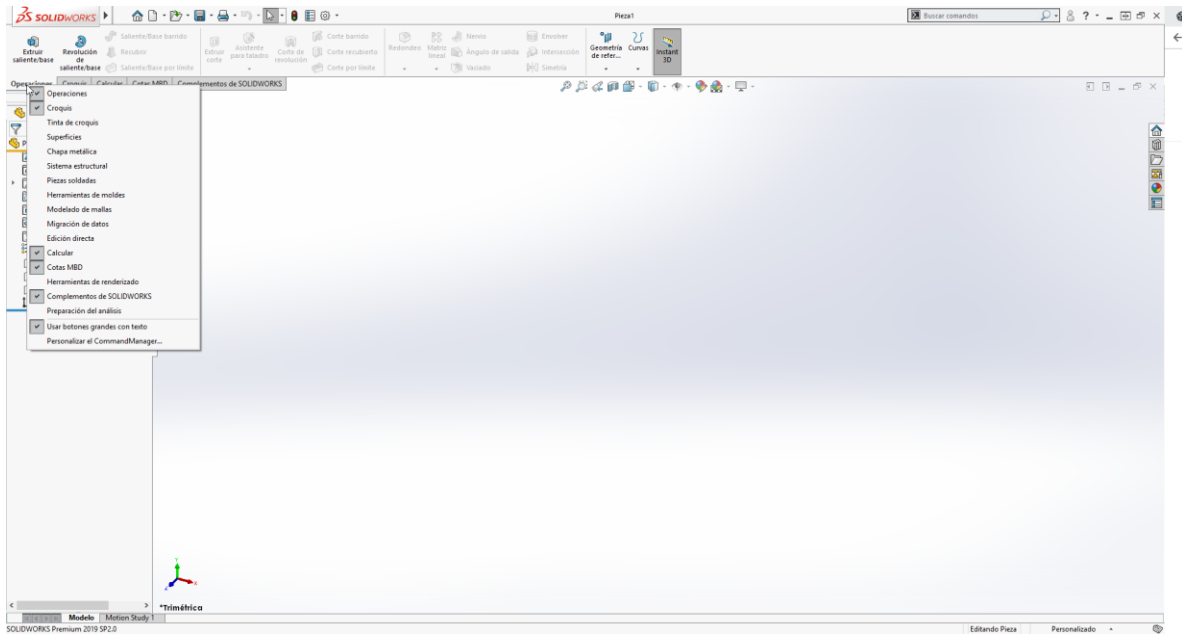


Fig. 42 Área de trabajo de SolidWorks

4.2.1 Módulos de SolidWorks

Módulos de SolidWorks. Los módulos de SolidWorks son un entorno de diseño en el que trabaja para lograr sus objetivos, son [20]:

1. El módulo de piezas tiene un conjunto de funciones e iconos que le permiten crear un modelo 3D a partir de una geometría de croquis 2D. Esto le permite obtener sólidos, superficies, estructuras metálicas, piezas de chapa y piezas multicuerpo en particular.
2. Ensamblaje. Consiste en un entorno de trabajo preparado para crear un kit o conjunto mediante la inserción de un modelo 3D creado en un módulo de pieza. Los ensamblajes se definen estableciendo relaciones geométricas entre los componentes.
3. Plano o dibujo. Esto le permite crear automáticamente un dibujo de vista de ensamblaje o modelo en muy poco tiempo. Puede usar este módulo para obtener proyecciones ortográficas, secciones transversales y transversales, dimensiones, listas de materiales, imágenes explosionadas y muchas otras características.



Otras aplicaciones de SolidWorks. Estas aplicaciones están vinculadas a la parte de modelado de SolidWorks, son:

4. Modelado y movimiento (SolidWorks Motion). Es una herramienta de modelado de física que puede evaluar cómo las formas, las dimensiones y las relaciones geométricas establecidas entre las piezas determinan la cinemática general. Puede agregar rotación, motores lineales, efectos de gravedad, resortes, amortiguadores, contactos de componentes, análisis de elementos finitos, optimización y más. La simulación de movimiento resultante proporciona resultados útiles de carga y desplazamiento para analizar la población simulada. Se utiliza para simular con precisión los efectos de las piezas móviles (fuerzas, resortes, amortiguadores, fricción, etc.) en el ensamblaje.
5. Análisis de movimiento.
6. Análisis de interferencia. Consta de las herramientas incluidas en el módulo de montaje. Su finalidad es detectar posibles interferencias entre las piezas que componen un conjunto.
7. SmartMates (tecnología inteligente). La herramienta SmartMates ahorra tiempo de diseño al crear automáticamente relaciones de posición en el ensamblaje.

4.2.2 Estudio de Movimiento

Un estudio de movimiento es una simulación numérica del movimiento de un ensamblaje. Se pueden personalizar configurando la perspectiva, la iluminación y las propiedades visuales. En este entorno, las propiedades físicas del ensamblaje no cambian, solo se simula y anima el movimiento del modelo. Puede utilizar las restricciones de SolidWorks para restringir el movimiento de los componentes de un ensamblaje al modelar el movimiento. En la (Fig.43) muestra la interfaz SolidWorks Motion.

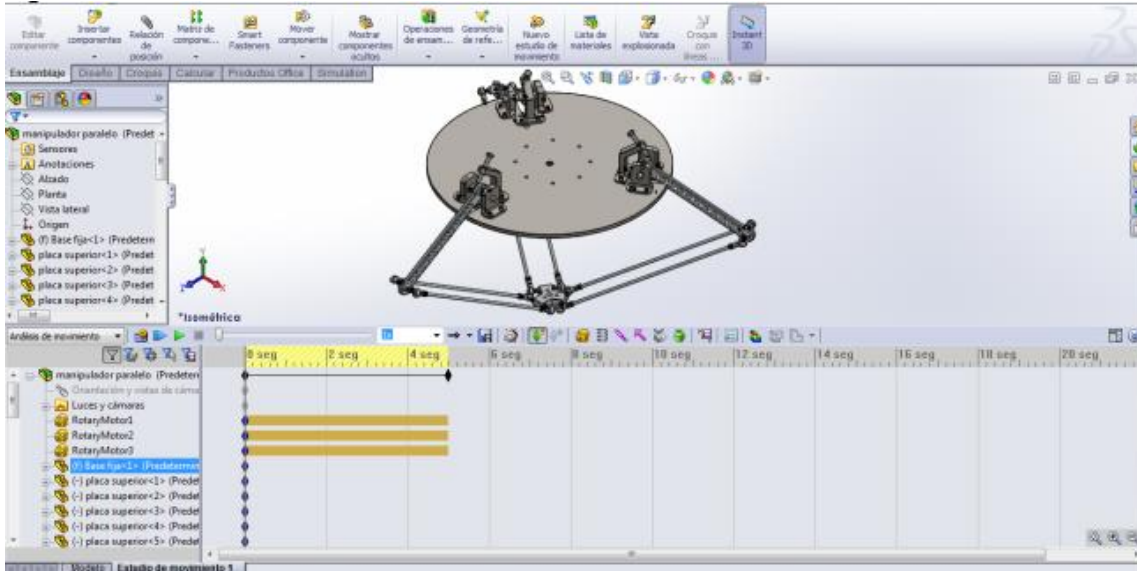


Fig. 43 Interfaz de SolidWorks Motion

4.2.2.1 Tipos de Estudio de Movimiento

En un estudio de movimiento se pueden utilizar tres herramientas para realizar la respectiva simulación (Fig.44):

1. Animación. Se utiliza esta herramienta para animar el movimiento de ensamblajes, en este modo no es posible utilizar fuerzas, resortes, contactos, gravedad, entre otros. Su uso se limita únicamente a observar el movimiento del ensamblaje. En esta herramienta se permite el uso de motores para conducir el movimiento del ensamblaje.
2. Movimiento básico. Se utiliza movimiento básico en ensamblajes para simular los efectos de motores, resortes, colisiones y gravedad. A la hora de calcular el movimiento, “Movimiento básico” tiene en cuenta la masa. El cálculo es relativamente rápido, por lo que puede utilizar un estudio de movimiento para crear animaciones tipo presentación mediante simulaciones basadas en leyes físicas.
3. Análisis de movimiento. Este tipo de movimiento se calcula mediante un *solver* (solucionador) que utiliza tres clases de integradores, GSTIFF, WSTIFF y SI2_GSTIFF, teniendo en cuenta las propiedades del material, así

como la masa y la inercia. También se puede utilizar para trazar los resultados de la simulación para su posterior análisis.

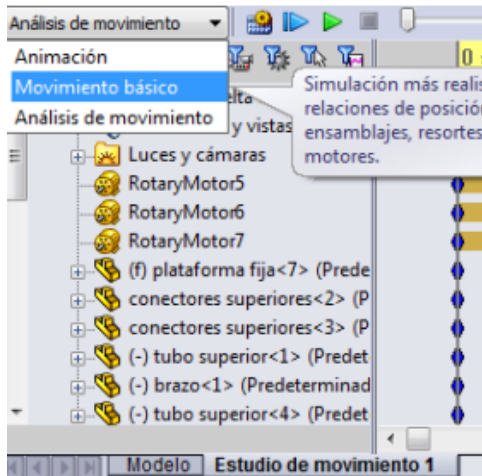


Fig. 44 Tipos de estudio de movimiento

4.3 Blender

Blender es un paquete de creación de contenido 3D completamente integrado que proporciona una variedad de herramientas importantes como modelado, renderizado, animación y edición, edición de video, efectos visuales, composición, texturizado y diferentes tipos de simulaciones. Es multiplataforma y tiene una interfaz gráfica de usuario (GUI) OpenGL compatible con todas las plataformas principales (configurable mediante comandos de Python) [19].

Además de estructura digital usando la técnica procesal de nodos, versión de vídeo, escultura (incluye topología dinámica) y pintura digital. En Blender, además, tienen la posibilidad de desarrollar vídeo juegos debido a que tiene un motor de juegos interno. El programa ha sido al inicio compartido de manera gratuita, empero sin el código fuente, con un manual disponible para la comercialización, aunque después pasó a ser programa independiente.

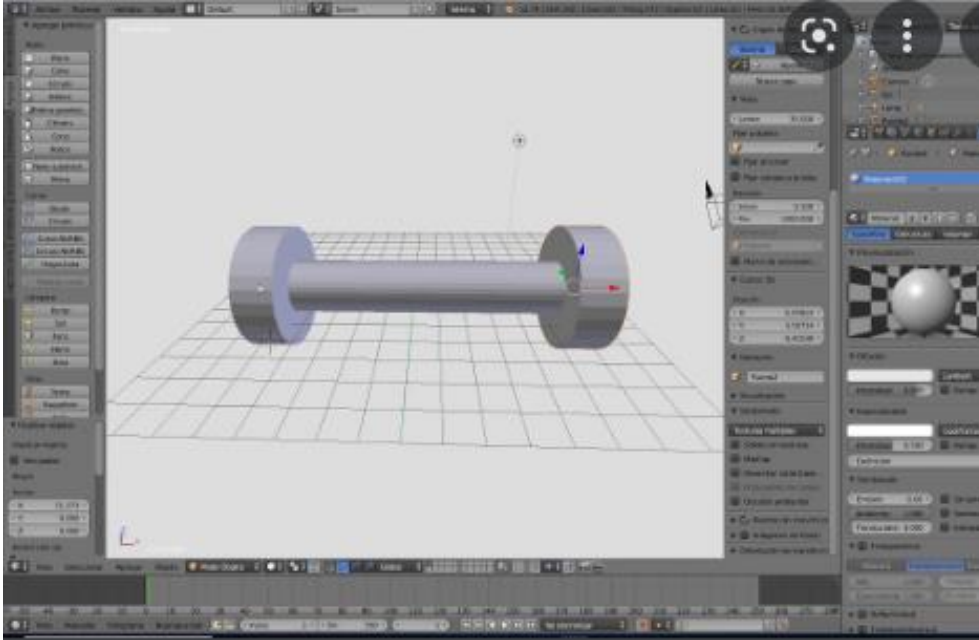


Fig. 45 Modelando en Blender. Unir y separar objetos

4.4 Integración con LabVIEW

LabVIEW (Laboratory Instrument Engineering Workbench) es un lenguaje de programación. Programación básica para el diseño de sistemas de adquisición de datos, instrumentación. Con LabVIEW, puede utilizar para diseñar su interfaz de usuario una consola de software interactivo. Se puede ampliar especificando sistema funcional, diagrama de bloques o notación de diseño técnico. LabVIEW es compatible con otras herramientas de desarrollo similares y Puede trabajar con programas en otros dominios de aplicación, como Matlab y SolidWorks [22].

Incluso si usa LabVIEW para desarrollar su programa, todavía trabaja en un sistema llamado VI o instrumento virtual. Los VI se pueden crear en función de las especificaciones funcionales requeridas por el desarrollador. Los VI se pueden usar en otras aplicaciones como una sub-característica de los programas comunes. La máquina virtual consta de tres partes [22].

1. Panel frontal. La interfaz de usuario ha sido desarrollada. El panel frontal contiene todos los indicadores y controles que el usuario puede ver mientras se ejecuta el programa (Fig. 56).

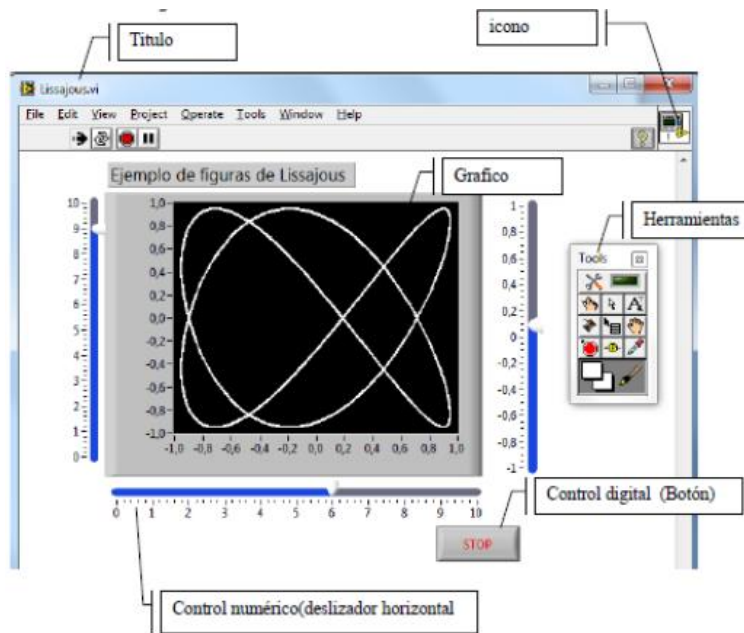


Figura 56. Panel frontal de LabVIEW

2. Panel de programación. Puede diseñar gráficamente el funcionamiento del sistema creado. Se basa en realizar una operación asignando datos numéricos a realizar y un icono que representa el procedimiento.
3. Este diagrama utiliza un diagrama de bloques, una estructura de programación y un flujo de datos entre varias entradas y salidas a lo largo de la línea (Fig. 57).

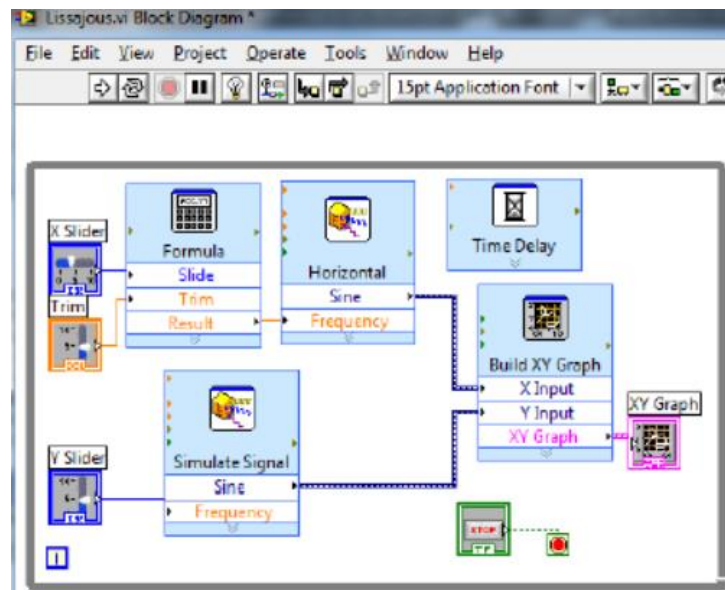


Figura 57. Ejemplo de diagrama de bloques en LabVIEW



4.4.1 Toolkits y módulos de LabVIEW

Los siguientes módulos y toolkits ayudan a optimizar el uso de los recursos de LabVIEW para ayudar a su computadora a manejar toda la información que manipula el software. No se requiere que SolidWorks se comunique con LabVIEW, pero es muy útil al procesar datos. Además, permiten diagramas de flujo en bucles cerrados o abiertos y utilizan representaciones como funciones de transferencia [22].

- Módulo LabVIEW Real Time
- Real time execution trace toolkit
- Módulo NI control design and simulation

Los módulos y toolkits que permiten la vinculación entre SolidWorks y LabVIEW, los encontramos a continuación:

- NI Motion Assistant. Asistente de movimiento NI. Puede desarrollar aplicaciones de control de movimiento en un entorno 3D interactivo y convertirlas a código C para un compilador C o LabVIEW Virtual Instrument (VI). También puede exponer funciones de contorno de pendiente inteligente a través de API que los usuarios pueden llamar desde C, Visual Basic y LabVIEW, e importar perfiles de movimiento desde CAD y programas de dibujo. También tiene una función para controlar el motor paso a paso y el motor paso a paso servo motor y actuadores piezoeléctricos para aplicaciones de un solo eje o de varios ejes.
- NI LabVIEW SoftMotion. Proporciona desarrollo gráfico de aplicaciones personalizadas de control de movimiento. Esto le permite usar su proyecto de LabVIEW para configurar todos los ejes de movimiento, probar la configuración y ajustar el motor. Después de completar la configuración del hardware, puede usar el panel de prueba interactivo para revisar la configuración y mover los ejes individuales para probar la configuración del hardware. NI SoftMotion proporciona la conveniencia de la programación de perfiles de movimiento con bloques de funciones API



de alto nivel basados en la biblioteca de control de movimiento definida en PLCopen. Incluye bloques funcionales de tipo de movimiento, como lineal, arco y contorno, así como bloques funcionales opcionales avanzados, como engranajes electrónicos y levas. También tiene funciones avanzadas para diseñar aplicaciones de movimiento personalizadas, como generación de trayectorias, interpolación de spline, control de posición y velocidad e implementación de codificador.

- NI LabVIEW SoftMotion para SolidWorks. National Instruments ofrece esta versión de software que simula diseños creados en SolidWorks utilizando perfiles de movimiento desarrollados con el bloque de construcción NI SoftMotion, por lo que los usuarios no pueden gastar dinero en prototipos físicos. Además, puede visualizar y optimizar su diseño y evaluar diferentes conceptos de diseño. Otro toolkit que también puede ser muy útiles.
- Módulo LabVIEW Robotics. El Módulo de Robótica de LabVIEW proporciona herramientas de desarrollo de hardware y software para diseñar sistemas de control robótico. El módulo de robótica de LabVIEW es un software opcional para el entorno de programación de LabVIEW. Proporciona una extensa biblioteca de robótica con conexiones integradas a actuadores y sensores robóticos, algoritmos centrales para operación inteligente y detección robusta, y capacidades de movimiento para mover robots y vehículos. Los algoritmos y códigos de control se pueden implementar en tiempo real en hardware integrado de NI. LabVIEW Robotics Module diseña y crea prototipos de vehículos autónomos y semiautónomos como sistemas agrícolas y militares, plataformas de rescate de robots, vehículos submarinos y aéreos, robots personales, servicios, dispositivos de robots médicos y robots académicos. ideal para científicos e ingenieros.



Capítulo 5

5 KUKA youBot

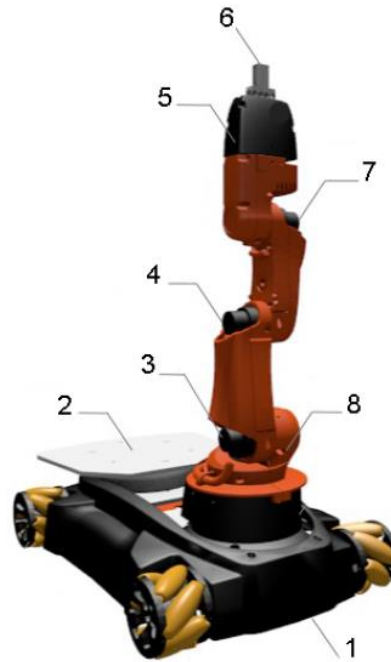
5.1 Descripción general

El KUKA youBot está formado por un chasis con movimiento omnidireccional y por uno o 2 brazos robóticos montados en el chasis. En el chasis permanecen incluidos una computadora industrial y una batería, la cual se comunica en tiempo real (ciclo de 1 ms), por medio de EtherCAT, con un total de 9 motores sin escobillas que tienen la posibilidad de funcionar con regulación de corriente, velocidad y posición. La plataforma y el brazo robóticos tienen la posibilidad de emplearse independientemente el uno de la otra. El chasis del youBot tiene tan solo 53 centímetros de longitud, 36 centímetros de anchura y apenas 11 centímetros de elevación, y se desplaza sobre 4 ruedas Mecanum. Hablamos de ruedas especiales, cuya área de movimiento está compuesta por rodillos instalados en un ángulo de 45° en toda su circunferencia, lo cual posibilita al youBot hacer toda clase de combinaciones de movimientos de traslación y rotación. Tal es viable llevar a cabo movimientos omnidireccionales, integrados desplazamientos laterales y diagonales.

Sobre la plataforma hay montado un extenso brazo de 66 centímetros, en cuyo extremo está una pinza con 2 dedos, con la que el robot puede desplazar objetos con un tamaño de hasta 70 mm y un peso de hasta 500 gramos. El brazo robótico está formado por 5 articulaciones, cada una de ellas accionadas por sistemas motores sin escobillas de maxon motor. Para esta labor se emplean combinaciones de motor-reductor-encoder de maxon motor [21].

Fig. 58 **KUKA youBot**:

1. Base,
2. Plataforma,
3. Articulación dos,
4. Articulación tres,
5. Articulación cinco,
6. Pinza,
7. Articulación 4,
8. Articulación 1. *KUKA Robot Group 2013.*



Para las articulaciones del brazo robótico se emplean un total de 5 motores brushless planos maxon motor sin escobillas (EC45 flat, EC32 flat) en conjunción con reductores especiales y encoders. Para la plataforma se usan 4 motores brushless planos sin escobillas EC 45 de la fábrica de maxon motor. El motor sin escobillas maxon EC 45 flat destaca por un sencillo diseño y una alta resistencia al par en un formato compacto. Los motores brushless suministran una potencia de entre 15 y 50 W, empero solo pesan entre 46 gramo y 110 gramo. Las articulaciones del brazo robótico permanecen conectadas a transmisores de postura maxon motor (encoders) para permitir la medición de ángulos [21].

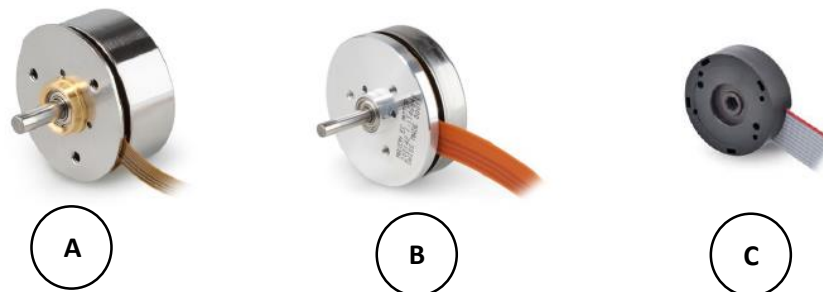


Fig. 59 Motor de la fábrica maxon motor. A) maxon EC32 flat, B) maxon EC45 flat, C) maxon Encoder MR, typo L

El KUKA youBot puede ya considerarse como un logro en la investigación y el desarrollo robótico. Con él, KUKA es consciente de haber abierto un nuevo camino dirigido al mercado de la indagación y al impulso de la transferencia de tecnología en la sociedad de la robótica. Claramente el término de un robot de código abierto con base en Linux da maneras hasta ahora inimaginables para estudiosos y científicos.



Fig. 60 El software de control del robot dispone de una serie de interfaces abiertas; es un sistema de código abierto basado en Linux

5.2 Modelo cinemático de la plataforma

El robot manipulador móvil Kuka youBot cuenta con una plataforma omnidireccional de 4 ruedas tipo Mecanum que permiten el independiente movimiento del sistema en el plano Cartesiano, no obstante, se necesita tener en cuenta la disposición de los rodillos de esta clase de ruedas para obtener un modelo cinemático fidedigno. Las medidas de la plataforma móvil se presentan en la (Fig. 61) y permanecen expresadas en milímetros [15].

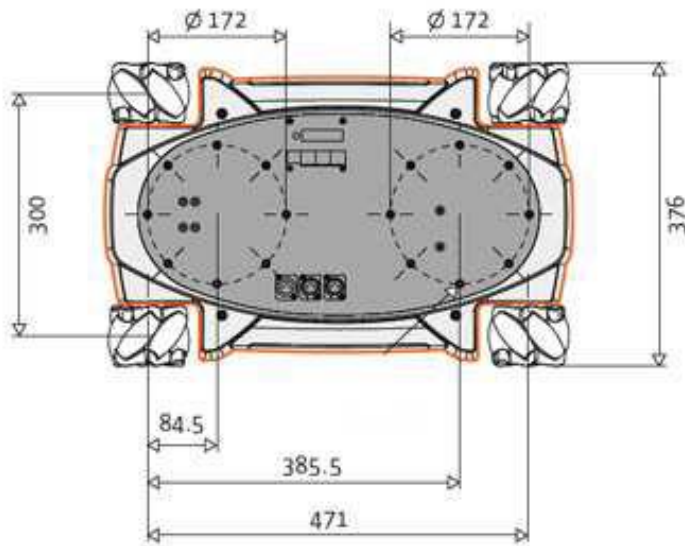


Fig. 61 Medidas de la plataforma móvil

La configuración cinemática de la parte móvil omnidireccional del robot ésta dada por el conjunto de ecuaciones:

$$\dot{x} = v_l = \frac{1}{4}(d_1 + d_2 + d_3 + d_4)$$

$$\dot{y} = v_t = \frac{1}{4}(-d_1 + d_2 + d_3 - d_4)\tan(\alpha_b)$$

$$\dot{\theta}_b = v_a = \omega = \frac{1}{4}(d_1 + d_2 + d_3 + d_4)\beta$$

\dot{x} y \dot{y} : son la velocidad en los ejes X y Y ;

v_l y v_t : son las velocidades longitudinal y transversal;

$\dot{\theta}_b$: la velocidad angular;

v_a : es la plataforma móvil omnidireccional;

d_i : es la velocidad lineal de cada rueda de la plataforma móvil;

α_b y β : calculados experimentalmente debido a que depende del ángulo en que estén dispuestos los rodillos de las ruedas mecanum.



Fig.62 Rueda Mecanum de KUKA youBot.

El desplazamiento del robot móvil en el plano dependerá entonces de la mezcla de las velocidades lineales de cada rueda de la plataforma, especialmente los movimientos de traslación transversal y rotación se ilustran en la Figura 3 donde la plataforma difuminada sugiere hacia donde se moverá el robot móvil de acuerdo con la conjunción de velocidades lineales de las 4 ruedas [15].

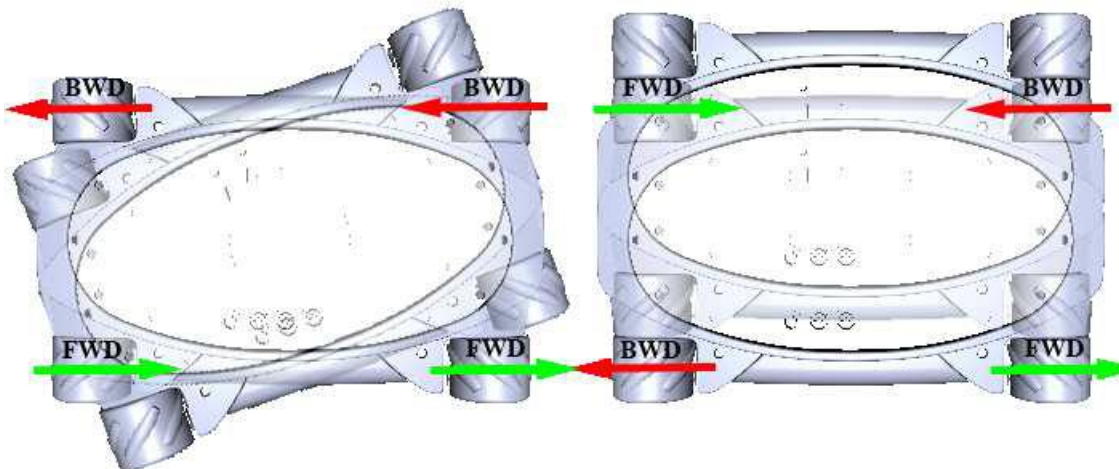


Fig. 63 Plataforma móvil. Movimiento.

5.3 Modelo cinemático del manipulador

El brazo manipulador del Kuka youBot es un robot de 5 grados de libertad (rotacionales), las medidas de cada eslabón, así como sus parámetros articulares se muestran en la (Fig. 64), el efector final es una pinza de 2 dedos que le posibilita sujetar y manipular objetos pequeños. El modelo cinemático del robot manipulador se recibe por medio de el algoritmo de Denavit Hartenberg [15].

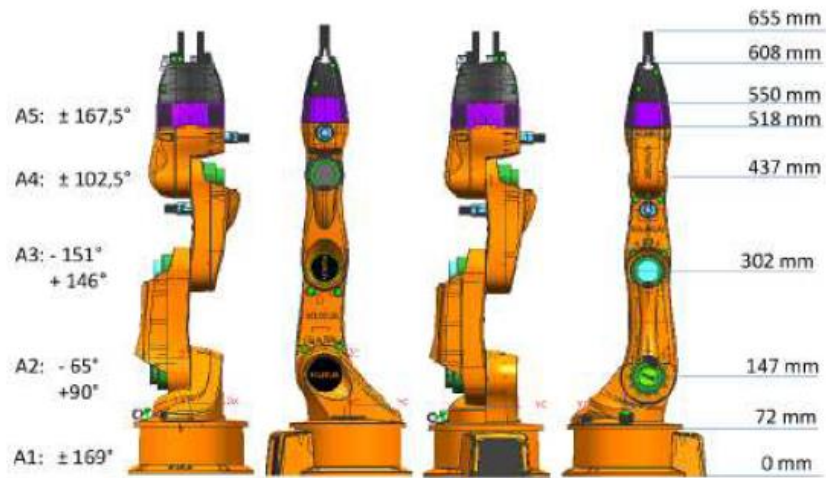


Fig. 64 Parámetros articulares de la parte del manipulador

La siguiente figura muestra los marcos de referencia para cada una de las articulaciones.

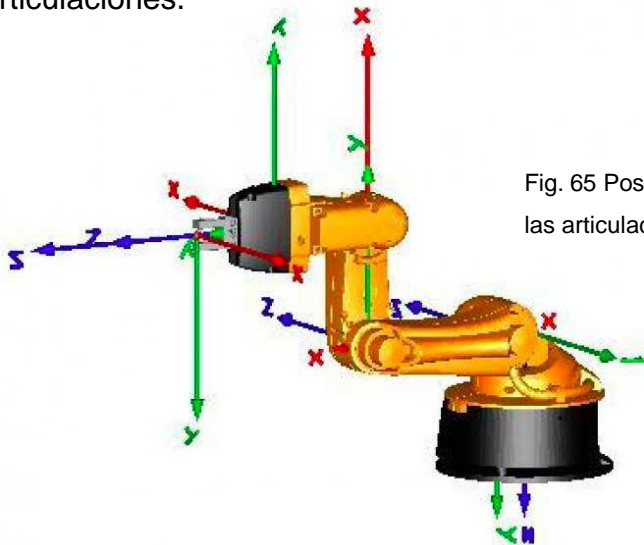


Fig. 65 Posición de los marcos de referencia para cada una de las articulaciones.

En la siguiente tabla se encuentran los parámetros de Denavit-Hartenberg obtenidos para el manipulador:

Eslabón θ	d	a	α	Posición de los marcos de referencia para cada una articulationes
q_1	0.147	0.0330	$\frac{\pi}{2}$	
q_2	0	0.1550	0	
q_3	0	0.1350	0	
$q_4 + \frac{\pi}{2}$	0	0	$\frac{\pi}{2}$	
q_5	0.2175	0	0	

Tabla A1. Posición de los marcos de referencia del manipulador

En esta otra tabla a continuación se muestran las distancias y ángulos entre los marcos de referencia para el brazo manipulador.

Marco anterior		Traslación (cm)			Rotación (grados)		
		x	y	z	x	y	z
Base	Articulación 1	2.4	0	11.5	180°	0°	0°
Articulación 1	Articulación 2	3.3	0	0	90°	0°	-90°
Articulación 2	Articulación 3	15.5	0	0	0°	0°	-90°
Articulación 3	Articulación 4	0	13.5	0	0°	0°	0°
Articulación 4	Articulación 5	0	11.36	0	-90°	0°	0°
Articulación 5	Pinza	0	0	5.716	90°	0°	180°

Tabla A2. Cadena cinemática del manipulador

Con los datos de las Tablas anteriores es viable obtener las matrices de transformación homogéneas:

$$A_i^{i-1} = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & -S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Los términos S y C denotan las funciones $\sin()$ y $\cos()$. La matriz de transformación total se saca de la multiplicación sucesiva de cada una de las matrices homogéneas.

$$T_n^0 = A_1^0 \dots A_n^{n-1}$$

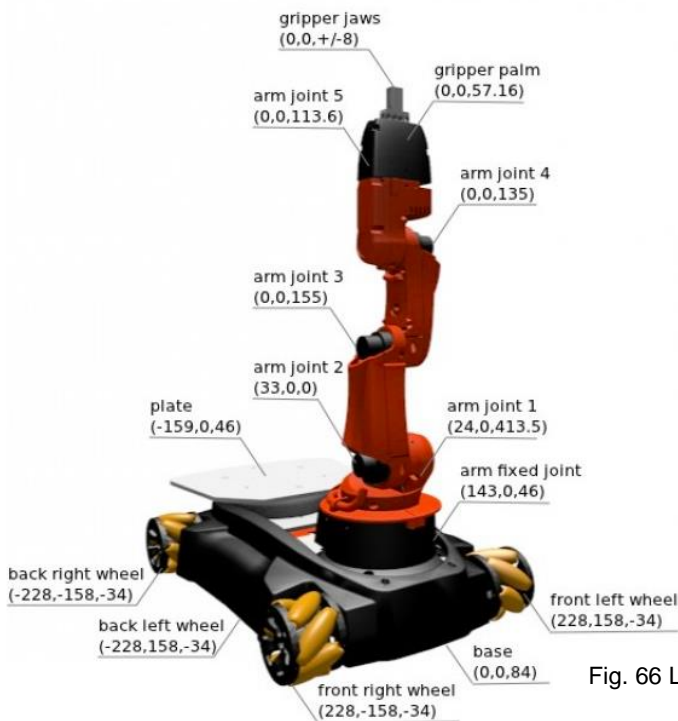


Fig. 66 Las posiciones relativas de las juntas (mm) del modelo 3D. Kuka youBot

5.4 Dinámica del robot

El comportamiento dinámico del youBot (robot manipulador móvil) se puede descomponer, sin perder la generalización, en tres tipos de componentes: ruedas Mecanum, articulaciones del manipulador y eslabones del manipulador [16].

5.4.1 Descripción de la plataforma

Para este tutorial paso a paso, se utiliza como ejemplo el modelado de la plataforma robótica youBot (Fig.67 A) La estructura cinemática del manipulador robótico youBot se muestra en la (Fig.67 B)



Figura 67 A) Modelado

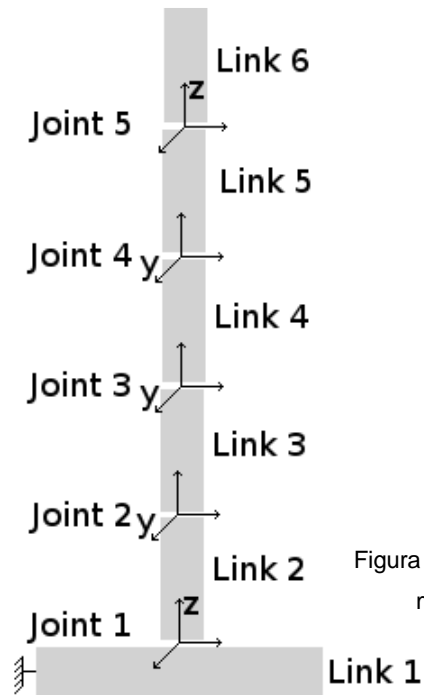


Figura 67 B) Cinemática manipulador

Fig. 67 Representación esquemática del KUKA youBot

El manipulador robótico tiene seis enlaces conectados por 5 juntas rotacionales accionadas. El eje de rotación de las articulaciones 1 y 5 es el eje z en el cuadro representado, para las articulaciones 2, 3 y 4 el eje de rotación es el eje z.

Cuatro ruedas Mecanum están montadas en el primer eslabón del actuador robótico para hacerlo móvil, esto se muestra en la (Fig. 68)



Figura 68 A) Plataforma youBot

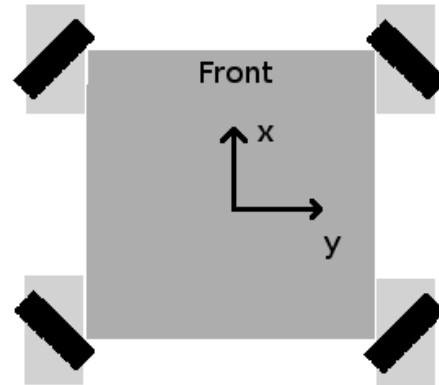


Figura 68 B) Modelado

Fig. 68 La base youBot con ruedas Mecanum

Un proceso de modelado comienza con la definición de un comportamiento dominante del componente. Para aplicaciones robóticas, el comportamiento dominante de los eslabones del manipulador es el comportamiento de un cuerpo al que se aplican llaves externas. Siempre que el youBot se utilice dentro de las especificaciones, es válido suponer que los enlaces son rígidos y se tratan como cuerpos rígidos. Sin embargo, el sub-modelo se puede adaptar para incluir otro comportamiento sin afectar la interfaz con otras partes del modelo youBot.

El modelo de dinámica de cuerpo rígido consiste en efectos de inercia y giroscópicos. De acuerdo con la teoría del tornillo, la cinemática del movimiento de un cuerpo rígido se puede representar como una rotación (ω) alrededor de un eje junto con una traslación (v) a lo largo del mismo eje [16]:

$$-T = \begin{pmatrix} \omega \\ v \end{pmatrix} : girar$$



$W = \begin{pmatrix} \tau \\ F \end{pmatrix}$: llave,

$I = \begin{pmatrix} J & 0 \\ 0 & M \end{pmatrix}$: tensor de inercia,

F : fuerza,

P : momento de impulso,

τ : par,

M : matriz de masa,

J : matriz de inercia.

El balance de llave que representa los efectos de inercia, expresados en el marco de inercia principal k , (en el centro de gravedad y orientado a lo largo de los tres ejes de inercia primaria) tendrá la forma:

$$I^k \dot{T}_a^{k,0} = \begin{pmatrix} \tilde{P}_\omega^k & \tilde{P}_v^k \\ \tilde{P}_v^k & 0 \end{pmatrix} \dot{T}_a^{k,0} + (W^k)^T$$

k : marco principal de inercia del cuerpo,

0 : marco de inercia,

a : el cuerpo.

En la relación anterior, el componente de una inercia se puede reconocer en el lado izquierdo del signo igual. El primer componente en el lado derecho del signo igual representa las fuerzas y pares ficticios (llaves), incluido el efecto giroscópico, el segundo componente en el lado derecho del signo igual representa las llaves aplicadas externamente. Las llaves aplicadas externamente incluyen reacciones articulares, colisiones, gravedad, etc. Las más significativas, que tienen un efecto constante en la dinámica del eslabón, son las reacciones articulares y la gravedad. Las uniones se conectarán a un eslabón por medio de puertos de alimentación, a través de los cuales las uniones aplican llaves a los eslabones. Sin embargo, estas llaves se expresan en el marco que corresponde a la junta accionada [21].



Cada cuerpo rígido tiene una posición y orientación relativa al marco base. La dirección está representada por los ángulos de Euler. La regla de rotación compuesta Yaw Pitch Roll (ZYX). El tipo de junta (conexión) describe el tipo de restricción, que puede ser uno de los siguientes:

- Revolvente (girar). Una articulación que gira a lo largo de un eje y está limitada por el rango definido por los límites superior e inferior.
- Continua. Torneado continuo sin límites superior e inferior girando alrededor del eje.
- Prismático. Una junta deslizante que se desliza a lo largo de un eje y tiene un rango limitado definido por límites superior e inferior. El eje de bisagra es el eje de rotación para juntas rotatorias y el eje de traslación para juntas prismáticas. El eje de conexión se especifica en el marco de unión. Los parámetros relacionados dinámicamente son la masa, la posición relativa del centro de gravedad y el momento de inercia del tensor. Los parámetros cinemáticos y dinámicos se mencionan a continuación:
 - Marco base
 - Posición relativa XYZ = “0mm 0mm 84mm”
 - Masa = “19.803kg”

Nota: La masa de la base incluye el marco metálico, la cubierta de plástico, la electrónica, la batería, los motores y los ejes. Sólo excluye las ruedas y la placa.

- Rueda delantera izquierda
 - Articulación parental = “marco base”;
 - Posición relativa XYZ = “228mm 158mm -34mm”;
 - Tipo de junta = “continuous”;
 - Eje articular = “0 1 0”;
 - Masa = “1.40kg”.

Nota: La masa de la rueda excluye el motor y el eje. El motor y el eje ya están incluidos en la base.

- Rueda delantera derecha
 - Articulación parental = “marco base”;
 - Posición relativa XYZ = “228mm -158mm -34mm”;



- Tipo de junta = “continua”;
 - Eje articular = “0 1 0”;
 - Masa = “1.40kg”.
- Rueda trasera izquierda
 - Articulación parental = “marco base”;
 - Posición relativa XYZ = “-228mm 158mm -34mm”;
 - Tipo de junta = “continua”;
 - Eje articular = “0 1 0”;
 - Masa = “1.40kg”
 - Rueda trasera derecha
 - Articulación parental = “marco base”;
 - Posición relativa XYZ = “-228mm -158mm -34mm”;
 - Tipo de junta = “continua”;
 - Eje articular = “0 1 0”;
 - Masa = “1.40kg”.
 - Plancha o placa
 - Articulación parental = “marco base”;
 - Posición relativa XYZ = “-159mm 0mm 46mm”;
 - Tipo de junta = “fija”;
 - Masa = “2.397kg”.

KUKA youBot brazo de 5 GDL

- Marco de la base del brazo
 - Articulación parental = “base”;
 - Posición relativa XYZ = “143mm 0mm 46mm”;
 - Tipo de junta = “fija”;
 - Masa = “0.961kg”.

Nota: Si el sistema de brazo KUKA youBot es operado sin una plataforma móvil, el bastidor base del brazo es el marco de origen. En este caso, su posición relativa XYZ es "0,0,0". Si el brazo está montado en la plataforma móvil, el bastidor de base del brazo representa una transformación de coordenadas con respecto al bastidor base de la plataforma móvil.



- Primera articulación del brazo
 - Articulación parental = “marco base del brazo”;
 - Posición relativa XYZ = “24mm 0mm 115mm”;
 - Orientación ZYX = “0° 0° 180°”;
 - Tipo de junta = “revolute”;
 - Eje articular = “0 0 1”;
 - Limites articulares = “-169° 169°”;
 - Masa = “1.390kg”, max. torque = “9.5Nm”;
 - Eje principal de inercia:
 - Posición XYZ = “15.16mm 3.59mm 31.05mm”;
 - Orientación ZYX = “180° 20° 0°”;
 - Inercia = “xx = 0.0029525kg·m² yy = 0.0060091kg·m² zz=0.0058821kg·m²”.

- Segunda articulación del brazo
 - Articulación parental = “primera articulación del brazo”;
 - Posición relativa XYZ = “33mm 0mm 0mm”;
 - Orientación ZYX = “-90° 0° 90°”;
 - Tipo de junta = “revolute”;
 - Eje articular = “0 0 1”;
 - Limites articulares = “-65° 90°”;
 - Masa = “1.318kg”, max. torque = “9.5Nm”;
 - Eje principal de inercia:
 - Posición XYZ = “113.97mm 15.0mm -19.03mm”;
 - Orientación ZYX = “-90° 0° -90°”;
 - Inercia = “xx = 0.0031145kg·m² yy = 0.0005843kg·m² zz=0.0031631kg·m²”.

- Tercera articulación del brazo
 - Articulación parental = “segunda articulación del brazo”;
 - Posición relativa XYZ = “155mm 0mm 0mm”;
 - Orientación ZYX = “-90° 0° 0°”;
 - Tipo de junta = “revolute”;
 - Eje articular= “0 0 1”;
 - Limites articulares = “-151° 146°”;
 - Masa = “0.821kg”, max. torque = “6.0Nm”;
 - Eje principal de inercia:
 - Posición XYZ = “0.13mm 104.41mm 20.22mm”;
 - Orientación ZYX = “0° 0° 90°”;
 - Inercia = “xx = 0.00172767kg·m² yy = 0.00041967kg·m² zz=0.0018468kg·m²”.



- Cuarta articulación del brazo
 - Articulación parental = “tercera articulación del brazo”;
 - Posición relativa XYZ = “0mm 135mm 0mm”;
 - Orientación ZYX = “0° 0° 0°”;
 - Tipo de junta = “revolute”;
 - Eje articular = “0 0 1”;
 - Limites articulares = “-102.5° 102.5°”;
 - Masa = “0.769kg”, max. torque = “2.0Nm”;
 - Eje principal de inercia:
 - Posición XYZ = “0.15mm 53.53mm -24.64mm”;
 - Orientación ZYX = “0° 180° 40°”;
 - Inercia = “xx = 0.0006764kg·m² yy = 0.0010573kg·m² zz=0.0006610kg·m²”

- Quinta articulación del brazo
 - Articulación parental = “cuarta articulación del brazo”;
 - Posición relativa XYZ = “0mm 113.6mm 0mm”;
 - Orientación ZYX = “0° 0° -90°”;
 - Tipo de junta = “revolute”;
 - Eje articular = “0 0 -1”;
 - Limites articulares = “-165° 165°”;
 - Masa = “0.687kg”, max. torque = “1.0Nm”;
 - Eje principal de inercia:
 - Posición XYZ = “0mm 1.2mm -16.48mm”;
 - Orientación ZYX = “0°90°0°”;
 - Inercia = “xx = 0.0001934kg·m² yy = 0.0001602kg·m² zz=0.0000689kg·m²”.

Pinza de 2 dedos KUKA youBot

- Marco de la base de la pinza
 - Articulación parental = “quinta articulación del brazo”;
 - Posición relativa = “0mm 0mm 57.16mm”;
 - Orientación ZYX = “180° 0° 0°”;
 - Tipo de junta = “fixed”;
 - Masa = “0.199kg”;
 - Eje principal de inercia:
 - Posición XYZ = “0mm 0mmq 28.9mm”;
 - Orientación ZYX = “180° 0° 90°”;
 - Inercia = “xx = 0.0002324kg·m² yy = 0.0003629kg·m² zz=0.0002067kg·m²”.



- Articulación del dedo izquierdo de la pinza
 - Articulación parental = “base de la pinza”;
 - Posición relativa = “0mm 8.2mm 0mm”;
 - Tipo de junta = “prismatic”;
 - Eje articular = “0 1 0”;
 - Limites articulares = “0mm 12.5mm”;
 - Masa = “0.010kg”.

- Articulación del dedo derecho de la pinza
 - Articulación parental = “base de la pinza”;
 - Posición relativa = “0mm -8.2mm 0mm”;
 - Tipo de junta = “prismatic”;
 - Eje articular = “0 -1 0”;
 - Limites articulares = “0mm 12.5mm”;
 - Masa = “0.010kg”.

Capítulo 6

6 Plataforma Experimental de Simulación

6.1 Modelado en Blender

En la siguiente se presenta el modelado del Kuka youBot.

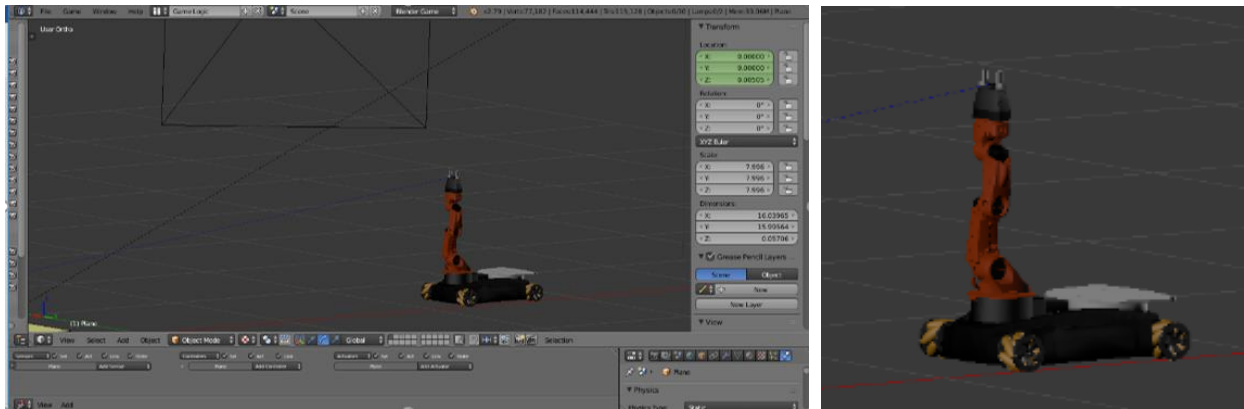


Fig. 69 Modelado del Kuka youBot en Blender

Con BGE (Blender Game Engine) o motor de juegos, puede expresar imágenes y sonidos especiales y simular movimientos físicos y colisiones. Manipule toda la lógica **en tiempo real** a través de propiedades lógicas de objetos y componentes y programación Python. B (figura. 70) Conoce a Kuka youBot en BGE Blender.

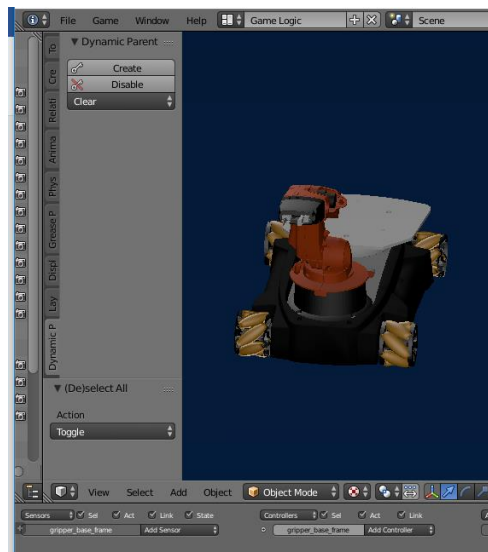


Fig. 70 Kuka YouBot en BGE

En las siguientes figuras de la (Fig. 71) a la (Fig. 4.6), se muestra la simulación de movimiento realizada, utilizando drivers y respetando la cinemática del modelo.

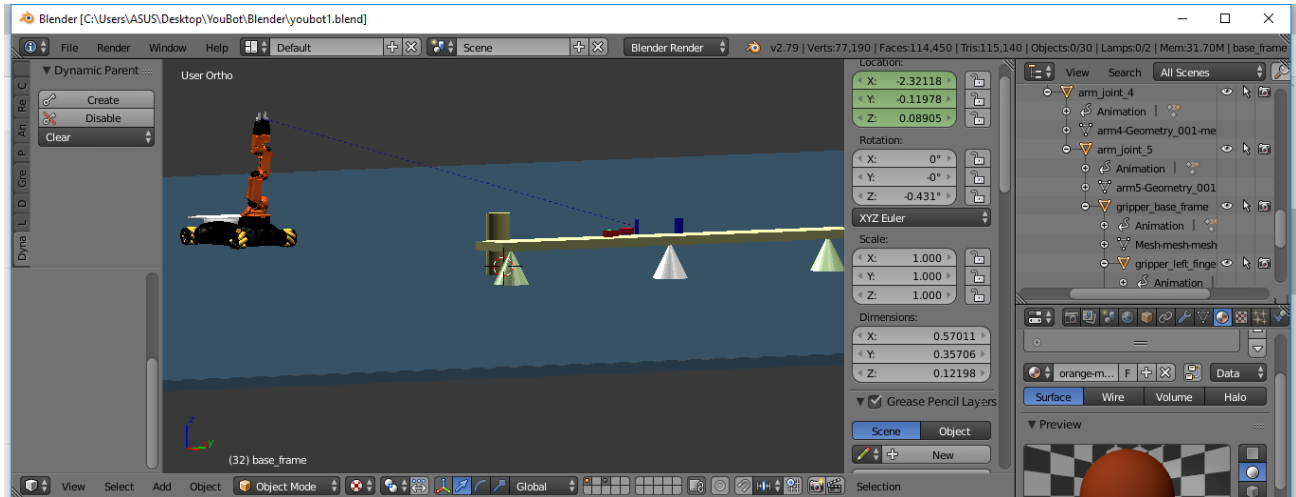


Fig. 71 Entorno de la simulación



Fig. 72 Parte de la secuencia de la simulación

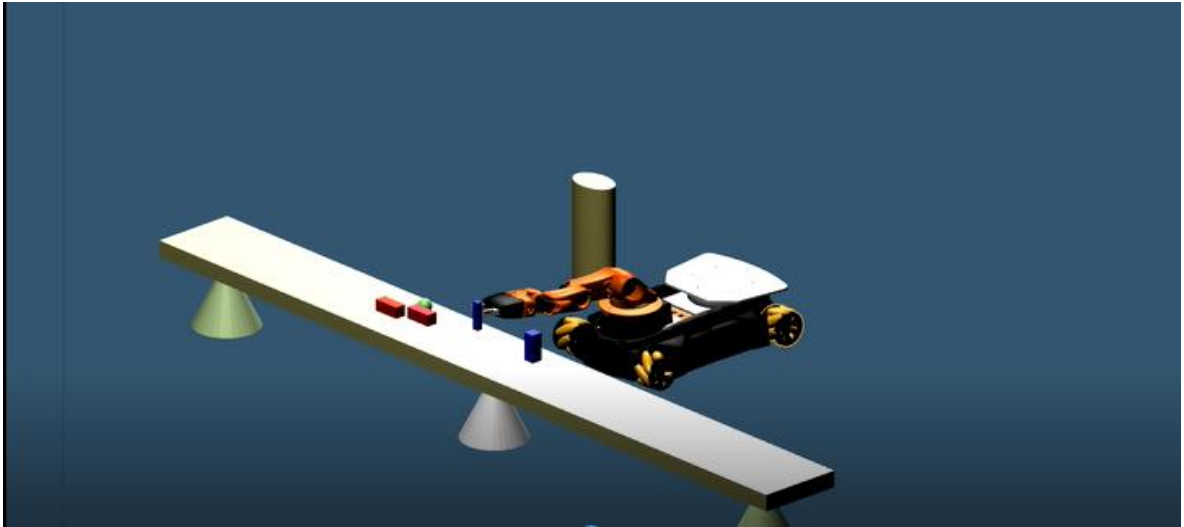


Fig. 73 El gripper toma el objeto

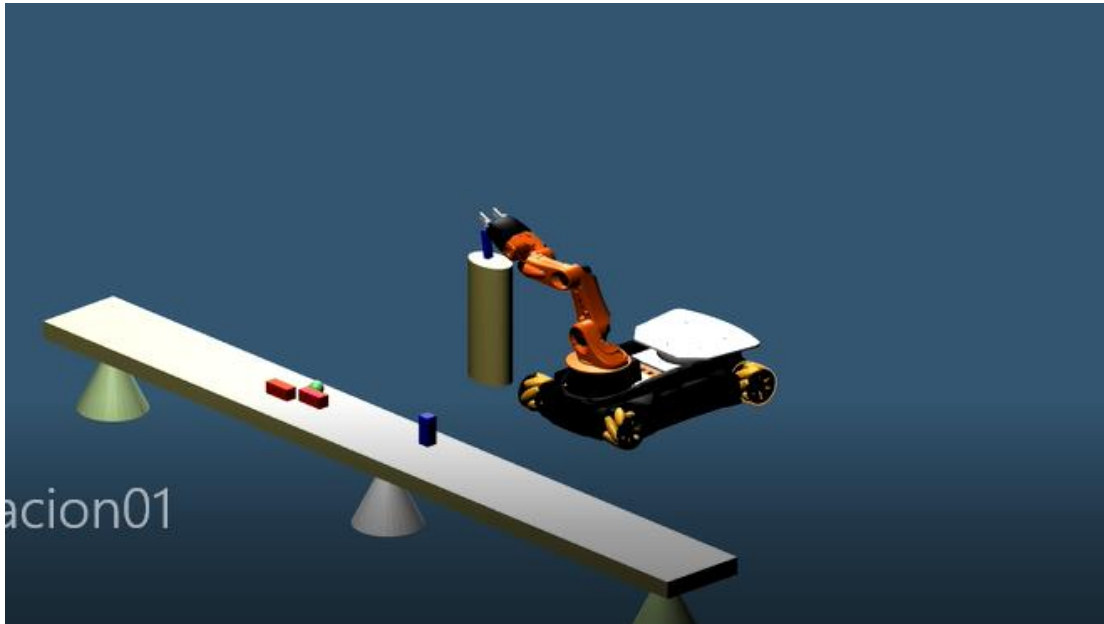


Fig. 74 Parte final de la secuencia de la simulación de movimiento. Deja el objeto en otro lugar



A continuación, en la (Fig. 75) a la (Fig. 76), se muestra la programación lógica para controlar los movimientos del robot MM, servomotores, sensores de presencia para evitar obstáculos, de velocidad. El entorno el espacio donde se mueve el MM, lo estamos llevando a cabo con una cámara, a través de un sensor de visión y la lógica podemos hacer múltiples actividades.

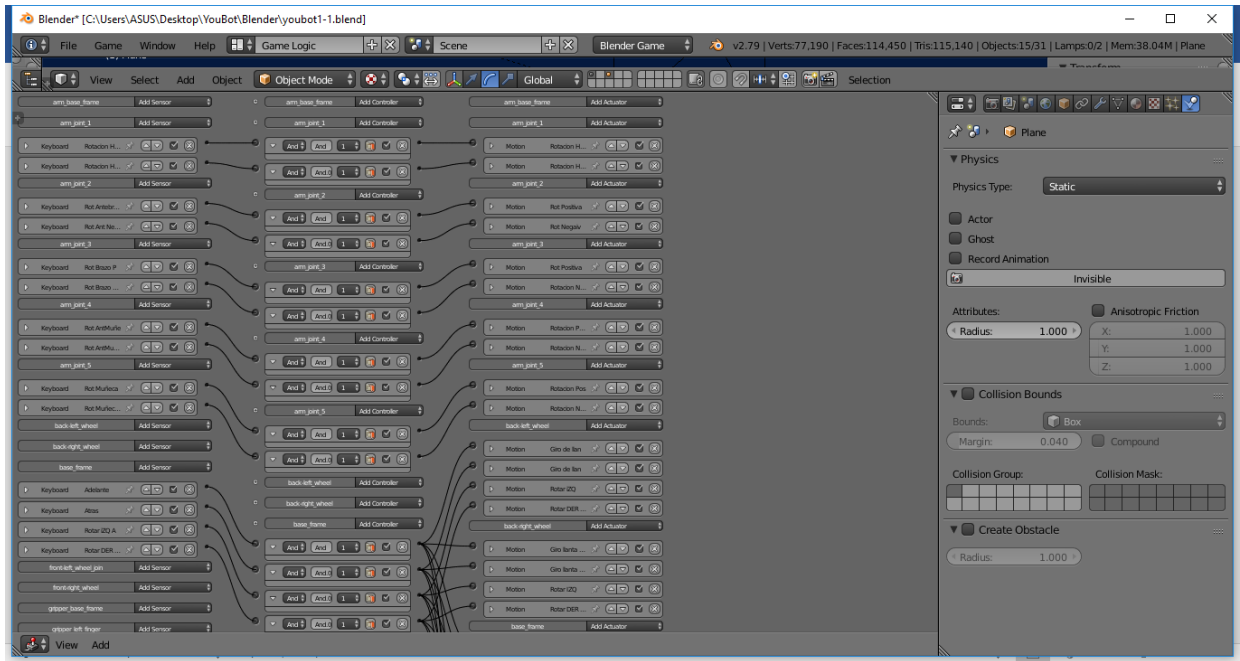


Fig. 75 Lógica realizada en el BGE de Blender

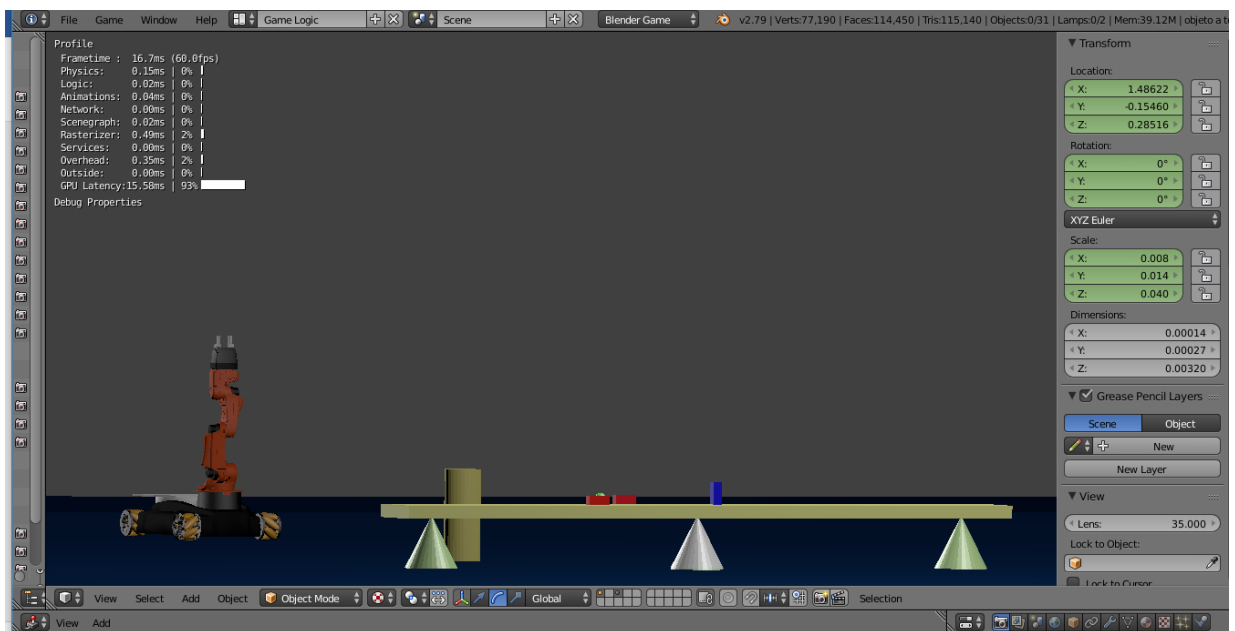


Fig. 76 a) Lógica de simulación, control del Kuka YouBot por teclado, coordenadas de trayectoria

6.2 Modelado V- Realm (VRML) – Simulink

En este caso, el sujeto se creó en un entorno simple formado por YouBot y el suelo sobre el que se mueve. En Simulink, puede acceder al "mundo" creado a través de bloques. Cuando se ejecuta en VRML-Builder, se asigna a bloques específicos en Simulink Matlab para programarlos para simular el entorno creado, y Simulink le permite agregar los controles necesarios. En la figura. La (Fig. 77) muestra cómo los dos programas están vinculados para formar una interfaz.

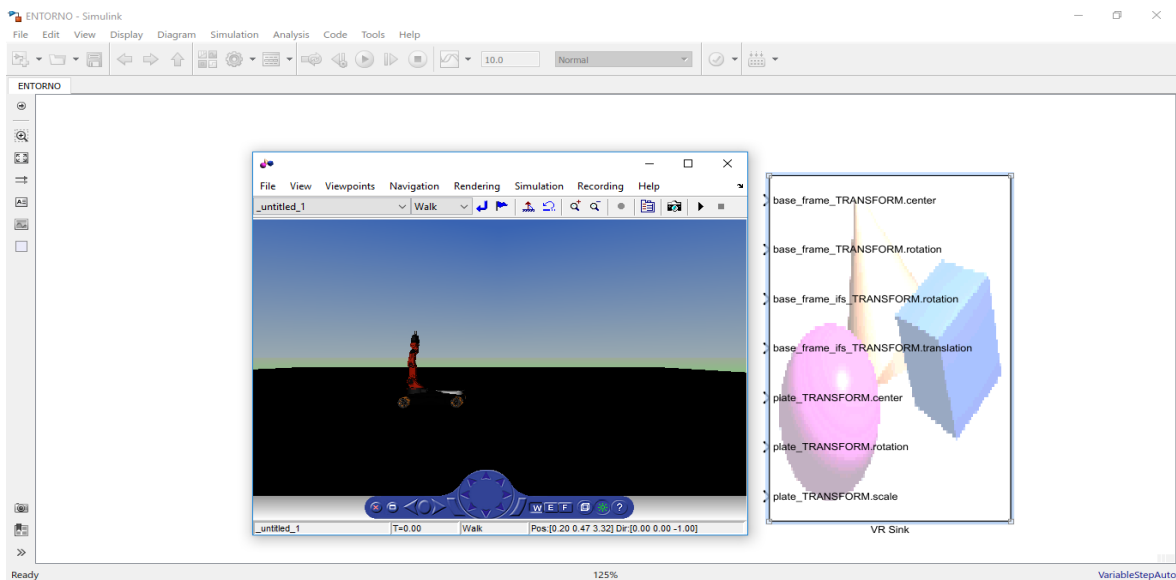


Fig. 77 Simulink- Entrando al sistema "mundo" VRML-Builder

(Figura 78) muestra cómo se ve el sistema físico asociado con Kuka youBot en el mundo VRML-Builder.

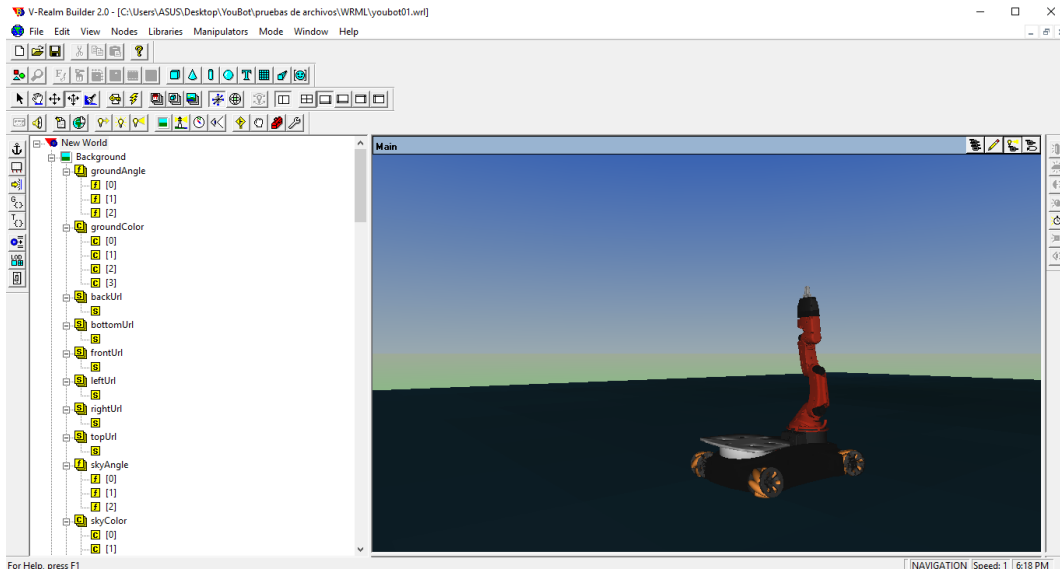


Fig. 78 “Mundo” VRML. Sistema físico YouBot.

6.3 SolidWorks- Simscape Multibody

El modelo se crea primero en SolidWorks y luego se exporta a SimMechanics, pero ambos programas deben estar vinculados para realizar este movimiento. Cuando se ejecuta con el tipo de código Matlab, se llama para mostrar SimMechanics y el diagrama de bloques en él. Por lo tanto, puede aplicar el control. Las pruebas de simulación están en curso en el entorno de SolidWorks Motion Study, pero actualmente se están probando con enlaces para abrir la plataforma y los diagramas de bloques de brazos móviles por separado. Estos se muestran en las (Fig. 80) y (Fig. 81).

Siendo igual de importante no hay que olvidar las pruebas de estudio de movimiento en Study Motion se siguen llevando a cabo, se debe tener cierto nivel para poder manejar este apartado de SolidWorks, Ya se lograron mover los 4 motores de la parte móvil, en el modelado sólido, el gripper no se ha colocado (Fig. 79).

6.3.1 Procedimiento de enlace de SolidWorks-Simulink (MATLAB)

Para poder enlazar los dos programas, se deben de tener en cuenta lo siguiente.



- Matlab y SolidWorks deben de ser del mismo año, por ejemplo: Matlab 2018a, SolidWorks 2018.
- Para descargar los paquetes adecuados tomar en cuenta la versión del Matlab, ejemplo: Matlab 2018a, es versión a.
- Que el equipo de cómputo revisar si es de 32 o 64 bits.

El procedimiento es el siguiente:

1. Tener instalados Matlab y SolidWorks.
2. Ingresar a la página oficial de Matlab, buscar y descargar los paquetes para Matlab Sim Multibody.
3. Una vez descargarlos dejarlos en un lugar fácil de ubicar.
4. Abrir Matlab.
5. En la ventana *Command Window*, escribir el siguiente comando para instalar los archivos descargados: `install_addon ('nombre archivo')`, ejemplo (`'smlink.r2018a.win64.zip'`). Enter. Al final te sale un cuadro donde te indica que la instalación fue correcta, *Installation of Simulink complete*.
6. Vincular Matlab con SolidWorks, escribir en *Command Window* de Matlab: `smlink_linksw`. Al final debe salir el siguiente mensaje: *Registering dll... correct...*
7. Abrir SolidWorks. En complementos (aparece un engrane de símbolo) dar click y en las opciones te debe aparecer *SimMechanics Link*, seleccionarlo, click en OK.
8. Ya se pueden exportar modelos ensambles a MATLAB.

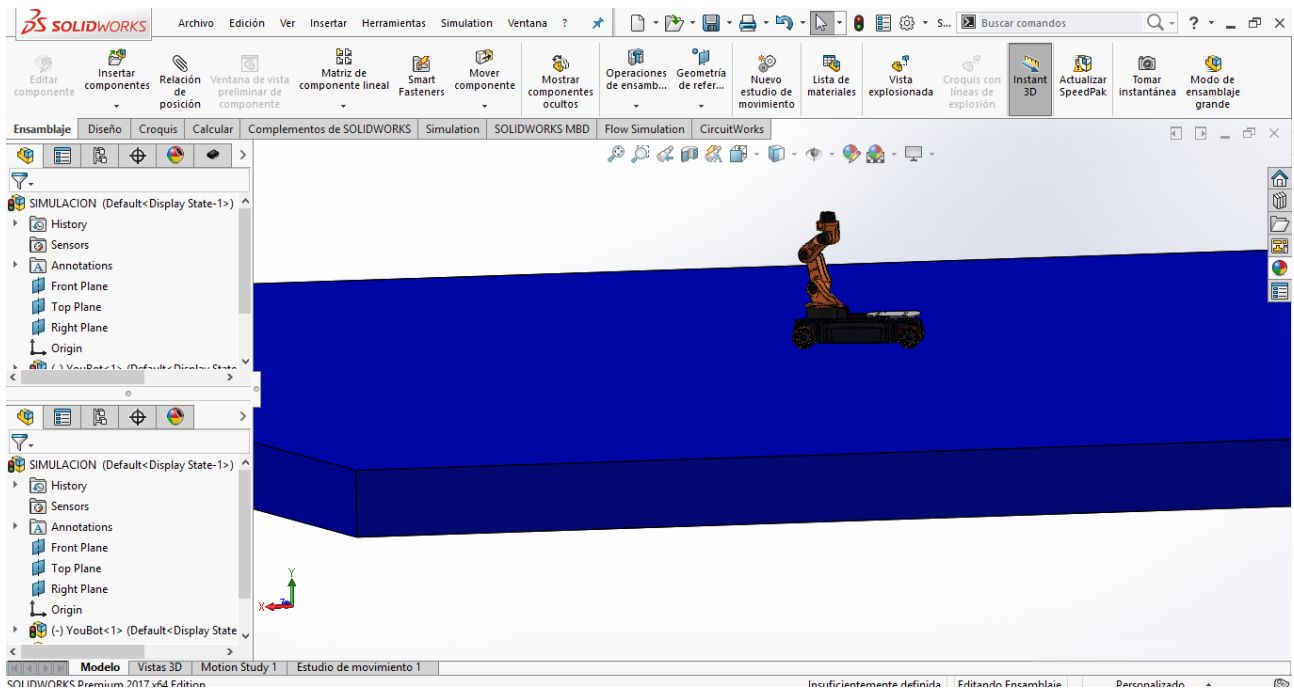


Fig. 79 Motion Study. Entorno y sistema físico YouBot en

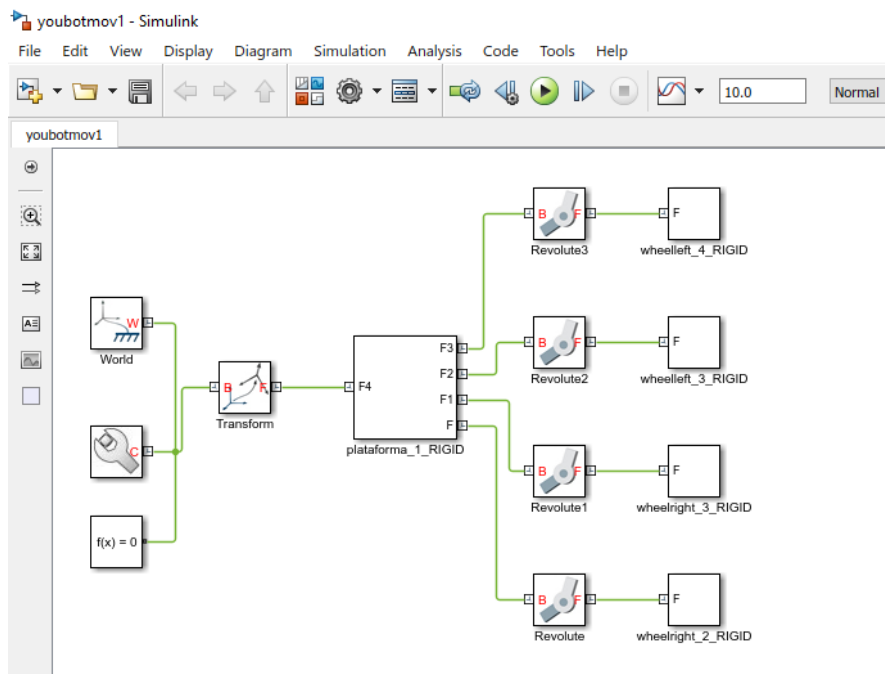


Fig. 80 Sistema físico en SimMechanics parte móvil

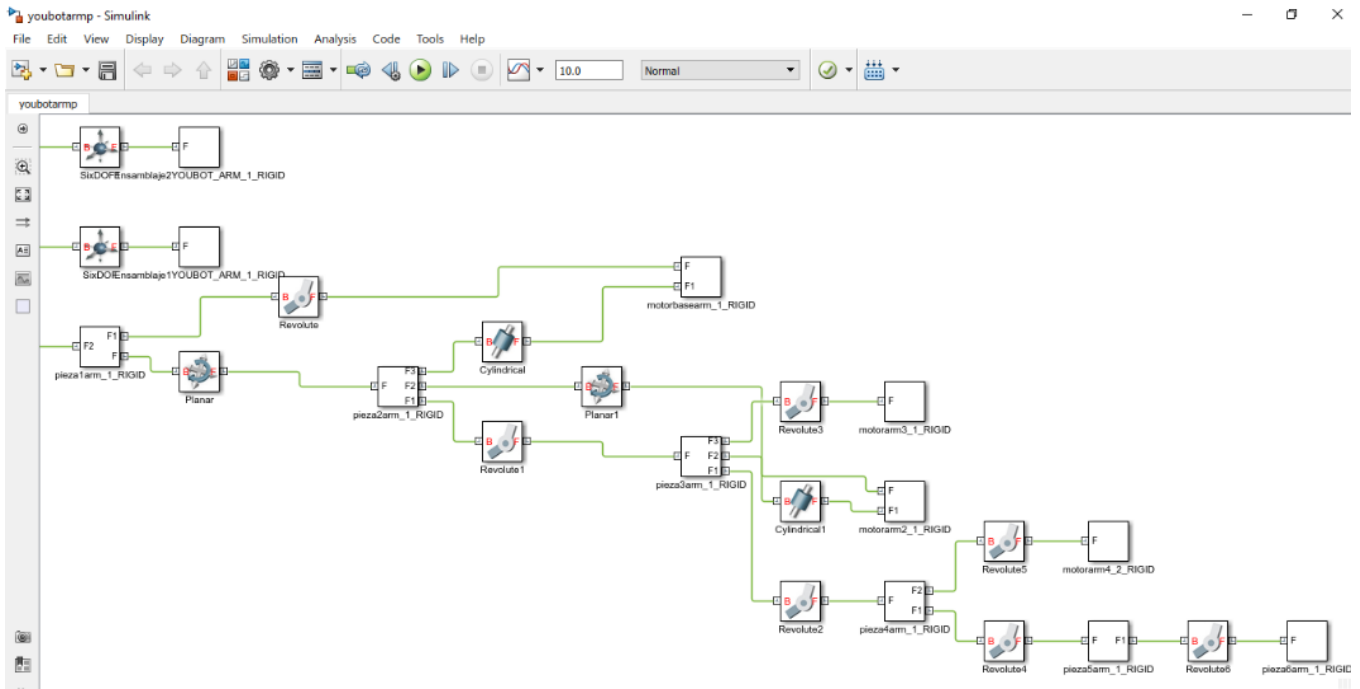


Fig. 81 Sistema físico SimMultibody del MM

Capítulo 7

7. Resultados de la Simulación

En el primer caso de Blender está completa la simulación, aunque pueden realizarse mejoras. De forma interna Blender calcula matemáticamente la posición que se encontrará. Respeta en todos los sentidos la geometría y cinemática del objeto, ya que, si sales fuera de los parámetros o si estas aplicando un resultado de manera incorrecta se notará en el resultado de la simulación, en un movimiento mal realizado, donde podría afectar de manera física al MM y un consumo mayor de energía. El programa es muy amigable y no requiere tanta experiencia para poder empezar a crear cosas.

VRML parece ser fácil demasiado amigable pero muy “antiguo”, para la actualidad se vuelve tedioso, ya que la forma de sus controles es muy austero y ambiguo; aunque para su uso en Simulink es esencial, continuará trabajando desde Simulink, comenzando con el controlador PID y trabajando en simulaciones basadas en comandos.

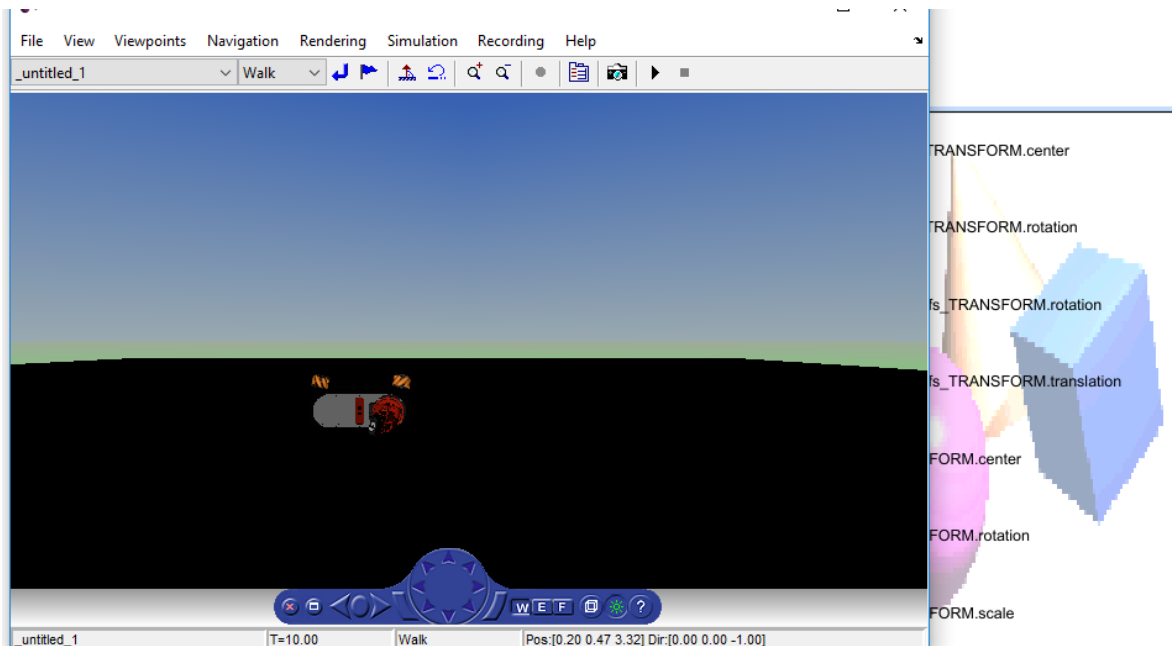


Fig. 82 Kuka youBot desde el mundo VRML, el robot cae.



En el caso en SolidWorks se tiene en un 85% por ciento el modelado del objeto ya que faltan de realizar el gripper (efecto final), el enlace esta correcto entre éste y Simulink.

7.1 Conclusiones

Blender es un software muy amable con controles muy amigables son fáciles de entender, puedes simular en tiempo real y respeta los parámetros del modelado cinemático. Si existen errores te los marca y puedes ver cómo le afectan al modelo.

VRLM-Builder es un poco ambiguo no te deja manejar al modelo con libertad y el modelado es algo tedioso de realizar, pero en contraste junto con Matlab-Simulink lo hacen una herramienta muy poderosa.

SolidWorks es muy eficiente, sobre todo en el enlace con SimMultibody, para el estudio de controles para los sistemas robóticos, pero si requiere de ciertas bases de conocimientos de diseño y programación, se continuará con el presente objeto de estudio para completar la simulación.

Se espera continuar las pruebas en Motion Study de SolidWorks; se debe completar el modelo en SimMechanics, la parte móvil y la parte del gripper del brazo manipulador del Kuka youBot una vez exportado de SolidWorks, para agregar los controles y poder simular y completar la interface desde Matlab.

Aún no se han realizado pruebas con LabVIEW – SolidWorks.

7.2 Trabajos Futuros

- Validar las tres plataformas experimentales de simulación aplicando diferentes tipos de controles para el movimiento dinámico, así mismo, realizar la comparación de eficiencia entre el control PID estándar y un control inteligente; terminar el gripper en SolidWorks del MM, programar desde Simulink el bloque de enlace con el mundo del robot de VRML programar el sistema físico del MM en Simulink a partir la importación del CAD desde SolidWorks.



- Realizar investigación documental de los índices de consumo de energía y de los índices de estabilidad en condiciones de trabajo no convencionales de los robots móviles con brazo manipulador, para después, desarrollar un programa para calcular la energía que utiliza en trabajo a diario.
- Trabajar con códigos de control para reducir la cantidad de fallas de comunicación del robot móvil con brazo manipulador.
- Una vez que se encuentre el tipo de control más adecuado se realizará la verificación de los materiales a necesitar para construir el controlador para su requisición.
- Construir el controlador.
- Realizar pruebas en físico en laboratorio.
- Realizar pruebas en físico en condiciones de campo.



Referencias Bibliográficas

- [1] Olledo Anibal. *Róbotica Manipuladores y Robots Móviles*. Alfaomega Marcombo, México, 2007.
- [2] Army Technology. "Telerob 2", Consultado Febrero 2012. <http://www.armytechnology.com/contractors/mines/telerob/telerob2.html>.
- [3] Army Technology. "Telerob 3", Consultado Febrero 2012. <http://www.armytechnology.com/contractors/mines/telerob/telerob3.html>.
- [4] Universidad de Argentina. "TeleMAX", Consultado Febrero 2012. <http://noticias.universia.com.ar/enportada/noticia/2005/01/14/373162/ubadesarrolla-ra-primer-robot-manipular-explosivos.html>.
- [5] RoboTiCan. "Komodo", Consultado Mayo 2014. <http://www.robotican.net/#komodo/c9sa>.
- [6] García O., Solaque L., Niño P., and Avilés. O. "Hardware and software architecture of a mobile robot with anthropomorphic arm". 2010 IEEE ANDEAN CONFERENCE – ANDESCON, Pages Bogotá, Colombia, Septiembre 15 – 17 2010.
- [7] Bogotá Periódico de la Universidad Militar de Nueva Granada. "El Neogranadino", Consultado Noviembre 2013.
- [8] Laboratoire d'analyse et d'architecture des systèmes (LAAS). Centre National de la Recherche Scientifique (CNR). "Hilare 2 bis", Consultado Febrero 2015. <http://homepages.laas.fr/matthieu/robots/h2bis.shtml>.
- [9] Hvilshoj M., Bogh S., Madsen O., and Kristiansen M. "The Mobile Robot "Little Helper": Concepts, ideas and working principles. Department of Production".



Universidad de Aalborg, Dinamarca, Consultado Febrero 2012.
http://www.machinevision.dk/joomla/images/artikler/etfa09_concepts.pdf.

[10] Spyros G. Tzafestas Web-Based Controls and Robotic Educations, Athens, Greece, 2009. Volumen 38.

[11] Seelinger M, Yoder JD, Baumgartner ET, Skaar BR. High-precision visual control of mobile manipulators. IEEE Trans Robot Autom 2002;18(6):957_65.

[12] De Luca A, Oriolo G, Giordano PR. Image-based visual servoing schemes for nonholonomic mobile manipulators. Robotica 2007; 25:131_45.

[13] Galarce Acevedo Patricio. Control de Trayectoria de Robots Manipuladores Mviles Utilizando Retroalimentación Linealizante. Enero 2015.

[14] Villela, V. J. G. and R. M. Parkin (2005). Evadiendo obstáculos con robots móviles. Revista Digital Universitaria. México, DGSCA-UNAM. Vol. 6: 2-9.

[15] Control Servovisual de un Kuka youBot para manipulación y traslado de objetos. F. Mirelez- Delgado; A. Morales-Díaz; R. Ríos Cabrera; H. Pérez- Villeda. Congreso Nacional de Control Automático. AMCA 2015.

[16] Unified Closed Form Inverse Kinematics for the Kuka youBot. Shashak Sharona, Gerhad K. Kraetzshmar, Christian Scheuner, Rainer Birschoff. Robotik 2012.

[17] <http://www.matworks.com>

[18] <https://www.matlab.com>

[19] <https://www.blender.org>

[20] <https://www.solidworks.com>

[21] <http://www.youbot-store.com>

[22] <https://www.ni.com>

Anexos

Control del Péndubot: El Caso del Seguimiento de Trayectoria Mediante el Regulador Lineal

Jalomo Cuevas Jaime

Instituto Tecnológico de Colima

jajmejo@itcc.edu.mx

Lázaro Flores Erika Berenice

Institución Tecnológico de Colima

erika.lflorez@umail.com

Resumen

En este trabajo se presenta la solución al problema del seguimiento de trayectorias usando la teoría del regulador lineal aplicado al sistema subactuado conocido como péndubot. Se describen las pruebas de simulación en donde se validan las propiedades estabilidad y de robustez ante perturbaciones en el estado. También se presenta la visualización virtual creada para este sistema para agregarle un enfoque.

Palabra(s) Clave: Seguimiento de trayectorias, simulación, péndubot, estabilidad, robustez.

Abstract

In this paper, is presented the solution to the trajectories tracking problem using the linear regulator theory applied to the subactuated system known as péndubot. Are described the simulation tests in where are validated the stability and robustness properties against disturbances in the state. Is also presented, the virtual visualization created for this system to add a focus.

Keyword(s): Trajectories tracking, simulation, pendulum, stability, robustness.



HERRAMIENTAS VIRTUALES PARA LA SIMULACION Y ESTUDIO DE MOVIMIENTO DEL ROBOT KUKA YUBOT

Lázaro Flores Erika Berenice

Institución Tecnológico de Ciudad Guzmán

erika.lazflo@gmail.com

Jalomo Cuevas Jaime

Instituto Tecnológico de Ciudad Guzmán

jaimelc@itcg.edu.mx

Resumen

En este artículo se presenta las herramientas actuales de simulacion de movimiento de sistemas robóticos, en este caso un robot móvil manipulador, en específico, como objeto de estudio es el Kuka youBot, para su análisis y comparar la eficiencia del estudio realizado.

En general para realizar la simulación de movimiento ROS, Gazebo, en Linux, se presentan como únicos software para poder realizarlo, pero se tiene poca accesibilidad al sistema operativo ya que es muy costoso, se requiere tener desde una computadora de un tipo específico y conocimientos de programación de código más avanzado, dejando a los usuarios sin opciones o con mucha limitación; pero no es así, en Windows se tienen opciones, en este caso se presenta el uso del software Blender, de código abierto, también VRML-Matlab y Simulink-SimMechanics, donde se puede trabajar desde el modelado, pasando por el proceso de simulación y puedes agregar controles al modelo, no necesitas una computadora especial ni tampoco ser un experto para poder entrar en el mundo de la simulación.

Palabra(s) Clave: Simulación, movimiento, Blender, VRML, Sim, SimMechanics.



Abstract

In this paper is presented the current simulation tools of robotic systems movement, in this case a mobile robot manipulator, specifically, as the object of study is the YouBot, for analysis and compare the study.

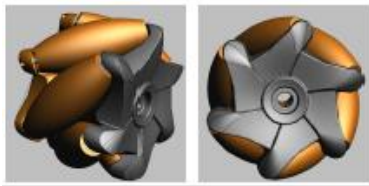
In general to perform the simulation of ROS movement, Gazebo, in Linux, is presented as the only software to be able to do it, but it has little access to the operating system since it is very expensive, it is required to have from a computer of a specific type and knowledge more advanced code programming, leaving users without options or with much limitation; but this is not the case, in Windows there are options, in this case the use of Blender software, open source, also VRML-Matlab and Simulink-SimMechanics, where you can work from modeling, through the simulation process and you can add controls to the model, you don't need a special computer or be an expert to enter the world of simulation.

Keyword(s): *Simulation, movement, Blender, VRML, Sim, SimMechanics.*

Anexos de Tablas y Figuras

Technical Specifications

youBot platform



General characteristics: youBot platform

Omnidirectional kinematics	4 kuka omniWheels
Length	580 mm
Width	380 mm
Height	140 mm
Clearance	20 mm
Weight	20 kg
Payload	20 kg
Structure	Steel
Speed	0.8 m/s
Communication	EtherCAT
Voltaje connection	24 v

Tabla A3. platform Kuka youBot characteristics

Energy supply

Maintenance-free lead acid rechargeable batteries: 24 v, 5 Ah, 4 kg. Approximate runtime of youBot mobile manipulator: 90 minutes power adapter: 200 w, 24 v

On-board PC

Mini iT X PC- Board with embedded CPU (Intel Atom Dual Core), 2 GB RAM, 32 GB Solid State Disk, 2 LAN Ports, 8 x USB, 6 x COM

Tabla A4. platform Kuka youBot characteristics, part 2

youBot arm + gripper

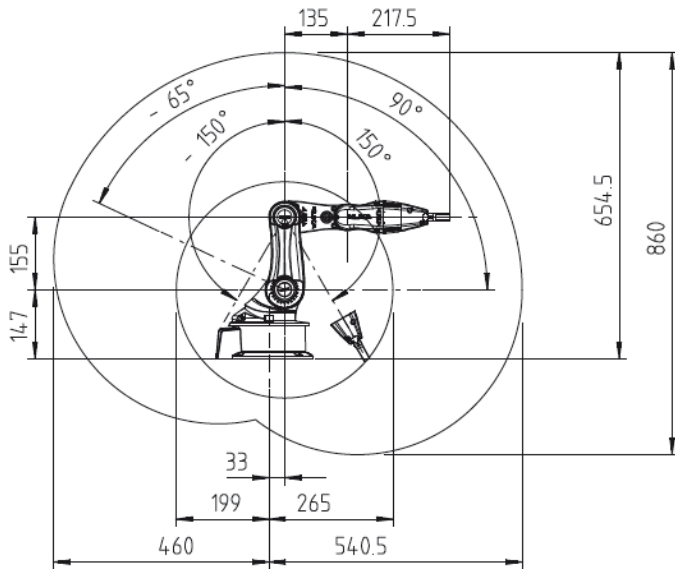


Figura A2. Kuka youBot arm

General characteristics: youBot arm

Serial kinematics	5 axes
Height	655 mm
Work envelope	0.513 m ³
Weight	6.3 kg
Payload	0.5 kg
Structure	Magnesium cast
Positioning repeatability	1 mm
Communication	EtherCAT
Voltaje connection	24 v
Power can be limited to	80 w



Figura A3. Kuka youBot arm

Gripper data

	Detachable, 2 fingers
Stroke	20 mm
Range	70 mm

Tabla A6. Kuka youBot, gripper data



Definiciones y Abreviaturas

Holonómico y No-Holonómico: Una forma posible de clasificar un robot es determinar si es o no holonómico. Esta diferencia se debe a su movilidad. En pocas palabras, un robot o sistema holonómico puede cambiar de dirección instantáneamente (suponiendo masa cero) sin necesidad de una rotación previa. Por ejemplo, un vehículo con un sistema de dirección, como un automóvil, no lo es porque requiere algunas operaciones preliminares para moverse lateralmente. De manera similar, los robots de dos ruedas no son holonómicos porque no pueden moverse de un lado a otro. Siempre lo hace hacia adelante en la dirección determinada por la velocidad de sus ruedas.

AUVs: Autonomous Underwater Vehicles.

CC: Corriente Continua (electricidad).

GDL: Grados de Libertad.

LMRS: Legged Mobile Robots.

MM: Manipuladores Móviles.

PD: Proporcional Derivativo.

UAVs: Unmanned Aerial Vehicles.

WMRs: *Wheeled Mobile Robots.*