



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**  
**PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA**  
**MECÁNICA – MECATRÓNICA**

**ROBOT MÓVIL OMNIDIRECCIONAL REDUNDANTE: MODELADO, CONTROL E  
IMPLEMENTACIÓN**

**TESIS**  
**QUE PARA OPTAR POR EL GRADO DE:**  
**MAESTRO EN INGENIERÍA**

**PRESENTA:**  
**EMMANUEL DORANTES MALAGÓN**

**TUTOR PRINCIPAL:**  
**DR. VÍCTOR JAVIER GONZÁLEZ VILLELA,**  
**FACULTAD DE INGENIERÍA**

**MÉXICO, D. F. SEPTIEMBRE 2015**

**JURADO ASIGNADO:**

Presidente: Dr. López Parra Marcelo

Secretario: Dr. Díaz Hernández Octavio

1 er. Vocal: Dr. González Villela Víctor Javier

2 do. Vocal: Dra. Corona Lira María Del Pilar

3 er. Vocal: Dr. Martínez Zamudio Patricio

Lugar o lugares donde se realizó la tesis:  
Facultad de Ingeniería

**TUTOR DE TESIS:**

DR. VÍCTOR JAVIER GONZÁLEZ VILLELA

-----  
**FIRMA**

## AGRADECIMIENTOS

---

*A mis padres Rosalio y María quienes me dieron la vida, por su apoyo durante toda mi formación profesional y en el cumplimiento de mis metas; por sus regaños, comprensión, cariño y confianza.*

*A mis hermanos Josué y Mariana por su compañía durante las horas de estudio, por hacerlas más amenas.*

*A mi novia Angie por todos estos años juntos; por su aliento y apoyo para continuar; por ser una inspiración; por su paciencia y cariño.*

*A mis tíos Jesús, Agustín, Rosa y Guadalupe; y a mi abuela Trinidad por todo su apoyo y cariño.*

*A todo el grupo MRG, por los conocimientos, la experiencia y el ánimo brindado.*

*Al Dr. Víctor, por su motivación, orientación y confianza en la realización de esta Tesis; y por sus enseñanzas tanto dentro como fuera del aula.*

*A la Universidad Nacional Autónoma de México y a la Facultad de Ingeniería por haber haberme otorgado la oportunidad de ampliar mi formación profesional*

*Agradezco en lo que corresponde a DGAPA, por el apoyo brindado para la realización de este trabajo, a través del proyecto PAPIIT IN117614: “Robótica intuitiva, adaptable, reactiva, híbrida y móvil aplicada al servicio, el rescate y la medicina”*

*Así también, se extiende un agradecimiento al programa de becas de CONACYT por el apoyo económico que permitió dirigir la atención directamente a la realización de este proyecto.*

## RESUMEN

---

El presente trabajo describe el análisis y el desarrollo de una nueva configuración de robot móvil omnidireccional, que no emplea llantas o mecanismos especiales. El robot presentado consiste en una plataforma rectangular central unida en sus esquinas a cuatro robots móviles diferenciales; que le otorgan la característica de ser redundante. Se presenta la postura del robot; el análisis cinemático y dinámico; el análisis de la omnidireccionalidad; se analiza la característica de redundancia del robot; se implementa un controlador basado en campos potenciales artificiales y se realizan los experimentos de regulación y seguimiento de postura. Los experimentos realizados se efectuaron en tiempo real, mediante el empleo de Simulink Matlab y de ReacTIVision. Se comprueba tanto en teoría como en práctica que el robot presentado es omnidireccional; la omnidireccionalidad se debe al uso mínimo de dos robots móviles diferenciales unidos a la plataforma central; y la redundancia al empleo de tres o más robots móviles diferenciales.

## ABSTRACT

---

A new type of omnidirectional mobile robot, without special wheels or mechanisms, with the characteristic of be redundant is presented. The robot is made of four differentially driven mobile robots and one platform. The kinematics of the robot; the degree of mobility, steerability and maneuverability are presented. An approach of artificial potential fields is applied to the state space representation of the kinematic model, in order to control the movements of the robot. Finally it was proved, theoretically and practically, that the mobile robot shown in this work is omnidirectional and redundant; the omnidirectionality is due to the minimal use of two differential mobile robots; and the redundancy is due to the combined use of three or more differential robots, in this particular case four.

## PREFACIO

---

La importancia de los robots móviles radica en que poseen un espacio de trabajo ilimitado, a diferencia de los robots manipuladores fijos los cuales están restringidos a cumplir tareas dentro de un espacio de trabajo determinado por sus dimensiones físicas. Es debido a esto que se busca aumentar la movilidad de los robots mediante la anexión de un sistema locomotor que le permita desplazarse libremente en un área de trabajo. Los robots móviles tienen la capacidad de adaptarse a una gran diversidad de terrenos y ambientes no estructurados.

Las aplicaciones de los robots móviles incluyen minería, construcción, misiones espaciales, exploración planetaria, mantenimiento, vigilancia, seguridad, etcétera. Las tareas más comunes que puede realizar un robot, ya sea móvil o no, son la de llegar a una posición fija con determinada orientación y la de seguimiento de postura; esta última consiste en estabilizarse alrededor de una trayectoria variante en el tiempo.

Los robots móviles se pueden clasificar de acuerdo con el tipo de locomoción utilizado; los tres más comunes son ruedas, patas y orugas. En este trabajo se habla de los robots móviles que emplean ruedas, en particular de los robots móviles omnidireccionales; definiéndolos como un vehículo capaz de moverse de manera autónoma o semiautónoma sobre una superficie, mediante la acción de las ruedas montadas sobre el robot, en cualquier dirección permisible por el medio.

En el primer Capítulo se mencionan los principales trabajos previos realizados en el ámbito de los robots móviles omnidireccionales y de los robots móviles redundantes. En el Capítulo 2 se describe la postura del robot móvil omnidireccional redundante; se realizan los cálculos cinemáticos y dinámicos mediante el método de propagación de velocidades; se analizan los grados de libertad del robot móvil; y se describe la redundancia del sistema. El Capítulo 3 describe por completo al sistema conformado por el robot móvil omnidireccional redundante y el sistema de visión; empezando por una descripción general de su funcionamiento; continuando con una descripción de cada elemento que lo constituye desde los microcontroladores hasta los drivers de los motores; se describen los sistemas en tiempo real y sus características; y finalmente se describen los bloques de códigos que se encargan de procesar toda la información que se necesita para realizar las tareas de seguimiento de postura y de regulación. En

el Capítulo 4 se analizan dos tipos de controladores que pueden ser empleados en este tipo de robots; primero se describe el controlador por dinámica inversa y después el controlador basado en campos potenciales artificiales, siendo este último el que se implementa en el robot. El Capítulo 5 analiza los experimentos realizados; empezando por definir las características bases sobre las cuales se ejecutaron todos los experimentos; después describiendo y mostrando los resultados de los experimentos de regulación; y por último describiendo y mostrando los resultados de los experimentos de seguimiento de postura. Finalmente el capítulo 6 muestra las conclusiones obtenidas, así como el trabajo a futuro que queda por realizar.

# TABLA DE CONTENIDO

<b>AGRADECIMIENTOS</b> .....	<b>III</b>
<b>RESUMEN</b> .....	<b>IV</b>
<b>ABSTRACT</b> .....	<b>IV</b>
<b>PREFACIO</b> .....	<b>IV</b>
<b>TRABAJOS PREVIOS</b> .....	<b>1</b>
<b>JUSTIFICACIÓN</b> .....	<b>3</b>
<b>HIPÓTESIS</b> .....	<b>3</b>
<b>OBJETIVOS</b> .....	<b>4</b>
<b>APORTACIONES</b> .....	<b>4</b>
<b>INTRODUCCIÓN</b> .....	<b>6</b>
<b>ROBOTS MÓVILES CON RUEDAS</b> .....	<b>6</b>
<i>Tipos de ruedas</i> .....	7
<i>Robot móvil diferencial</i> .....	7
<i>Robots móviles omnidireccionales</i> .....	7
<i>Robot móvil omnidireccional redundante</i> .....	8
<b>MODELADO CINEMÁTICO</b> .....	<b>8</b>
<b>POSTURA DEL ROBOT</b> .....	<b>9</b>
<b>PROPAGACIÓN DE VELOCIDADES</b> .....	<b>10</b>
<i>Propagación de velocidad: marco coordenado P al marco coordenado 1</i> .....	11
<i>Propagación de velocidad: Marco coordenado 1 al 11</i> .....	13
<b>RESTRICCIONES CINEMÁTICAS</b> .....	<b>13</b>
<i>Restricción sobre el plano de giro <math>(x_{11}, z_{11})</math> de la rueda</i> .....	14
<i>Restricción ortogonal al plano de giro de la rueda <math>(y_{11}, z_{11})</math></i> .....	15
<b>CINEMÁTICA DIFERENCIAL INVERSA</b> .....	<b>15</b>
<b>MODELO DINÁMICO</b> .....	<b>17</b>
<b>OMNIDIRECCIONALIDAD Y REDUNDANCIA</b> .....	<b>18</b>
<b>OMNIDIRECCIONALIDAD</b> .....	<b>18</b>
<i>Grado de movilidad</i> .....	19
<i>Grado de direccionalidad</i> .....	19

<i>Grado de maniobrabilidad</i> .....	20
REDUNDANCIA .....	20
<i>Redundancia en el robot móvil omnidireccional</i> .....	21
<i>Implicaciones de la Redundancia en el robot móvil omnidireccional</i> .....	22
<b>DISCUSIÓN</b> .....	<b>23</b>
<b>DESCRIPCIÓN GENERAL</b> .....	<b>24</b>
<b>ROBOT MÓVIL OMNIDIRECCIONAL REDUNDANTE</b> .....	<b>25</b>
PLATAFORMA CENTRAL .....	25
ROBOTS MÓVILES DIFERENCIALES .....	26
DESCRIPCIÓN DE ARDUINO .....	27
DESCRIPCIÓN DE WIFI SHIELD .....	28
DESCRIPCIÓN DE LOS MOTORES Y LA TARJETA CONTROLADORA .....	29
<b>SISTEMAS EN TIEMPO REAL</b> .....	<b>32</b>
TIPOS DE SISTEMAS EN TIEMPO REAL .....	32
TIEMPO REAL EN MATLAB .....	33
<i>Real Time Windows Target</i> .....	34
<i>Conexiones físicas de dispositivos</i> .....	34
<i>Real-Time Windows Target Kernel</i> .....	35
<b>SISTEMA DE VISIÓN ARTIFICIAL</b> .....	<b>36</b>
CONTROL SERVO VISUAL .....	36
REACTIVISION .....	37
SOFTWARE DE ENLACE .....	38
<i>Marco coordinado mundo</i> .....	40
<b>MATLAB SIMULINK</b> .....	<b>41</b>
BLOQUE DE VISIÓN .....	41
BLOQUE DE REFERENCIA .....	42
BLOQUE DEL CONTROL .....	42
BLOQUE DE LECTURA DE ENCODERS .....	43
BLOQUE DE CINEMÁTICA .....	44
BLOQUE DE PROCESAMIENTO .....	44
BLOQUE DE ENVÍO DE DATOS .....	45
<b>DISCUSIÓN</b> .....	<b>46</b>
<b>SEGUIMIENTO DE POSTURA</b> .....	<b>47</b>

<b>SEGUIMIENTO DE PUNTO .....</b>	<b>47</b>
<b>CONTROL DE ROBOTS MÓVILES OMNIDIRECCIONALES .....</b>	<b>48</b>
CONTROL DINÁMICO INVERSO .....	48
CAMPOS POTENCIALES ARTIFICIALES .....	49
<i>Planeación de trayectorias</i> .....	51
<i>Implementación</i> .....	53
<b>DISCUSIÓN .....</b>	<b>55</b>
<b>CARACTERÍSTICAS DEL EXPERIMENTO .....</b>	<b>56</b>
<b>REGULACIÓN .....</b>	<b>58</b>
ERRORES PROMEDIO .....	58
<b>SEGUIMIENTO DE POSTURA .....</b>	<b>62</b>
ERRORES PROMEDIO .....	62
<b>DISCUSIÓN .....</b>	<b>69</b>
<b>CONCLUSIÓN .....</b>	<b>70</b>
<b>TRABAJO A FUTURO .....</b>	<b>71</b>
<b>ANEXO A: MATRIZ JACOBIANA.....</b>	<b>72</b>
<b>ANEXO B: MATRIZ A.....</b>	<b>73</b>
<b>ANEXO C: MATRICES DE MASAS E INCERCIAS Y DE ACELERACIONES.....</b>	<b>74</b>
<b>BIBLIOGRAFÍA .....</b>	<b>81</b>

## TABLA DE ILUSTRACIONES

---

Ilustración 1: Robot móvil omnidireccional redundante.....	8
Ilustración 2: Postura de un robot móvil.....	9
Ilustración 3: Marcos coordinados del robot móvil omnidireccional redundante .....	10
Ilustración 4: Propagación de velocidad del eslabón $i$ al eslabón $i+1$ .....	10
Ilustración 5: Medidas del robot móvil omnidireccional redundante.....	12
Ilustración 6: descripción del sistema .....	25
Ilustración 7: Robot móvil omnidireccional redundante.....	26
Ilustración 8: Arduino UNO .....	27
Ilustración 9: WiFi shield .....	28
Ilustración 10: Descripción wifi shield .....	29
Ilustración 11: Motor EMg30 .....	29
Ilustración 12: MD25 .....	30
Ilustración 13: Descripción MD25 .....	31
Ilustración 14: Marco coordinado cámara.....	37
Ilustración 15: Fiducials .....	37
Ilustración 16: bloques de visión artificial.....	38
Ilustración 17: Marco coordinado imagen .....	39
Ilustración 18: Marco coordinado mundo.....	40
Ilustración 19: Bloque completo de visión.....	42
Ilustración 20: Bloque de variables de referencia .....	42
Ilustración 21: Bloque de control.....	43
Ilustración 22: Bloque de lectura de encoders .....	43
Ilustración 23: Bloque de cinemática .....	44
Ilustración 24: Bloque de procesamiento de velocidades .....	45
Ilustración 25: Bloque de envío de datos .....	45
Ilustración 26: Campo potencial atractivo .....	50
Ilustración 27: Campo potencial repulsivo .....	50
Ilustración 28: Campos potenciales artificiales .....	51
Ilustración 29: Diagrama de las ecuaciones de control.....	53

Ilustración 30: Condición inicial de orientación .....	57
Ilustración 31: Orientación a 90 grados del fiducial .....	58
Ilustración 32: E1 Trayectoria en el eje x.....	59
Ilustración 33: E1 Trayectoria en el eje y.....	59
Ilustración 34: E1 Trayectoria en el eje z.....	59
Ilustración 35: E2 Trayectoria en el eje X.....	60
Ilustración 36: E2 Trayectoria en el eje y.....	60
Ilustración 37: E2 Trayectoria en el eje Z.....	60
Ilustración 38: E3 Trayectoria en el eje X.....	61
Ilustración 39: E3 Trayectoria en el eje Y.....	61
Ilustración 40: E3 Trayectoria en el eje Z.....	61
Ilustración 41: E1 Seguimiento de postura .....	63
Ilustración 42: E1 Trayectoria en el eje x.....	63
Ilustración 43: E1 Trayectoria en el eje y.....	64
Ilustración 44: E1 Trayectoria en el eje z.....	64
Ilustración 45: E2 Seguimiento de postura .....	65
Ilustración 46: E2 Trayectoria en el eje x.....	65
Ilustración 47: E2 Trayectoria en el eje y.....	66
Ilustración 48: E2 Trayectoria en el eje z.....	66
Ilustración 49: E3 Seguimiento de postura .....	67
Ilustración 50: E3 Trayectoria en el eje x.....	67
Ilustración 51: E3 Trayectoria en el eje y.....	68
Ilustración 52: E3 Trayectoria en el eje z.....	68
Ilustración 53: Matriz Jacobiana parte 1 .....	72
Ilustración 54: Matriz Jacobiana parte 2 .....	72
Ilustración 55: Matriz que describe las restricciones cinemáticas de las ruedas .....	73
Ilustración 56: Columna 1, matriz de masas e inercias.....	74
Ilustración 57: Columna 2, matriz de masas e inercias.....	74
Ilustración 58: Columna 3.a, matriz de masas e inercias.....	75
Ilustración 59: Columna 3.b, matriz de masas e inercias.....	75
Ilustración 60: Columna 4, matriz de masas e inercias.....	76

Ilustración 61: Columna 5, matriz de masas e inercias.....	76
Ilustración 62: Columna 6, matriz de masas e inercias.....	77
Ilustración 63: Columnas 7, 8 y 9; matriz de masas e inercias .....	77
Ilustración 64: Columnas 10:15, matriz de masas e inercias .....	78
Ilustración 65: Columnas 1, 2, 5 y 5; matriz de aceleraciones .....	78
Ilustración 66: Columna 3, matriz de aceleraciones .....	79
Ilustración 67: Columnas 6, 7 y 8; matriz de aceleraciones.....	79
Ilustración 68: Columnas 9:15, matriz de aceleraciones.....	80





---

# INTRODUCCIÓN

---

Desde los años ochenta, los robots móviles con ruedas han sido ampliamente estudiados; esto debido a su capacidad de desplazarse libremente en su ambiente, en comparación con los robots manipuladores, los cuales poseen un área fija de trabajo.

La movilidad de los robots móviles está restringida por las características de las ruedas que poseen. Por ejemplo, un robot con ruedas como las de un coche requiere realizar diversas maniobras para poder re-orientarse; y un robot diferencial, que ocupa ruedas fijas, solo necesita variar la velocidad de sus ruedas.

De los diferentes tipos de ruedas, aquellas que ofrecen mayor movilidad son las omnidireccionales, ya que permiten a un robot desplazarse en cualquier dirección permisible por el medio. Es por esta razón que desde que se empezaron a estudiar los robots con ruedas, ha habido un interés en encontrar una forma de dotarlos con la cualidad de ser omnidireccionales. En esta búsqueda se han diseñado diversos tipos de ruedas omnidireccionales como las universales, las suecas y las mecanum. También se han diseñado varios mecanismos para hacer omnidireccionales a los robots.

En el ámbito de la redundancia de los robots móviles, las investigaciones se han enfocado en los diseños particulares realizados por cada investigador. La redundancia se ha empleado para realizar diseños a prueba de fallas, para evitar singularidades o incluso para dotar de mayor flexibilidad al robot.

A continuación se muestran algunas investigaciones que se han realizado tanto en el ámbito de los robots omnidireccionales y en la redundancia.

---

## TRABAJOS PREVIOS

---

En 1987 Campion [1] demostró la omnidireccionalidad de un robot móvil constituido por tres ruedas tipo castor o tres ruedas suecas. Otros investigadores han diseñado varios mecanismos basados en el "concepto de la rueda universal", las cuales

proporcionan una combinación de movimiento restringido y no restringido al girar [2]. El diseño inicial de la rueda universal se utilizó para llevar a cabo movimientos omnidireccionales sin cambiar la dirección de las ruedas, pero este tipo de rueda sufre de choques sucesivos cuando los rodillos individuales hacen contacto con el suelo. La rueda Mecanum [3], la rueda de Sueca [4] y la rueda alternativa [5], [6], [7] se propusieron para remediar la brecha entre los rodillos. En el 2000 Wada [8] desarrolló un robot móvil con dos ruedas castoras y un actuador rotacional. Yi y Kim [9] desarrollaron el modelo cinemático del robot móvil omnidireccional con ruedas castores activas. En el 2006 Salih [10] desarrolló un robot móvil omnidireccional usando cuatro ruedas tipo mecanum. Moore y Flann [11] y Berkermeier y Ma [12] implementaron un robot móvil omnidireccional constituido de seis ruedas y un sistema especial conocido como "Sistema de inspección omnidireccional" con una rueda inteligente y una rueda castor respectivamente. Ushimi [13] desarrolló un robot móvil omnidireccional con dos ruedas tipo castor. Finalmente en el grupo MRG de la Facultad de Ingeniería de la UNAM [14] se desarrolló un nuevo tipo de robot móvil omnidireccional, el cual consiste de dos robots móviles diferenciales conectados a una plataforma triangular.

Hasta ahora los trabajos previos presentados se han enfocado en las diferentes maneras que se han propuesto para dotar de omnidireccionalidad a los robots móviles. Sin embargo, otro aspecto importante para el trabajo de investigación que se aborda en esta tesis es la redundancia; en particular para los robots móviles omnidireccionales. Es necesario recalcar que los trabajos previos en cuanto a redundancia en robots móviles omnidireccionales son escasos y se enfocan en diseños particulares de los autores.

Tadakuma [15] propuso un robot móvil que funciona mediante tres ruedas y pose redundancia cinemática en cada cadena, sin embargo el modelo matemático no fue desarrollado y el robot no era omnidireccional. En [16] se presenta un robot móvil omnidireccional con múltiples redundancias cinemáticas y varios algoritmos para aprovechar dichas redundancias. En [17] se propone un esquema de distribución de carga libre de singularidades, para un robot móvil omnidireccional de tres ruedas accionado de forma redundante.

## JUSTIFICACIÓN

---

Con base en la investigación de los trabajos previos realizados en el ámbito de los robots móviles omnidireccionales, se determinó que existen numerosas investigaciones en cuestión del desarrollo de ruedas y mecanismos especiales, sin embargo pocos trabajos se han enfocado en emplear solamente ruedas fijas para hacer que un robot sea omnidireccional.

La redundancia se ha empleado para casos particulares de cada artículo; algunos autores la han utilizado para evitar singularidades en sus diseños particulares; otros la emplean para accionar a los robots de forma redundante y realizar diseño a prueba de fallas, al asegurar que si un actuador falla queda otro que realice la tarea; y también ha sido empleada para dotar de mayor flexibilidad a los robots móviles mediante diversas cadenas cinemáticas. A pesar de una exhaustiva investigación, pocos fueron los artículos encontrados que abordaban la redundancia y la omnidireccionalidad de los robots móviles; por lo que se determinó que sigue siendo un campo de oportunidad para nuevas investigaciones.

También considerando el antecedente del grupo MRG [14] se propuso la continuación en la investigación del nuevo robot móvil omnidireccional; revisando los cálculos de la cinemática, obteniendo el modelo dinámico, controlando de forma independiente los tres grados de libertad del robot, implementando los experimentos en tiempo real y agregándole la característica de ser redundante.

## HIPÓTESIS

---

Bajo estas premisas se estableció la siguiente hipótesis: ¿Es posible obtener el modelo cinemático y dinámico de un robot móvil omnidireccional redundante, que está constituido por una plataforma y cuatro robots móviles diferenciales, y estabilizarlo alrededor de una trayectoria?

## OBJETIVOS

---

Con la finalidad de responder la hipótesis planteada, se estableció como objetivo general obtener un robot móvil omnidireccional que sea redundante; que esté formado por cuatro robots móviles diferenciales unidos a una plataforma rectangular; y que sea capaz de realizar tareas en tiempo real.

Para alcanzar este objetivo general se establecieron los siguientes objetivos particulares:

- Obtener el modelo cinemático del robot móvil omnidireccional redundante.
- Obtener el modelo dinámico del robot móvil omnidireccional redundante.
- Instrumentar cada robot diferencial para que sea capaz de recibir instrucciones de control desde un procesador central.
- Realizar la medición de la orientación de cada robot móvil diferencial con respecto a la plataforma rectangular.
- Realizar los algoritmos para leer la orientación y posición del robot móvil omnidireccional redundante con respecto a un sistema coordenado fijo.
- Realizar los algoritmos de envío de datos hacia el robot móvil omnidireccional redundante.
- Realizar los algoritmos para la recepción de los datos enviados por el robot móvil omnidireccional redundante.
- Realizar los algoritmos de control para el robot móvil omnidireccional redundante.
- Implementar la tarea de regulación.
- Implementar la tarea de seguimiento de trayectorias.

## APORTACIONES

---

Las aportaciones de este trabajo de investigación son: el empleo de ruedas fijas para constituir un robot móvil omnidireccional; el modelo cinemático y dinámico de un nuevo tipo de robot móvil omnidireccional; que tiene la característica de ser redundante; un robot capaz de realizar tareas en tiempo real; que funciona de manera modular, de tal

forma que es posible implementarle una gama diversa de controladores con solo alterar un documento de texto.

Estas aportaciones serán analizadas y desarrolladas en los capítulos posteriores.

---

# ROBOT MÓVIL OMNIDIRECCIONAL REDUNDANTE

---

---

## INTRODUCCIÓN

---

La robótica móvil es una disciplina científica que aborda la investigación y desarrollo de una clase particular de sistemas mecatrónicos: los robots; estudia aquellos robots que pueden desplazarse dentro de su medio, ya sea en el agua, el aire o sobre la superficie terrestre. Los robots móviles se pueden definir como una plataforma mecánica dotada de un sistema de locomoción capaz de navegar a través de un determinado ambiente de trabajo, dotado de cierto nivel de autonomía para su desplazamiento portando cargas.

Los robots móviles se clasifican de acuerdo al tipo de locomoción que emplean:

- Ruedas. Se utilizan sobre superficies lisas.
- Orugas. Se emplean sobre superficies irregulares
- Patas. Se usan para superficies muy irregulares

---

## ROBOTS MÓVILES CON RUEDAS

---

La rueda ha sido el mecanismo de locomoción más utilizado en la robótica móvil. Un móvil con ruedas es un vehículo capaz de moverse sobre una superficie mediante la acción de ruedas montadas sobre el robot. Al emplear ruedas se asumen como ciertas las siguientes condiciones:

- El robot móvil se mueve sobre una superficie plana horizontal, es decir, la energía potencial es constante.
- Los ejes de guiado son perpendiculares al suelo.
- No existen elementos flexibles en la estructura del robot, incluyendo las ruedas.
- El contacto entre cada rueda y el suelo se reduce a un solo punto.

- No existe deslizamiento.

---

## TIPOS DE RUEDAS

---

Existen dos clases básicas de ruedas: las ruedas convencionales y las ruedas suecas. Para las ruedas convencionales, el contacto entre la rueda y el suelo satisface las condiciones de no deslizamiento y rotación pura a lo largo del eje de movimiento. Por lo tanto la velocidad de contacto es cero e implica que las componentes tanto paralela como ortogonal al plano de la rueda son igual a cero. A su vez las ruedas convencionales se dividen en: rueda fija, rueda direccionable y rueda castor

Para la rueda sueca, solo un componente de la velocidad de contacto con el suelo es cero. La dirección del vector no cero de velocidad es conocida de antemano y es fija con respecto a la orientación de la rueda.

---

## ROBOT MÓVIL DIFERENCIAL

---

Son aquellos que no tienen ruedas direccionables y poseen una o varias ruedas fijas, las cuales deben de encontrarse centradas sobre un mismo eje. Su movimiento está restringido por la velocidad a la que giran las ruedas [18].

---

## ROBOTS MÓVILES OMNIDIRECCIONALES

---

En [18] se describe a los robots omnidireccionales como aquellos que no tienen ruedas fijas o direccionables, sino que poseen solo ruedas tipo castor o suecas. Tienen movilidad completa en el plano, lo que significa que pueden moverse en cualquier dirección en cualquier instante de tiempo sin necesidad de alguna reorientación. Para este trabajo solo se toma la segunda parte de la definición; un robot móvil omnidireccional es aquel que tiene movilidad completa en el plano, por lo tanto puede desplazarse en cualquier dirección en cualquier instante de tiempo sin la necesidad de reorientarse.

---

## ROBOT MÓVIL OMNIDIRECCIONAL REDUNDANTE

---

En comparación a la definición convencional de un robot móvil omnidireccional [1], el robot que se describirá a continuación emplea únicamente ruedas fijas para obtener la omnidireccionalidad. Este robot está conformado por cuatro robots móviles diferenciales unidos a una plataforma rectangular (ilustración 1). En las siguientes imágenes se muestran las medidas indispensables para obtener el modelo cinemático del robot.

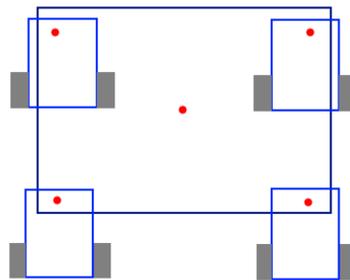


ILUSTRACIÓN 1: ROBOT MÓVIL OMNIDIRECCIONAL REDUNDANTE

---

## MODELADO CINEMÁTICO

---

Cada rueda individual contribuye al movimiento del robot, y al mismo tiempo, le impone restricciones en su movimiento. Las ruedas están unidas entre sí con base en la geometría del chasis del robot, por lo tanto sus restricciones individuales se combinan para formar las restricciones sobre el movimiento global del robot.

Para determinar el modelo cinemático de un robot móvil, las restricciones se expresan con base en los movimientos de las ruedas, debido a que los movimientos individuales de las ruedas influyen en el movimiento global del robot.

Las fuerzas y restricciones de cada rueda deben ser expresadas con respecto a un marco de referencia global que se encuentre sobre el robot, para ello a parte del marco principal del robot se seleccionan diversos marcos coordenados que permitan propagar la velocidad del marco de referencia global hacia las ruedas.

## POSTURA DEL ROBOT

Para describir la posición de un robot móvil con ruedas en el plano, se establece de manera arbitraria un marco coordenado O (marco coordenado mundo) en el plano del movimiento; mientras que se establece un segundo marco coordenado P sobre el robot. Por lo tanto la postura del robot se puede describir en términos de las dos coordenadas  $(x,y)$  del origen P y de la orientación  $\theta_p$  con respecto al marco coordenado O.

$$(2.1) \quad \xi = [x \quad y \quad \theta_p]^T$$

Mientras que la orientación del marco coordenado P con respecto al marco coordenado O puede determinarse por la matriz de rotación.

$$(2.2) \quad R(\theta_p) = \begin{pmatrix} \cos \theta_p & \sin \theta_p & 0 \\ -\sin \theta_p & \cos \theta_p & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

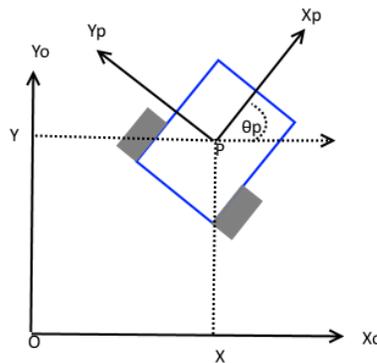


ILUSTRACIÓN 2: POSTURA DE UN ROBOT MÓVIL

Con la finalidad de obtener el modelo cinemático del robot es necesario describir su postura en términos de un marco coordenado mundo; también se requiere describir las velocidades a las cual se ve sometido el robot en el punto P con respecto a las velocidades de sus ruedas. Para ello colocan diversos marcos coordenados sobre el robot (ilustración 3). Se coloca un marco coordenado en cada rueda, en los puntos de

unión de los robots móviles diferenciales con la plataforma y un marco sobre el punto P. Las ecuaciones 2.1 y 2.2 siguen teniendo validez para este tipo de robot.

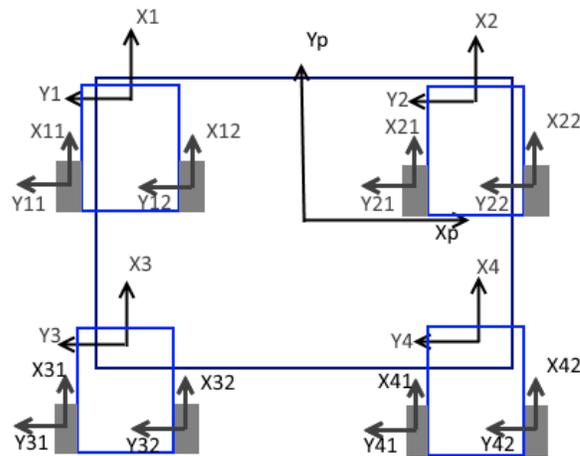


ILUSTRACIÓN 3: MARCOS COORDENADOS DEL ROBOT MÓVIL OMNIDIRECCIONAL REDUNDANTE

### PROPAGACIÓN DE VELOCIDADES

[19]Este método se emplea para propagar velocidades del eslabón  $i$  al eslabón  $i + 1$  (ilustración 4)

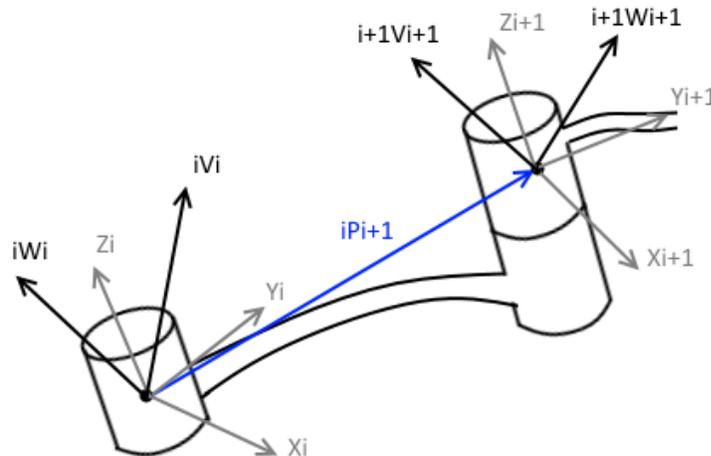


ILUSTRACIÓN 4: PROPAGACIÓN DE VELOCIDAD DEL ESLABÓN  $i$  AL ESLABÓN  $i+1$

Lo que este método sugiere es que la velocidad angular del eslabón  $i + 1$  relativa a su propio sistema de referencia, es igual a la velocidad angular del eslabón  $i$  con respecto al sistema  $i + 1$  más una componente propia del sistema  $i + 1$ .

$$(2.3) \quad {}^{i+1}w = {}_i^{i+1}R \times {}_i^i w + {}_{i+1}^{i+1} \dot{\theta}$$

Donde  ${}_i^{i+1}R$  es la matriz de rotación del sistema  $i$  al sistema  $i+1$ . Por lo tanto al multiplicar la velocidad angular de  $i$  con respecto al sistema  $i$  por la matriz de rotación, se obtiene la velocidad angular de  $i$  con respecto a  $i+1$ .

Por otro lado la velocidad lineal del eslabón  $i+1$  con respecto a su propio sistema de referencia es igual a la velocidad lineal del eslabón  $i$  más la velocidad producida por el brazo de palanca  ${}_i^{i+1}P$  y la velocidad angular de  $i$ . Todo esto multiplicado por la matriz de rotación del sistema  $i$  al sistema  $i+1$ .

$$(2.4) \quad {}^{i+1}v = {}_i^{i+1}R \times ({}_i^i v + {}_i^i w \times {}_i^{i+1}P)$$

Para el robot móvil omnidireccional redundante el punto de interés es donde se encuentra el marco coordenado P. Por lo tanto se emplea el método de propagación de velocidades de dicho punto hasta las ruedas, empleando las ecuaciones 2.3 y 2.4; de esta manera se obtiene la cinemática inversa para el robot.

Para obtener la cinemática inversa del robot móvil omnidireccional redundante se debe propagar las velocidades del marco coordenado P, mediante las ecuaciones 2.3 y 2.4, a los marcos coordenados 1, 2, 3 y 4 (ilustración 3). El procedimiento para la propagación de las velocidades del marco coordenado P es el mismo para los cuatro marcos coordenados; por lo tanto solo se mostrará el procedimiento para propagar la velocidad del marco coordenado P al marco coordenado 1.

---

### PROPAGACIÓN DE VELOCIDAD: MARCO COORDENADO P AL MARCO COORDENADO 1

---

Sea la velocidad angular del sistema P con respecto a si mismo representada por:

$$(2.5) \quad {}_P^P w = [0 \quad 0 \quad \dot{\theta}_p]^T$$

La velocidad lineal se representa por el siguiente vector:

$$(2.6) \quad {}_P^P v = [\dot{x}_1 \quad \dot{y}_1 \quad 0]^T$$

La matriz de rotación del marco coordenado P a 1 se representa como:

$$(2.7) \quad {}^1_P R(\theta_p) = \begin{pmatrix} \cos\left(\theta_1 + \frac{\pi}{2}\right) & \sin\left(\theta_1 + \frac{\pi}{2}\right) & 0 \\ -\sin\left(\theta_1 + \frac{\pi}{2}\right) & \cos\left(\theta_1 + \frac{\pi}{2}\right) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

La velocidad angular del marco coordenado 1 está representada por la ecuación:

$$(2.8) \quad {}^1\dot{\theta} = [0 \quad 0 \quad \dot{\theta}_1]^T$$

Por medio de la ilustración 5 se observa que el vector de posición del punto 1 con respecto al punto P está dado por:

$$(2.9) \quad {}^P_1 P = [-a \quad b \quad 0]^T$$

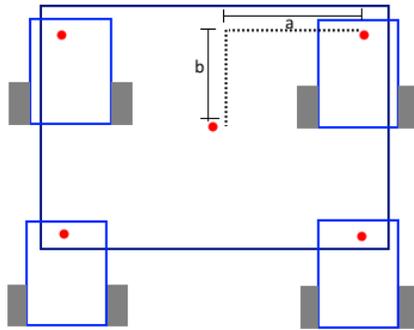


ILUSTRACIÓN 5: MEDIDAS DEL ROBOT MÓVIL OMNIDIRECCIONAL REDUNDANTE

Sustituyendo  ${}^P_w, {}^1_P R, {}^1\dot{\theta}$  en la ecuación 2.3, se obtiene:

$$(2.10) \quad {}^1w = [0 \quad 0 \quad \dot{\theta}_1 + \dot{\theta}_p]^T$$

Donde  ${}^1w$  es la velocidad angular del robot 1 con respecto a si mismo. Para obtener las ecuaciones de la velocidad lineal se sustituyen las ecuaciones 2.5, 2.6, 2.7 y 2.9 en la ecuación 2.4 y se obtiene:

$$(2.11) \quad {}^1v = \begin{bmatrix} \dot{y}_1 \cos(\theta_1) - \dot{x}_1 \sin(\theta_1) - a\dot{\theta}_p \cos(\theta_1) + b\dot{\theta}_p \sin(\theta_1) \\ a\dot{\theta}_p \sin(\theta_1) - \dot{y}_1 \sin(\theta_1) - \dot{x}_1 \cos(\theta_1) + b\dot{\theta}_p \cos(\theta_1) \\ 0 \end{bmatrix}$$

Una vez obtenidas la velocidad lineal y angular del robot 1 con respecto a si mismo, se continua con la propagación de dichas velocidades a las ruedas del robot 1.

---

### PROPAGACIÓN DE VELOCIDAD: MARCO COORDENADO 1 AL 11

---

Se define la matriz de rotación del sistema 1 al 11 cómo:

$$(2.12) \quad {}^1_1R(\theta_p) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

El vector de velocidad angular propio de la rueda 1 es:

$$(2.13) \quad {}^{11}\theta = [0 \quad \dot{\phi}_{11} \quad 0]^T$$

El vector de posición del marco 11 con respecto al marco 1 es define con apoyo de la ilustración 6, como:

$$(2.14) \quad {}^1_1P = [-c \quad d \quad 0]^T$$

Sustituyendo las ecuaciones 2.12, 2.13, 2.14, 2.10 y 2.11 en las ecuaciones 2.3 y 2.4 se obtienen la velocidad angular y lineal para la rueda 1.

$$(2.15) \quad {}^{11}w = [0 \quad \dot{\phi}_{11} \quad \dot{\theta}_1 + \dot{\theta}_p]^T$$

$$(2.16) \quad {}^{11}v = \begin{bmatrix} \dot{y}_1 \cos(\theta_1) - \dot{x}_1 \sin(\theta_1) - a\dot{\theta}_p \cos(\theta_1) + b\dot{\theta}_p \sin(\theta_1) - d\dot{\theta}_p - d\dot{\theta}_1 \\ a\dot{\theta}_p \sin(\theta_1) - \dot{y}_1 \sin(\theta_1) - \dot{x}_1 \cos(\theta_1) + b\dot{\theta}_p \cos(\theta_1) - c\dot{\theta}_p - c\dot{\theta}_1 \\ 0 \end{bmatrix}$$

El mismo procedimiento se aplica para los marcos coordenados de las ruedas: 12, 21, 22, 31, 32, 41 y 42 (Ilustración3).

---

### RESTRICCIONES CINEMÁTICAS

---

Para el cálculo de las restricciones se considera que no hay deslizamiento; que no hay desplazamiento lateral, en el eje (y); y que la rueda solo se mueve en la dirección del movimiento, la cual en este caso es sobre el eje (x) (ilustración 3). Esto conlleva a que las velocidades de la ecuación 2.16 sean igual al siguiente vector:

$$(2.17) \quad {}^{11}v = [r\dot{\phi}_{11} \quad 0 \quad 0]^T$$

Donde  $r$  es el radio de la rueda y  $\dot{\phi}_{11}$  es la velocidad angular de la rueda del marco coordinado 11. El término  $r\dot{\phi}_{11}$  se debe a que la velocidad lineal de un punto en una circunferencia, está dada por la velocidad angular de la rueda multiplicada por el radio de giro. En este caso esta velocidad actúa en el eje (x).

Al igualar la ecuación 2.17 con la ecuación 2.16 se obtiene la restricción cinemática de la rueda con el marco coordinado 11, la cual se expresa mediante las ecuaciones:

$$(2.18) \quad \begin{bmatrix} -\sin(\theta_1) & \cos(\theta_1) & -a\cos(\theta_1) + b\sin(\theta_1) - d & -d \end{bmatrix} \xi_{11} - r\dot{\phi}_{11} = 0$$

$$(2.19) \quad \begin{bmatrix} -\cos(\theta_1) & -\sin(\theta_1) & a\sin(\theta_1) + b\cos(\theta_1) - c & -c \end{bmatrix} \xi_{11} = 0$$

La ecuación 2.18 representa la restricción sobre el plano de giro  $(x_{11}, z_{11})$  de la rueda, suponiendo rotación pura sin deslizamiento. La ecuación 2.19 representa la restricción ortogonal al plano de giro de la rueda  $(y_{11}, z_{11})$  suponiendo que no hay deslizamiento lateral. El vector  $\xi_{11}$  está determinado por la ecuación 2.20.

$$(2.20) \quad \xi_{11} = \begin{bmatrix} \dot{x}_1 & \dot{y}_1 & \dot{\theta}_p & \dot{\theta}_1 \end{bmatrix}^T$$

Se emplea el mismo procedimiento de manera recursiva para las demás ruedas y se obtienen las siguientes restricciones cinemáticas.

---

### RESTRICCIÓN SOBRE EL PLANO DE GIRO $(x_{11}, z_{11})$ DE LA RUEDA

---

$$(2.21) \quad \begin{bmatrix} -\sin(\theta_1) & \cos(\theta_1) & -a\cos(\theta_1) + b\sin(\theta_1) + d & d \end{bmatrix} \xi_{12} - r\dot{\phi}_{12} = 0$$

$$(2.22) \quad \begin{bmatrix} -\sin(\theta_1) & \cos(\theta_1) & a\cos(\theta_1) + b\sin(\theta_1) - d & -d \end{bmatrix} \xi_{21} - r\dot{\phi}_{21} = 0$$

$$(2.23) \quad \begin{bmatrix} -\sin(\theta_1) & \cos(\theta_1) & a\cos(\theta_1) + b\sin(\theta_1) + d & d \end{bmatrix} \xi_{22} - r\dot{\phi}_{22} = 0$$

$$(2.24) \quad \begin{bmatrix} -\sin(\theta_1) & \cos(\theta_1) & -a\cos(\theta_1) - b\sin(\theta_1) - d & -d \end{bmatrix} \xi_{31} - r\dot{\phi}_{31} = 0$$

$$(2.25) \quad \begin{bmatrix} -\sin(\theta_1) & \cos(\theta_1) & -a\cos(\theta_1) - b\sin(\theta_1) + d & d \end{bmatrix} \xi_{32} - r\dot{\phi}_{32} = 0$$

$$(2.26) \quad \begin{bmatrix} -\sin(\theta_1) & \cos(\theta_1) & a\cos(\theta_1) - b\sin(\theta_1) - d & -d \end{bmatrix} \xi_{41} - r\dot{\phi}_{41} = 0$$

$$(2.27) \quad \begin{bmatrix} -\sin(\theta_1) & \cos(\theta_1) & a\cos(\theta_1) - b\sin(\theta_1) + d & d \end{bmatrix} \xi_{42} - r\dot{\phi}_{42} = 0$$

---

RESTRICCIÓN ORTOGONAL AL PLANO DE GIRO DE LA RUEDA ( $y_{11}, z_{11}$ )

---

$$(2.28) \quad \begin{bmatrix} -\cos(\theta_1) & -\sin(\theta_1) & a\sin(\theta_1) + b\cos(\theta_1) - c & -c \end{bmatrix} \xi_{12} = 0$$

$$(2.29) \quad \begin{bmatrix} -\cos(\theta_1) & -\sin(\theta_1) & -a\sin(\theta_1) + b\cos(\theta_1) - c & -c \end{bmatrix} \xi_{21} = 0$$

$$(2.30) \quad \begin{bmatrix} -\cos(\theta_1) & -\sin(\theta_1) & -a\sin(\theta_1) + b\cos(\theta_1) - c & -c \end{bmatrix} \xi_{22} = 0$$

$$(2.31) \quad \begin{bmatrix} -\cos(\theta_1) & -\sin(\theta_1) & a\sin(\theta_1) - b\cos(\theta_1) - c & -c \end{bmatrix} \xi_{31} = 0$$

$$(2.32) \quad \begin{bmatrix} -\cos(\theta_1) & -\sin(\theta_1) & a\sin(\theta_1) - b\cos(\theta_1) - c & -c \end{bmatrix} \xi_{32} = 0$$

$$(2.33) \quad \begin{bmatrix} -\cos(\theta_1) & -\sin(\theta_1) & -a\sin(\theta_1) - b\cos(\theta_1) - c & -c \end{bmatrix} \xi_{41} = 0$$

$$(2.34) \quad \begin{bmatrix} -\cos(\theta_1) & -\sin(\theta_1) & -a\sin(\theta_1) - b\cos(\theta_1) - c & -c \end{bmatrix} \xi_{42} = 0$$

CINEMÁTICA DIFERENCIAL INVERSA

---

Una vez encontradas las ecuaciones que definen las restricciones sobre las ruedas, estas pueden organizarse de la siguiente manera:

$$(2.35) \quad A\dot{q}_1 = 0$$

Dónde  $A$  es una matriz de  $16 \times 15$  que representa las restricciones cinemáticas de las ruedas y  $\dot{q}_1$  un vector columna de velocidades que describen completamente al robot (ecuación 2.35).

$$(2.36) \quad \dot{q}_1 = \begin{bmatrix} \dot{x}_1 & \dot{y}_1 & \dot{\theta}_p & \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_3 & \dot{\theta}_4 & \dot{\phi}_{11} & \dot{\phi}_{12} & \dot{\phi}_{21} & \dot{\phi}_{22} & \dot{\phi}_{31} & \dot{\phi}_{32} & \dot{\phi}_{41} & \dot{\phi}_{42} \end{bmatrix}^T$$

Dónde  $\begin{bmatrix} \dot{x}_1 & \dot{y}_1 & \dot{\theta}_p \end{bmatrix}$  son las velocidades lineales y angulares del punto P, (el robot) con respecto a sí mismo; las variables  $\begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_3 & \dot{\theta}_4 \end{bmatrix}$  representan las velocidades angulares de los robots móviles diferenciales con respecto a la plataforma; y las variables representadas por  $\dot{\phi}$  hacen referencia a las velocidades angulares de las ruedas.

Una vez re-expresadas las ecuaciones de restricciones, se resuelve el sistema de ecuaciones  $A\dot{q}_1 = 0$  con respecto a las variables de velocidad del robot  $[\dot{x}_1 \ \dot{y}_1 \ \dot{\theta}_p]$ .

El resultado de resolver el sistema de ecuaciones se expresa como:

$$(2.37) \quad \dot{q}_1 = S_{q1}u$$

Donde  $S_{q1}$  es una matriz jacobiana de  $15 \times 3$  y  $u$  es el vector de velocidades de entrada.

$$(2.38) \quad u = [\dot{x}_1 \ \dot{y}_1 \ \dot{\theta}_p]^T$$

El significado de la ecuación 2.37, es que se obtiene una ecuación que expresa las velocidades de las ruedas  $\dot{\phi}$  con respecto a las velocidades lineales y angulares del robot en el punto P  $[\dot{x}_1 \ \dot{y}_1 \ \dot{\theta}_p]$ . Esto por esto que esta ecuación representa la cinemática diferencial inversa.

El vector de velocidades de entrada  $u$  está expresado con respecto al robot; por lo tanto es necesario transformar la ecuación 2.37 para que quede re-expresada en términos de las velocidades del robot con respecto al sistema coordenado mundo. Esto se debe a que es de interés tener un marco coordenado mundo a partir del cual el robot se desplace y realice las tareas para las cuales se ha diseñado. Para lograrlo se multiplica  $S_{q1}$  de la ecuación 2.37 por una matriz de rotación similar a la ecuación 2.2.

$R(\theta_p)$  es una matriz de  $15 \times 15$

$$(2.39) \quad R(\theta_p) = \begin{bmatrix} \cos \theta_p & -\sin \theta_p & 0 & \dots & 0 \\ \sin \theta_p & \cos \theta_p & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$(2.40) \quad S_q = R(\theta_p)S_{q1}$$

Al emplear la ecuación 2.40, se obtiene:

$$(2.41) \quad \dot{q} = S_q u$$

Donde  $\dot{q}$  queda expresado como:

$$(2.42) \quad \dot{q} = [\dot{x} \ \dot{y} \ \dot{\theta}_p \ \dot{\theta}_1 \ \dot{\theta}_2 \ \dot{\theta}_3 \ \dot{\theta}_4 \ \dot{\phi}_{11} \ \dot{\phi}_{12} \ \dot{\phi}_{21} \ \dot{\phi}_{22} \ \dot{\phi}_{31} \ \dot{\phi}_{32} \ \dot{\phi}_{41} \ \dot{\phi}_{42}]^T$$

La diferencia entre la ecuación 2.42 y 2.36 es que los dos primeros términos, los cuales se refieren a la velocidad lineal del robot, se modifican por medio de la matriz de rotación (ecuación 2.39). Por lo tanto en la ecuación 2.41 las velocidades lineales del robot ya no están expresadas con respecto a si mismo, sino con respecto al marco de referencia global (marco coordenado mundo). En el Anexo A se encuentra la matriz  $S_q$ .

## MODELO DINÁMICO

La dinámica se encarga de estudiar las fuerzas que producen el movimiento de los cuerpos. Por lo tanto, el modelo dinámico de un robot tiene como objetivo conocer las ecuaciones que establecen la relación entre el movimiento del robot y las fuerzas implicadas en el mismo. Usando el método de Euler-Lagrange es posible determinar estas ecuaciones de movimiento por medio de un análisis de energía del robot móvil. Para realizar el análisis se parte de la ecuación conocida como Lagrangiano.

$$(2.43) \quad L = K - U$$

Dónde  $K$  es la energía cinética del sistema y  $U$  la energía potencial.

$$(2.44) \quad K_i = \frac{1}{2} m_i \left( {}^i v + {}^i w \times P_{ci}^i \right)^T \left( {}^i v + {}^i w \times P_{ci}^i \right) + \frac{1}{2} w_1^T I_i^{ci} w_1$$

Dónde  $m_i$  es la masa;  ${}^i v$  es la velocidad lineal;  ${}^i w$  es la velocidad angular;  $P_{ci}^i$  es un vector posición del centro de inercia; e  $I_i^{ci}$  es la matriz de inercia. El cálculo de la energía cinética se hace para los marcos coordenados de interés; es decir, para el punto P del robot móvil y para los marcos de las ruedas. La energía potencial no se calculó debido a que cuando se deriva su valor es cero; esto se debe a que se asume que el robot se mueve sobre una superficie horizontal, por lo tanto la energía potencial es constante.

Una vez calculado el Lagrangiano se procede a aplicar la ecuación conocida como Euler-Lagrange.

$$(2.45) \quad \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} \right) - \left( \frac{\partial L}{\partial q} \right) = \tau$$

Siendo  $\tau$  el vector del par motor. Desarrollando la ecuación 2.45 se obtiene parte del modelo dinámico del robot móvil, el cual queda expresado de la siguiente forma:

$$(2.46) \quad M_q \ddot{q} + C_q = \tau_l$$

$$(2.47) \quad \tau_l = \left[ \tau_x \quad \tau_y \quad \tau_{\theta_p} \quad \tau_{\theta_1} \quad \tau_{\theta_2} \quad \tau_{\theta_3} \quad \tau_{\theta_4} \quad \tau_{\phi_{11}} \quad \tau_{\phi_{12}} \quad \tau_{\phi_{21}} \quad \tau_{\phi_{22}} \quad \tau_{\phi_{31}} \quad \tau_{\phi_{32}} \quad \tau_{\phi_{41}} \quad \tau_{\phi_{41}} \right]^T$$

Dónde  $M_q$  es la matriz de masas e inercias y  $C_q$  representa las fuerzas centrífugas y de Coriolis. Sin embargo, como ya se mencionó la ecuación permanece incompleta; esto se debe a que falta agregar un término que represente las reacciones que se ven ejercidas en las ruedas debido a su contacto con el suelo. También se debe especificar cuáles son los pares de control del sistema; es decir, cuales pares se emplearan para mover al robot móvil. Al agregar estos términos extras a la ecuación 2.46 queda:

$$(2.48) \quad M_q \ddot{q} + C_q = \tau_e + A^T \lambda$$

$$(2.49) \quad \tau_e = \left[ 0 \quad \tau_{\phi_{11}} \quad \tau_{\phi_{12}} \quad \tau_{\phi_{21}} \quad \tau_{\phi_{22}} \quad \tau_{\phi_{31}} \quad \tau_{\phi_{32}} \quad \tau_{\phi_{41}} \quad \tau_{\phi_{41}} \right]^T$$

Dónde  $A$  se refiere a la matriz de la ecuación 2.35 y  $\lambda$  es una matriz de multiplicadores de Lagrange, que expresan las reacciones sobre las ruedas del robot móvil. Revisar Anexo B y C.

## OMNIDIRECCIONALIDAD Y REDUNDANCIA

---

El robot móvil que se presenta en esta tesis tiene entre sus características el ser un robot omnidireccional que a su vez es redundante. A continuación se describen estos rasgos distintivos.

### OMNIDIRECCIONALIDAD

---

Un robot móvil omnidireccional es aquel que puede moverse en cualquier dirección permisible por el medio. Al ser un robot con ruedas solo puede moverse en el plano, por lo tanto tiene tres movimientos posibles: en dirección (x), en dirección (y), y rotando alrededor del eje (z). Al analizar la ecuación 2.38 se observa que el robot que se presenta tiene tres velocidades de entrada, las cuales corresponden a las indicadas

para un movimiento completo en el plano. Es por esta razón que el robot móvil presentado en este escrito es omnidireccional.

Otra forma de determinar que el robot es omnidireccional es analizando el tipo de robot que es; de acuerdo a los grados de movilidad, direccionalidad y maniobrabilidad que posea [18]. Estos grados están definidos por la matriz  $A$  de la ecuación 2.35.

---

### GRADO DE MOVILIDAD

---

El grado de movilidad corresponde a los grados de libertad instantáneos que posee un robot y se obtiene a partir de la siguiente ecuación:

$$(2.50) \quad r_m = \dim(\xi) - \text{rank}(o_{f_c})$$

Donde  $\dim(\xi)$  corresponde a la dimensión de las coordenadas generalizadas que definen la postura de la plataforma móvil (ecuación 2.51), y  $\text{rank}(o_{f_c})$  corresponde al rango de la submatriz  $o_{f_c}$  perteneciente a la matriz  $A$ . Dicha submatriz corresponde a la sección de las restricciones cinemáticas ortogonales al plano de la rueda comprendida por las ruedas fijas y las ruedas centradas direccionables.

$$(2.51) \quad \xi = [x \quad y \quad \theta_p \quad \theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4]^T$$

$$(2.52) \quad o_{f_c} = \begin{pmatrix} A_{11} & \dots & A_{17} \\ \vdots & \ddots & \vdots \\ A_{81} & \dots & A_{87} \end{pmatrix}$$

De las ecuaciones 2.51 y 2.52;  $\dim(\xi) = 7$  y  $\text{rank}(o_{f_c}) = 4$ . Por lo tanto  $r_m = 3$

---

### GRADO DE DIRECCIONALIDAD

---

El grado de direccionalidad define el número de elementos que cambian la dirección del vehículo y se obtiene a partir de la siguiente ecuación:

$$(2.53) \quad r_s = \text{rank}(o_c)$$

Donde  $\text{rank}(o_c)$  corresponde al rango de la submatriz  $o_c$  perteneciente a la matriz  $A$  definida por los elementos de las ruedas direccionables de las restricciones

cinemáticas ortogonales al plano de las ruedas. En este caso dado que no hay ninguna rueda direccionable el rango es 0, por lo que  $r_s = 0$

---

### GRADO DE MANIOBRABILIDAD

---

El grado de maniobrabilidad corresponde a los grados de libertad totales del robot y se define por la ecuación:

$$(2.54) \quad r_M = r_m + r_s$$

Por lo que se obtiene que  $r_M = 3$

Esto significa que la plataforma móvil tiene tres grados de libertad. Utilizando la clasificación de Campion [1] el tipo de robot se expresa por la forma  $(r_m, r_s)$ , por lo tanto el robot móvil corresponde al tipo  $(3,0)$ ; esto comprueba que el robot es omnidireccional en el plano.

Otro aspecto a considerar es que en [18] se menciona que los robots móviles omnidireccionales no poseen ruedas fijas; en este caso el robot que se presenta está desarrollado únicamente con ruedas fijas y sin la necesidad de emplear algún mecanismo especial.

---

### REDUNDANCIA

---

La redundancia en un robot se refiere, a que este tiene más grados de libertad de los que requiere para una movilidad completa en el medio en el que se desenvuelve; es decir, un robot es redundante si tiene más grados de libertad de los que requiere para desempeñar una tarea. Por ejemplo, un manipulador serial se considera redundante cuando este posee más de seis actuadores (tres de posición y tres de orientación); debido a que un objeto en el espacio, con completa movilidad, se describe a partir de su desplazamiento con respecto a los ejes  $(x,y,z)$  y su orientación con respecto a los mismos ejes.

A continuación se muestra una tabla con algunas tareas y los grados de libertad necesarios para realizarlas.

TABLA 1: RELACIÓN ENTRE GRADOS DE LIBERTAD Y TAREA DESEMPEÑADA

<b>Tarea</b>	<b>Grados de libertad requeridos</b>
Posicionamiento en el plano	2
Posicionamiento en el espacio	3
Orientación en el plano	1
Posicionamiento y orientación en el plano	3
Posicionamiento y orientación en el espacio	6

Analizando la Tabla 1 se observa que si la tarea de un robot móvil con ruedas es posicionarse y orientarse en el plano, entonces cualquier robot con un grado de maniobrabilidad igual a tres puede llegar a ser redundante.

Entre las diversas consecuencias que conllevan el uso de robots redundantes se encuentran el aumento del número de actuadores y sensores; el aumento del costo del sistema; y principalmente una mayor complejidad al momento de realizar el análisis cinemático, junto con la necesidad de emplear algoritmos especiales de coordinación.

---

### REDUNDANCIA EN EL ROBOT MÓVIL OMNIDIRECCIONAL

---

En párrafos anteriores se mencionó que la condición requerida para que un robot se considere redundante es que posea más grados de libertad de los necesarios para realizar una tarea específica; esta condición prevalece en su mayoría para robots manipuladores; ya que en [18] se menciona que el valor máximo que alcanza el grado de maniobrabilidad de un robot móvil es de tres. Por lo tanto un robot móvil no puede poseer más grados de libertad que los requeridos para moverse en el plano; esto se ve de manifiesto en la ecuación 2.54, cuyo resultado es tres para el robot móvil presentado en este escrito y para el robot móvil presentado en [14]. De hecho el vector  $u$  (ecuación 2.38) es el mismo para ambos robots; y sus grados de direccionalidad y movilidad son los mismos. Pero una de las diferencias principales entre ambos robots es que el que aquí se presenta posee cuatro robots móviles diferenciales en lugar de dos; lo cual significa que posee más actuadores (considerando a cada robot móvil diferencial como un actuador).

En [19] se establece que el número de grados de libertad de un mecanismo es el número de variables de posición independientes que tienen que ser especificadas para

poder localizar todas las partes del mecanismo; es decir, los grados de libertad de un robot manipulador están determinados por el número de variables articulares independientes requeridas para posicionar el efector final. Aplicando el mismo concepto al robot móvil omnidireccional presentado; y considerando que posee dos actuadores más que su antecesor [14] se concluye que es redundante.

---

## IMPLICACIONES DE LA REDUNDANCIA EN EL ROBOT MÓVIL OMNIDIRECCIONAL

---

Como ya se mencionó, uno de los aspectos negativos de un robot redundante es la complejidad al momento de calcular la cinemática; debido a la necesidad de decidir cómo será el movimiento de los actuadores del robot para que desempeñen la tarea deseada. En el caso particular del robot móvil omnidireccional redundante, el mismo método empleado para calcular la cinemática de su antecesor [14] es el empleado en esta tesis; por lo tanto la redundancia de este robot no presenta complicaciones, ya que el mismo procedimiento empleado obtiene las ecuaciones de coordinación tanto de un robot no redundante, como de uno redundante (las cuales son las ecuaciones de la cinemática inversa diferencial).

El efecto de la redundancia en el robot móvil se puede observar mediante la comparación de los valores de las ecuaciones 2.47 y 2.42, con los valores de las mismas ecuaciones para el robot antecesor (ecuaciones 2.55 y 2.56).

$$(2.55) \quad \dot{q} = [\dot{x} \quad \dot{y} \quad \dot{\theta}_p \quad \dot{\theta}_1 \quad \dot{\theta}_2 \quad \dot{\phi}_{11} \quad \dot{\phi}_{12} \quad \dot{\phi}_{21} \quad \dot{\phi}_{22}]^T$$

$$(2.56) \quad \tau_l = [\tau_x \quad \tau_y \quad \tau_{\theta_p} \quad \tau_{\theta_1} \quad \tau_{\theta_2} \quad \tau_{\phi_{11}} \quad \tau_{\phi_{12}} \quad \tau_{\phi_{21}} \quad \tau_{\phi_{22}}]^T$$

Comparando los pares de ecuaciones, se observa que la única diferencia son los términos extra que aparecen. Debido a que los términos que aparecen en ambos pares de ecuaciones son exactamente iguales; es decir  $\dot{\theta}_1$  de 2.55 es igual a  $\dot{\theta}_1$  de 2.42, y lo mismo para los demás términos iguales.

De lo anterior se puede concluir que la única implicación de la redundancia en la cinemática y dinámica del robot es la aparición de los términos extras que representan a los robots móviles diferenciales extra.

Sin embargo cabe mencionar que si se hiciera un análisis estático de las fuerzas que se ejercen sobre el robot (lo cual está fuera del alcance de la tesis), se encontraría que hay una disminución en el par requerido por cada rueda en comparación con el antecesor, debido a la distribución de fuerzas en un número mayor de soportes.

## DISCUSIÓN

---

El robot móvil descrito en este capítulo presenta como características principales el ser omnidireccional y redundante. Al ser omnidireccional es capaz de moverse libremente en cualquier dirección permisible con el ambiente; y al ser redundante se plantea que es capaz de soportar grandes cargas debido a la distribución de las fuerzas, este hecho queda por comprobar mediante un análisis estático de las fuerzas ejercidas sobre el robot.

A diferencia de la mayoría de los robots móviles omnidireccionales, el robot descrito no necesita de ruedas especiales (mecanum, suecas, castoras, universales) o de mecanismos especiales para funcionar; por el contrario emplea ruedas comunes del mismo tipo que emplean los robots móviles diferenciales.

El análisis cinemático y dinámico presentados son únicos en su clase al describir a un robot móvil omnidireccional que emplee ruedas convencionales. Además, describen a un robot móvil redundante.

---

# DESCRIPCIÓN DEL SISTEMA

---

---

## DESCRIPCIÓN GENERAL

---

El sistema global está constituido por un robot móvil omnidireccional redundante, un sistema de cómputo y un sistema de visión.

El sistema de cómputo es el encargado de procesar toda la información proveniente del robot y del sistema de visión, para tomar las acciones requeridas para que el robot se mueva como se le ordena. En él se encuentran los softwares de visión, el procesamiento y acoplamiento de la información de la visión para indicar la posición en la que se encuentra el robot; la lectura de las posiciones deseadas para el robot, los cálculos cinemáticos y el controlador, que determinan que movimientos y a qué velocidad debe ir el robot para alcanzar la referencia deseada; el acoplamiento y envío de la información de velocidad para que el robot pueda leerla; y la lectura de la información que el robot envía al sistema para cerrar el lazo de control.

El sistema de visión se conforma por una cámara web, la cual determina el área de trabajo del robot y se comunica con el sistema de cómputo para procesar la información.

El robot móvil omnidireccional redundante se encarga de recibir los valores de velocidad para cada una de sus llantas, provenientes del sistema de cómputo, y de asegurar que cada llanta se mueva a dicha velocidad. A su vez, también envía información proveniente de encoders absolutos, al sistema de cómputo, para cerrar el lazo de control.

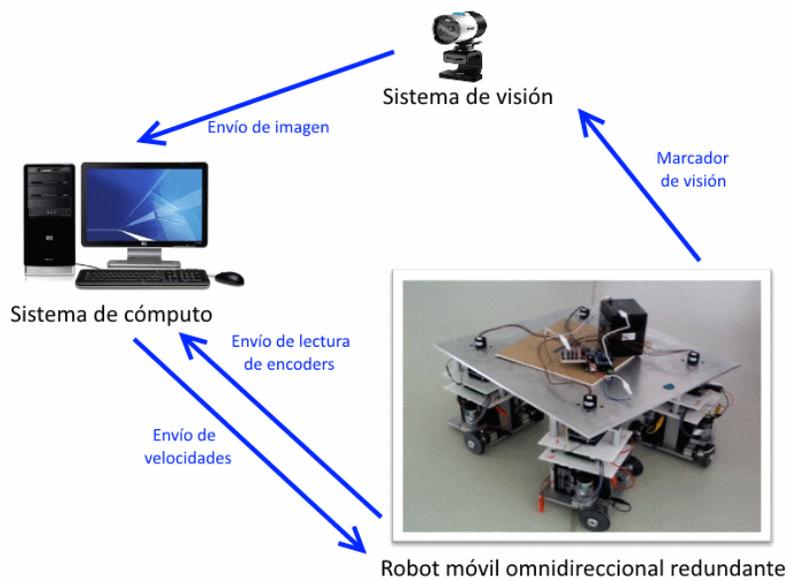


ILUSTRACIÓN 6: DESCRIPCIÓN DEL SISTEMA

## ROBOT MÓVIL OMNIDIRECCIONAL REDUNDANTE

El robot móvil omnidireccional redundante se puede dividir en dos partes para su descripción. La primera parte consiste en la plataforma central y la electrónica de la que dispone. La segunda parte está compuesta por los cuatro robots móviles diferenciales; los cuales son idénticos entre sí.

### PLATAFORMA CENTRAL

El robot está compuesto por una plataforma rectangular de 28x32.5 cm de lado, en cada esquina la plataforma se une a un robot móvil diferencial por medio de un acoplamiento rígido. En la parte de la plataforma donde está el acoplamiento, también se encuentra un encoder absoluto el cual es el encargado de medir la orientación de los robots móviles diferenciales con respecto a la plataforma. Para poder leer la información de los encoders se dispone de un Arduino uno y de un WiFi shield para realizar el envío de datos hacia el procesador central; finalmente para alimentar la electrónica se emplea una batería de 12V.

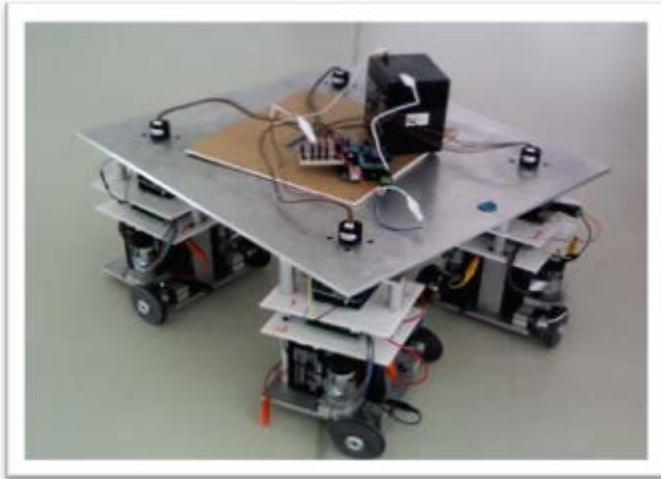


ILUSTRACIÓN 7: ROBOT MÓVIL OMNIDIRECCIONAL REDUNDANTE

## ROBOTS MÓVILES DIFERENCIALES

---

Cada robot está formado por una estructura mecánica de aluminio y acrílico; incluye un soporte para los motores, las tarjetas de control y la batería; y un mecanismo de conexión con la plataforma central y los encoders.

Cada robot móvil tiene un shield WiFi para recibir la velocidad que deben alcanzar sus motores; una tarjeta Arduino para procesar la información de velocidad recibida y comunicarse con una tarjeta MD25, la cual es la encargada de controlar la velocidad de los motores; dos motores EMG30, una batería de 12 [V] para alimentar la electrónica, un par de ruedas y engranes de para conectar las ruedas con los motores.

### Descripción de los encoders

Los encoders absolutos empleados, para medir la orientación de los robots móviles diferenciales con respecto a la plataforma, son de Avago Technologies y su número de serie es el AEAT – 6600 –T16. A continuación se muestran sus características principales.

TABLA 2: TABLA DE ESPECIFICACIONES

Compañía	Avago Technologies
Resolución	10 to 16 bits
Salidas	SSI, ABI, UVW, PWM outputs
Voltaje de entrada	3.3 o 5 Volts
Velocidad mecánica de lectura máxima	30000 rpm
Tipo de contacto	Magnético
Tasa de muestreo	12 kHz

## DESCRIPCIÓN DE ARDUINO

Arduino [20] es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Para este trabajo de investigación se emplea la Arduino Uno.



ILUSTRACIÓN 8: ARDUINO UNO

El Arduino Uno es una plataforma electrónica basada en el ATmega328. Cuenta con 14 pines digitales de entrada/salida (de los cuales 6 pueden utilizarse para salidas PWM), 6 entradas analógicas, un resonador cerámico 16[MHz], una conexión USB, un

conector de alimentación, una cabecera ICSP, y un botón de reinicio. En la tabla siguiente se muestran sus características principales.

TABLA 3: CARACTERÍSTICAS DE ARDUINO UNO

Microcontrolador	Atmega328
Voltaje operativo	5[V]
Voltaje de entrada (recomendado)	7-12[V]
Voltaje de entrada (límites)	6-20[V]
Entradas/Salidas digitales	14 (6 PWM)
Entradas analógicas	6
Corriente directa por pin	40[mA]
Velocidad de reloj	16[MHz]

## DESCRIPCIÓN DE WIFI SHIELD

El WiFi shield de Arduino [21] permite conectar tarjetas Arduinos compatibles a redes inalámbricas, utilizando la especificación inalámbrica 802.11, por medio de conexión TCP/IP o UDP. Para el caso de este proyecto se emplea la comunicación UDP debido a que es compatible con el software de Matlab.

Arduino se comunica con el WiFi shield mediante el bus SPI (a través de la cabecera ICSP). Esto es en los pines digitales 11, 12 y 13 en Arduino Uno y los pines 50, 51, y 52 en Arduino Mega. En ambas tarjetas, el pin 10 se utiliza para seleccionar el HDG204 y el pin 4 de la tarjeta SD. Estos pines no se pueden utilizar. En los Arduinos Mega, el pin 53 debe estar en desuso.

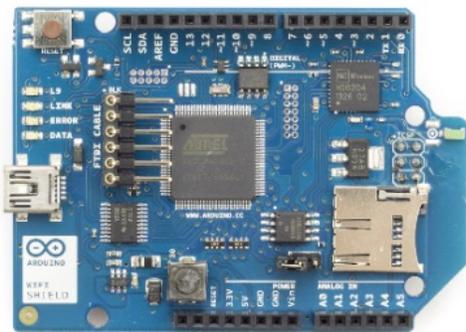
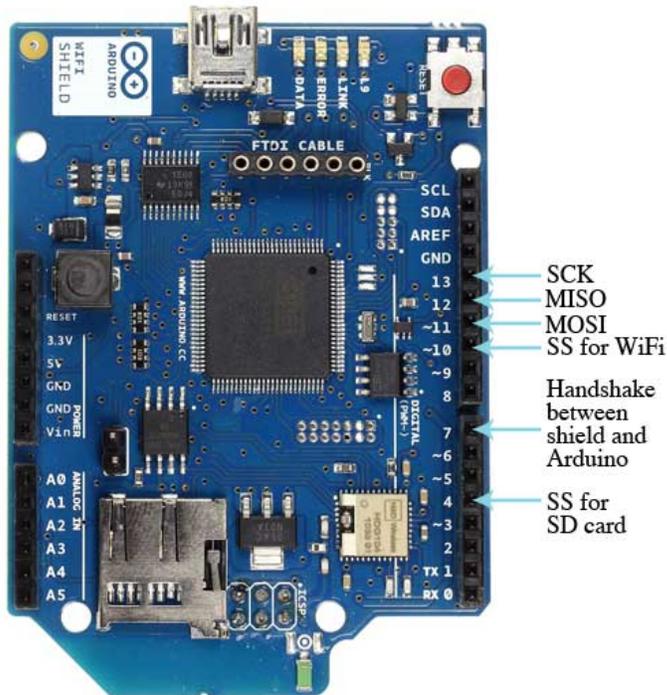


ILUSTRACIÓN 9: WIFI SHIELD



SCL  
 SDA  
 AREF  
 GND  
 13  
 12  
 11  
 10  
 9  
 8  
 7  
 6  
 5  
 4  
 3  
 2  
 1  
 TX 1  
 RX 0

SCK  
 MISO  
 MOSI  
 SS for WiFi  
 Handshake between shield and Arduino  
 SS for SD card

ILUSTRACIÓN 10: DESCRIPCIÓN WIFI SHIELD

## DESCRIPCIÓN DE LOS MOTORES Y LA TARJETA CONTROLADORA

La tracción de las ruedas está dada por ocho motores EMG30 [22], dos por cada robot móvil diferencial. Son motores que funcionan con 12 [V] de corriente directa, tienen integrado un encoder de efecto hall y una reducción de 30:1. Su velocidad máxima sin carga, utilizando la tarjeta MD25 como controlador, es de 200 revoluciones por minuto y sus especificaciones nominales se muestran en la siguiente tabla.



ILUSTRACIÓN 11: MOTOR EMG30

TABLA 4: CARACTERÍSTICAS DEL MOTOR EMG30

<b>Voltaje nominal</b>	<b>12 [V]</b>
<b>Torque nominal</b>	1.5 [Kg/cm]
<b>Velocidad nominal</b>	170 [rpm]
<b>Corriente nominal</b>	530 [mA]
<b>Velocidad sin carga</b>	216 [rpm]
<b>Corriente sin carga</b>	150 [mA]
<b>Corriente a motor bloqueado</b>	2.5 [A]
<b>Potencia nominal</b>	4.22[W]
<b>Pulsos por vuelta del encoder</b>	360
<b>Velocidad máxima sin carga a 12[V]</b>	200 rpm (20rad/s)

La tarjeta MD25 [23] es un controlador de doble puente H desarrollada para el control de dos motores de corriente directa EMG30. Sus características principales son:

- Lectura de los encoders para determinar el desplazamiento de los motores y el sentido de giro de estos.
- Alimentación con voltaje único de 12[V] de corriente directa.
- Regulador de +5[Vcc] a 300[mA] para alimentar circuitería externa.
- Suministro de hasta 3[A] de corriente para cada motor.
- Emplea comunicación serial o I2C con posibilidad de conectar hasta ocho controladores MD25 en el mismo bus.
- Control de potencia y aceleración.



ILUSTRACIÓN 12: MD25

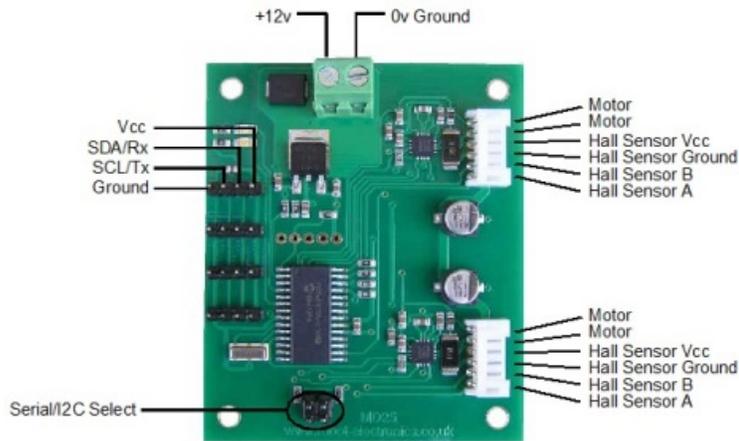


ILUSTRACIÓN 13: DESCRIPCIÓN MD25

Esta tarjeta permite el control de velocidad de los motores y otras aplicaciones como lectura de encoders, lectura de voltaje de entrada, etc. Mediante el envío de comandos a determinados registros. La comunicación con la tarjeta MD25 se puede realizar mediante el envío de comando en forma serial o mediante comunicación I2C. Para este trabajo se decidió emplear el protocolo de comunicación I2C para comunicar la tarjeta MD25 con Arduino; debido a que se emplea el protocolo de comunicación serial para corroborar que la información se envíe y se reciba de forma adecuada, en caso de ser necesario.

Para el control de los motores, existen distintas opciones configurables por medio de los registros; la inversión del sentido de giro de los motores, la forma de enviar las velocidades deseadas, etc. La configuración deseada, dependerá de la aplicación para la cual se esté empleando la tarjeta MD25; sin embargo para este proyecto en particular los motores se manejan en modo independiente, con datos de entrada que van de 0 a 255, siendo 0 la velocidad máxima en un sentido, 255 la velocidad máxima en el sentido opuesto y 128 alto total.

Debido a que la información que se escribe en los registros de la tarjeta MD25 está entre 0 y 255; y la velocidad de la llantas se encuentra entre -20 rad/s y 20rad/s se requiere hacer una transformación lineal. Esta transformación queda expresada por la siguiente ecuación.

$$(3.1) \quad v(x) = 6.4080x + 127.6666$$

Donde  $v(x)$  es la velocidad expresada en el intervalo cerrado  $[0,255]$  y  $x$  es la velocidad en rad/s.

## SISTEMAS EN TIEMPO REAL

---

Un sistema se puede definir como un mapeo de un conjunto de elementos de entrada a un conjunto de elementos de salida. Al tiempo transcurrido entre que el sistema recibe la información de entrada, la transforma y pone a disposición el conjunto completo de salidas, se le denomina tiempo de respuesta del sistema. Por lo tanto un sistema en tiempo real es aquel que debe satisfacer una restricción de tiempo explícita, tiene un tiempo de respuesta limitado; o de lo contrario el sistema podría comportarse de forma errática e incluso producir fallos [24].

Un sistema que mide la temperatura de una habitación exactamente cada minuto, el sistema de control de un dron que permite estabilizarlo en el aire, y un robot que realiza tareas de servicio retroalimentándose por medio de información visual son ejemplos de sistemas en tiempo real; debido a que todos ellos se ven sometidos a determinadas restricciones de tiempo, sin importar que la restricción varíe de sistema en sistema.

Los Sistemas en tiempo real son a menudo sistemas reactivos o embebidos. Los sistemas reactivos son aquellos en los que la programación es impulsada por la interacción continua con su entorno; por ejemplo, un sistema de control de fuego reacciona a botones presionados por un piloto. Los sistemas embebidos son aquellos que se encuentran en un sistema que no es en sí un ordenador, como ejemplo están: Raspberry pi, Galileo, Edison y Beaglebone.

## TIPOS DE SISTEMAS EN TIEMPO REAL

---

Los sistemas en tiempo real se clasifican [24] [25] de acuerdo al tipo de fallas que pueden presentarse al ocurrir en el incumplimiento de los plazos de tiempo impuestos por las restricciones:

- Tiempo real blando: Un sistema en tiempo real blando es uno en el que el rendimiento se ve degradado, pero no destruido por no cumplir con las restricciones de tiempo impuestas.
- Tiempo real duro: Un sistema en tiempo real duro es aquel en que el incumplimiento de la restricción de tiempo, aunque sea solo un plazo, puede conducir a un fallo completo del sistema.
- Tiempo real sólido: Es aquel en el que el incumplimiento de algunos plazos de tiempo no darán lugar a un fracaso total, pero incumplir más que algunos plazos puede conducir a un fallo completo y catastrófico del sistema.

Otra forma de clasificar los sistemas en tiempo real es por el tipo de hardware empleado:

- Tiempo real duro: En esta división la tarea a realizar en tiempo real debe ser completada en un lapso de tiempo garantizado. Para este tipo de tiempo real es necesario el empleo de hardware dedicado exclusivamente a la realización de la tarea deseada.
- Tiempo real blando: En este tipo de tiempo real no se hace empleo de hardware dedicado, sino que a las tareas a ser realizadas en tiempo real se les asigna una mayor prioridad que a las demás.

## TIEMPO REAL EN MATLAB

---

Matlab posee una aplicación para hacer simulaciones en tiempo real, el toolbox Real Time Windows Target. Esta herramienta permite realizar aplicaciones de control y simulaciones en tiempo real para plantas físicas [26].

Una aplicación en tiempo real tiene las siguientes características:

- Código compilado: Es el resultado de compilar el código fuente, en el caso de Simulink el código del modelo matemático es traducido y compilado en lenguaje C.
- Relación con el modelo de Simulink: El ejecutable contiene una relación de todos los componentes del modelo, es decir conexión entre bloques, dependencias de tiempo y variables.

- Relación con el kernel: el modelo tiene que ser cargado y ejecutado directamente por el kernel del Real Time Windows Target.
- Cheksum: el modelo y el ejecutable de Simulink contienen un valor de Cheksum. El kernel utiliza este valor para comparar el modelo y el ejecutable, si estos son coherentes permitirá realizar la ejecución.

---

## REAL TIME WINDOWS TARGET

---

El Real Time Windows Target proporciona controladores de dispositivos de Entrada/Salida que admiten una amplia selección de tarjetas O/I, lo que le permite interactuar con sensores, actuadores y otros dispositivos para desarrollar, experimentar y probar los sistemas en tiempo real [27].

Incluye un motor en tiempo real que se ejecuta en modo kernel de Windows. Este motor de tiempo real carga controladores de dispositivos de Entrada/Salida y establece una conexión con Simulink. En el modo normal, los controladores de dispositivo de Entrada/Salida se ejecutan en tiempo real, en paralelo con una simulación Simulink. Real Time Windows Target sincroniza los datos entre el motor en tiempo real y Simulink; alcanzando un rendimiento superior a 500 Hz en el modo normal. En modo externo el motor en tiempo real carga el archivo binario, resultante de la compilación, y los controladores de dispositivos de Entradas/Salidas, y establece una conexión con Simulink. El modelo, el compilador y las Entradas/Salidas de todos los controladores de dispositivos se ejecutan en tiempo real; alcanzando un rendimiento cercano a 20 kHz.

---

## CONEXIONES FÍSICAS DE DISPOSITIVOS

---

Real Time Windows Target incluye una librería de bloques de controladores de Entrada/Salida que proporcionan conexiones entre los dispositivos físicos de Entrada/Salida y los modelos en tiempo real. Permite la comunicación usando ASCII o protocolos de datos binarios de formato arbitrario. Los protocolos de comunicación compatibles incluyen UDP para las conexiones de red estándar, RS-232 para líneas de

serie, y la CAN para redes de área de controlador. Los bloques de controladores de Entrada/Salida disponibles son:

- Entrada analógica: Define los convertidores Analógico/Digital para digitalizar las señales analógicas como las señales de entrada a un modelo Simulink.
- Salida analógica: Configura convertidores Digital/Analógico para digitalizar las señales analógicas como las señales de salida de un modelo Simulink.
- Entrada digital: Conecta las señales de entrada digitales de hardware externo con el modelo de Simulink.
- Salida digital: Conecta señales digitales de salida del modelo de Simulink con hardware externo.
- Entrada del Contador: cuenta impulsos o mide frecuencia utilizando contadores de hardware de Entradas/Salidas.
- Frecuencia de salida: Genera modulación de ancho de pulso (PWM).
- Entrada de encoder: Incluye retroalimentación de encoders ópticos.
- Paquete de entrada: Recibe datos binarios sin formato.
- Salida de Paquetes: envía datos binarios sin formato.
- Corriente de entrada: recibe el formato de datos ASCII.
- Corriente de salida: Envía datos con formato ASCII.

---

### REAL-TIME WINDOWS TARGET KERNEL

---

El Real-Time Windows Target Kernel asigna la más alta prioridad de la ejecución al ejecutable de tiempo real, lo que permite que se ejecute sin interferencias en la frecuencia de muestreo seleccionada. Durante la ejecución en tiempo real del modelo, el kernel interviene para dar al modelo prioridad para utilizar la CPU con la finalidad de ejecutar cada actualización del modelo en los tiempos de muestreo prescritos. Una vez que una actualización del modelo se completa, el núcleo libera la CPU para ejecutar otras aplicaciones basadas en Windows que podrían necesitar mantenimiento [28].

El kernel intercepta la interrupción del reloj de la computadora antes que el sistema operativo Windows la reciba, y bloquea cualquier llamada al sistema operativo, mientras el kernel utiliza la interrupción para iniciar la ejecución del modelo compilado.

Para garantizar un periodo de muestreo preciso el kernel reprograma el reloj de la computadora a una frecuencia mayor, Debido a que el reloj del PC es también la principal fuente de tiempo para el sistema operativo, el kernel envía una interrupción al sistema manteniendo la tasa de interrupción inicial. Durante la ejecución de la aplicación en tiempo real almacena los datos en buffers, y posteriormente el contenido de los buffers es recuperado por Simulink para imprimirlas en pantalla.

## SISTEMA DE VISIÓN ARTIFICIAL

---

La visión artificial es la capacidad de la máquina para ver el mundo que le rodea, más precisamente para reducir la estructura y las propiedades del mundo tridimensional a partir de una o más imágenes bidimensionales [29] .

El proceso fundamental de un sistema de visión puede definirse como sigue:

1. Adquisición de imágenes: Por medio de una cámara la imagen es adquirida.
2. Procesamiento de la imagen: La imagen pasa por una serie de fases que transforman y filtran la información para facilitar la extracción de parámetros de importancia.
3. Extracción de información: Se obtienen los parámetros importantes para la realización de una determinada tarea, como ejemplo está la posición y orientación de un robot manipulador con respecto a un objeto a transportar.

## CONTROL SERVO VISUAL

---

El término *Visual Servoing* (control servo visual) se refiere al control de un robot empleando visión artificial. Es una técnica que utiliza la información extraída de un sensor de visión para controlar los movimientos de un robot.

Existen dos acercamientos fundamentales para el control servo visual [30]: Control servo visual basado en posición y control servo visual basado en imagen. El primero emplea características visuales observadas, una cámara calibrada y un modelo geométrico conocido del objetivo para determinar su posición y orientación con respecto a la cámara; el robot se mueve hacia la posición indicada y el controlador

realiza la tarea en el espacio tarea. El segundo omite el paso de estimación de la posición y emplea las características de la imagen directamente; el control es realizado en coordenadas imagen.

## REACTIVISION

ReactIVision [31] es una multiplataforma de código abierto de visión por computador, para el seguimiento rápido y robusto de marcadores de referencia colocados en objetos físicos. ReactIVision es una aplicación que emite mensaje TUIO utilizando el protocolo de comunicación UDP en el puerto 3333 a cualquier cliente que se encuentre habilitado, se comunica directamente con la computadora que alberga el programa. Funciona bajo plataformas Windows, Linux y MacOS.

ReactIVision reconoce marcadores especialmente diseñados llamados fiducials, el software es capaz de obtener la posición en coordenadas  $(x,y)$  y la orientación del marcador con respecto al eje coordenado  $(z)$  de la cámara.



ILUSTRACIÓN 14: MARCO COORDENADO CÁMARA

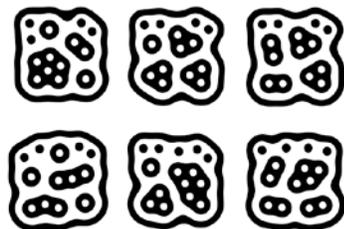


ILUSTRACIÓN 15: FIDUCIALS

## SOFTWARE DE ENLACE

Es un software realizado por Pedro Javier Gálvez Valadez con la finalidad de conectar ReactIVision con Matlab Simulink en tiempo real. El software lee la información, de hasta tres fiducials, que envía ReactIVision a la computadora y la redirecciona hacia un puerto UDP que se conecta específicamente con Matlab Simulink. La intención de esta redirección de información es que Matlab Simulink sea capaz de leer en tiempo real la información que le llega al puerto UDP.

El software está compuesto por un código en Processing y unos bloques de códigos en Matlab Simulink (ilustración 16), para su funcionamiento se requiere que primero se ejecute ReactIVision, luego el software de enlace y finalmente el bloque de código de Matlab Simulink.

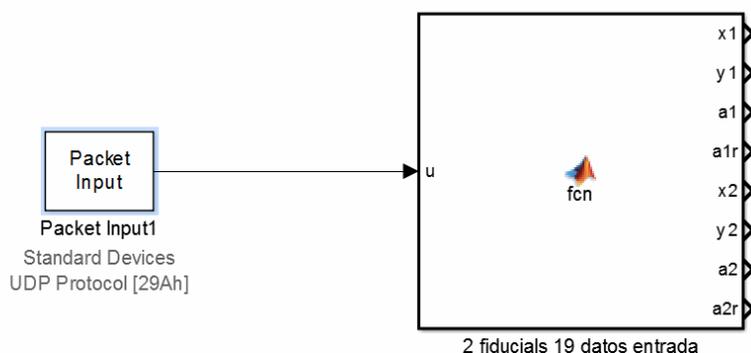


ILUSTRACIÓN 16: BLOQUES DE VISIÓN ARTIFICIAL

La información que se proporciona a la salida de los bloques de Matlab Simulink es la posición y orientación de hasta tres fiducials, ya que existen un bloque para dos fiducials y un bloque para tres; siendo el código en Processing el mismo para cada bloque.

La posición de los fiducials no está en píxeles, sino que esta escalada en un valor de 0 al 100; donde el 0 corresponde a la posición inicial del marco coordenada de la imagen y 100 corresponde a la posición máxima que alcanza la imagen.

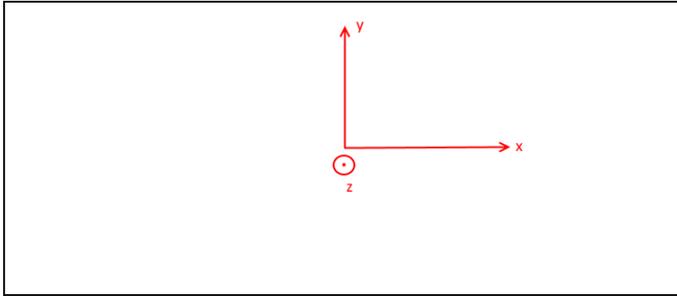


ILUSTRACIÓN 17: MARCO COORDENADO IMAGEN

Es debido a esto que es necesario transformar los valores del 0 al 100 a un equivalente a centímetros; para ello se orienta la cámara paralela al suelo y se mide la distancia que abarca la imagen físicamente en el suelo, tanto para  $x$  como  $y$ ; a continuación se emplea una ecuación que transforma a centímetros la posición del fiducial de acuerdo a un nuevo marco coordenado (marco coordenado mundo).

$$(3.2) \quad \begin{aligned} x_s &= \frac{x}{100} * (-2l) + l \\ y_s &= \frac{y}{100} * (2a) - a \end{aligned}$$

Donde  $l$  es la distancia absoluta del origen del sistema coordenado (ilustración 18), medida desde el eje ( $x$ ), al perímetro del área de trabajo;  $a$  es la distancia absoluta del origen del sistema coordenado (ilustración 18), medida desde el eje ( $y$ ), al perímetro del área de trabajo y;  $x_s$  y  $y_s$  son la posición del fiducial en [cm], con respecto al marco coordenado mundo (ilustración 18).



Área de trabajo del robot móvil omnidireccional redundante

#### ILUSTRACIÓN 18: MARCO COORDENADO MUNDO

ReacTIVision envía la orientación con respecto al marco coordenado Z de la cámara (ilustración 14), y para la realización de los experimentos es necesario que esta orientación sea con respecto al marco coordenado mundo. Debido a esto, en el código de Processing se aplica una transformación para realizar dicho cambio.

---

### MARCO COORDENADO MUNDO

---

Para el caso particular de este proyecto el marco coordenado imagen y el marco coordenado cámara (ilustraciones 17 y 14) son los mismos. La cámara se encuentra mirando hacia abajo y por lo tanto su eje (z) apunta hacia esa dirección.

El marco coordenado mundo se refiere al sistema coordenado (x,y,z) que describe la posición, en cm, del robot móvil omnidireccional redundante en su área de trabajo (ilustración 18). El marco coordenado mundo se obtiene por medio de dos procesos:

1. Trasladar el marco coordenado imagen hacia el centro de la imagen, invertir la dirección del eje (y) y transformar los valores del 0 a 100 en su equivalente en cm;
2. Invertir la dirección del eje (z)

El primer paso se realiza por medio de la ecuación 3.2, el segundo es realizado por Processing.

## MATLAB SIMULINK

---

Es en Matlab, específicamente en Simulink, donde la mayoría del procesamiento es realizado. El programa principal, que es ejecutado en Simulink, está constituido por diferentes bloques de programación; donde cada uno realiza una actividad específica. El programa en general se encarga de leer la información proveniente de la cámara, de procesarla y convertirla a cm; de leer la información proveniente de los encoders absolutos del robot móvil omnidireccional redundante; de leer la referencia deseada en posición para el robot; de ejecutar los algoritmos de control y los cálculos cinemáticos para determinar las velocidades de las llantas requeridas para alcanzar la referencia; de procesar la información para que pueda ser leída por Arduino y los controladores de velocidad; y se encarga de enviar las velocidades correspondientes a cada robot móvil diferencial. A continuación se presenta cada bloque de Simulink así como su descripción.

### BLOQUE DE VISIÓN

---

El bloque de visión está constituido por tres secciones. La primera es el Packet Input, la cual se encarga de leer la información proveniente de reactIVision y del código de Processing. La segunda sección es un bloque de función de Matlab, el cual se encarga de leer la información de exactamente dos fiducials, procesarla y enviarla a la última sección; las salidas de este bloque son la posición en (x,y) la orientación en grados y radianes; todo con respecto al marco coordenado cámara. Finalmente la tercera sección se encarga de procesar la información para transformarla del marco coordenado cámara al marco coordenado mundo ecuación 3.2.

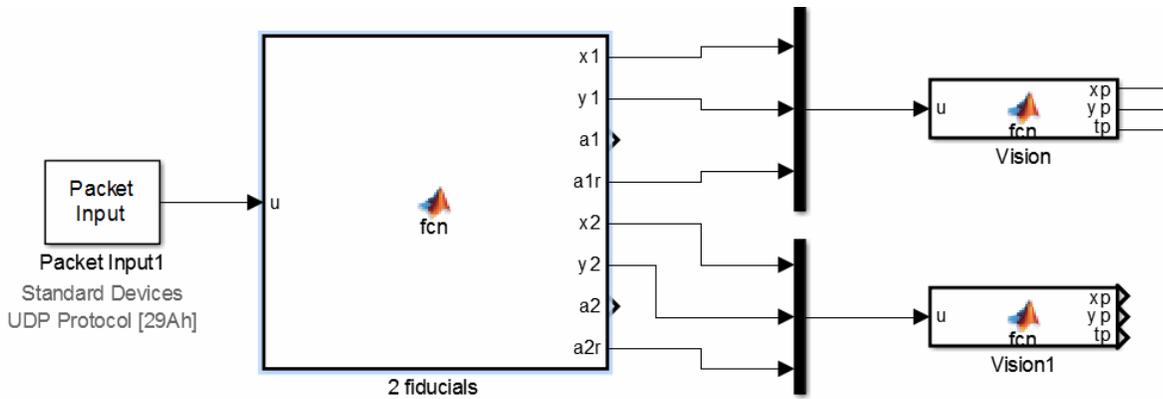


ILUSTRACIÓN 19: BLOQUE COMPLETO DE VISIÓN

### BLOQUE DE REFERENCIA

Este bloque está constituido por las posiciones deseadas para el robot, así como su orientación. El primer bloque Sine Wave corresponde a la referencia en el eje (x), el segundo a la referencia en el eje (y); y los últimos tres bloques corresponden a la referencia en orientación con respecto al eje z.

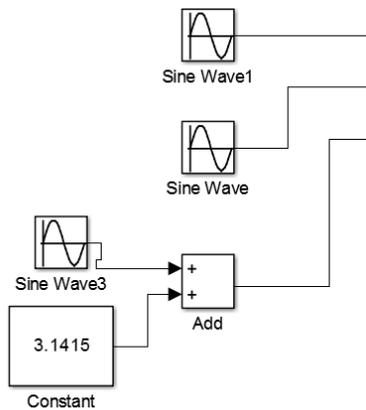


ILUSTRACIÓN 20: BLOQUE DE VARIABLES DE REFERENCIA

### BLOQUE DEL CONTROL

Este bloque se encarga de recibir la información proveniente del bloque de visión y del bloque de referencia; con la cual ejecuta un algoritmo de control, el cual determina las

velocidades lineales y angulares que el robot debe tener para alcanzar la referencia, con respecto al marco coordenado mundo.

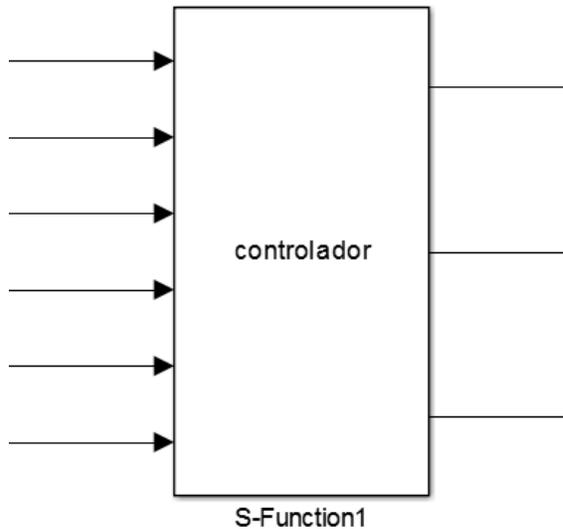


ILUSTRACIÓN 21: BLOQUE DE CONTROL

### BLOQUE DE LECTURA DE ENCODERS

Este bloque se encarga de leer la información proveniente de los encoders que se encuentran sobre el robot. Consta de un bloque Packet Input, el cual lee la información enviada por el robot; y un bloque que procesa la información y la transforma en radianes.

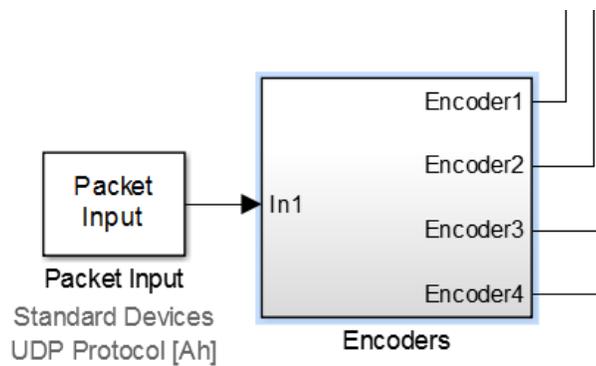


ILUSTRACIÓN 22: BLOQUE DE LECTURA DE ENCODERS

## BLOQUE DE CINEMÁTICA

---

En este bloque se calcula la cinemática del robot móvil omnidireccional redundante; emplea como entrada las orientaciones de cada robot móvil diferencial con respecto a la plataforma, así como la información de las velocidades lineales y angular proveniente del bloque del controlador. Este bloque se encarga de leer las velocidades del robot con respecto al marco coordenado mundo, y las orientaciones respectivas para calcular la velocidad de cada llanta de cada robot móvil diferencial.

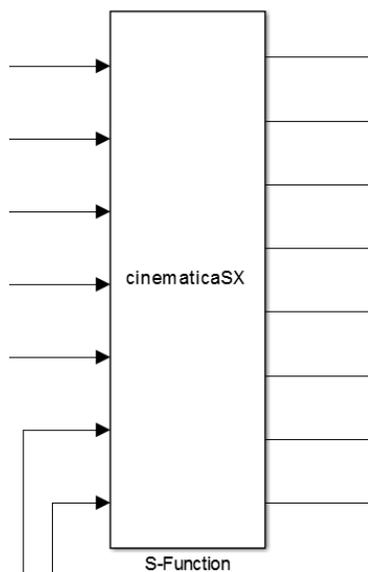


ILUSTRACIÓN 23: BLOQUE DE CINEMÁTICA

## BLOQUE DE PROCESAMIENTO

---

Este bloque se encarga de procesar la información de velocidad de las llantas y modificarla para que pueda ser leída por Arduino y por la tarjeta de control MD25 (ecuación 3.1). Ejecuta la ecuación tal y le aplica una transformación para convertir los datos en enteros sin signos.

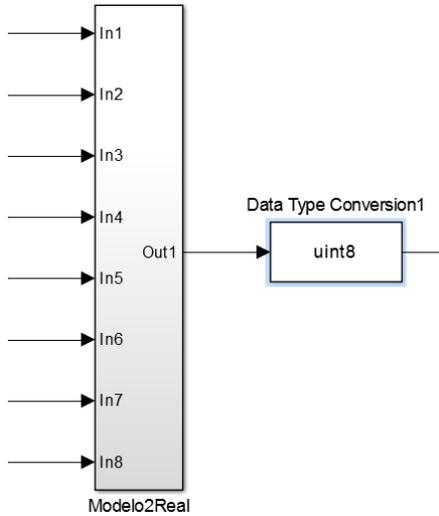


ILUSTRACIÓN 24: BLOQUE DE PROCESAMIENTO DE VELOCIDADES

## BLOQUE DE ENVÍO DE DATOS

Este bloque se encarga de recibir la información del bloque de procesamiento y enviarla directamente a las tarjetas Arduinos de los robots móviles diferenciales.

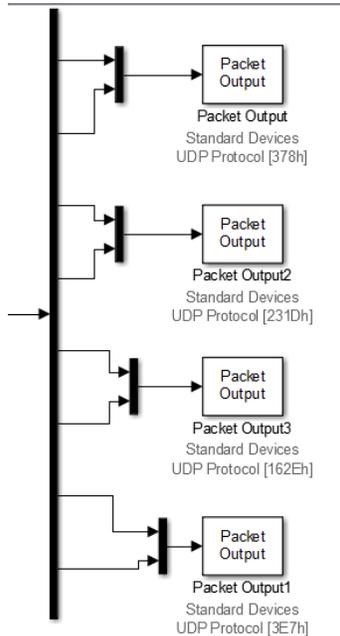


ILUSTRACIÓN 25: BLOQUE DE ENVÍO DE DATOS

## DISCUSIÓN

---

Todo sistema mecatrónico debe de cumplir con restricciones de tiempo, principalmente los sistemas robóticos: sin embargo el sistema descrito funciona parcialmente en tiempo real. Esto es debido a que el sistema de visión empleado no garantiza en ningún momento el cumplimiento de alguna restricción, para este tipo de sistemas el tiempo de muestreo debe estar garantizado como mínimo cada 33,33 milisegundos (tiempo de muestreo de una cámara web normal); pero al emplear ReactIVision los cuadros por segundos varían desde los veinte hasta los treinta. También el uso de Arduino como microcontrolador no garantiza en ningún momento el cumplimiento de alguna restricción de tiempo, simplemente funciona a la mayor velocidad posible; la cual no corresponde a la del microcontrolador que Arduino emplea debido a las capas de código que emplea el software especial requerido para programar dicho microcontrolador. La única parte del sistema que funciona en tiempo real son los códigos de programas implementados en Simulink, ya que este garantiza que cada determinado periodo se ejecute el código completo.

Un sistema de que funcione en tiempo real y tenga implementado un lazo cerrado de control, debe ejecutarse cada milisegundo para un adecuado funcionamiento. Esta característica ha sido comprobada por años de experimentación y por consejo de diversos Doctores especializados en el tema, incluyendo al Dr. Víctor Javier González Villela. Por lo tanto se deja como recomendación al lector implementar o cuando menos intentar hacer todos sus sistemas con dicha restricción de tiempo.

---

# CONTROLADORES

---

Un controlador de lazo cerrado es aquel que emplea como parámetros de entrada la información de referencia requerida para el sistema y el estado actual del sistema. En robótica móvil existen dos problemas principales en controladores de lazo cerrado: seguimiento de postura y seguimiento de punto.

---

## SEGUIMIENTO DE POSTURA

---

Consiste en encontrar un controlador de estados en lazo cerrado que pueda realizar el seguimiento de una referencia móvil (postura)  $\xi_r(t)$ , la cual puede asumirse dos veces diferenciable. El objetivo es encontrar una ley de control  $v$  tal que:

- El seguimiento del error  $\tilde{\xi}(t) = \xi(t) - \xi_r(t)$  y la ley de control  $v$  están delimitados para todo  $t$ .
- El seguimiento del error converge asintóticamente a cero, es decir:  
$$\lim_{t \rightarrow \infty} (\xi(t) - \xi_r(t)) = 0.$$
- Si  $\xi(0) = \xi_r(0)$ , entonces  $\xi(t) = \xi_r(t)$  para toda  $t$ .

---

## SEGUIMIENTO DE PUNTO

---

En algunos casos el control total de la postura del robot no es indispensable, y solo se requiere controlar la posición de un punto fijo  $P$  ubicado en el robot. El seguimiento de punto consiste en encontrar un controlador de estados en lazo cerrado que pueda realizar el seguimiento de una referencia de posición móvil  $x_r(t), y_r(t)$ , la cual puede asumirse dos veces diferenciable. El objetivo es encontrar una ley de control  $v$  tal que:

- El seguimiento del error  $\tilde{x}(t) = x(t) - x_r(t)$ ,  $\tilde{y}(t) = y(t) - y_r(t)$  y la ley de control  $v$  están delimitados para todo  $t$ .

- El seguimiento del error converge asintóticamente a cero, es decir:  

$$\lim_{t \rightarrow \infty} (x(t) - x_r(t)) = 0 \text{ y } \lim_{t \rightarrow \infty} (y(t) - y_r(t)) = 0$$
- Si  $x(0) = x_r(0)$  y  $y(0) = y_r(0)$ , entonces  $x(t) = x_r(t)$  y  $y(t) = y_r(t)$  para toda  $t$ .

## CONTROL DE ROBOTS MÓVILES OMNIDIRECCIONALES

---

El grado de movilidad de un robot móvil omnidireccional es igual a tres [18]. Al ser un robot con ruedas este valor coincide con la dimensión de estado en la que se encuentra; es decir, un robot móvil omnidireccional es capaz de moverse en tres direcciones permitidas por el movimiento en un plano: traslación en los ejes (x) y (y), y rotación alrededor del eje (z). Esto implica que una aproximación lineal del modelo dinámico de la postura de un robot móvil omnidireccional es completamente controlable. Por lo tanto para los robots móviles omnidireccionales el problema del seguimiento de postura puede ser resuelto por un control linealizado estático suave de estados en lazo cerrado. Para robots móviles restringidos el problema puede ser resuelto por un control linealizado dinámico de estados en lazo cerrado [18]. Un robot móvil omnidireccional es completamente linealizabile en lazo cerrado, mientras que un robot móvil restringido solo es parcialmente linealizabile.

El hecho de que una aproximación lineal del modelo dinámico de la postura de un robot móvil omnidireccional sea completamente controlable, implica que es posible implementar en este tipo de robots controladores tales como PD, PID, Par Calculado, y cualquier tipo de controladores que sean lineales. En este escrito se abordan el control dinámico inverso y el control por campos potenciales artificiales, siendo este último el que se implementó en el robot móvil omnidireccional redundante.

### CONTROL DINÁMICO INVERSO

---

El modelo de un robot móvil omnidireccional se representa cómo:

$$(4.1) \quad \begin{aligned} \dot{\xi} &= R^T(\theta)\eta \\ \dot{\eta} &= v \end{aligned}$$

Donde  $\dot{\xi}$  es la derivada de la postura del robot (ecuación 2.1).

$$(4.2) \quad \dot{\xi} = [\dot{x} \quad \dot{y} \quad \dot{\theta}_p]^T$$

$R(\theta)^T$  es la transpuesta de la matriz de la ecuación 2.2. Y  $\eta$  es el vector de velocidades del punto P sobre el robot móvil.

$$(4.3) \quad \eta = [\dot{x}_p \quad \dot{y}_p \quad \dot{\theta}_p]^T$$

La ecuación 4.1 se diferencia dos veces y se obtiene:

$$(4.4) \quad \ddot{\xi} = R^T(\theta)\dot{\eta} + \dot{R}^T(\theta)\eta$$

Se escoge la ley de control  $v$  tal que:

$$(4.5) \quad v = R^{-T}(\theta) \left( -R^T(\theta)\eta + \ddot{\xi}_r - (\Lambda_1 + \Lambda_2)\dot{\xi} - \Lambda_1\Lambda_2\xi \right)$$

Donde  $\tilde{\xi}$  es el error de la postura y donde  $\Lambda_1, \Lambda_2$  son matrices diagonales definidas positivas de  $(3 \times 3)$ .

Al sustituir la ecuación 4.5 en la ecuación 4.4; es decir, al introducir la ley de control en las ecuaciones del robot móvil se obtiene:

$$(4.6) \quad \ddot{\tilde{\xi}} + (\Lambda_1 + \Lambda_2)\dot{\tilde{\xi}} + \Lambda_1\Lambda_2\tilde{\xi} = 0$$

La ecuación 4,6 es una ecuación diferencial cuyo resultado expresa que conforme pase el tiempo el error  $\tilde{\xi}$  tiende a cero.

## CAMPOS POTENCIALES ARTIFICIALES

Los Campos Potenciales Artificiales (CPA) son campos de fuerzas generados por obstáculos y metas. Esta teoría surge de hacer una analogía con la electrostática, en donde cargas de signos contrarios se atraen y cargas de igual signo se repelen, en el caso de los campos potenciales artificiales y los robots móviles, el robot presenta una "carga" opuesta a la meta (punto que se desea alcanzar) e igual a los obstáculos (punto o región que se desea evitar).

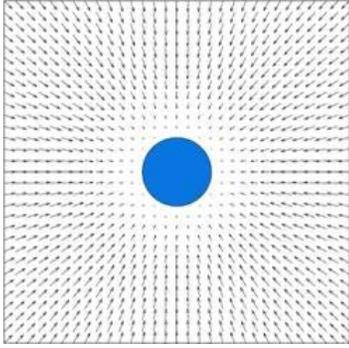


ILUSTRACIÓN 26: CAMPO POTENCIAL ATRACTIVO

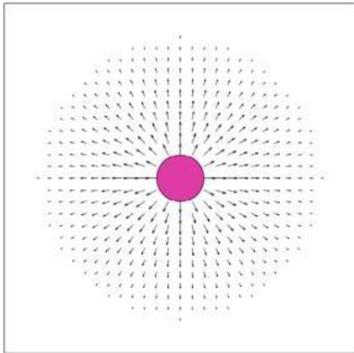


ILUSTRACIÓN 27: CAMPO POTENCIAL REPULSIVO

Debido a los signos de las cargas se crea una fuerza atractiva entre el efector final y el objetivo, y una fuerza repulsiva entre el efector final y los obstáculos. Estas fuerzas no son debidas a interacciones físicas, es decir no existen realmente cargas eléctricas en los puntos mencionados, pero se modelan análogamente a las cargas eléctricas. El modelado no es idéntico al de las cargas eléctricas porque el comportamiento de los campos generados puede ser propuesto a conveniencia, siempre conservando como principio básico las fuerzas atractivas para las metas y fuerzas repulsivas para los obstáculos. La suma de las fuerzas tanto atractivas como repulsivas da como resultado la fuerza total ejercida sobre el robot móvil, a la cual se le calcula el gradiente con la finalidad de obtener la máxima dirección de cambio; es decir, al calcular el gradiente se obtiene la dirección hacia la cual va cambiando la fuerza con lo que se obtiene un vector dirección que guiará al robot móvil al punto deseado.

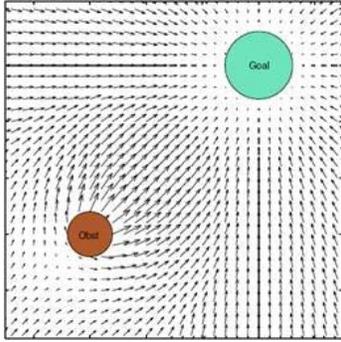


ILUSTRACIÓN 28: CAMPOS POTENCIALES ARTIFICIALES

Pero existe un inconveniente con esta forma de navegación, los campos potenciales pueden dar lugar a mínimos locales. Los cuales son puntos donde la suma de fuerzas (atractivas y repulsivas) se vuelve cero con lo que el robot queda en un estado de reposo aún sin haber llegado a la posición deseada; esto provoca que el gradiente sea cero y no haya un cambio en la dirección por lo que no habrá movimiento.

---

### PLANEACIÓN DE TRAYECTORIAS

---

Los campos potenciales artificiales permiten una planeación reactiva, es decir es una planeación hecha en el momento y si intervención humana con la finalidad de llevar al robot móvil a su destino. Existen tres propuestas para una planeación de movimiento haciendo uso de los campos potenciales artificiales [32]. Estas se enlistan a continuación:

La primera posibilidad es considerar la fuerza producida por los CPA como la entrada de control al sistema  $\tau = f(q)$ . Siendo  $\tau$  el par motor ejercido sobre cada articulación y  $f(q)$  la fuerza producida por los campos potenciales artificiales expresada en términos de las variables articulares.

La segunda posibilidad consiste en considerar al robot como una masa unitaria moviéndose bajo la influencia de  $f(q)$ , es decir, sobre el robot se ejercerá una aceleración deseada equivalente a la fuerza producida por los campos potenciales artificiales  $\ddot{q} = f(q)$ .

La tercera posibilidad es considerar la fuerza  $f(q)$  generada por los campos potenciales artificiales como una velocidad deseada para el robot, esto es posible debido a que los campos potenciales artificiales no representan físicamente una fuerza  $\dot{q} = f(q)$ .

En principio, se puede utilizar cualquiera de los 3 métodos mencionados para la planeación del movimiento; en la primera posibilidad se calculan directamente los valores del par motor para el robot, en la segunda se requiere del cálculo de la dinámica inversa del robot para calcular el par motor necesario, y finalmente en la tercera se puede utilizar un esquema de control cinemático.

- La ecuación  $\tau = f(q)$  genera trayectorias más suaves, esto debido a que la dinámica del robot “filtra” naturalmente las reacciones que el robot pueda tener en presencia de obstáculos.
- La implementación de la ecuación  $\ddot{q} = f(q)$  es más rápida que la ecuación  $\tau = f(q)$  en la corrección de los movimientos indicados por  $f(q)$ .
- Otro aspecto a considerar es que la ecuación  $\dot{q} = f(q)$  garantiza que el robot alcanzará la posición deseada con velocidad 0 lo cual no es necesariamente cierto para las otras dos ecuaciones.

## FUNCIONES DE NAVEGACIÓN

---

Las funciones de navegación están definidas en función a la distancia a los obstáculos, solo tienen un mínimo (global) y aplican para una limitada clase de espacios de configuración. Una función de navegación queda definida por cuatro características:

- Es suave.
- Tiene un solo mínimo en la meta, en el componente conectado del espacio libre de la meta.
- Es uniformemente máxima en los límites del espacio libre.

- Es Morse. Una función es Morse si todos sus puntos críticos son no generativos.

Los espacios de configuración más comunes para el empleo de navegación son el esférico y la estrella. En el primero se asume que el espacio está limitado por una esfera con centro en  $q_0$  el cual es un punto central definido por conveniencia; en el espacio estrella se considera un espacio de trabajo con forma de estrella y obstáculos con la misma forma.

---

### IMPLEMENTACIÓN

---

Las ecuaciones empleadas en esta tesis se basan en [33] y consideran la fuerza  $f(q)$  generada por los campos potenciales artificiales como una velocidad deseada para el robot. En las ecuaciones de control el objetivo a alcanzar representa un vector de velocidad en dirección del robot hacia este. La magnitud de la velocidad es directamente proporcional a la distancia existente entre ambos cuerpos. Para evitar vectores de velocidad de magnitud superiores a las que el robot acepta, se define una velocidad máxima y un área de evasión alrededor del objetivo; a partir de la cual la velocidad comienza a disminuir.

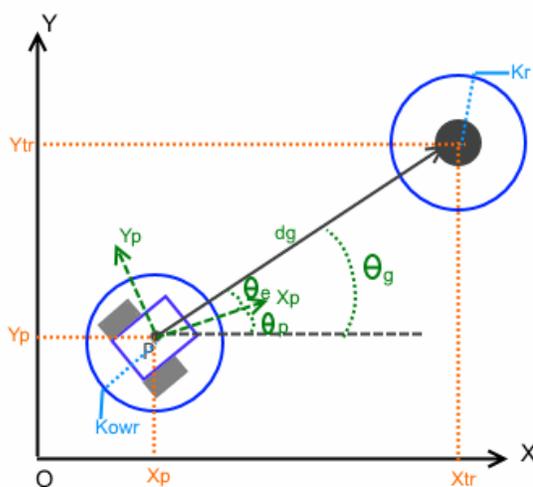


ILUSTRACIÓN 29: DIAGRAMA DE LAS ECUACIONES DE CONTROL

En la ilustración 29 se muestra a  $d_g$  como el vector de atracción del robot al objetivo,  $\theta_g$  como el ángulo de dicho vector,  $\theta_p$  como la inclinación del robot con respecto al marco coordenado mundo y  $\theta_e$  como el error entre ambos ángulos. El área de evasión de la meta se define como  $k_r$  y se define arbitrariamente dependiendo de la aplicación.

También se define un área de seguridad para el robot móvil denominada  $k_{owr}$ , la cual se acostumbra sea el área que abarca el robot móvil, sin embargo también puede ampliarse dependiendo de la aplicación. Dado que el robot sobre el cual se emplearán estas ecuaciones es un robot móvil omnidireccional redundante se requieren tres ecuaciones: para la velocidad lineal en (x), para la velocidad lineal en (y) y para la velocidad angular en (z).

A partir de la información anterior se definen las siguientes ecuaciones:

$$(4.7) \quad \begin{aligned} v_x &= v_{x\max} \cos \theta_e \quad \text{si } \|d_g\| > (k_r + k_{owr}) \\ v_x &= \frac{v_{x\max}}{k_r} d_g \cos \theta_e \quad \text{si } \|d_g\| < (k_r + k_{owr}) \end{aligned}$$

$$(4.8) \quad \begin{aligned} v_x &= 0 \quad \text{si } \|d_g\| \leq k_{owr} \\ v_y &= v_{y\max} \sin \theta_e \quad \text{si } \|d_g\| > (k_r + k_{owr}) \\ v_y &= \frac{v_{y\max}}{k_r} d_g \sin \theta_e \quad \text{si } \|d_g\| < (k_r + k_{owr}) \\ v_y &= 0 \quad \text{si } \|d_g\| \leq k_{owr} \end{aligned}$$

El robot móvil al ser omnidireccional tiene la capacidad de alcanzar la meta con cualquier orientación, debido a de grado de movilidad, por lo que no se requiere reducir el valor del ángulo  $\theta_e$  a cero. Sin embargo, para orientar la plataforma con la referencia deseada se emplea la siguiente ecuación:

$$(4.9) \quad w = w_{\max} \sin \theta_i$$

$$(4.10) \quad \theta_i = \theta_p - \theta_d$$

Donde  $\theta_p$  es la orientación del robot móvil y  $\theta_d$  es la orientación deseada. Este conjunto de ecuaciones no contempla la evasión de obstáculos, ya que no se encuentra dentro de los objetivos de este trabajo.

## DISCUSIÓN

---

Una de las ventajas de un robot móvil omnidireccional es que la linealización de su dinámica es completamente controlable, con lo cual es posible implementarle una gran gama de controladores desde un PID hasta campos potenciales artificiales. A su vez este tipo de robots presenta la característica de moverse con tres grados de libertad en el plano; por lo tanto el seguimiento de postura para este tipo de robots es completamente factible, ya que en comparación un robot móvil diferencial (el cual tiene dos grados de libertad en el plano) es capaz de llegar a un punto establecido pero no con una orientación establecida.

---

# ANÁLISIS DE RESULTADOS

---

---

## CARACTERÍSTICAS DEL EXPERIMENTO

---

Los experimentos que se describirán a continuación tienen en común las siguientes características:

- Tiempo de muestreo de Matlab Simulink igual a 0.003ms.
- El muestreo máximo de la cámara es de 30 cuadros por segundo.
- La velocidad lineal máxima real del robot en el eje (x) y (y) es: 20cm/s.
- La velocidad angular máxima real del robot en el eje (z) es: 0.4 rad/s.
- El valor de  $k_{ovr} = 50$  y de  $k_r = 1$ .
- Las dimensiones de la plataforma son:  $a = \frac{32.5}{2} \text{ cm}$  y  $b = 14 \text{ cm}$ .
- Las dimensiones de los robots móviles diferenciales son:  $c = 5 \text{ cm}$ ,  $d = 9 \text{ cm}$  y  $r = 3 \text{ cm}$ .
- El puerto UDP de lectura de encoders tiene los siguientes valores:
  - Puerto local: 10
  - Dirección remota: 192.168.0.105
  - Puerto UDP remoto: 56000
- El puerto UDP de visión artificial tiene los siguientes valores:
  - Puerto local: 666
  - Dirección remota: 127.0.0.1
  - Puerto UDP remoto: 1000
- El puerto UDP del robot móvil omnidireccional 1 (ilustración 3) tiene los siguientes valores:
  - Puerto local: 888
  - Dirección remota: 192.168.0.101
  - Puerto UDP remoto: 6780
- El puerto UDP del robot móvil omnidireccional 2 (ilustración 3) tiene los siguientes valores:

- Puerto local: 8989
  - Dirección remota: 192.168.0.102
  - Puerto UDP remoto: 555
- El puerto UDP del robot móvil omnidireccional 3 (ilustración 3) tiene los siguientes valores:
  - Puerto local: 5678
  - Dirección remota: 192.168.0.103
  - Puerto UDP remoto: 12000
- El puerto UDP del robot móvil omnidireccional 4 (ilustración 3) tiene los siguientes valores:
  - Puerto local: 999
  - Dirección remota: 192.168.0.104
  - Puerto UDP remoto: 15000
- La dirección IP de la computadora central es: 192.168.0.100
- El controlador empleado es el mencionado en las ecuaciones 4.7, 4.8 y 4.9.
- El área de trabajo del robot móvil omnidireccional es: 220cm x 170cm.
- El sentido de lectura de los encoders absolutos es en sentido contrario a las manecillas del reloj y su posición inicial está dada por la posición inicial de los robots móviles diferenciales (ilustración 3).
- La orientación de los fiducials se mide en sentido contrario a las manecillas del reloj. La orientación inicial del fiducial, es la que reconoce ReactIVision por default; y es la orientación con la que aparecen en la hoja de impresión [31].

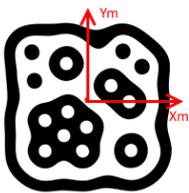


ILUSTRACIÓN 30: CONDICIÓN INICIAL DE ORIENTACIÓN

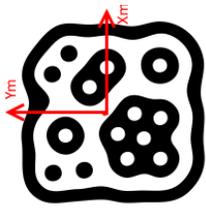


ILUSTRACIÓN 31: ORIENTACIÓN A 90 GRADOS DEL FIDUCIAL

## REGULACIÓN

El experimento consistió en colocar al robot en una posición inicial elegida al azar e indicarle que se dirigiera a una posición final aleatoria sin seguir algún perfil preestablecido de velocidad; todo dentro de su área de trabajo. La finalidad de este experimento es medir el error promedio de posición del robot móvil omnidireccional así como su distribución media.

### ERRORES PROMEDIO

El error promedio de posición en el eje (x) fue  $e_x = 0.529032cm$ , en el eje (y) fue de  $e_y = 0.516129cm$  y el error promedio de orientación con respecto al eje (z) fue de  $e_w = 0.010813rad$ . La distribución media del error en el eje (x) fue de  $d_{mx} = 0.472425cm$ , en el eje (y) fue  $d_{my} = 0.398335cm$  y la distribución media con respecto al error en el eje (z) fue de  $d_{mw} = 0.0104rad$ . A continuación se muestran los resultados obtenidos de algunos experimentos. La tabla muestra los errores obtenidos y los términos E1, E2 y E3 hacen referencia a un conjunto de gráficas pertenecientes a un mismo experimento. El tiempo promedio estimado en alcanzar el error fue  $t = 15.09929s$ , la distribución media del tiempo fue  $d_{mt} = 2.708092$ .

TABLA 5: EXPERIMENTOS DE REGULACIÓN

	Error de posición en el eje x cm	Error de posición en el eje y cm	Error de orientación en el eje z rad	Ilustraciones
<b>E1</b>	0	0	0.0001	32, 33 y 34
<b>E2</b>	0	0.6	0.0001	35, 36 y 37
<b>E3</b>	0	0.6	0.0001	38, 39 y 40

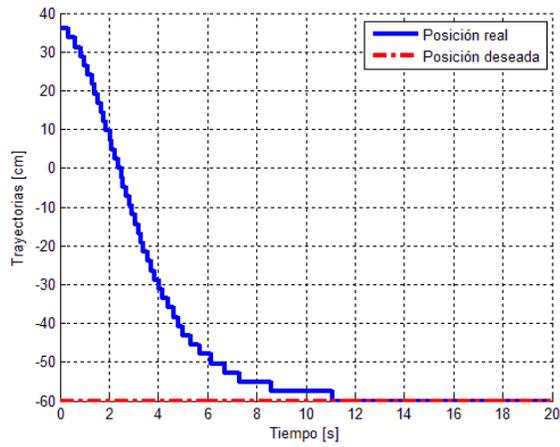


ILUSTRACIÓN 32: E1 TRAYECTORIA EN EL EJE X

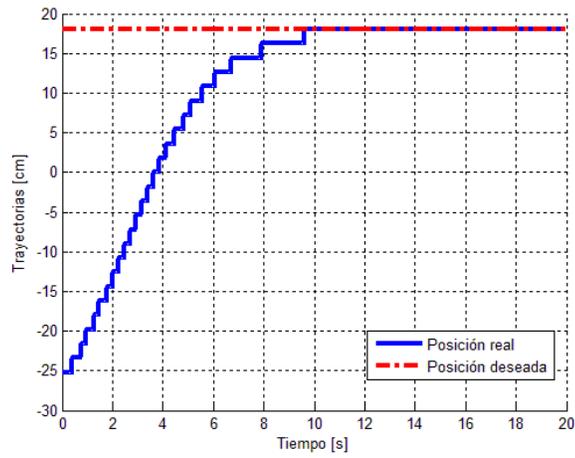


ILUSTRACIÓN 33: E1 TRAYECTORIA EN EL EJE Y

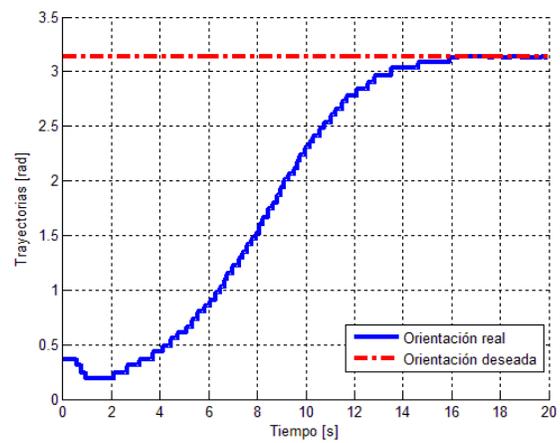


ILUSTRACIÓN 34: E1 TRAYECTORIA EN EL EJE Z

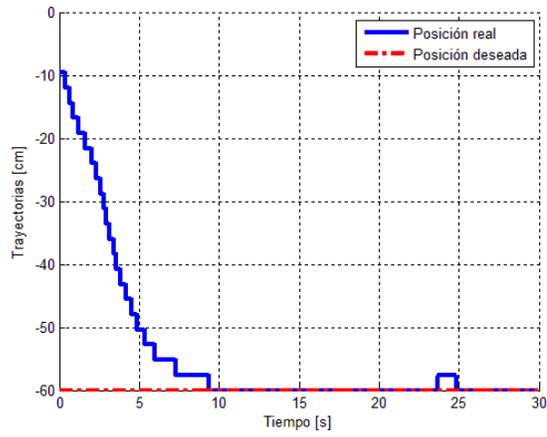


ILUSTRACIÓN 35: E2 TRAYECTORIA EN EL EJE X

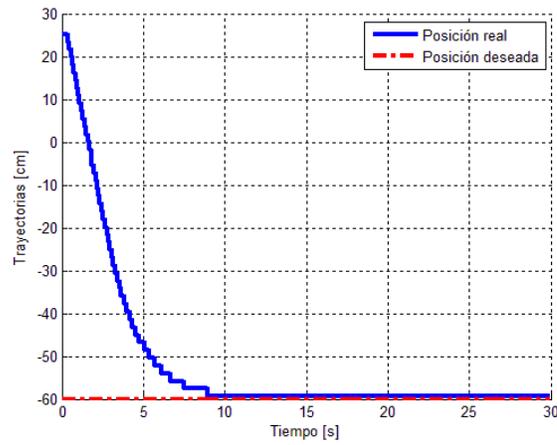


ILUSTRACIÓN 36: E2 TRAYECTORIA EN EL EJE Y

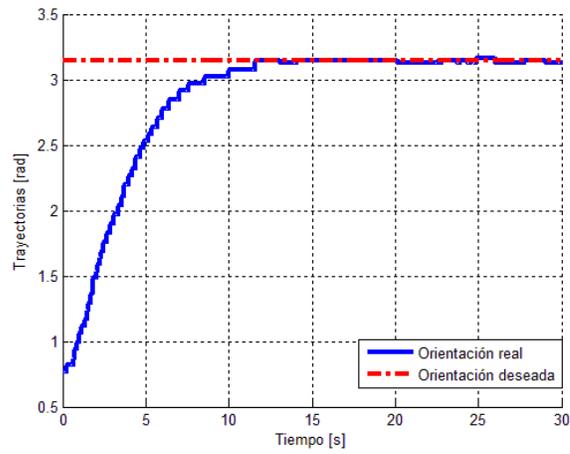


ILUSTRACIÓN 37: E2 TRAYECTORIA EN EL EJE Z

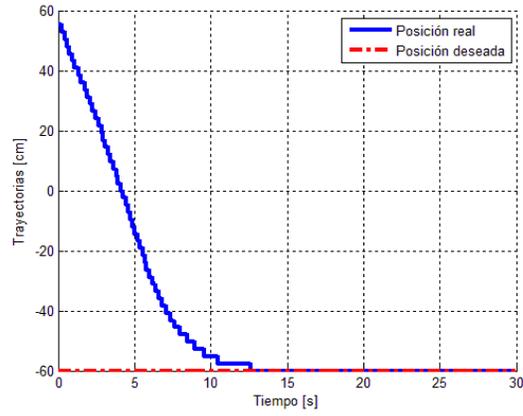


ILUSTRACIÓN 38: E3 TRAYECTORIA EN EL EJE X

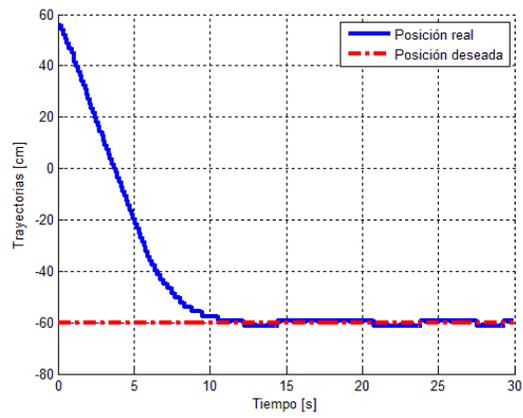


ILUSTRACIÓN 39: E3 TRAYECTORIA EN EL EJE Y

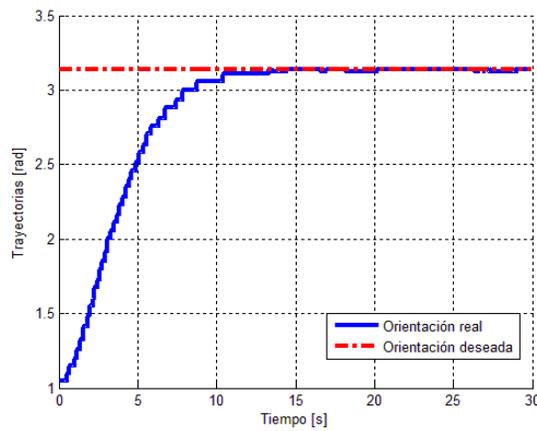


ILUSTRACIÓN 40: E3 TRAYECTORIA EN EL EJE Z

## SEGUIMIENTO DE POSTURA

El experimento consistió en seguir una trayectoria circular en posición tanto para x como para y (ecuaciones 4.11 y 4.12); y para la orientación una trayectoria senoidal (ecuación 4.13). Se eligió este tipo de trayectoria debido a que son suaves y diferenciables al menos dos veces. La finalidad de este experimento es observar al robot realizando movimientos independientes en sus tres grados de libertad, y comprobar que es capaz de realizar el seguimiento de postura.

$$(4.11) \quad x = 50 \cos\left(\frac{2\pi}{120}\right)$$

$$(4.12) \quad y = 50 \sin\left(\frac{2\pi}{120}\right)$$

$$(4.13) \quad \theta = 3.14 \cos\left(\frac{2\pi}{120}\right) + \pi$$

### ERRORES PROMEDIO

El error promedio de posición en el eje (x) fue  $e_x = 4.8014\text{cm}$ , en el eje (y) fue de  $e_y = 4.9468\text{cm}$  y el error promedio de orientación con respecto al eje (z) fue de  $e_w = 0.33404\text{rad}$ . La distribución media del error en el eje (x) fue de  $d_{mx} = 0.00184\text{cm}$ , en el eje (y) fue  $d_{my} = 0.00252\text{cm}$  y la distribución media con respecto al error en el eje (z) fue de  $d_{mw} = 0.013064\text{rad}$ . A continuación se muestran los resultados obtenidos de algunos experimentos. La tabla muestra los errores obtenidos y los términos E1, E2 y E3 hacen referencia a un conjunto de gráficas pertenecientes a un mismo experimento.

TABLA 6: EXPERIMENTOS DE SEGUIMIENTO

	<b>Error de posición en el eje x cm</b>	<b>Error de posición en el eje y cm</b>	<b>Error de orientación en el eje z rad</b>	<b>Ilustraciones</b>
<b>E1</b>	4.8014	4.9454	0.3181	41, 42, 43 y 44
<b>E2</b>	4.8014	4.9594	0.3531	45, 46, 47 y 48
<b>E3</b>	4.8114	4.9454	0.3355	49, 50, 51 y 52

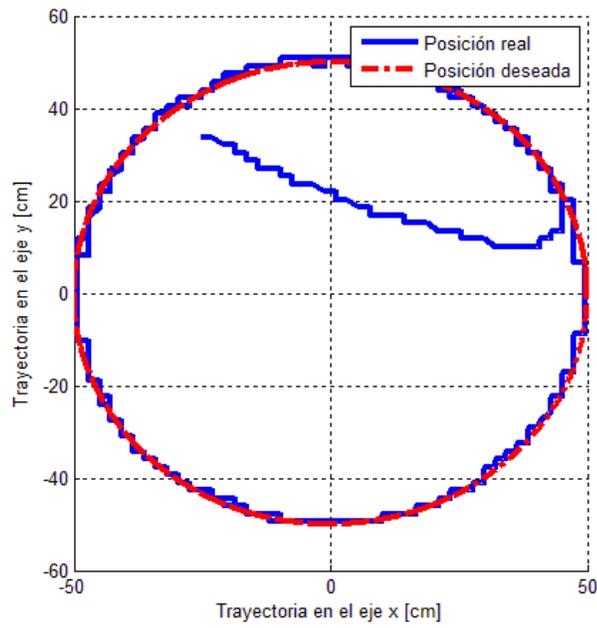


ILUSTRACIÓN 41: E1 SEGUIMIENTO DE POSTURA

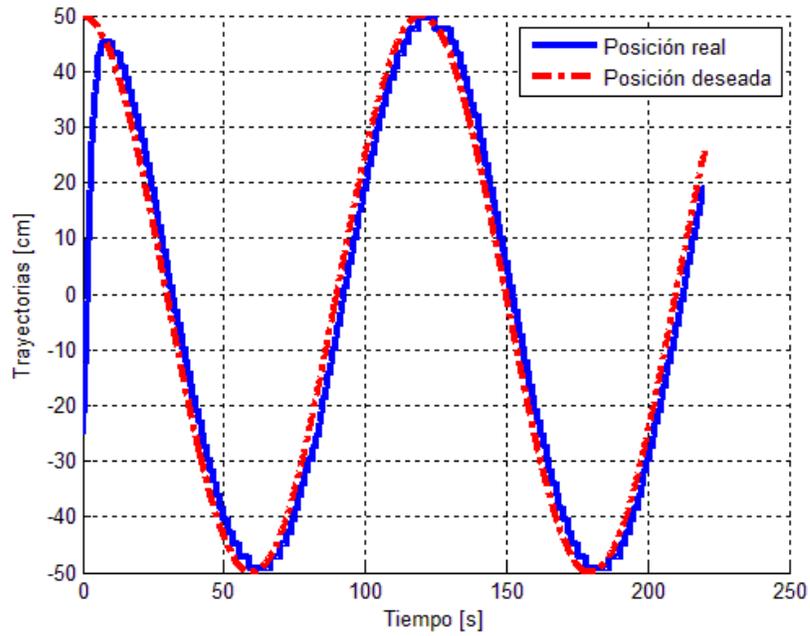


ILUSTRACIÓN 42: E1 TRAYECTORIA EN EL EJE X

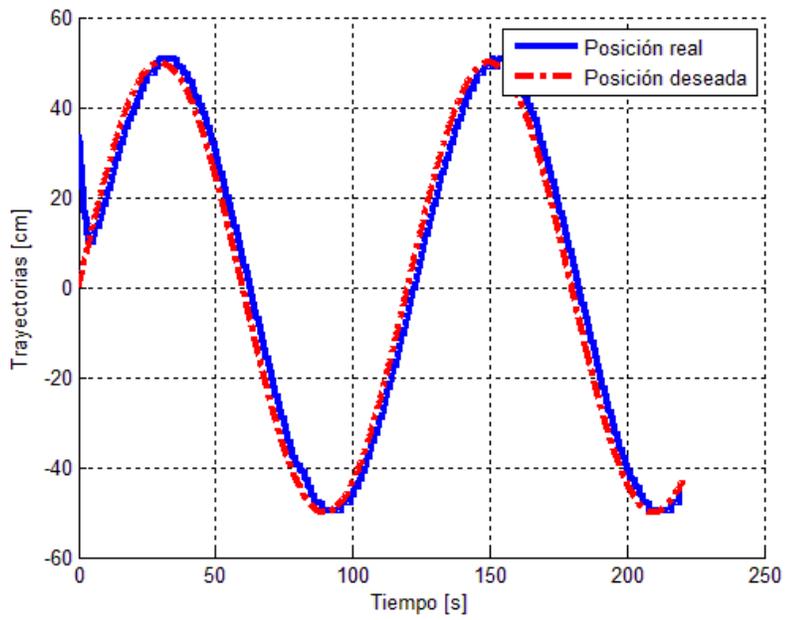


ILUSTRACIÓN 43: E1 TRAYECTORIA EN EL EJE Y

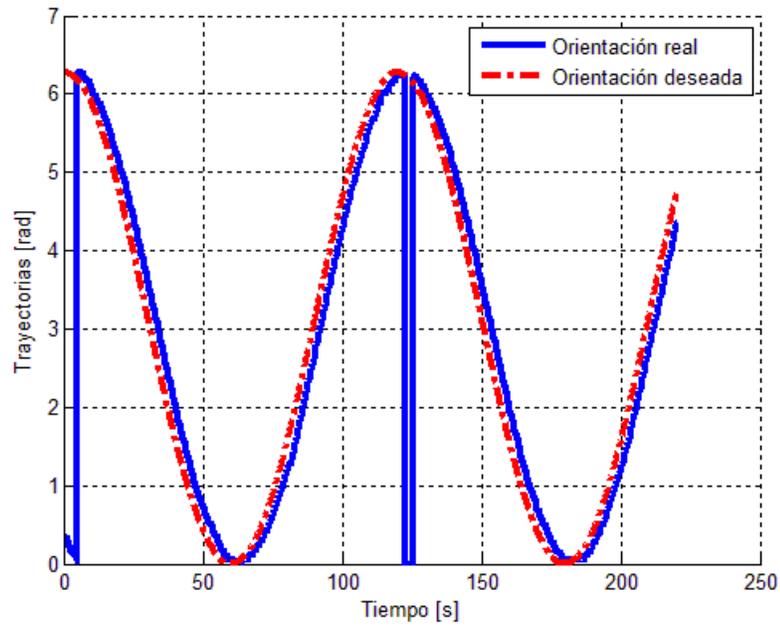


ILUSTRACIÓN 44: E1 TRAYECTORIA EN EL EJE Z

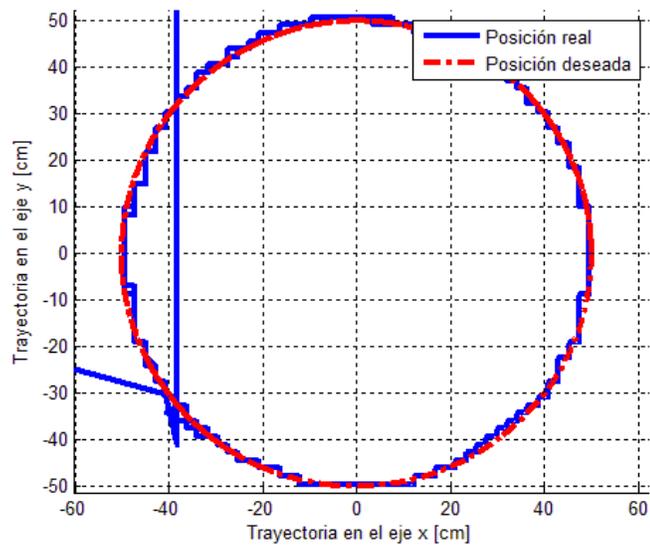


ILUSTRACIÓN 45: E2 SEGUIMIENTO DE POSTURA

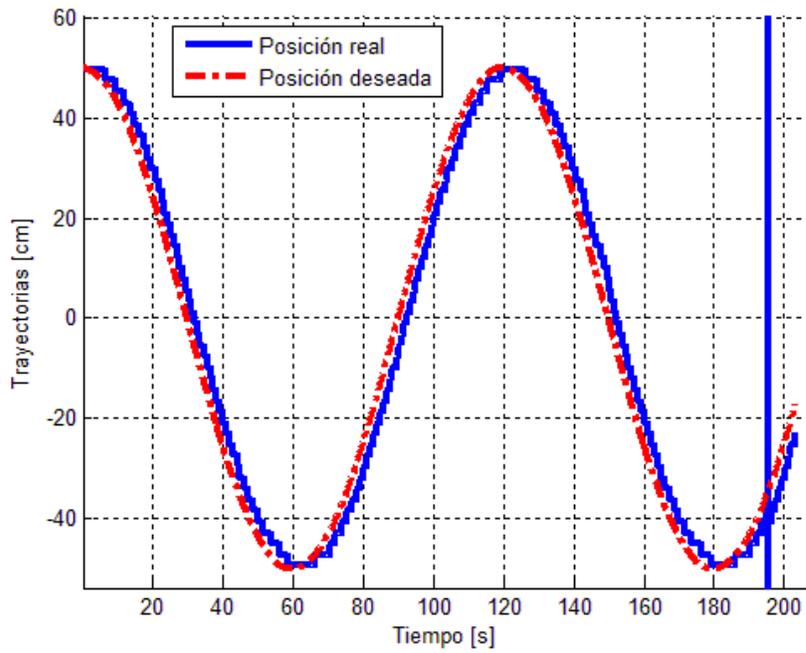


ILUSTRACIÓN 46: E2 TRAYECTORIA EN EL EJE X

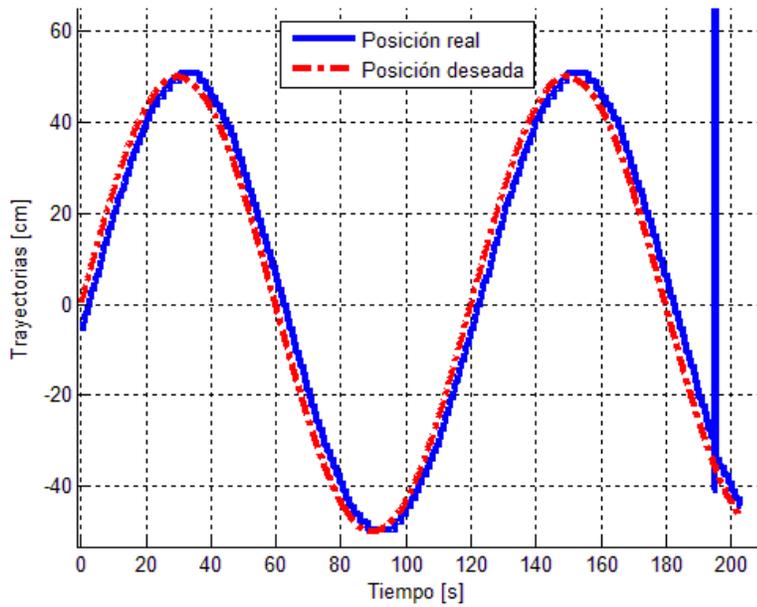


ILUSTRACIÓN 47: E2 TRAYECTORIA EN EL EJE Y

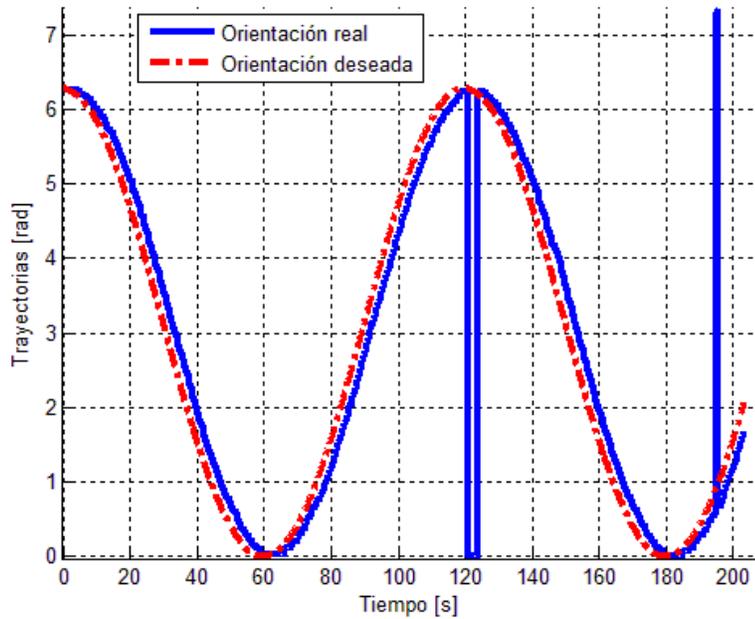


ILUSTRACIÓN 48: E2 TRAYECTORIA EN EL EJE Z

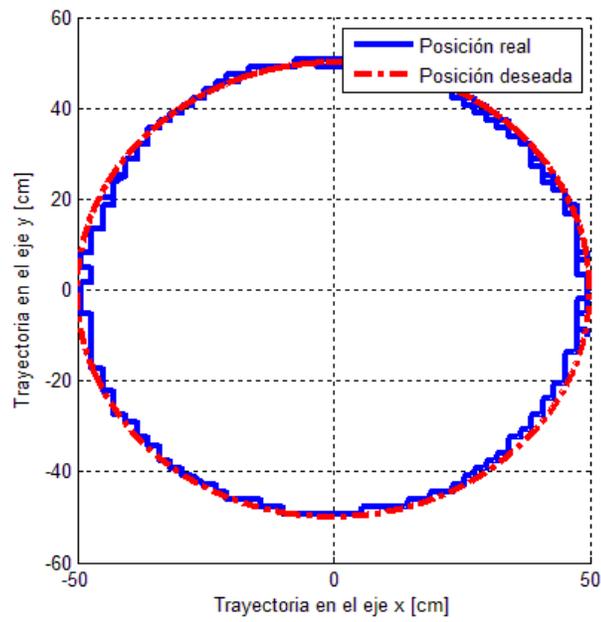


ILUSTRACIÓN 49: E3 SEGUIMIENTO DE POSTURA

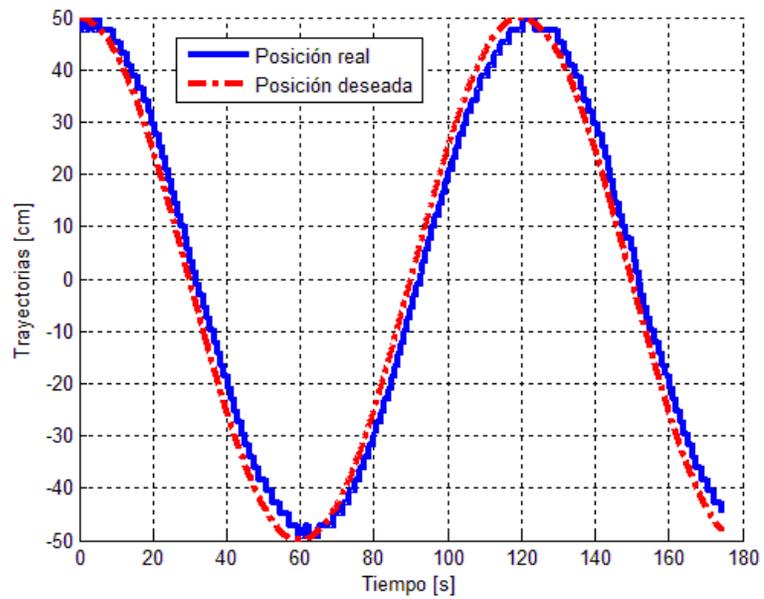


ILUSTRACIÓN 50: E3 TRAYECTORIA EN EL EJE X

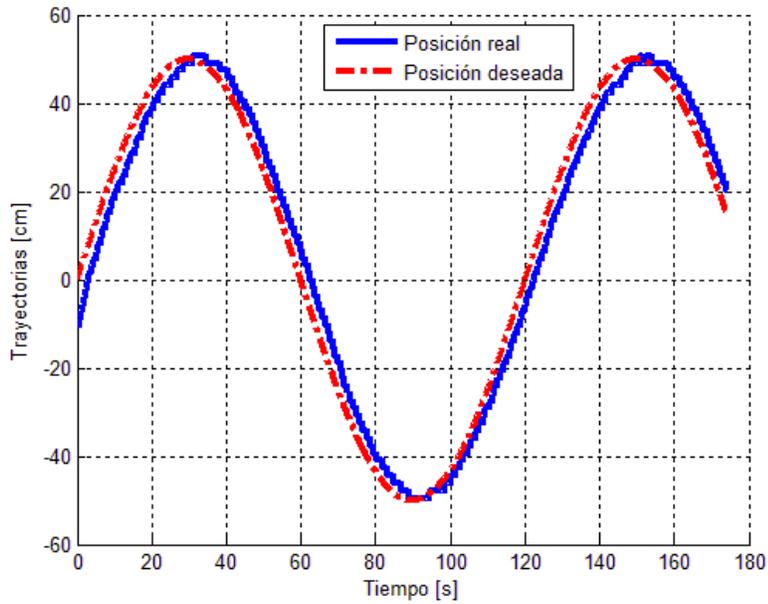


ILUSTRACIÓN 51: E3 TRAYECTORIA EN EL EJE Y

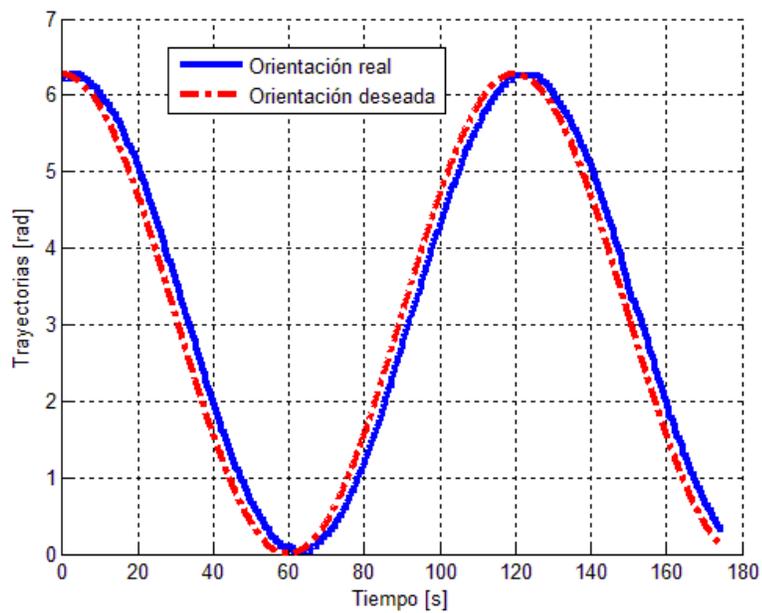


ILUSTRACIÓN 52: E3 TRAYECTORIA EN EL EJE Z

## DISCUSIÓN

---

La magnitud de los errores varía considerablemente de acuerdo al tipo de experimento realizado. En el caso de regulación el error promedio de posición es menor a un centímetro sin embargo en el experimento de seguimiento el error promedio se mantiene alrededor de los cinco centímetros en posición; el error promedio de orientación en el experimento de postura es cercano a 0.57 grados, y para el experimento de seguimiento es alrededor de veinte grados. La desviación media también varía de experimento a experimento; en el caso de regulación no es mayor a medio centímetro, para seguimiento la desviación media no es mayor a tres milímetros. Se puede observar que en el experimento de regulación el error promedio es menor que en el de seguimiento, pero la desviación media es considerablemente mayor que la del experimento de seguimiento.

En el caso de los experimentos de seguimiento, se observa que hay ruido en el cálculo de la posición y orientación del robot, por los cambios bruscos que se presentan en las gráficas. Observando el conjunto E1 de gráficas de seguimiento de postura, en específico la ilustración 44 entre los cien y ciento cincuenta segundos se ve un salto en la orientación del robot; de hecho en todos los conjuntos de gráficas se ve este salto. Esta variación en la orientación y/o posición del robot no es ocasionada por una falla en la cámara o en el sistema de visión, ya que de ser así el salto se presentaría tanto en posición y orientación en el mismo tiempo para todas las gráficas, lo cual no ocurre. Al ser una falla aislada. El ruido que se presenta se debe al envío de datos de ReactIVision a Matlab.

Sin embargo en el conjunto de gráficas E2 del seguimiento de postura, se presenta ruido ocasionado por una falla en el sistema de visión alrededor del tiempo doscientos segundos; esto se deduce debido a que es una falla presente en todas las gráficas, coincidiendo con un error común de los sistemas de visión cuando el objeto que se está rastreando desaparece del campo de visión o el sistema es incapaz de captarlo. También en ese mismo conjunto de gráficas aparece ruido debido a una falla en el envío de datos, al ser un ruido aislado solo a la gráfica de la ilustración 48, nuevamente en el intervalo de tiempo entre los cien y ciento cincuenta segundos.

---

# CONCLUSIONES

---

---

## CONCLUSIÓN

---

Se comprobó en el capítulo 2, teórica y prácticamente, que el robot móvil se muestra en este trabajo es omnidireccional y redundante; mediante el análisis de los grados de movilidad, direccionalidad y maniobrabilidad que el robot presentado es omnidireccional. La ley de control implementada permite el control independiente de los tres grados de libertad del robot; lo que significa que este tipo de robots puede mover al mismo tiempo que giran. La omnidireccionalidad es debida al uso mínimo de dos robots móviles diferenciales; y la redundancia es debida al uso combinado de tres o más robots diferenciales, en este caso particular cuatro. El análisis cinemático resultó ser una extensión del análisis del antecesor, al incluir solo los términos nuevos de los robots móviles diferenciales extras; se presentó por primera vez el análisis dinámico para esta clase de robots.

La redundancia en la robótica por lo general implica la aplicación de técnicas para coordinar los grados de libertad adicionales. Sin embargo, en este caso, la forma en que se calculan las ecuaciones cinemáticas del robot permite que las mismas ecuaciones se encarguen de la coordinación de los robots.

A diferencia del antecesor, los experimentos realizados fueron efectuados en tiempo real lo cual permitió obtener errores menores a un centímetro para la tarea de regulación y errores de cinco centímetros para la tarea de seguimiento. También el empleo de Matlab permite que el sistema sea modular, de esta manera se pueden implementar diversos controladores solo con alterar el algoritmo del bloque de control.

El empleo de comunicación UDP para el envío de datos presenta el error común a este tipo de comunicación, ya que al enviar la información no se toma alguna medida para garantizar la integridad de la misma; por lo tanto en algún punto la información se distorsiona y pareciera que el robot se transporta de forma inmediata de un punto a otro. Afortunadamente el error solo ocurre en un intervalo de tres milisegundos, por lo que no afecta de manera substancial el rendimiento del robot. También el sistema de

visión presenta el problema de la precisión, ya que hay valores que no puede leer; por ejemplo si al robot se le indica que vaya a la posición 7.5, este permanecerá oscilando alrededor de ella ya que solo es capaz de leer los valores 7.4 y 7.6; esto se verifica observando las gráficas del experimento de regulación.

## TRABAJO A FUTURO

---

Cómo trabajo a futuro queda la implementación del controlador dinámico inverso, así como otro tipo de controladores en el sistema y verificar el error resultando; con la finalidad de encontrar un controlador óptimo para este tipo de robots. También existe la posibilidad de ampliar el robot móvil y convertirlo en un manipulador móvil omnidireccional redundante; esto gracias a la magnitud del error que se presenta en el experimento de regulación. Pero principalmente queda por verificar mediante un análisis estático de fuerzas la ventaja que presenta la redundancia para la tarea de soportar cargas pesadas.

En el aspecto de diseño está la opción de rediseñar el robot para que pueda soportar alguna carga, ya que actualmente no es capaz de soportar peso alguno debido a la forma en que se sujetan los motores. También se debe de substituir la tarjeta Arduino para la lectura de los encoders, con la finalidad de aumentar la velocidad angular del sistema, ya que se ve limitada por la velocidad de lectura de los puertos de entrada de Arduino.

Por último en cuanto al sistema de visión, este debe modificarse para que tenga una mejor precisión e incluso se aumenten los cuadros por segundo; esto le brindaría al robot una mayor velocidad y disminuiría significativamente los errores mostrados en este trabajo.

# ANEXOS

## ANEXO A: MATRIZ JACOBIANA

$$S_{qa} \rightarrow \begin{pmatrix} \cos(\theta_p) & \sin(\theta_p) & 0 \\ -\sin(\theta_p) & \cos(\theta_p) & 0 \\ 0 & 0 & 1 \\ \frac{\cos(\theta_1)}{c} & \frac{\sin(\theta_1)}{c} & \frac{a \cdot \sin(\theta_1) - c + b \cdot \cos(\theta_1)}{c} \\ \frac{\cos(\theta_2)}{c} & \frac{\sin(\theta_2)}{c} & \frac{b \cdot \cos(\theta_2)}{c} - \frac{a \cdot \sin(\theta_2)}{c} - 1 \\ \frac{\cos(\theta_3)}{c} & \frac{\sin(\theta_3)}{c} & \frac{a \cdot \sin(\theta_3)}{c} - \frac{b \cdot \cos(\theta_3)}{c} - 1 \\ \frac{\cos(\theta_4)}{c} & \frac{\sin(\theta_4)}{c} & \frac{c + a \cdot \sin(\theta_4) + b \cdot \cos(\theta_4)}{c} \\ \frac{d \cdot \cos(\theta_1)}{c \cdot r} - \frac{\sin(\theta_1)}{r} & \frac{c \cdot \cos(\theta_1) + d \cdot \sin(\theta_1)}{c \cdot r} & \frac{a \cdot c \cdot \cos(\theta_1) + a \cdot d \cdot \sin(\theta_1) - b \cdot c \cdot \sin(\theta_1) + b \cdot d \cdot \cos(\theta_1)}{c \cdot r} \\ \frac{c \cdot \sin(\theta_1) + d \cdot \cos(\theta_1)}{c \cdot r} & \frac{c \cdot \cos(\theta_1) - d \cdot \sin(\theta_1)}{c \cdot r} & \frac{a \cdot d \cdot \sin(\theta_1) - a \cdot c \cdot \cos(\theta_1) - b \cdot c \cdot \sin(\theta_1) + b \cdot d \cdot \cos(\theta_1)}{c \cdot r} \end{pmatrix}$$

ILUSTRACIÓN 53: MATRIZ JACOBIANA PARTE 1

$$S_{qb} \rightarrow \begin{pmatrix} \frac{c \cdot \sin(\theta_2) - d \cdot \cos(\theta_2)}{c \cdot r} & \frac{c \cdot \cos(\theta_2) + d \cdot \sin(\theta_2)}{c \cdot r} & \frac{a \cdot c \cdot \cos(\theta_2) + a \cdot d \cdot \sin(\theta_2) + b \cdot c \cdot \sin(\theta_2) - b \cdot d \cdot \cos(\theta_2)}{c \cdot r} \\ \frac{c \cdot \sin(\theta_2) + d \cdot \cos(\theta_2)}{c \cdot r} & \frac{c \cdot \cos(\theta_2) - d \cdot \sin(\theta_2)}{c \cdot r} & \frac{a \cdot c \cdot \cos(\theta_2) - a \cdot d \cdot \sin(\theta_2) + b \cdot c \cdot \sin(\theta_2) + b \cdot d \cdot \cos(\theta_2)}{c \cdot r} \\ \frac{c \cdot \sin(\theta_3) - d \cdot \cos(\theta_3)}{c \cdot r} & \frac{c \cdot \cos(\theta_3) + d \cdot \sin(\theta_3)}{c \cdot r} & \frac{a \cdot c \cdot \cos(\theta_3) + a \cdot d \cdot \sin(\theta_3) + b \cdot c \cdot \sin(\theta_3) - b \cdot d \cdot \cos(\theta_3)}{c \cdot r} \\ \frac{c \cdot \sin(\theta_3) + d \cdot \cos(\theta_3)}{c \cdot r} & \frac{c \cdot \cos(\theta_3) - d \cdot \sin(\theta_3)}{c \cdot r} & \frac{a \cdot c \cdot \cos(\theta_3) - a \cdot d \cdot \sin(\theta_3) + b \cdot c \cdot \sin(\theta_3) + b \cdot d \cdot \cos(\theta_3)}{c \cdot r} \\ \frac{c \cdot \sin(\theta_4) - d \cdot \cos(\theta_4)}{c \cdot r} & \frac{c \cdot \cos(\theta_4) + d \cdot \sin(\theta_4)}{c \cdot r} & \frac{a \cdot c \cdot \cos(\theta_4) + a \cdot d \cdot \sin(\theta_4) - b \cdot c \cdot \sin(\theta_4) + b \cdot d \cdot \cos(\theta_4)}{c \cdot r} \\ \frac{c \cdot \sin(\theta_4) + d \cdot \cos(\theta_4)}{c \cdot r} & \frac{c \cdot \cos(\theta_4) - d \cdot \sin(\theta_4)}{c \cdot r} & \frac{a \cdot d \cdot \sin(\theta_4) - a \cdot c \cdot \cos(\theta_4) + b \cdot c \cdot \sin(\theta_4) + b \cdot d \cdot \cos(\theta_4)}{c \cdot r} \end{pmatrix}$$

ILUSTRACIÓN 54: MATRIZ JACOBIANA PARTE 2

## ANEXO B: MATRIZ A

$$A \rightarrow \begin{pmatrix}
 -\cos(\theta_1) & -\sin(\theta_1) & a \cdot \sin(\theta_1) - c + b \cdot \cos(\theta_1) & -c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -\cos(\theta_1) & -\sin(\theta_1) & a \cdot \sin(\theta_1) - c + b \cdot \cos(\theta_1) & -c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -\cos(\theta_2) & -\sin(\theta_2) & b \cdot \cos(\theta_2) - a \cdot \sin(\theta_2) - c & 0 & -c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -\cos(\theta_2) & -\sin(\theta_2) & b \cdot \cos(\theta_2) - a \cdot \sin(\theta_2) - c & 0 & -c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -\cos(\theta_3) & -\sin(\theta_3) & a \cdot \sin(\theta_3) - c - b \cdot \cos(\theta_3) & 0 & 0 & -c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -\cos(\theta_3) & -\sin(\theta_3) & a \cdot \sin(\theta_3) - c - b \cdot \cos(\theta_3) & 0 & 0 & -c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -\cos(\theta_4) & -\sin(\theta_4) & -c - a \cdot \sin(\theta_4) - b \cdot \cos(\theta_4) & 0 & 0 & 0 & -c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -\cos(\theta_4) & -\sin(\theta_4) & -c - a \cdot \sin(\theta_4) - b \cdot \cos(\theta_4) & 0 & 0 & 0 & -c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -\sin(\theta_1) & \cos(\theta_1) & b \cdot \sin(\theta_1) - a \cdot \cos(\theta_1) - d & -d & 0 & 0 & 0 & -r & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -\sin(\theta_1) & \cos(\theta_1) & d - a \cdot \cos(\theta_1) + b \cdot \sin(\theta_1) & d & 0 & 0 & 0 & 0 & -r & 0 & 0 & 0 & 0 & 0 & 0 \\
 -\sin(\theta_2) & \cos(\theta_2) & a \cdot \cos(\theta_2) - d + b \cdot \sin(\theta_2) & 0 & -d & 0 & 0 & 0 & 0 & -r & 0 & 0 & 0 & 0 & 0 \\
 -\sin(\theta_2) & \cos(\theta_2) & d + a \cdot \cos(\theta_2) + b \cdot \sin(\theta_2) & 0 & d & 0 & 0 & 0 & 0 & 0 & -r & 0 & 0 & 0 & 0 \\
 -\sin(\theta_3) & \cos(\theta_3) & -d - a \cdot \cos(\theta_3) - b \cdot \sin(\theta_3) & 0 & 0 & -d & 0 & 0 & 0 & 0 & 0 & -r & 0 & 0 & 0 \\
 -\sin(\theta_3) & \cos(\theta_3) & d - a \cdot \cos(\theta_3) - b \cdot \sin(\theta_3) & 0 & 0 & d & 0 & 0 & 0 & 0 & 0 & 0 & -r & 0 & 0 \\
 -\sin(\theta_4) & \cos(\theta_4) & a \cdot \cos(\theta_4) - d - b \cdot \sin(\theta_4) & 0 & 0 & 0 & -d & 0 & 0 & 0 & 0 & 0 & 0 & -r & 0 \\
 -\sin(\theta_4) & \cos(\theta_4) & d + a \cdot \cos(\theta_4) - b \cdot \sin(\theta_4) & 0 & 0 & 0 & d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -r
 \end{pmatrix}$$

ILUSTRACIÓN 55: MATRIZ QUE DESCRIBE LAS RESTRICCIONES CINEMÁTICAS DE LAS RUEDAS

## ANEXO C: MATRICES DE MASAS E INCERCIAS Y DE ACELERACIONES

La representación de estas matrices se realiza en forma de vectores columna para una adecuada visualización.

$$M_1 \rightarrow \begin{pmatrix} m_p + 8 \cdot m_w \\ 0 \\ 2 \cdot c \cdot m_w \cdot \cos(\theta_1 - \theta_p) + 2 \cdot c \cdot m_w \cdot \cos(\theta_2 - \theta_p) + 2 \cdot c \cdot m_w \cdot \cos(\theta_3 + \theta_p) + 2 \cdot c \cdot m_w \cdot \cos(\theta_4 + \theta_p) - c_y \cdot m_p \cdot \cos(\theta_p) - c_x \cdot m_p \cdot \sin(\theta_p) \\ 2 \cdot c \cdot m_w \cdot \cos(\theta_1 - \theta_p) \\ 2 \cdot c \cdot m_w \cdot \cos(\theta_2 - \theta_p) \\ 2 \cdot c \cdot m_w \cdot \cos(\theta_3 + \theta_p) \\ 2 \cdot c \cdot m_w \cdot \cos(\theta_4 + \theta_p) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

ILUSTRACIÓN 56: COLUMNA 1, MATRIZ DE MASAS E INCERCIAS

$$M_2 \rightarrow \begin{pmatrix} 0 \\ m_p + 8 \cdot m_w \\ 2 \cdot c \cdot m_w \cdot \sin(\theta_1 - \theta_p) + 2 \cdot c \cdot m_w \cdot \sin(\theta_2 + \theta_p) + 2 \cdot c \cdot m_w \cdot \sin(\theta_3 + \theta_p) + 2 \cdot c \cdot m_w \cdot \sin(\theta_4 + \theta_p) + c_x \cdot m_p \cdot \cos(\theta_p) - c_y \cdot m_p \cdot \sin(\theta_p) \\ 2 \cdot c \cdot m_w \cdot \sin(\theta_1 - \theta_p) \\ 2 \cdot c \cdot m_w \cdot \sin(\theta_2 + \theta_p) \\ 2 \cdot c \cdot m_w \cdot \sin(\theta_3 + \theta_p) \\ 2 \cdot c \cdot m_w \cdot \sin(\theta_4 + \theta_p) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

ILUSTRACIÓN 57: COLUMNA 2, MATRIZ DE MASAS E INCERCIAS



$$M_4 \rightarrow \begin{pmatrix} 2 \cdot c \cdot m_W \cdot \cos(\theta_1 + \theta_p) \\ 2 \cdot c \cdot m_W \cdot \sin(\theta_1 + \theta_p) \\ 2 \cdot I_{ZZW} + 2 \cdot c^2 \cdot m_W + 2 \cdot d^2 \cdot m_W - 2 \cdot a \cdot c \cdot m_W \cdot \sin(\theta_1) - 2 \cdot b \cdot c \cdot m_W \cdot \cos(\theta_1) \\ 2 \cdot m_W \cdot c^2 + 2 \cdot m_W \cdot d^2 + 2 \cdot I_{ZZW} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

ILUSTRACIÓN 60: COLUMNA 4, MATRIZ DE MASAS E INERCIAS

$$M_5 \rightarrow \begin{pmatrix} 2 \cdot c \cdot m_W \cdot \cos(\theta_2 + \theta_p) \\ 2 \cdot c \cdot m_W \cdot \sin(\theta_2 + \theta_p) \\ 2 \cdot I_{ZZW} + 2 \cdot c^2 \cdot m_W + 2 \cdot d^2 \cdot m_W + 2 \cdot a \cdot c \cdot m_W \cdot \sin(\theta_2) - 2 \cdot b \cdot c \cdot m_W \cdot \cos(\theta_2) \\ 0 \\ 2 \cdot m_W \cdot c^2 + 2 \cdot m_W \cdot d^2 + 2 \cdot I_{ZZW} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

ILUSTRACIÓN 61: COLUMNA 5, MATRIZ DE MASAS E INERCIAS

$$M_6 \rightarrow \begin{pmatrix} 2 \cdot c \cdot m_W \cdot \cos(\theta_3 + \theta_p) \\ 2 \cdot c \cdot m_W \cdot \sin(\theta_3 + \theta_p) \\ 2 \cdot I_{ZZW} + 2 \cdot c^2 \cdot m_W + 2 \cdot d^2 \cdot m_W - 2 \cdot a \cdot c \cdot m_W \cdot \sin(\theta_3) + 2 \cdot b \cdot c \cdot m_W \cdot \cos(\theta_3) \\ 0 \\ 0 \\ 2 \cdot m_W \cdot c^2 + 2 \cdot m_W \cdot d^2 + 2 \cdot I_{ZZW} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

ILUSTRACIÓN 62: COLUMNA 6, MATRIZ DE MASAS E INERCIAS

$$M_7 \rightarrow \begin{pmatrix} 2 \cdot c \cdot m_W \cdot \cos(\theta_4 + \theta_p) \\ 2 \cdot c \cdot m_W \cdot \sin(\theta_4 + \theta_p) \\ 2 \cdot I_{ZZW} + 2 \cdot c^2 \cdot m_W + 2 \cdot d^2 \cdot m_W + 2 \cdot a \cdot c \cdot m_W \cdot \sin(\theta_4) + 2 \cdot b \cdot c \cdot m_W \cdot \cos(\theta_4) \\ 0 \\ 0 \\ 0 \\ 2 \cdot m_W \cdot c^2 + 2 \cdot m_W \cdot d^2 + 2 \cdot I_{ZZW} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad M_8 \rightarrow \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad M_9 \rightarrow \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

ILUSTRACIÓN 63: COLUMNAS 7, 8 Y 9; MATRIZ DE MASAS E INERCIAS



$$C_3 \rightarrow \begin{pmatrix} c_y \cdot m_p \cdot \sin(\theta_p) - 2 \cdot c \cdot m_w \cdot \sin(\theta_2 + \theta_p) - 2 \cdot c \cdot m_w \cdot \sin(\theta_3 + \theta_p) - 2 \cdot c \cdot m_w \cdot \sin(\theta_4 + \theta_p) - c_x \cdot m_p \cdot \cos(\theta_p) - 2 \cdot c \cdot m_w \cdot \sin(\theta_1 + \theta_p) \\ 2 \cdot c \cdot m_w \cdot \cos(\theta_1 + \theta_p) + 2 \cdot c \cdot m_w \cdot \cos(\theta_2 + \theta_p) + 2 \cdot c \cdot m_w \cdot \cos(\theta_3 + \theta_p) + 2 \cdot c \cdot m_w \cdot \cos(\theta_4 + \theta_p) - c_y \cdot m_p \cdot \cos(\theta_p) - c_x \cdot m_p \cdot \sin(\theta_p) \\ 0 \\ 2 \cdot a \cdot c \cdot m_w \cdot \cos(\theta_1) - 2 \cdot b \cdot c \cdot m_w \cdot \sin(\theta_1) \\ -2 \cdot a \cdot c \cdot m_w \cdot \cos(\theta_2) - 2 \cdot b \cdot c \cdot m_w \cdot \sin(\theta_2) \\ 2 \cdot a \cdot c \cdot m_w \cdot \cos(\theta_3) + 2 \cdot b \cdot c \cdot m_w \cdot \sin(\theta_3) \\ 2 \cdot b \cdot c \cdot m_w \cdot \sin(\theta_4) - 2 \cdot a \cdot c \cdot m_w \cdot \cos(\theta_4) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

ILUSTRACIÓN 66: COLUMNA 3, MATRIZ DE ACELERACIONES

$$C_6 \rightarrow \begin{pmatrix} -2 \cdot c \cdot m_w \cdot \sin(\theta_3 + \theta_p) \\ 2 \cdot c \cdot m_w \cdot \cos(\theta_3 + \theta_p) \\ -2 \cdot a \cdot c \cdot m_w \cdot \cos(\theta_3) - 2 \cdot b \cdot c \cdot m_w \cdot \sin(\theta_3) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad C_7 \rightarrow \begin{pmatrix} -2 \cdot c \cdot m_w \cdot \sin(\theta_4 + \theta_p) \\ 2 \cdot c \cdot m_w \cdot \cos(\theta_4 + \theta_p) \\ 2 \cdot a \cdot c \cdot m_w \cdot \cos(\theta_4) - 2 \cdot b \cdot c \cdot m_w \cdot \sin(\theta_4) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad C_8 \rightarrow \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

ILUSTRACIÓN 67: COLUMNAS 6, 7 Y 8; MATRIZ DE ACELERACIONES

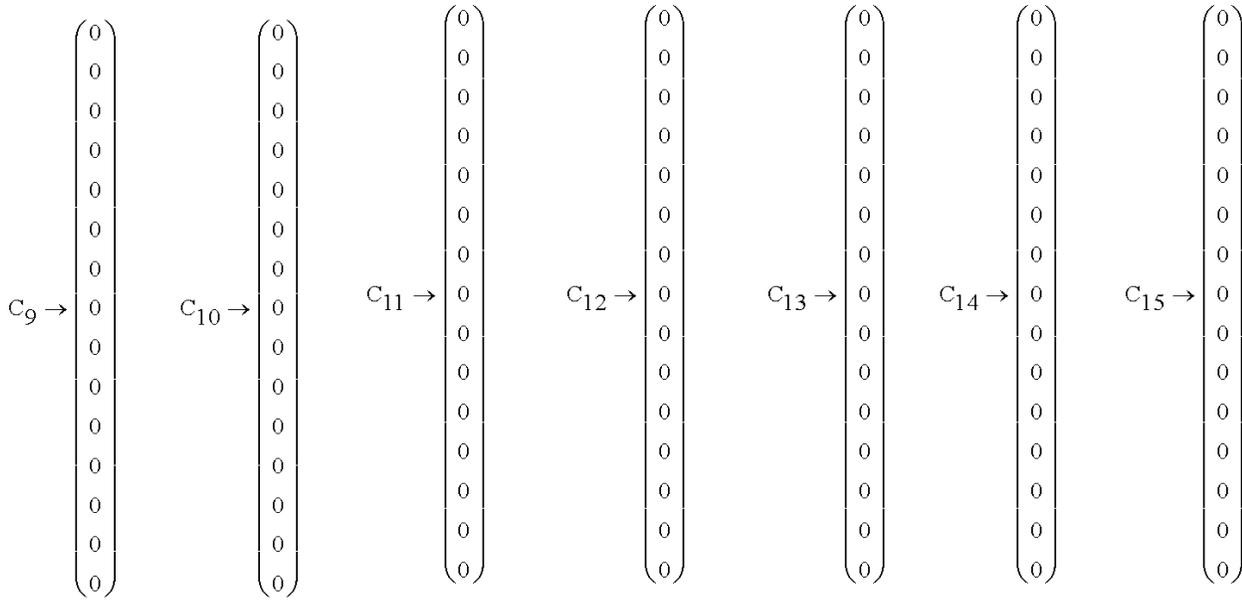


ILUSTRACIÓN 68: COLUMNAS 9:15, MATRIZ DE ACELERACIONES

## BIBLIOGRAFÍA

---

- [1] G. B. a. B.-N. C. Campion, «"Structural properties and classification of kinematics and dynamics models of wheeled Mobile Robot",» *IEEE Trans. on Robotics and Automation*,, vol. 4, nº 2, p. 281–340, 1987.
- [2] P. F. M. a. C. P. Neuman, «"Kinematic modeling of wheeled mobile robots",» *Journal of Robotic Systems*, vol. 4, nº 2, p. 281–340, 1987.
- [3] B. Ilou, «"Wheels for a course stable self-propelling vehicle movable in any desired direction on the ground or some other base"». USA 1975.
- [4] G. Indiveri, «"Swedish wheeled omnidirectional mobile robots: Kinematics analysis and control",» *IEEE Transactions on Robotics*, vol. 25, nº 1, p. 164 –171, Feb. 2009.
- [5] K.-S. B. a. J.-B. Song, «"Design and construction of continuous alternate wheels for an omnidirectional mobile robot",» *Journal of Robotic Systems*, vol. 20, nº 9, p. 569–579, 2003.
- [6] J.-B. S. a. K.-S. Byun, «"Design and control of a four-wheeled omnidirectional mobile robot with steerable omnidirectional wheels",» *Journal of Robotic Systems*, vol. 21, nº 4, p. 193–208, 2004.
- [7] M. S. N. G. H. K. A. M. a. I. E. H. Asama, «"Mutual transportation of cooperative mobile robots using forklift mechanisms",» *IEEE International Conference on Robotics and Automation*, vol. 2, p. 1754 –1759., Apr. 1996.
- [8] A. T. a. S. M. M. Wada, «"Caster drive mechanism for holonomic and omnidirectional mobile platforms with no over constraint",» *IEEE Int. Conf. on Robotics and Automation*, pp. 1531-1538, 2000..
- [9] B.-J. Y. a. W. Kim, «"The kinematics for redundantly actuated omni-directional

mobile robots",» *Journal of Robotic Systems*, vol. 19, nº 6, pp. 255-267, 2002.

[10] J. Salih, «"Designing omni-directional mobile robot with mecanum wheel",» *American Journal of Applied Sciences*, vol. 3, nº 5, pp. 1831-1835, 2006.

[11] K. M. a. N. Flann, «"A six-wheeled omnidirectional autonomous mobile robot",» *IEEE Control System Magazine*, vol. 20, nº 6, pp. 53-66, 2000.

[12] M. B. a. L. Ma, «"Discrete control for visual servoing the ODIS robot to parking lot lines",» *IEEE Int. Conf. on Robotics and Automation*, pp. 3149-3154, 2005.

[13] M. Y. a. A. M. N. Ushimi, «"Two wheels caster type odometer for omni-directional vehicles",» *EEE Int. Conf. On Robotics and Automation*, pp. 497-502, 2003.

[14] V. Dr. González-Villela, «"Omnidirectional mobile platform driven by two differentially driven wheeled mobile robots for moving heavy loads",» *MEMORIAS DEL XIX CONGRESO INTERNACIONAL ANUAL DE LA SOMIM*, pp. 1064-1069, 2013.

[15] M. M. a. S. H. K. Tadakuma, «"Mechanical design of joint braking and underactuated mechanism of "Tri-Star3";horizontal polyarticular arm equipped 3-wheeled expandable mobile robot",» *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 4252-4259, 2006.

[16] S. M. K. B.-J. Y. Eui-Jung Jung, «"Two-Leveled Obstacle Avoidance Scheme Using a Kinematically Redundant Omni-directional Mobile Robot",» *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3949-3954, 2010.

[17] W. K. K. Byung-Ju Yi, «"The Kinematics for Redundantly Actuated Omnidirectional Mobile Robots",» *Journal of Robotic Systems*, pp. 255-267, 2002.

[18] C. Canudas, "Theory of Robot Control", Springer, 1996.

[19] J. J. Craig, "Introduction to Robotics mechanics and control", Pearson, 2005.

[20] «Arduino,» Arduino, 2015. [En línea]. Available: <http://www.arduino.cc>. [Último acceso: 20 marzo 2014].

[21] «Arduino,» Ardubo, 2015. [En línea]. Available:

- <http://arduino.cc/en/Main/ArduinoWiFiShield>. [Último acceso: 30 marzo 2015].
- [22] «Robot electronics,» Robot electronics, [En línea]. Available: <http://www.robot-electronics.co.uk/htm/emg30.htm>. [Último acceso: 30 marzo 2015].
- [23] «Robot electronics,» Robot electronics, [En línea]. Available: <http://www.robot-electronics.co.uk/htm/md25tech.htm>. [Último acceso: 30 marzo 2015].
- [24] P. A. Laplante, "Real-Time Systems Design and Analysis", Wiley , 2004.
- [25] P. K. M. Al-Aubidy, "*Classification of Real-Time Systems*", Computer Engineering Department: Philadelphia University, 2011.
- [26] «Mathworks,» Mathworks, [En línea]. Available: <http://www.mathworks.com/products/rtwt/index.html>. [Último acceso: 30 marzo 2015].
- [27] «Mathworks,» Mathworks, [En línea]. Available: <http://www.mathworks.com/products/rtwt/features.html#key-features>. [Último acceso: 30 marzo 2015].
- [28] «Mathworks,» Mathworks, [En línea]. Available: <http://www.mathworks.com/help/rtwin/ug/real-time-windows-target-kernel.html>. [Último acceso: 30 marzo 2015].
- [29] G. P. Martinsanz, "Visión por computador imágenes digitales y aplicaciones ", México D.F.: Alfaomega Ra-Ma , marzo 2008.
- [30] P. Corke, "Robotics, vision and control", Springer, 2011.
- [31] «ReacTIVision,» ReacTIVision, 21 octubre 2014. [En línea]. Available: <http://reactivision.sourceforge.net/>. [Último acceso: 30 marzo 2015].
- [32] B. Siciliano, Robotics: Modelling, Planning and Control, Springer, 2010.
- [33] M. V. J. G. Villela, «"Evadiendo obstáculos con robots móviles",» *Revista Digital Universitaria*, vol. 6, nº 1, 2005.